Office of
Information Security
Securing One HHS

Health Sector Cybersecurity
Coordination Center

HC3: Sector Alert

September 4, 2024 TLP:CLEAR Report: 202409041500

Apache Tomcat Vulnerabilities

Executive Summary

Tomcat is one of the most popular and widely-deployed web servers and Java-based application servers in the world, heavily leveraged by the U.S. health sector. Like any other software platform, vulnerabilities in Tomcat are constantly being discovered that can make it open to exploitation by cyberattack. Due to its functionality, it is usually exposed directly to the Internet, making it accessible to countless threat actors. This bulletin will provide an overview of Apache Tomcat vulnerabilities, as well as mitigation strategies and an overall approach to keeping it secure.

Tomcat Security Overview

Tomcat is an open-source web server maintained by the nonprofit Apache Corporation. It is often used for hosting electronic health record (HER) systems, running health information exchange (HIE) systems, hosting laboratory information management systems, hosting and running custom healthcare applications, and supporting telemedicine applications, among other functions. Because Tomcat is so frequently deployed, it has attracted the attention of threat actors.

Historically Common Tomcat Vulnerability Categories

As Apache Tomcat is both a commonly deployed platform around the world, and its functionality ensures that it is deployed in a way that makes it Internet accessible, it has drawn the attention of vulnerability researchers and cyber threat actors. As a result, it is not uncommon for vulnerabilities in it to be identified and exploited. Historically, there are categories of vulnerabilities that are most commonly found in Tomcat, and those are listed below with examples. The vulnerabilities below are historic vulnerabilities and should have already been mitigated by vulnerable organizations; our purpose for reviewing them is to demonstrate that they represent some of the historically common Tomcat vulnerabilities.

REMOTE CODE EXECUTION

Remote code execution vulnerabilities allow attackers to execute code on the victim system, which in effect allows them to extract any data from it, distrupt its operations, or use it as a staging point for further cyberattacks. Remote code execution attacks are particularly egregious, as they can lead to complete system compromise. This is identified by MITRE ID TA0002.

Examples:

- CVE-2017-12617: A vulnerability in Tomcat (applicable to versions 9.0.0.M1 to 9.0.0, 8.5.0 to 8.5.22, 8.0.0.RC1 to 8.0.46 and 7.0.0 to 7.0.81) that allows remote code execution by uploading a JSP file via a crafted HTTP request if HTTP PUTs are enabled.
- CVE-2020-9484: A vulnerability in Tomcat (applicable to 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) that allows remote code execution. Exploitation of this vulnerability requires that all four of the below conditions be true:
  1. The attacker must be able to control the contents and name of a file on the server.
  2. The server must be configured to use the PersistenceManager with a FileStore.
  3. PersistenceManager is configured with sessionAttributeValueClassNameFilter="null", which is the default unless a SecurityManager is used, or a sufficiently lax filter allows the attacker-provided object to be deserialized.
  4. The attacker identifies the relative file path from the FileStore storage location to the attacker-

controlled file, and subsequently leveraging a crafted request, the attacker triggers the remote code execution via deserialization of the file under their control.

INFORMATION DISCLOSURE

These vulnerabilities result in the exposure of sensitive information to unauthorized parties. This information can feed the reconaissance process, allowing an attacker to gather further information on the infrastructure they are targeting, and continue launching further stages of their attack. This is identified by MITRE ID TA1426.

Examples:

- CVE-2023-28708: A vulnerability in Tomcat (applicable to Apache Tomcat versions 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) which can allow for the passing of a session cookie over an insecure channel.
- CVE-2017-12616: An information disclosure vulnerability (applicable to Apache Tomcat versions 7.0.0 to 7.0.80) in the way Tomcat handles certain requests. Exploitation of this vulnerability potentially allows an attacker to bypass security constraints and/or view the source code of JSPs for resources served by the VirtualDirContext using a specially crafted request.

CROSS-SITE SCRIPTING (XSS)

Cross-site scripting vulnerabilities allow attackers to insert malicious scripts into legitimate web pages that executed when a victim surfs to the website. Because Apache Tomcat is first and foremost a webserver, special attention should be paid to the potential for cross-site scripting attacks and vulnerabilities as they are released. An older Apache information page is here, and the Open Web Application Security Project has a Cross-Site Scripting page here, which includes several resources. Cross-site scripting is applicable to a number of elements in the MITRE framework, depending on the context and details of the attack, including the following tactics: MITRE ID TA0001, MITRE ID TA0002, MITRE ID TA0003, MITRE ID TA0010, and MITRE ID TA0040.

Examples:

- CVE-2016-6816: A vulnerability in the Apache Tomcat Manager (Applicable to Apache Tomcat versions 9.0.0.M1 to 9.0.0.M11, 8.5.0 to 8.5.6, 8.0.0.RC1 to 8.0.38, 7.0.0 to 7.0.72, and 6.0.0 to 6.0.47) that could lead to the poisioning of a web cache, a cross-site scripting attack and/or the exfiltration of sensitive information.
- CVE-2022-34305: A vulnerability in Apache Tomcat (Applicable to Apache Tomcat versions 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) that can display user-provided data without filtering when exploited.

DENIAL OF SERVICE (DoS)

A denial-of-service vulnerability exposes a victim system to attempts to flood it with illgitimate traffic, which by overwhelming it, can then render intended services unavailable to legitimate users. This is identified by MITRE ID T1499.

Examples:

- CVE-2020-11996: A vulnerability in Apache Tomcat (Applicable to versions 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) which, if triggered with a specially crafted sequence of

HTTP/2 requests, could prompt a spike in CPU usage and ultimately render the victim system unresponsive.
- CVE-2023-24998: Apache Commons FileUpload—which is a component that allows for multipart file upload functionality to servlets and web applications—does not limit the number of request parts to be processed. This applies to all versions of FileUpload before 1.5, and if exploited, can result in the triggering a denial-of-service with a malicious upload.

INSECURE DESERIALIZATION

An insecure deserialization vulnerability allows an attacker to potentially manipulate or craft malicious serialized data in such a way that an application will deserialize and process, potentially leading to unauthorized actions or code execution. It can allow untrusted data to be used to instantiate objects. Insecure deserialization is applicable to a number of elements in the MITRE framework, depending on the context and details of the attack, including and of the following tactics: MITRE ID TA0001, MITRE ID TA0003, MITRE ID TA0002, MITRE ID TA0004, MITRE ID TA0005, MITRE ID TA0011 and MITRE ID TA0040.

Examples:

- CVE-2019-0232: Insecure deserialization vulnerability in Apache Tomcat (applicable to Tomcat versions 9.0.0.M1 to 9.0.17, 8.5.0 to 8.5.39 and 7.0.0 to 7.0.93) related to the the Java Runtime Environment passes command line arguments to Windows that could be exploited for the remote execution of code, depending on the specific configuraiton. Exploitation of this vulnerability requires the target system to be running on Windows with enableCmdLineArguments enabled.
- CVE-2020-9484: This vulnerability was detailed above, in the Remote Code Execution section. Exploitation is possible due to the insufficient validation of a cached session file before deserialization, also making it an insecure deserialization vulnerability.

SECURITY MISCONFIGURATIONS

Security misconfigurations occur when any aspect of an enterprise infrastructure, including servers, databases, and applications, are not securely configured, leading to potential exploitation. This can also include both infrastructurew—which is on-prem—as well as outsourced cloud services.

Examples:

- CVE-2018-8014: This is a vulnerability in Apache Tomcat (applicable to Tomcat versions 9.0.0.M1 to 9.0.8, 8.5.0 to 8.5.31, 8.0.0.RC1 to 8.0.52, 7.0.41 to 7.0.88) due to the CORS filter default setting enabling 'supportsCredentials'. The CORS (Cross-Origin Resource Sharing) filter in Apache Tomcat is a component designed to handle headers for web applications running. CORS is a security feature implemented by many web browsers, used to restrict the way web pages can make requests to a different domain than the one that served the web page. It is particularly important for preventing malicious websites from making unauthorized requests to a server.
- CVE-2018-8034: This is a vulnerability in Apache Tomcat (applicable to Tomcat versions 9.0.0.M1 to 9.0.9, 8.5.0 to 8.5.31, 8.0.0.RC1 to 8.0.52, and 7.0.35 to 7.0.88) associated with a missing host name verification when using TLS with the WebSocket client. After patching, host name verification is enabled by default.

SESSION FIXATION

These vulnerabilities occur when an attacker is able to find or set a user's session identifier, allowing them to hijack the user's session.

Examples:

- CVE-2008-5515: This vulnerability in Apache Tomcat (applicable to Tomcat versions 4.1.0 through 4.1.39, 5.5.0 through 5.5.27, 6.0.0 through 6.0.18, and possibly unidentified earlier versions as well) normalizes the target pathname before filtering the query string when using the RequestDispatcher method. If exploited, an attacker may bypass access restrictions and execute a directory traversal attack, allowing them to access sensitive areas of the directory structure.
- CVE-2015-5346: This vulnerability in Apache Tomcat (applicable to Tomcat versions 7.0.0 through 7.0.20, 6.0.0 through 6.0.33, 5.5.0 through 5.5.33, and possibly unidentified versions as well) which can allow for remote attackers to hijack web sessions by leveraging use of a requestedSessionSSL field for an unintended request.

DIRECTORY TRAVERSAL

Directory traversal vulnerabilities enable attackers to access files and directories outside any intended directory. Often, exploitation of a directory traversal vulnerability leads to unintended information disclosure.

Examples:

- CVE-2011-3190: This vulnerability in Apache Tomcat (applicable to Tomcat versions 7.0.0 through 7.0.20, 6.0.0 through 6.0.33, 5.5.0 through 5.5.33, and possibly unidentified versions as well) potentially allows a remote, unauthenticated attacker access to sensitive information. This vulneraiblity is enabled by the spoofing of an AJP request, which is the Apache JServ Protocol, a protocol which can serve as a proxy to inbound requests from a webserver.
- CVE-2021-30640: This vulnerability in Apache Tomcat (applicable to Tomcat versions 10.0.0-M1 to 10.0.5; 9.0.0.M1 to 9.0.45; 8.5.0 to 8.5.65) is specifically in the Java Naming and Directory Interface (JNDI), which is a Java application programming interface that facilitates requests from Java clients for data resources.

Resources

We recommend the following general resources for securing Apache Tomcat vulnerabilities:

Apache Foundation: Security Resources and Vulnerability Reporting
https://www.apache.org/security/

University of California Lawrence Berkeley National Laboratory: Securing Apache Web Servers
https://commons.lbl.gov/display/cpp/Securing+Apache+Web+Servers

Cybersecurity & Infrastructure Security Agency: Cyber Threats and Advisories
https://www.cisa.gov/topics/cyber-threats-and-advisories

NIST: Publications
https://www.nist.gov/publications

Tomcat Users Mailing Lists
https://tomcat.apache.org/lists.html

Apache Software Foundation: Security Considerations for Apache Tomcat 7
https://tomcat.apache.org/tomcat-7.0-doc/security-howto.html

Apache Software Foundation: Security Considerations for Apache Tomcat 8
https://tomcat.apache.org/tomcat-8.0-doc/security-howto.html

Apache Software Foundation: Security Considerations for Apache Tomcat 9
https://tomcat.apache.org/tomcat-9.0-doc/security-howto.html

Red Hat Linux: Securing the Apache HTTP Server
https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/security_guide/sect-security_guide-server_security-securing_the_apache_http_server

References

Apache Software Foundation
https://www.apache.org/

Apache Tomcat homepage
https://tomcat.apache.org/

Contact Information

If you have any additional questions, we encourage you to contact us at HC3@hhs.gov.

We want to know how satisfied you are with the resources HC3 provides. Your answers will be anonymous, and we will use the responses to improve all future updates, features, and distributions. Share Your Feedback