****

# 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** | **11/10/2022** | **Brandie Sheppard** |  |

## 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

Brandie Sheppard

## Interpreting Client Needs

By using secure communication platforms, you can not only protect your data from being compromised, but it also minimizes your risks of a data breach, protects your company’s reputation and finances. Being that Global Rain operates internationally, they are even more susceptible to cyber-attacks. The government encourages companies to have secure communication to help protect the user’s data as well as their privacy, so there are no restrictions on secure communication. Being that the computer, internet, and technology as a whole is constantly changing, as well as hackers just keep getting better and more advanced, there are always new threats coming and going, so the company’s data is always at risk. Some modernization requirements to consider might be having an IT analyst, or team, always staying on top of the security advancements, cloud computing, and Agile development.

## Areas of Security

API – Need to make sure the APIs are all up to date so they’re not open to being compromised or leaving any backdoors open.

Letters – Using the API to ensure confidentiality to all users as well as the clients.

Client/Server – We are going to need to make sure the correct certificates are being used if we want to implement API access.

Verification of entry – We will need to make sure that no more than half of the ASVS can be completed without human assistance, finding flaws in business logic and access control shouldn’t be left to automated tools alone.

Error codes – Having a “Second Person Verification” process could help with error codes, and not leaving any windows open to attack. The second pair of eyes look over the code and make sure there aren’t any mistakes or weaknesses left open.

Encapsulation – Ensuring we use it correctly and not use it in the data format, will help protect the system if our command operation needs access to data on the system itself.

## Manual Review

In the pom.xml, we need to update to the current version to make sure we have all the latest upgrades and security. We should also edit the length of the string in GreetingController.java to avoid any errors in our system later, or any tampering with the format.

## Static Testing

|  |  |
| --- | --- |
| [CVE-2020-10693](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2020-25649](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2020-36518](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-42003](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-42004](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-9488](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 |
| [CVE-2019-17571](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17571) | Included in Log4j 1.2 is a SocketServer class that is vulnerable to deserialization of untrusted data which can be exploited to remotely execute arbitrary code when combined with a deserialization gadget when listening to untrusted network traffic for log data. This affects Log4j versions up to 1.2 up to 1.2.17. |
| [CVE-2021-42550](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2017-18640](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-25857](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-38749](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-38751](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-38752](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-38750](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-27772](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2022-22965](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22965) | A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it. |
| [CVE-2021-22118](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22118) | In Spring Framework, versions 5.2.x prior to 5.2.15 and versions 5.3.x prior to 5.3.7, a WebFlux application is vulnerable to a privilege escalation: by (re)creating the temporary storage directory, a locally authenticated malicious user can read or modify files that have been uploaded to the WebFlux application, or overwrite arbitrary files with multipart request data. |
| [CVE-2020-5421](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5421) | In Spring Framework versions 5.2.0 - 5.2.8, 5.1.0 - 5.1.17, 5.0.0 - 5.0.18, 4.3.0 - 4.3.28, and older unsupported versions, the protections against RFD attacks from CVE-2015-5211 may be bypassed depending on the browser used through the use of a jsessionid path parameter. |
| [CVE-2020-5398](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5398) | In Spring Framework, versions 5.2.x prior to 5.2.3, versions 5.1.x prior to 5.1.13, and versions 5.0.x prior to 5.0.16, an application is vulnerable to a reflected file download (RFD) attack when it sets a "Content-Disposition" header in the response where the filename attribute is derived from user supplied input. |
| [CVE-2022-22950](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22950) | n Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition. |
| [CVE-2022-22971](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22971) | In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user. |
| [CVE-2022-22968](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22968) | In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path. |
| [CVE-2022-22970](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22970) | In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object. |
| [CVE-2021-22060](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22060) | In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase. |
| [CVE-2021-22096](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22096) | In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. |
| [CVE-2021-22060](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22060) | In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase. |
| [CVE-2016-1000027](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-1000027) | Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data. |
| [CVE-2022-22965](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22965) | A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it. |
| [CVE-2021-22118](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22118) | In Spring Framework, versions 5.2.x prior to 5.2.15 and versions 5.3.x prior to 5.3.7, a WebFlux application is vulnerable to a privilege escalation: by (re)creating the temporary storage directory, a locally authenticated malicious user can read or modify files that have been uploaded to the WebFlux application, or overwrite arbitrary files with multipart request data. |
| [CVE-2020-5421](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5421) | In Spring Framework versions 5.2.0 - 5.2.8, 5.1.0 - 5.1.17, 5.0.0 - 5.0.18, 4.3.0 - 4.3.28, and older unsupported versions, the protections against RFD attacks from CVE-2015-5211 may be bypassed depending on the browser used through the use of a jsessionid path parameter. |
| [CVE-2020-5398](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5398) | In Spring Framework, versions 5.2.x prior to 5.2.3, versions 5.1.x prior to 5.1.13, and versions 5.0.x prior to 5.0.16, an application is vulnerable to a reflected file download (RFD) attack when it sets a "Content-Disposition" header in the response where the filename attribute is derived from user supplied input. |
| [CVE-2022-22950](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22950) | n Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition. |
| [CVE-2022-22971](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22971) | In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user. |
| [CVE-2022-22968](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22968) | In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path. |
| [CVE-2022-22970](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-22970) | In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object. |
| [CVE-2021-22060](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22060) | In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase. |
| [CVE-2021-22096](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22096) | In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. |
| [CVE-2021-22060](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22060) | In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase. |
| [CVE-2020-1938](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2020-1928](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1928) | An information disclosure vulnerability was found in Apache NiFi 1.10.0. The sensitive parameter parser would log parsed values for debugging purposes. This would expose literal values entered in a sensitive property when no parameter was present. |
| [CVE-2020-10569](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-10569) | \*\* DISPUTED \*\* SysAid On-Premise 20.1.11, by default, allows the AJP protocol port, which is vulnerable to a GhostCat attack. Additionally, it allows unauthenticated access to upload files, which can be used to execute commands on the system by chaining it with a GhostCat attack. NOTE: This may be a duplicate of CVE-2020-1938. |
| [CVE-2020-1938](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2020-11996](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-13934](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-14384](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-14384) | A flaw was found in JBossWeb in versions before 7.5.31.Final-redhat-3. The fix for CVE-2020-13935 was incomplete in JBossWeb, leaving it vulnerable to a denial of service attack when sending multiple requests with invalid payload length in a WebSocket frame. The highest threat from this vulnerability is to system availability. |
| [CVE-2020-13935](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-17527](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-25122](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-41079](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-29885](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-42252](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-42252) | If Apache Tomcat 8.5.0 to 8.5.52, 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. |

## Search Results

There are **4** CVE Records that match your search.

|  |  |
| --- | --- |
| **Name** | **Description** |
| [CVE-2022-23181](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-23181) | The fix for bug CVE-2020-9484 introduced a time of check, time of use vulnerability into Apache Tomcat 10.1.0-M1 to 10.1.0-M8, 10.0.0-M5 to 10.0.14, 9.0.35 to 9.0.56 and 8.5.55 to 8.5.73 that allowed a local attacker to perform actions with the privileges of the user that the Tomcat process is using. This issue is only exploitable when Tomcat is configured to persist sessions using the FileStore. |
| [CVE-2021-25329](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-9494](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-9494) | Apache Traffic Server 6.0.0 to 6.2.3, 7.0.0 to 7.1.10, and 8.0.0 to 8.0.7 is vulnerable to certain types of HTTP/2 HEADERS frames that can cause the server to allocate a large amount of memory and spin the thread. |
| [CVE-2020-9484](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 Security Manager 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-2021-25329](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-9494](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-9494) | Apache Traffic Server 6.0.0 to 6.2.3, 7.0.0 to 7.1.10, and 8.0.0 to 8.0.7 is vulnerable to certain types of HTTP/2 HEADERS frames that can cause the server to allocate a large amount of memory and spin the thread. |
| [CVE-2020-9484](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 Security Manager 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-2021-30640](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-34305](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-24122](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 behavior of the JRE API File.getCanonicalPath() which in turn was caused by the inconsistent behavior of the Windows API (FindFirstFileW) in some circumstances. |
| [CVE-2021-33037](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 honored the identify encoding; and - Tomcat did not ensure that, if present, the chunked encoding was the final encoding. |
| [CVE-2020-1935](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1935) | In Apache Tomcat 9.0.0.M1 to 9.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99 the HTTP header parsing code used an approach to end-of-line parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2019-17569](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17569) | The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2020-1935](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1935) | In Apache Tomcat 9.0.0.M1 to 9.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99 the HTTP header parsing code used an approach to end-of-line parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2019-17569](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17569) | The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2020-13943](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-43980](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-1938](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2020-1928](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1928) | An information disclosure vulnerability was found in Apache NiFi 1.10.0. The sensitive parameter parser would log parsed values for debugging purposes. This would expose literal values entered in a sensitive property when no parameter was present. |
| [CVE-2020-10569](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-10569) | \*\* DISPUTED \*\* SysAid On-Premise 20.1.11, by default, allows the AJP protocol port, which is vulnerable to a GhostCat attack. Additionally, it allows unauthenticated access to upload files, which can be used to execute commands on the system by chaining it with a GhostCat attack. NOTE: This may be a duplicate of CVE-2020-1938. |
| [CVE-2020-8022](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-8022) | An 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-2020-1938](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |
| [CVE-2020-11996](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-13934](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-14384](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-14384) | A flaw was found in JBossWeb in versions before 7.5.31.Final-redhat-3. The fix for CVE-2020-13935 was incomplete in JBossWeb, leaving it vulnerable to a denial of service attack when sending multiple requests with invalid payload length in a WebSocket frame. The highest threat from this vulnerability is to system availability. |
| [CVE-2020-13935](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-17527](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-25122](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-41079](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-29885](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-42252](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-42252) | If Apache Tomcat 8.5.0 to 8.5.52, 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-2022-23181](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-23181) | The fix for bug CVE-2020-9484 introduced a time of check, time of use vulnerability into Apache Tomcat 10.1.0-M1 to 10.1.0-M8, 10.0.0-M5 to 10.0.14, 9.0.35 to 9.0.56 and 8.5.55 to 8.5.73 that allowed a local attacker to perform actions with the privileges of the user that the Tomcat process is using. This issue is only exploitable when Tomcat is configured to persist sessions using the FileStore. |
| [CVE-2021-25329](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-9494](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-9494) | Apache Traffic Server 6.0.0 to 6.2.3, 7.0.0 to 7.1.10, and 8.0.0 to 8.0.7 is vulnerable to certain types of HTTP/2 HEADERS frames that can cause the server to allocate a large amount of memory and spin the thread. |
| [CVE-2020-9484](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 Security Manager 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-2021-25329](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2020-9494](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-9494) | Apache Traffic Server 6.0.0 to 6.2.3, 7.0.0 to 7.1.10, and 8.0.0 to 8.0.7 is vulnerable to certain types of HTTP/2 HEADERS frames that can cause the server to allocate a large amount of memory and spin the thread. |
| [CVE-2020-9484](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 Security Manager 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-2021-30640](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2022-34305](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-24122](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 behavior of the JRE API File.getCanonicalPath() which in turn was caused by the inconsistent behavior of the Windows API (FindFirstFileW) in some circumstances. |

|  |  |
| --- | --- |
| [CVE-2021-33037](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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 honored the identify encoding; and - Tomcat did not ensure that, if present, the chunked encoding was the final encoding. |
| [CVE-2020-1935](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1935) | In Apache Tomcat 9.0.0.M1 to 9.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99 the HTTP header parsing code used an approach to end-of-line parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2019-17569](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17569) | The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2020-1935](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1935) | In Apache Tomcat 9.0.0.M1 to 9.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99 the HTTP header parsing code used an approach to end-of-line parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2019-17569](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17569) | The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely. |
| [CVE-2020-13943](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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-2021-43980](https://cve.mitre.org/cgi-bin/cvename.cgi?name=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. |

## Mitigation Plan

* Update the data to the latest version
* Add input validation on the GreetingController input
* Add a library for parsing input to control string pattern errors
* Consider a “Second Person Verification”
* Add an IT team to help ensure we are within ASVS for secure verifications and the two-person code verification.