# [Discontinued support for TLSv1.1 and below](https://central.sonatype.org/articles/2018/May/04/discontinued-support-for-tlsv11-and-below/" \o "Permalink to Discontinued support for TLSv1.1 and below)

Fri 04 May 2018 - Terry Yanko

# Question

Which TLS versions are supported by Central?

# Answer

As of June 18th 2018, Central (repo1.maven.org & repo.maven.apache.org) supports TLS 1.2 only. If you encounter the error message

"Received fatal alert: protocol\_version"

or

"peer not authenticated"

you have four options for remediation:

1. Upgrade your Java runtime, for example with OpenJDK builds or Oracle paying support
2. Configure your Java runtime to enable TLS 1.2 by adding -Dhttps.protocols=TLSv1.2
3. Use a repository manager that uses a Java version supporting TLS 1.2
4. Revert back to http until you can acheive one of the above remediation steps.

# Question

Why did this happen?

# Answer

TLS 1.1 is inherently insecure and in order to maintain compliance, our provider removed the last vestiges of support:

By June 18th, 2018, all customers, including those on dedicated endpoints, should have converted to TLS-1.2. Due to the PCI Security Standards Council mandate, older TLS implementations are no longer supported on Fastly infrastructure on shared or dedicated endpoints.

To read more on this, [see the GitHub blog](https://githubengineering.com/crypto-deprecation-notice/).

# Question

Who is affected?

# Answer

If you are using http to access Central, or Java 8+ with https urls, you are not affected.

If you are on Java 7, it supports TLS 1.2 but it is disabled by default in versions before 1.7.0\_131-b31, and the following Apache Maven workaround functions for older Java 7 users:

mvn -Dhttps.protocols=TLSv1,TLSv1.1,TLSv1.2 <goals>

or add the following to your environment or build script:

export MAVEN\_OPTS=-Dhttps.protocols=TLSv1,TLSv1.1,TLSv1.2

If you are on Java 6, you will need to switch back to http or upgrade to a more modern Java version. Very recent versions of Java 6 claim to have TLS 1.2 support, but we have not confirmed it yet. See here for more information:

[Changes in 6u115 b32 TLS v1.2 support now available](http://www.oracle.com/technetwork/java/javase/overview-156328.html" \l "6u115-b32)

# Question

I cannot implement the above required changes in my environment -- what are my options?

# Answer

We don't recommend this, but [http://repo1.maven.org](http://repo1.maven.org/) is, and will continue to be supported in the long term.

<https://central.sonatype.org/articles/2018/May/04/discontinued-support-for-tlsv11-and-below/>

Sonatype [no longer supports TLSv1.1 and below](https://central.sonatype.org/articles/2018/May/04/discontinued-support-for-tlsv11-and-below/) (effective, June 18th, 2018). My guess is that you are using TLSv1.1 protocol or below.

The documentation I listed gives you 4 options:

1. Upgrade your Java runtime, for example with OpenJDK builds or Oracle paying support
2. Configure your Java runtime to enable TLS 1.2 by adding -Dhttps.protocols=TLSv1.2
3. Use a repository manager that uses a Java version supporting TLS 1.2
4. Revert back to http until you can acheive one of the above remediation steps.

I fixed it myself by just using -Dhttps.protocols=TLSv1.2 as a VM argument.

<https://stackoverflow.com/questions/50946420/could-not-transfer-artifact-https-repo-maven-apache-org-maven2-received-fat>