****

# 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 5](#_Toc32574613)

[4. Static Testing 5](#_Toc32574614)

[5. Mitigation Plan 11](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/15/2023** | **Benjamin Dowell** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for 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 choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Benjamin Dowell

## Interpreting Client Needs

Artemis Financial is looking to modernize its operations. They specialize in individualized financial plans and require a software solution that supports various aspects of financial planning, including savings, retirement, investments, and insurance. They want to make sure that their application is protected from external threats like cyberattacks and data breaches.

The client wants Global Rain to provide guidance on implementing effective software security measures. This includes patching security vulnerabilities in the existing software and the new solution is built with security as the primary focus. Artemis Financial requires that their existing RESTful API is secure to protect sensitive financial data and prevent unauthorized access. This should be outlined in a vulnerability assessment report containing detail on any security vulnerabilities found in the current software application and provide recommendations on a strategy for mitigating them.

The value of secure communications to the company is customer base trust. This trust is earned by protecting their sensitive information such as customer data and financial records. This includes during transmissions and while stored within the company. If the company were to lose its reputation, this could potentially be costly to recover from.

The company does make international transactions. Some government restrictions about secure communications to consider are the use of encryption technologies, data localization laws, monitoring and surveillance laws, import/export laws, privacy laws, telecommunications, and national security laws. It is essential for companies to know where they are dealing and when international, other countries laws regarding as such.

Some relevant external threats could be malware, ransomware, phishing attacks, advanced persistent threats, cloud security risks, social engineering, and credential theft to name a few.

Modernizing a software system involves updating and improving various aspects of the software to align with current technology trends and business needs.

Open-source Libraries and Components need to be scanned for known vulnerabilities and chosen from ones that have a healthy user community that generally makes them more actively maintained.

Evolving web application technologies can include considering more modern frameworks for enhancing user experience and better performance.

Assess whether cloud adoption could improve reliability or implementing DevOps practices to automate software development, testing, and deployment processes. Lastly, regulatory compliance and data privacy through compliance with relevant data protection regulations such as PCI-DSS (credit card management) or Gramm-Leach-Bliley (financial information management).

## Areas of Security

Input Validation – Artemis Financial’s API probably involves untrusted user input for various reasons. Input validation is a huge concern to avoiding SQL injection, XSS, and command injection attacks.

Secure API – Given that Artemis Financial already has a working RESTfull API, it is essential in combing it over for security flaws.

Code Error – Validating that the written code is handling exceptions, errors, and unexpected situations securely. Ensuring any attempts that throw errors, do not cause leakage or leave the application vulnerable.

Client/Server – Due to the vast geographic distance between Artemis and it’s customers and potential environments data must travel through, it is crucial that communication is secure between server-side and client-side.

Cryptography – Given the nature of financial data, the value that is sought of it, and communication being international, to protect customer information and records, it is prudent to ensure the encryption of this data in transit and while it is stored.

Encapsulation – Ensuring proper access controls as many roles will be filled by the application, protecting data structures from unauthorized access and manipulation is of great importance.

## Manual Review

Architecture Review: appears to be a spring boot application

Input Validation: While reviewing the GreetingController, I noticed there are no input validation measures in the code.

APIs: The API GreetingController does not appear to be very secure in interactions with other APIs.

Cryptography: In the pom.xml file, cryptography appears to be in use by org.bouncycastle:bcprov-jdk15on.

Client/Server: The GreetingController is a server-side component of the spring boot application and handles HTTP requests securely if configured appropriately.

Code Error: The code lacks quite a bit in error handling and validation.

Code Quality: Nothing wrong with the code quality.

Encapsulation: The GreetingController class does show encapsulation, though nothing in the code is overly complex.

## Static Testing

|  |  |
| --- | --- |
| **VULNERABILITY CODES** | **DESCRIPTION** |
| bcprov-jdk15on-1.46.jar | |
| CVE-2016-1000352 | In the Bouncy Castle JCE Provider version 1.55 and earlier the ECIES implementation allowed the use of ECB mode. This mode is regarded as unsafe and support for it has been removed from the provider. |
| CVE-2016-1000346 | In the Bouncy Castle JCE Provider version 1.55 and earlier the other party DH public key is not fully validated. This can cause issues as invalid keys can be used to reveal details about the other party's private key where static Diffie-Hellman is in use. As of release 1.56 the key parameters are checked on agreement calculation. |
| CVE-2016-1000345 | In the Bouncy Castle JCE Provider version 1.55 and earlier the DHIES/ECIES CBC mode vulnerable to padding oracle attack. For BC 1.55 and older, in an environment where timings can be easily observed, it is possible with enough observations to identify when the decryption is failing due to padding. |
| CVE-2016-1000344 | In the Bouncy Castle JCE Provider version 1.55 and earlier the DHIES implementation allowed the use of ECB mode. This mode is regarded as unsafe and support for it has been removed from the provider. |
| CVE-2016-1000343 | In the Bouncy Castle JCE Provider version 1.55 and earlier the DSA key pair generator generates a weak private key if used with default values. If the JCA key pair generator is not explicitly initialised with DSA parameters, 1.55 and earlier generates a private value assuming a 1024 bit key size. In earlier releases this can be dealt with by explicitly passing parameters to the key pair generator. |
| CVE-2016-1000342 | In the Bouncy Castle JCE Provider version 1.55 and earlier ECDSA does not fully validate ASN.1 encoding of signature on verification. It is possible to inject extra elements in the sequence making up the signature and still have it validate, which in some cases may allow the introduction of 'invisible' data into a signed structure. |
| CVE-2016-1000341 | In the Bouncy Castle JCE Provider version 1.55 and earlier 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 k value and ultimately the private value as well. |
| 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 it turns out that if 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 although it was substantially less. AESEngine has been modified to remove any signs of leakage (testing carried out on Intel X86-64) and is now the primary AES class for the BC JCE provider from 1.56. Use of AESFastEngine is now only recommended where otherwise deemed appropriate. |
| CVE-2016-1000338 | In Bouncy Castle JCE Provider version 1.55 and earlier the DSA does not fully validate ASN.1 encoding of signature on verification. It is possible to inject extra elements in the sequence making up the signature and still have it validate, which in some cases may allow the introduction of 'invisible' data into a signed structure. |
| 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. Bouncy Castle release 1.47 changes the BKS format to a format which uses a 160 bit HMAC instead. This applies to any BKS keystore generated prior to BC 1.47. For situations where people need to create the files for legacy reasons a specific keystore type "BKS-V1" was introduced in 1.49. It should be noted that the use of "BKS-V1" is discouraged by the library authors and should only be used where it is otherwise safe to do so, as in where the use of a 16 bit checksum for the file integrity check is not going to cause a security issue in itself. |
| 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." |
| CVE-2013-1624 | The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 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 |
| spring-boot-2.2.4.RELEASE.jar | |
| CVE-2023-20883 | In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache. |
| CVE-2023-20873 | In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+. |
| CVE-2022-27772 | \*\* UNSUPPORTED WHEN ASSIGNED \*\* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer. |
| logback-core-1.2.3.jar | |
| CVE-2021-42550 | In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers. |
| log4j-api-2.12.1.jar | |
| CVE-2021-44832 | Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2. |
| CVE-2021-45105 | Apache Log4j2 versions 2.0-alpha1 through 2.16.0 (excluding 2.12.3 and 2.3.1) did not protect from uncontrolled recursion from self-referential lookups. This allows an attacker with control over Thread Context Map data to cause a denial of service when a crafted string is interpreted. This issue was fixed in Log4j 2.17.0, 2.12.3, and 2.3.1. |
| CVE-2021-45046 | It was found that the fix to address CVE-2021-44228 in Apache Log4j 2.15.0 was incomplete in certain non-default configurations. This could allows attackers with control over Thread Context Map (MDC) input data when the logging configuration uses a non-default Pattern Layout with either a Context Lookup (for example, $${ctx:loginId}) or a Thread Context Map pattern (%X, %mdc, or %MDC) to craft malicious input data using a JNDI Lookup pattern resulting in an information leak and remote code execution in some environments and local code execution in all environments. Log4j 2.16.0 (Java 8) and 2.12.2 (Java 7) fix this issue by removing support for message lookup patterns and disabling JNDI functionality by default. |
| CVE-2021-44228 | Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects. |
| 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. Fixed in Apache Log4j 2.12.3 and 2.13.1 |
| snakeyaml-1.25.jar | |
| CVE-2022-1471 | SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization. We recommend upgrading to version 2.0 and beyond. |
| CVE-2022-41854 | Those using Snakeyaml to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack overflow. This effect may support a denial of service attack. |
| CVE-2022-38752 | Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack-overflow. |
| CVE-2022-38751 | Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow. |
| CVE-2022-38750 | Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow. |
| CVE-2022-38749 | Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow. |
| CVE-2022-25857 | The package org.yaml:snakeyaml from 0 and before 1.31 are vulnerable to Denial of Service (DoS) due missing to nested depth limitation for collections. |
| CVE-2017-18640 | The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564. |
| CVE-2022-3064 | Parsing malicious or large YAML documents can consume excessive amounts of CPU or memory. |
| CVE-2021-4235 | Due to unbounded alias chasing, a maliciously crafted YAML file can cause the system to consume significant system resources. If parsing user input, this may be used as a denial of service vector. |
| jackson-databind-2.10.2.jar | |
| CVE-2023-35116 | \*\* DISPUTED \*\* jackson-databind through 2.15.2 allows attackers to cause a denial of service or other unspecified impact via a crafted object that uses cyclic dependencies. NOTE: the vendor's perspective is that this is not a valid vulnerability report, because the steps of constructing a cyclic data structure and trying to serialize it cannot be achieved by an external attacker. |
| CVE-2021-46877 | jackson-databind 2.10.x through 2.12.x before 2.12.6 and 2.13.x before 2.13.1 allows attackers to cause a denial of service (2 GB transient heap usage per read) in uncommon situations involving JsonNode JDK serialization. |
| CVE-2022-42004 | In FasterXML jackson-databind before 2.13.4, resource exhaustion can occur because of a lack of a check in BeanDeserializer.\_deserializeFromArray to prevent use of deeply nested arrays. An application is vulnerable only with certain customized choices for deserialization. |
| CVE-2022-42003 | In FasterXML jackson-databind before 2.14.0-rc1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP\_SINGLE\_VALUE\_ARRAYS feature is enabled. Additional fix version in 2.13.4.1 and 2.12.17.1 |
| CVE-2020-36518 | jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects. |
| CVE-2020-25649 | A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity. |
| tomcat-embed-core-9.0.30.jar | |
| CVE-2023-41080 | URL Redirection to Untrusted Site ('Open Redirect') vulnerability in FORM authentication feature Apache Tomcat.This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.0-M10, from 10.1.0-M1 through 10.0.12, from 9.0.0-M1 through 9.0.79 and from 8.5.0 through 8.5.92. The vulnerability is limited to the ROOT (default) web application. |
| CVE-2023-28708 | When using the RemoteIpFilter with requests received from a reverse proxy via HTTP that include the X-Forwarded-Proto header set to https, session cookies created by Apache Tomcat 11.0.0-M1 to 11.0.0.-M2, 10.1.0-M1 to 10.1.5, 9.0.0-M1 to 9.0.71 and 8.5.0 to 8.5.85 did not include the secure attribute. This could result in the user agent transmitting the session cookie over an insecure channel. |
| CVE-2022-42252 | If Apache Tomcat 8.5.0 to 8.5.82, 9.0.0-M1 to 9.0.67, 10.0.0-M1 to 10.0.26 or 10.1.0-M1 to 10.1.0 was configured to ignore invalid HTTP headers via setting rejectIllegalHeader to false (the default for 8.5.x only), Tomcat did not reject a request containing an invalid Content-Length header making a request smuggling attack possible if Tomcat was located behind a reverse proxy that also failed to reject the request with the invalid header. |
| CVE-2021-43980 | The simplified implementation of blocking reads and writes introduced in Tomcat 10 and back-ported to Tomcat 9.0.47 onwards exposed a long standing (but extremely hard to trigger) concurrency bug in Apache Tomcat 10.1.0 to 10.1.0-M12, 10.0.0-M1 to 10.0.18, 9.0.0-M1 to 9.0.60 and 8.5.0 to 8.5.77 that could cause client connections to share an Http11Processor instance resulting in responses, or part responses, to be received by the wrong client. |
| CVE-2022-34305 | In Apache Tomcat 10.1.0-M1 to 10.1.0-M16, 10.0.0-M1 to 10.0.22, 9.0.30 to 9.0.64 and 8.5.50 to 8.5.81 the Form authentication example in the examples web application displayed user provided data without filtering, exposing a XSS vulnerability. |
| CVE-2022-29885 | The documentation of Apache Tomcat 10.1.0-M1 to 10.1.0-M14, 10.0.0-M1 to 10.0.20, 9.0.13 to 9.0.62 and 8.5.38 to 8.5.78 for the EncryptInterceptor incorrectly stated it enabled Tomcat clustering to run over an untrusted network. This was not correct. While the EncryptInterceptor does provide confidentiality and integrity protection, it does not protect against all risks associated with running over any untrusted network, particularly DoS risks. |
| CVE-2021-41079 | Apache Tomcat 8.5.0 to 8.5.63, 9.0.0-M1 to 9.0.43 and 10.0.0-M1 to 10.0.2 did 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. |
| CVE-2021-33037 | Apache Tomcat 10.0.0-M1 to 10.0.6, 9.0.0.M1 to 9.0.46 and 8.5.0 to 8.5.66 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 honoured the identify encoding; and - Tomcat did not ensure that, if present, the chunked encoding was the final encoding. |
| CVE-2021-30640 | A vulnerability in the JNDI Realm of Apache Tomcat allows an attacker to authenticate using variations of a valid user name and/or to bypass some of the protection provided by the LockOut Realm. This issue affects Apache Tomcat 10.0.0-M1 to 10.0.5; 9.0.0.M1 to 9.0.45; 8.5.0 to 8.5.65. |
| CVE-2021-25329 | 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. |
| CVE-2021-25122 | When responding to new h2c connection requests, Apache Tomcat versions 10.0.0-M1 to 10.0.0, 9.0.0.M1 to 9.0.41 and 8.5.0 to 8.5.61 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. |
| CVE-2021-24122 | When serving resources from a network location using the NTFS file system, Apache Tomcat versions 10.0.0-M1 to 10.0.0-M9, 9.0.0.M1 to 9.0.39, 8.5.0 to 8.5.59 and 7.0.0 to 7.0.106 were susceptible to JSP source code disclosure in some configurations. The root cause was the unexpected behaviour of the JRE API File.getCanonicalPath() which in turn was caused by the inconsistent behaviour of the Windows API (FindFirstFileW) in some circumstances. |
| CVE-2020-17527 | While investigating bug 64830 it was discovered that Apache Tomcat 10.0.0-M1 to 10.0.0-M9, 9.0.0-M1 to 9.0.39 and 8.5.0 to 8.5.59 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. |
| CVE-2020-13943 | If an HTTP/2 client connecting to Apache Tomcat 10.0.0-M1 to 10.0.0-M7, 9.0.0.M1 to 9.0.37 or 8.5.0 to 8.5.57 exceeded the agreed maximum number of concurrent streams for a connection (in violation of the HTTP/2 protocol), 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. |
| CVE-2020-13935 | The payload length in a WebSocket frame was not correctly validated in Apache Tomcat 10.0.0-M1 to 10.0.0-M6, 9.0.0.M1 to 9.0.36, 8.5.0 to 8.5.56 and 7.0.27 to 7.0.104. Invalid payload lengths could trigger an infinite loop. Multiple requests with invalid payload lengths could lead to a denial of service. |
| CVE-2020-13934 | An h2c direct connection to Apache Tomcat 10.0.0-M1 to 10.0.0-M6, 9.0.0.M5 to 9.0.36 and 8.5.1 to 8.5.56 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. |
| CVE-2020-8022 | A Incorrect Default Permissions vulnerability in the packaging of tomcat on SUSE Enterprise Storage 5, SUSE Linux Enterprise Server 12-SP2-BCL, SUSE Linux Enterprise Server 12-SP2-LTSS, SUSE Linux Enterprise Server 12-SP3-BCL, SUSE Linux Enterprise Server 12-SP3-LTSS, SUSE Linux Enterprise Server 12-SP4, SUSE Linux Enterprise Server 12-SP5, SUSE Linux Enterprise Server 15-LTSS, SUSE Linux Enterprise Server for SAP 12-SP2, SUSE Linux Enterprise Server for SAP 12-SP3, SUSE Linux Enterprise Server for SAP 15, SUSE OpenStack Cloud 7, SUSE OpenStack Cloud 8, SUSE OpenStack Cloud Crowbar 8 allows local attackers to escalate from group tomcat to root. This issue affects: SUSE Enterprise Storage 5 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP2-BCL tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP2-LTSS tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP3-BCL tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP3-LTSS tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP4 tomcat versions prior to 9.0.35-3.39.1. SUSE Linux Enterprise Server 12-SP5 tomcat versions prior to 9.0.35-3.39.1. SUSE Linux Enterprise Server 15-LTSS tomcat versions prior to 9.0.35-3.57.3. SUSE Linux Enterprise Server for SAP 12-SP2 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server for SAP 12-SP3 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server for SAP 15 tomcat versions prior to 9.0.35-3.57.3. SUSE OpenStack Cloud 7 tomcat versions prior to 8.0.53-29.32.1. SUSE OpenStack Cloud 8 tomcat versions prior to 8.0.53-29.32.1. SUSE OpenStack Cloud Crowbar 8 tomcat versions prior to 8.0.53-29.32.1. |
| CVE-2020-11996 | A specially crafted sequence of HTTP/2 requests sent to Apache Tomcat 10.0.0-M1 to 10.0.0-M5, 9.0.0.M1 to 9.0.35 and 8.5.0 to 8.5.55 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. |
| CVE-2020-9484 | When using Apache Tomcat versions 10.0.0-M1 to 10.0.0-M4, 9.0.0.M1 to 9.0.34, 8.5.0 to 8.5.54 and 7.0.0 to 7.0.103 if a) an attacker is able to control the contents and name of a file on the server; and b) the server is configured to use the PersistenceManager with a FileStore; and c) the PersistenceManager is configured with sessionAttributeValueClassNameFilter="null" (the default unless a SecurityManager is used) or a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) 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 a) to d) must be true for the attack to succeed. |
| CVE-2020-1938 | When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising. In Apache Tomcat 9.0.0.M1 to 9.0.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required. This vulnerability report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - 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 (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible. It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31, 8.5.51 or 7.0.100 or later. A number of changes were made to the default AJP Connector configuration in 9.0.31 to harden the default configuration. It is likely that users upgrading to 9.0.31, 8.5.51 or 7.0.100 or later will need to make small changes to their configurations. |
| hibernate-validator-6.0.18.Final.jar | |
| CVE-2020-10693 | A flaw was found in Hibernate Validator version 6.1.2.Final. 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 (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages. |

## Mitigation Plan

|  |  |
| --- | --- |
| **VULNERABILITY CODES** | **MITIGATION** |
| bcprov-jdk15on-1.46.jar | |
| CVE-2016-1000352 | Red Hat Fuse 7.1  bouncycastle: ECIES implementation allowed the use of ECB mode (CVE-2016-1000352) |
| CVE-2016-1000346 | Red Hat Fuse 7.1  bouncycastle: Other party DH public keys are not fully validated (CVE-2016-1000346) |
| CVE-2016-1000345 | Red Hat Fuse 7.1  bouncycastle: DHIES/ECIES CBC modes are vulnerable to padding oracle attack (CVE-2016-1000345) |
| CVE-2016-1000344 | Red Hat Fuse 7.1  bouncycastle: DHIES implementation allowed the use of ECB mode (CVE-2016-1000344) |
| CVE-2016-1000343 | Red Hat Fuse 7.1  bouncycastle: DSA key pair generator generates a weak private key by default (CVE-2016-1000343) |
| CVE-2016-1000342 | Red Hat Fuse 7.1  bouncycastle: ECDSA improper validation of ASN.1 encoding of signature (CVE-2016-1000342) |
| CVE-2016-1000341 | Red Hat Fuse 7.1  bouncycastle: Information exposure in DSA signature generation via timing attack (CVE-2016-1000341) |
| CVE-2016-1000339 | Red Hat Fuse 7.1  bouncycastle: Information leak in AESFastEngine class (CVE-2016-1000339) |
| CVE-2016-1000338 | Red Hat Fuse 7.1  bouncycastle: DSA does not fully validate ASN.1 encoding during signature verification allowing for injection of unsigned data (CVE-2016-1000338) |
| CVE-2018-5382 | Red Hat Satellite 6.4 for RHEL 7 security, bug fix, and enhancement update  bouncycastle: BKS-V1 keystore files vulnerable to trivial hash collisions (CVE-2018-5382) |
| CVE-2017-13098 | Affected users and system administrators are encouraged to disable TLS RSA cyphers if possible. Please refer to your product's documentation or contact the vendor's customer service. |
| CVE-2013-1624 | release of Red Hat JBoss BPM Suite 6.0.1  It was discovered that bouncycastle leaked timing information when  decrypting TLS/SSL protocol encrypted records when CBC-mode cipher suites  were used. A remote attacker could possibly use this flaw to retrieve plain  text from the encrypted packets by using a TLS/SSL server as a padding  oracle. (CVE-2013-1624) |
| spring-boot-2.2.4.RELEASE.jar | |
| CVE-2023-20883 | Users of affected versions should apply the following mitigations:  3.0.x users should upgrade to 3.0.7+  2.7.x users should upgrade to 2.7.12+  2.6.x users should upgrade to 2.6.15+  2.5.x users should upgrade to 2.5.15+ |
| CVE-2023-20873 | Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. 2.6.x users should upgrade to 2.6.15+. 2.5.x users should upgrade to 2.5.15+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+. |
| CVE-2022-27772 | Update to version v2.2.11.RELEASE |
| logback-core-1.2.3.jar | |
| CVE-2021-42550 | Once the instance is back up visit the Cloud Manager UI to confirm that the version is now 3.9.14 or higher. |
| log4j-api-2.12.1.jar | |
| CVE-2021-44832 | Update to 2.17.1 |
| CVE-2021-45105 | Patch available |
| CVE-2021-45046 | Fixed in Log4j 2.17.1 (Java 8), 2.12.4 (Java 7) and 2.3.2 (Java 6) |
| CVE-2021-44228 | Patch available |
| CVE-2020-9488 | Upgrade to 2.13.2 which supports this feature. |
| snakeyaml-1.25.jar | |
| CVE-2022-1471 | Potential mitigations include, leveraging SnakeYaml's SafeConstructor while parsing untrusted content. |
| CVE-2022-41854 | Update to version 4.7.4 Security fix for CVE-2022-41854 |
| CVE-2022-38752 | All snakeyaml users should upgrade to the latest version |
| CVE-2022-38751 | All snakeyaml users should upgrade to the latest version |
| CVE-2022-38750 | All snakeyaml users should upgrade to the latest version |
| CVE-2022-38749 | All snakeyaml users should upgrade to the latest version |
| CVE-2022-25857 | Upgrade org.yaml:snakeyaml to version 1.31 or higher. |
| CVE-2017-18640 | All snakeyaml users should upgrade to the latest version |
| CVE-2022-3064 | Install package gopkg.in/yaml.v2. |
| CVE-2021-4235 | Install package gopkg.in/yaml.v2. |
| jackson-databind-2.10.2.jar | |
| CVE-2023-35116 | Refer to the solution of jackson-databind: Add the depth variable to record the current parsing depth. If the parsing depth exceeds a certain threshold, an exception is thrown. (fcfc499)  Refer to the GSON solution: Change the recursive processing on deeply nested arrays or JSON objects to stack+iteration processing.(（google/gson@2d01d6a20f39881c692977564c1ea591d9f39027）) |
| CVE-2021-46877 | Install v.2.13.1 |
| CVE-2022-42004 | Patch available |
| CVE-2022-42003 | Patch available |
| CVE-2020-36518 | Patch available |
| CVE-2020-25649 | Update to v.2.11.0 |
| tomcat-embed-core-9.0.30.jar | |
| CVE-2023-41080 | Users of the affected versions should apply one of the following  mitigations:  - Upgrade to Apache Tomcat 11.0.0-M11 or later  - Upgrade to Apache Tomcat 10.1.13 or later  - Upgrade to Apache Tomcat 9.0.80 or later  - Upgrade to Apache Tomcat 8.5.93 or later |
| CVE-2023-28708 | Users of the affected versions should apply one of the following  mitigations:  - Upgrade to Apache Tomcat 11.0.0-M3 or later  - Upgrade to Apache Tomcat 10.1.6 or later  - Upgrade to Apache Tomcat 9.0.72 or later  - Upgrade to Apache Tomcat 8.5.86 or later |
| CVE-2022-42252 | Users of the affected versions should apply one of the following  mitigations:  - Ensure rejectIllegalHeader is set to true  - Upgrade to Apache Tomcat 10.1.1 or later  - Upgrade to Apache Tomcat 10.0.27 or later  - Upgrade to Apache Tomcat 9.0.68 or later  - Upgrade to Apache Tomcat 8.5.83 or later |
| CVE-2021-43980 | Users of the affected versions should apply one of the following  mitigations:  - Upgrade to Apache Tomcat 10.1.0-M14 or later once released  - Upgrade to Apache Tomcat 10.0.20 or later once released  - Upgrade to Apache Tomcat 9.0.62 or later once released  - Upgrade to Apache Tomcat 8.5.78 or later once released  - Note 10.1.0-M13, 10.0.19 and 9.0.61 were not released |
| CVE-2022-34305 | Users of the affected versions should apply one of the following  mitigations:  - Remove the examples web application as documented in the Tomcat  security guide  - Upgrade to Apache Tomcat 10.1.0-M17 or later once released  - Upgrade to Apache Tomcat 10.0.23 or later once released  - Upgrade to Apache Tomcat 9.0.65 or later once released  - Upgrade to Apache Tomcat 8.5.82 or later once released |
| CVE-2022-29885 | Users running clustering over an untrusted network who require full  protection should switch to an alternative solution such as running the  clustering communication over a VPN. |
| CVE-2021-41079 | Users of the affected versions should apply one of the following  mitigations:  - Upgrade to Apache Tomcat 10.1.0-M6 or later  - Upgrade to Apache Tomcat 10.0.12 or later  - Upgrade to Apache Tomcat 9.0.54 or later  - Upgrade to Apache Tomcat 8.5.72 or later |
| CVE-2021-33037 | Users of the affected versions should apply one of the following  mitigations:  - Upgrade to Apache Tomcat 10.0.7 or later  - Upgrade to Apache Tomcat 9.0.48 or later  - Upgrade to Apache Tomcat 8.5.68 or later  Note that issue was fixed in 9.0.47 and 8.5.67 but the release votes for  those versions did not pass. |
| CVE-2021-30640 | Users of the affected versions should apply one of the following  mitigations:  - Upgrade to Apache Tomcat 10.0.6 or later  - Upgrade to Apache Tomcat 9.0.46 or later  - Upgrade to Apache Tomcat 8.5.66 or later  - Upgrade to Apache Tomcat 7.0.109 or later |
| CVE-2021-25329 | tomcat/site/trunk/docs/security-10.html  tomcat/site/trunk/docs/security-7.html  tomcat/site/trunk/docs/security-8.html  tomcat/site/trunk/docs/security-9.html  tomcat/site/trunk/xdocs/security-10.xml  tomcat/site/trunk/xdocs/security-7.xml  tomcat/site/trunk/xdocs/security-8.xml  tomcat/site/trunk/xdocs/security-9.xml |
| CVE-2021-25122 | tomcat/site/trunk/docs/security-10.html  tomcat/site/trunk/docs/security-7.html  tomcat/site/trunk/docs/security-8.html  tomcat/site/trunk/docs/security-9.html  tomcat/site/trunk/xdocs/security-10.xml  tomcat/site/trunk/xdocs/security-7.xml  tomcat/site/trunk/xdocs/security-8.xml  tomcat/site/trunk/xdocs/security-9.xml |
| CVE-2021-24122 | Users of the affected versions should apply one of the following  mitigations:  - Upgrade to Apache Tomcat 10.0.0-M10 or later  - Upgrade to Apache Tomcat 9.0.40 or later  - Upgrade to Apache Tomcat 8.5.60 or later  - Upgrade to Apache Tomcat 7.0.107 or later |
| CVE-2020-17527 | - Upgrade to Apache Tomcat 10.0.0-M10 or later  - Upgrade to Apache Tomcat 9.0.40 or later  - Upgrade to Apache Tomcat 8.5.60 or later |
| CVE-2020-13943 | - Upgrade to Apache Tomcat 10.0.0-M8 or later  - Upgrade to Apache Tomcat 9.0.38 or later  - Upgrade to Apache Tomcat 8.5.58 or later |
| CVE-2020-13935 | - Upgrade to Apache Tomcat 10.0.0-M7 or later  - Upgrade to Apache Tomcat 9.0.37 or later  - Upgrade to Apache Tomcat 8.5.57 or later |
| CVE-2020-13934 | - Upgrade to Apache Tomcat 10.0.0-M7 or later  - Upgrade to Apache Tomcat 9.0.37 or later  - Upgrade to Apache Tomcat 8.5.57 or later |
| CVE-2020-8022 | Update to tomcat-9.0.35-lp151.3.21.1 |
| CVE-2020-11996 | Upgrade Tomcat to 9.0.36 |
| CVE-2020-9484 | - Upgrade to Apache Tomcat 10.0.0-M5 or later  - Upgrade to Apache Tomcat 9.0.35 or later  - Upgrade to Apache Tomcat 8.5.55 or later  - Upgrade to Apache Tomcat 7.0.104 or later  Alternatively, users may configure the PersistenceManager with an  appropriate value for sessionAttributeValueClassNameFilter to ensure  that only application provided attributes are serialized and deserialized. |
| CVE-2020-1938 | tomcat/site/trunk/docs/security-10.html  tomcat/site/trunk/docs/security-8.html  tomcat/site/trunk/docs/security-9.html  tomcat/site/trunk/xdocs/security-10.xml  tomcat/site/trunk/xdocs/security-8.xml  tomcat/site/trunk/xdocs/security-9.xml |
| hibernate-validator-6.0.18.Final.jar | |
| CVE-2020-10693 | Update to v.6.0.20.Final |