# Managing Certificates

**Page Contents**

## Overview

Some remote repositories (e.g. Red Hat Networks) block access from clients that are not authenticated with an SSL/TLS certificate. Therefore, to use a remote repository to proxy such resources, Artifactory must be equipped with the corresponding SSL/TLS certificate.

To support this requirement when needed, from version 5.4, Artifactory lets you manage certificates and configure them for use by remote repositories.

**JAVA Cryptography Extension policy**

Some JDK versions and distributions exclude the permissions to use some cryptographic algorithms or SSL extensions, which could required for a successful SSL handshake between a client and a server.

As mentioned in the JDK 8 readme "*due to import control restrictions of some countries, the version of the JCE policy files that are bundled in the Java Runtime Environment, or JRE(TM), 8 environment allow "strong" but limited cryptography to be used*".

To Enable these, do one of the following:

- For Java version 1.8.0-151 and above it is possible to modify the **/usr/lib/jvm/jre-oracle/lib/security/java.security** file and setting it to allow the unlimited crypto policy as demonstrated below:

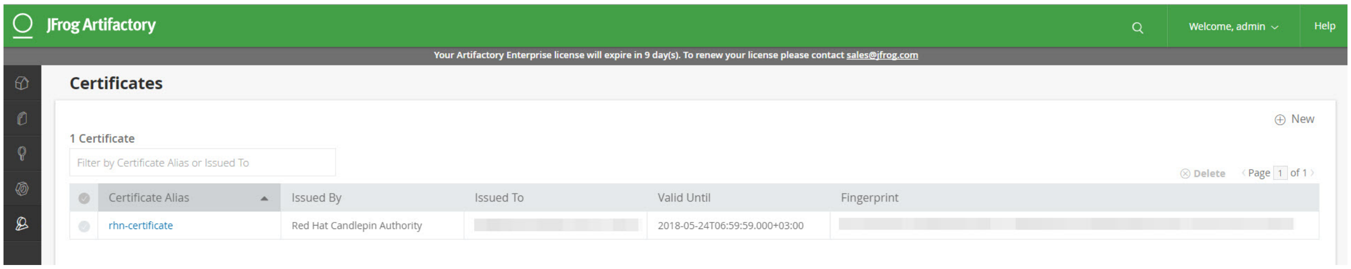
```
crypto.policy=unlimited
```

- Download and enable the Java Cryptography Extension (JCE) jar file that allows *Unlimited Strength Jurisdiction Policy* which contain no restrictions on cryptographic algorithms strength from here

## Adding Certificates

Certificates are managed in the **Admin** module under **Security | Certificates.**

A certificate entered into this module should be a PEM file that includes both a private key and its corresponding certificate.



Your Artifactory Enterprise license will expire in 9 day(s). To renew your license please contact sales@jfrog.com

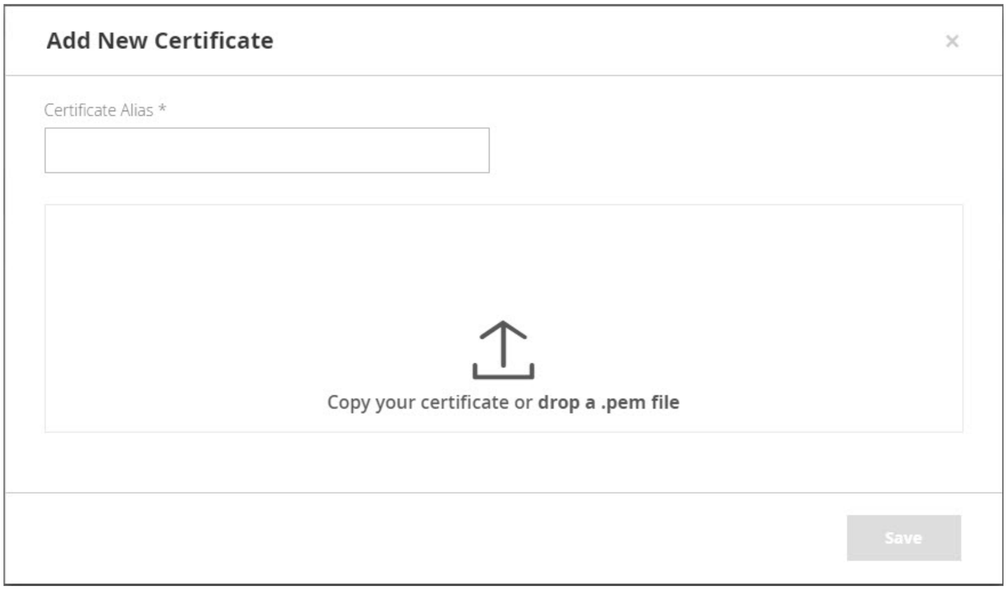
Certificates

1 Certificate

Filter by Certificate Alias or Issued To

| Certificate Alias | Issued By | Issued To | Valid Until | Fingerprint |
|---|---|---|---|---|
| rhn-certificate | Red Hat Candlepin Authority | [redacted] | 2018-05-24T06:59:59.000+03:00 | [redacted] |

To add a new certificate, click **New.**



Provide the **Certificate Alias** and copy the certificate contents into the designated area. Alternatively, you can drag and drop the corresponding PEM file into the designated area.

To avoid text errors, we recommend dragging and dropping the PEM file into the designated area

**Password-protected PEM files are not supported**

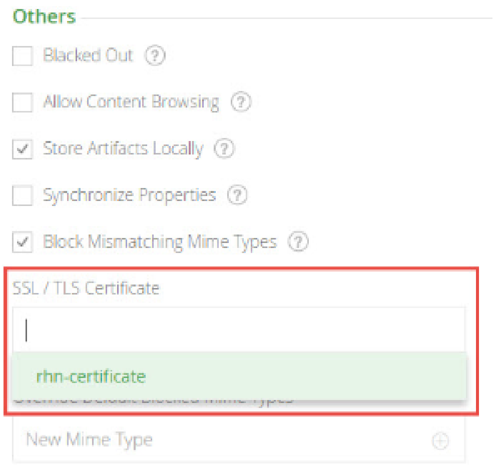
Make sure the PEM file you upload is not password-protected.

---

## Using a Certificate with a Remote Repository

When a remote repository proxy's a resource that requires authentication with a certificate, you need to obtain the certificate from the resource's owner and add it to the list of certificates as described above.

Under the remote repository's Other Settings, select the certificate you want to use from the list provided in the **SSL/TLS Certificate** field.


Others

☐ Blacked Out ?

☐ Allow Content Browsing ?

☑ Store Artifacts Locally ?

☐ Synchronize Properties ?

☑ Block Mismatching Mime Types ?

SSL / TLS Certificate

rhn-certificate

New Mime Type

## Proxying a Resource that Uses a Self-Signed Certificates

If the remote resource that your Artifactory remote repository is proxying (e.g. Red Hat Network's server) uses an **untrusted** server certificate (i.e. it is **self-signed** and not signed by any known Certificate Authority), you need to import the server's certificate into Artifactory's JVM truststore. To learn more about configuring a Self-Signed Certificate in Artifactory, please refer to Using a Self-Signed Certificate.

ⓘ **You cannot configure a self-signed certificate in Artifactory SaaS**

If you are using Artifactory SaaS (as opposed to an on-prem installation), you will not be able to proxy resources that use untrusted (i.e. self-signed) certificates since you do not have access to the Artifactory SaaS JVM truststore.

# REST API

Artifactory supports automated management of certificates using the REST API endpoints described below

## Get Certificates

Gets a list of installed SSL certificates.

For details, refer to the REST API documentation for Get Certificates.

## Add Certificate

Installs a new SSL certificate.

For details, refer to the REST API documentation for Add Certificate.

## Delete Certificate

Deletes the specified certificate.

For details, refer to the REST API documentation for Delete Certificate.