IdP Cert Migration

Migrating a Certificate in IdP Metadata

This article is for site administrators wishing to replace an old certificate with a new certificate in IdP metadata. Please read the overview Certificate Migration topic before continuing.

> **Handle the New Private Key Carefully!**
>
> The IdP private signing key must be handled with extreme care. Before generating a new private key, consult the IdP Key Handling topic for recommended practices.

In Federation metadata, all certificates in IdP metadata are contained in an `<md:KeyDescriptor use="signing">` element. Such a certificate may be used for signing and/or back-channel TLS. Usually there are identical key descriptors contained in the `<md:IDPSSODescriptor>` element (used as a signing key) and the `<md:AttributeAuthorityDescriptor>` element (used as a back-channel TLS key), in which case both certificates are migrated out of metadata at the same time.

> **Implementation Requirements**
>
> This procedure ultimately requires two `<md:KeyDescriptor use="signing">` elements to be bound to a single role descriptor in IdP metadata. Some SP software implementations will not consume such metadata (which is an implementation bug). Check with your federation partners before initiating the procedure below.

Regardless of the IdP implementation used, the general migration process is as follows.

Preconditions:

- There is a single `<md:KeyDescriptor use="signing">` element bound to each role descriptor in IdP metadata.
- The IdP software is configured to use the corresponding private key as a signing key and/or back-channel TLS key.

Procedure:

1. Add a new `<md:KeyDescriptor use="signing">` element to IdP metadata.
2. Wait for the newly updated metadata to propagate throughout the Federation. Two weeks is safe, although longer times may be needed, depending on the operational practices of your partners.
3. Configure the IdP software to use the new key (instead of the old key) as the signing key and/or back-channel TLS key.
4. Remove the old `<md:KeyDescriptor use="signing">` element from IdP metadata.

Procedural details:

At step 1, log into the Federation Manager, upload a new certificate, and bind that certificate to your metadata. Be sure to bind the certificate to each of the `<md:IDPSSODescriptor>` and `<md:AttributeAuthorityDescriptor>` elements. After doing so, your IdP's metadata will contain four (4) key descriptors, two of which are new.

> **Key Order in Metadata**
>
> When two verification keys are listed in IdP metadata, the old one is listed first. This is because the IdP is still signing with the old key as long as two keys are listed in metadata. This accommodates non-conforming SP implementations (such as EZProxy) that try the first key listed and then stop.

The configuration at step 3 depends on your particular IdP software implementation and how the key is used. Some implementations require separate configurations for signing and back-channel TLS. In particular, if your IdP supports artifact resolution or attribute query, it may require a separate back-channel TLS key configuration. Consult your software documentation for further instructions. (If you're using the Shibboleth IdP, refer to the next section.)

Finally, at step 4, remove the old key descriptors from metadata but leave the two newer key descriptors in the metadata. This completes the migration process.

Implementation-specific Details for IdP Deployers

The key rollover process for a Shibboleth 2.x IdP is partially documented in the Shib wiki. If you have questions about Shibboleth, please consult the Shib mailing lists.

Resources

- https://www.switch.ch/aai/docs/shibboleth/SWITCH/idp-certificate-rollover.html