**JENKINS**

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

**Vulnerability Assessment**

**Author**

Mahammad Azeem

# **DESCRIPTIONS**

HTTP/2 denial of service vulnerabilities in bundled Jetty

**Severity (CVSS):** High

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. This allows attackers to read arbitrary files on the Jenkins controller file system using the default character encoding of the Jenkins controller process.

Jenkins 2.427 and earlier, LTS 2.414.2 and earlier bundles versions of Jetty affected by the security vulnerabilities CVE-2023-36478 and 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.

**Mitigations:**

* Jenkins 2.428, LTS 2.414.3 updates the bundled Jetty to version 10.0.17, which is unaffected by these issues.
* Upgrade current Jenkins to LTS 2.414.3
* Jenkins Administrators unable to update to these releases of Jenkins (or newer) are advised to disable HTTP/2.

**How to apply fix:**

Upgrading current version of Jenkins from 2.387.3 (Source Jenkins version) to LTS 2.414.3 (target Jenkins version)

(Confirmation:

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

**Pre-Requisites:**

NOTE:

This upgrade exercise is to be conducted on a staging server before performing the activity against the live/production server.

Step 1. Check for plugin compatibility:

Verify and download all required version of plugins in accordance with target Jenkins version compatibility and transfer all these plugins to Jenkins master.

NOTE:

* We can make use of the plugin-installation-manager-tool as indicated at

<https://github.com/jenkinsci/plugin-installation-manager-tool>

to determine/download the plugins compatible with target Jenkins version.

* There are two plugins indicated in LTS 2.414.X upgrade notes. GCC Jenkins doesn’t have them.

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

Step 4. 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 master VM and Backup “$JENKINS\_HOME\plugins” folder and transfer all plugins downloaded in Pre-requisites Step 1) to “$JENKINS\_HOME\plugins” (Including latest Jakarta Mail API and Jakarta Activation API plugin)

Step 4. Upgrade to the latest version of remoting in Jenkins slave/agent connectivity command/script

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 5. Start Jenkins Service (Similar to Step 2, select “start” instead)

Step 6. Wait for Jenkins application to launch, log in and verify  
that Jenkins CLI is disabled by navigating to “Manage Jenkins” >> “Jenkins CLI”

You should see below error page

A screenshot of a computer

Description automatically generated

And try following command if SSH server is enabled and configured to use Fixed port (eg. 22) under “Manage Jenkins >> Configure Global/Security >> SSH Server”

“ssh -p 22 AHCMAP03@jenkins\_host\_url” or

“ssh AHCMAP03@jenkins\_host\_url”

You should see following error

In Below example Jenkins\_host\_url set to localhost

A black and white screen with white text

Description automatically generated

Step 6: Wait for Jenkins application to launch, log in and verify all existing functionalities, plugins, monitor builds, etc