**JENKINS**

**SECURITY-3291 / CVE-2023-36478 and CVE-2023-44487**

**Vulnerability Assessment**

**Author**

Mahammad Azeem

# **DESCRIPTIONS**

HTTP/2 denial of service vulnerabilities in bundled Jetty

**Severity (CVSS):** High

Jenkins has a built-in [command line interface (CLI)](https://www.jenkins.io/doc/book/managing/cli/) to access Jenkins from a script or shell environment.

Jenkins bundles Winstone-Jetty, a wrapper around Jetty, to act as HTTP and servlet server when started using java -jar jenkins.war. This is how Jenkins is run when using any of the installers or packages, but not when run using servlet containers such as Tomcat.

Jenkins 2.427 and earlier, LTS 2.414.2 and earlier bundles versions of Jetty affected by the security vulnerabilities [CVE-2023-36478](https://www.cve.org/CVERecord?id=CVE-2023-36478) and [CVE-2023-44487](https://www.cve.org/CVERecord?id=CVE-2023-44487). These vulnerabilities allow unauthenticated attackers to cause a denial of service.

Note:  
This only affects instances that enable HTTP/2, typically using the --http2Port argument to java -jar jenkins.war or corresponding options in service configuration files. It is disabled by default in all native installers and the Docker images provided by the Jenkins project.

Jenkins 2.428, LTS 2.414.3 updates the bundled Jetty to version 10.0.17, which is unaffected by these issues.

Administrators unable to update to these releases of Jenkins (or newer) are advised to disable HTTP/2.

**Affected Versions:**

* Jenkins weekly up to and including 2.427
* Jenkins LTS up to and including 2.414.2

**Mitigations:**

<https://www.jenkins.io/security/advisory/2023-10-18/>

* Jenkins weekly should be updated to version 2.428
* Jenkins LTS should be updated to version 2.414.3

Planning to upgrade to latest Jenkins LTS version 2.452.1 as the latest LTS contains fix for all security vulnerabilities as of the date it was released (15 May 2024).

**How to apply fix:**

Upgrading current version of Jenkins from 2.387.3 (Source Jenkins version) to LTS 2.452.1 (target Jenkins version)

(Confirmation:

<https://community.jenkins.io/t/upgrading-jenkins-from-lts-2-387-3-to-lts-2-426-3/14606/2>

<https://community.jenkins.io/t/upgrade-steps-for-jenkins-from-2-387-x-to-2-4410-x/14771>)

**Pre-Requisites:**

Step 1. Check for plugin compatibility:

Performed assessment on the list of plugins currently installed in GCC Jenkins need update as per going through the Jenkins LTS changelog from LTS version 2.387.3 to 2.452.1

List of plugins that require update are shared separately.

NOTE:

$JENKINS\_HOME variable value is different for Trial upgrade and actual upgrade steps.

Step 2. Backup:

* Backup the existing $JENKINS\_HOME for both Jenkins master (D:\DevOps\Jenkins) and slave (VMSS) (D:\DevOps)
* Take backup of “D:\script\” folder in Jenkins Slave
* Take VM level backup for both Jenkins master/slave from Azure Portal.
* Backup existing “Jenkins.war” file located D:\DevOps\Jenkins-exec\

Step 3. Download installers/plugins

* Download the Jenkins LTS 2.452.1 war <https://get.jenkins.io/war-stable/2.452.1/jenkins.war>

and transfer to Jenkins master

* Download all required versions of plugins from the urls indicated in the plugin.txt attachment file.
* Make sure to transfer all these files to GCC, get CPF devops approve/scan and transfer to Jenkins controller VM. (under \\shared-drive\DevOps\installers)

Step 4. Conduct the upgrade on a staging environment. (To arrest any issues that may arise due to plugins compatibility, maven project type builds etc)

* Upgrade to be conducted on existing Jenkins controller (Suggestion is to conduct the trial upgrade on some other VM without touching the existing Production Jenkins controller)
* For trial upgrade first, Install/configure to replicate the similar Jenkins setup viz, version(2.387.3), plugins, configurations etc as current GCC Jenkins controller
* Download the Jenkins LTS 2.387.3 installer (.msi) and download or copy all plugins from existing Jenkins controller’s $JENKINS\_HOME\plugins to a temporary folder ([\\K\DevOps\DUMMY\Jenkins\](file:///\\K\DevOps\DUMMY\Jenkins\)plugins).
* Run the installer and configure Jenkins to install on a location (viz., [\\K\DevOps\DUMMY\Jenkins\](file:///\\K\DevOps\DUMMY\Jenkins\)) and run the service on port 8085 (any location/port other then currently running prod Jenkins service)
* After successful installation, login and verify version, plugins, configurations etc
* Now, configure the Jenkins agent/slave to connect to controller by running the Jenkins java jnlp command. Make sure to set the working directory to “D:\DevOps\DUMMY\Jenkins” to avoid conflicting with already running production Jenkins agent/VMSS.
* Verify that, this Jenkins installation version/plugins is same as the actual production GCC Jenkins service.
* Configure some Jenkins jobs/Run some builds to verify further on the setup.
* Once verified the above, proceed to perform the upgrade as per the steps as indicated in below “**Upgrade Jenkins**” section (take note $JENKINS\_HOME will be [\\K\DevOps\DUMMY\Jenkins\](file:///\\K\DevOps\DUMMY\Jenkins\) for this trial upgrade)

Note:

In Jenkins LTS Version 2.452.1, Non-Pipeline builds interrupted by a controller restart will now be marked as aborted rather than failed

Step 5. Make announcement across the board regarding Jenkins upgrade for teams to plan their activities well in advance.

**Upgrade Jenkins:**

Step 1: Check if there are any active/running Jenkins builds. Stop all running builds. Do not proceed to next steps unless all active/running Jenkins builds are stopped.

Step 2: Disconnect/Disable Jenkins slave and Stop currently running Jenkins service (Windows search > Services > Look for Jenkins service > Right click “stop”)

Step 3. Login to Jenkins controller VM, take backup and remove the old jenkins.war file located at $JENKINS\_HOME and replace it with the latest downloaded war file in Pre-requisites Step 3.

NOTE:

$JENKINS\_HOME variable value is different for Trial upgrade and actual upgrade steps.

Step 4: Login to Jenkins controller VM, Backup “$JENKINS\_HOME\plugins” folder and transfer all plugins downloaded in Pre-requisites Step 3) to “$JENKINS\_HOME\plugins”

Step 5. Upgrade to the latest version of remoting in Jenkins slave/agent connectivity command/script

Login to agent VM and update the Jenkins jnlp command/script to use latest remoting version:

The -jnlpUrl ${JENKINS\_URL}/manage/computer/${AGENT\_NAME}/jenkins-agent.jnlp argument to the agent JAR has been deprecated; use -url ${JENKINS\_URL} and -name ${AGENT\_NAME} instead, in Jenkins master-Slave agent script (in “D:\script\create-and-connect-Jenkins-Slave.bat”)

Step 6: Wait for Jenkins application to launch, log in to Jenkins with your existing credentials and verify if Jenkins controller running is online and agent is connected/online.

**Post Upgrade activities:**

Step 1. Login to Jenkins url and verify if Jenkins agent is connected and accepting jobs to execute.

Step 2. Verify all Jenkins configurations viz., environment variables, global security, nodes, plugins, tools installations, ssh servers entries etc

Note: For below Jenkins jobs verification,

Non-Pipeline builds interrupted by a controller restart will now be marked as aborted rather than failed

Step 3. List down and verify all Jenkins non-maven jobs

Step 4. List down and verify all Jenkins maven jobs

Step 5. List down and verify all Infra Jenkins jobs

Step 6. List down and verify all DevOps automation jobs (VMSS m2 folder copy job)

**Further more on the changes in the upgraded Jenkins LTS 2.452.1**

* Remove the People view. Administrators can install the new People View plugin to restore this functionality. (<https://issues.jenkins.io/browse/JENKINS-18884>)

(People View plugin is considered and installed as part of jenkins upgrade assessment)

* Add a "copy to clipboard" button to the build console output
* Modernize progress bar UI in various locations
* Add a computer icon legend and a new icon for agents that are not accepting tasks
* Non-Pipeline builds interrupted by a controller restart will now be marked as aborted rather than failed
* Create an index page for heap dump creation
* Show an error message in progressive logs on 4xx status codes

Reference:

<https://www.jenkins.io/changelog-stable/>