Using Access as a Certificate Authority

Overview

There are three ways of using a Certificate Authority (CA):

- The Access server can act as CA, in which case you do not need to provide your own certificate.
- You can provide Access with your company's CA certificate and then let Access use that certificate to sign server certificates.
- Alternatively, you can provide your own signed server certificate without requiring Access to participate in the action.

Page Contents

Setting TLS

The steps required for setting the TLS are as follows..

- Enable TLS on the JFrog Platform
- Set the TLS certificates: indicating to the Platform which TLS certificate to use

The way to enable a secure cookie is by enabling TLS on access. When you have HTTPS, JDP will then block insecure access to the application (HTTP) and will adds the secure flag to all JDP cookies.

Step 1: Enabling TLS in the JFrog Platform

By default, TLS in the JFrog Platform is *disabled*. When TLS is enabled, all communications to the JFrog Platform are required to use TLS including service-to-service communication within the platform. In the JFrog Platform, Access acts as the CA and signs the TLS certificates used by all the different JFrog Platform services.

Any options you need to set in the TLS certificate will require you to enable TLS (see below).

To enable TLS, set the tls entry (under the security section) in the Access YAML Configuration file to '`true`' and rename it to `access.config.import.yaml`.

```
security:
  tls: true
```

- For Artifactory nodes, the root CA is distributed automatically via the database, and there is no need to copy the Access root CA manually.
  For every other JFrog product node, copy the Access root CA manually to the location, `$JFROG_HOME/{product}/var/etc/security/keys/trusted`. For example, copy the Access root CA to `$JFROG_HOME/xray/var/etc/security/keys/trusted` for Xray.
- If you need to set trust to the Access CA by an external server, for example a load balancer, you will need to load the Access root CA file to the external service key store.

Step 2: Setting the TLS Certificate

Option 1: Use Access as a Root CA with an Access-generated Self-signed Certificate

1. With TLS enabled (see step 1 above), restart the Artifactory node and let the router generate the self-signed certificate with Access.
2. Enable TLS on Artifactory by setting `artifactory.tomcat.httpsConnector.enabled` in the `system.yaml` file to *true*.
3. Restart the Artifactory node.

Option 2: Providing Your Own Signed Certificate

Prerequisites

When providing your own custom TLS certificate, you will need to provide the matching private key. The certificate will be used by ports 8081 (Artifactory) and 8082 (the Platform router).

By default the JFrog Platform (from Artifactory 7.x and above) requires two public ports. You will need to ensure that both ports are using the same certificate.

- 8081: served by Artifactory (i.e., Tomcat)
- 8082: served by the router

Using a Custom TLS Certificate with the Artifactory and Router Ports

If you have not started the application for the first time, you will need to create the /router/keys/ folder manually.

1. Copy the certificate and key files to the `bootstrap/router/keys/custom-server.crt` and `bootstrap/router/keys/custom-server.key`.
   - `custom-server.key` is the private key file
   - `custom-server.crt` is the cert file

   Important

   The files should be named exactly according to their names above.
2. Copy the CA of the custom TLS certificate in `etc/security/keys/trusted/`.
3. Restart the Artifactory node and let the router use the bootstrapped certificate.
4. Enable TLS on Artifactory by setting `artifactory.tomcat.httpsConnector.enabled` to *true* (in the `system.yaml` file).
5. Restart the Artifactory node again.
6. Copy the CA of the custom TLS certificate in `etc/security/keys/trusted/` of all the JFrog Products nodes installed in the same JPD.
7. If applicable, copy the CA to the load balancer.

Custom Certificate and CA Prerequisites

Your custom certificate must meet the following prerequisites:

- The private key must use the RSA algorithm
- The private key must be at least 1024-bit
- The certificate must match the provided private key
- The certificate's issuer must match the CA certificate subject
- The certificate's subject must match the property `shared.node.ip` from `system.yaml`
- The certificate's Subject Alternative Names (SAN) must include the certificate's subject
- Key usage extension should be marked CRITICAL
- Key usage `digitalSignature` extension should be enabled
- Key usage `keyEncipherment` extension should be enabled
- Extended key usage `tlsWebServerAuthentication` should be enabled
- Extended key usage `tlsWebClientAuthentication` should be enabled

Your custom CA certificate must meet the following prerequisites:

- The private key must use the RSA algorithm
- The private key must be at least 1024-bit
- The certificate must match the provided private key
- The certificate must be valid for the next 7 days at least
- The certificate must be marked with a CA basic constraint
- SAN should not be set
- Key usage extension should be marked CRITICAL
- Key usage `digitalSignature` extension should be enabled
- Key usage `keyCertSign` extension should be enabled
- The CN of the certificate should be an IP (and not domain name)
- The IP should match the IP set in the Subject Alternative Names (SAN)

TLS is Disabled

If TLS has not been enabled, you will not be required to take any steps, TLS will not be enabled on the router, nor on Artifactory.

Option 3: Providing a Custom CA Certificate to Access

You can provide a custom CA certificate and matching private key, to be used by Access, for signing the TLS certificates used by all the different JFrog Platform nodes.

Custom CA Prerequisites

Your custom CA certificate must meet the prerequisites described in Option 2 above.

To load a custom CA certificate and matching private key:

1. Create `ca.crt` and `ca.private.key` files and place them under `$JFROG_HOME/artifactory/var/bootstrap/etc/access/keys`.
2. Restart Artifactory.

Regenerating a New Access CA Certificate

In some scenarios you might want to force Access to generate a new CA Certificate. To force JFrog Access to regenerate the CA certificate and matching private key, do the following.

1. Create a `reset_ca_keys` file and place it under `$JFROG_HOME/artifactory/var/bootstrap/etc/access/keys`.
2. Restart Artifactory.
3. If you have already set TLS between Artifactory and other JFrog Platform nodes, copy the new `ca.crt` to the `trusted` directories on all the JFrog Platform nodes.