Metadata Consumption

This page introduces important policy and procedures associated with InCommon metadata. Other pages describe the availability of multiple metadata aggregates and provide guidance on how to configure specific metadata clients. General configuration issues, including the configuration of outbound firewalls, are discussed below.

Metadata Refresh Policy

InCommon expects participants to **refresh metadata daily** to ensure that SAML deployments have access to the most up-to-date keys and other registered information. Some software implementations (such as Shibboleth) handle metadata easily, but regardless of your software, please read this entire page to understand the requirements and pitfalls associated with metadata consumption.

> It is **strongly recommended** that InCommon SPs and IdPs **refresh and verify metadata** at least daily. An optimal configuration would attempt to **refresh metadata every hour** (assuming your client supports HTTP Conditional GET).

Participants are strongly encouraged to use metadata client software that properly handles metadata; failure to do so can have profound effects on the successful use of the Federation. In addition to maintaining the security of your own deployment, proper metadata use is critical to ensure that **other participants** can depend on your system behaving correctly when they make changes.

> **The security implications of metadata refresh!**
>
> Regular metadata refresh protects users against spoofing and phishing, and is a necessary precaution in the event of key compromise. Failure to refresh metadata exposes you, your users, and other Federation participants to unnecessary risk.

In addition, if you don't refresh your metadata regularly, it is likely that a software implementation will fail at some point since the XML document carries an expiration date (`validUntil`) that causes the metadata to expire in approximately two weeks. InCommon strongly recommends that you do **not** rely on the actual length of this validity interval in any way, and in fact, we reserve the right to shorten the validity interval with little or no notice.

Metadata Refresh Process

Here are the steps to deploy a secure, automated metadata refresh process:

1. Choose one of three Metadata Aggregates
2. Obtain an authentic copy of the Metadata Signing Certificate
3. Install and configure recommended Metadata Client Software:
    a. Refresh metadata at least daily (but more often if possible)
    b. Validate the expiration date on downloaded metadata
    c. Verify the XML signature on downloaded metadata
4. Adjust your outbound firewall rules (if necessary)

Refresh Interval

Deployments are strongly encouraged to *refresh metadata at least daily*. If your metadata client supports HTTP Conditional GET, configure the client to *refresh metadata every hour*. This strategy provides the best protection in the event of a key compromise.

Validity Check

Federation metadata has an expiration date, much like an X.509 certificate. It is important that expired metadata not be accepted, otherwise an attacker would be able to substitute expired metadata in conjunction with metadata refresh. In particular, a metadata file should **not** be accepted if any of the following conditions are true:

1. If the metadata file does not have a `validUntil` XML attribute on the root element.
2. If the `validUntil` date on the root element is expired.
3. If the `validUntil` date on the root element is too far into the future.

A metadata refresh process should check each of the above conditions before accepting the metadata. Alternatively, if your SAML implementation is known to ignore/reject expired metadata (a basic correctness requirement), it may be sufficient to ensure that a `validUntil` attribute exists and its date value is not unexpectedly far into the future.

> **Validate the expiration date on InCommon metadata!**
>
> Verifying the signature on a SAML metadata file does **not** validate the presence or value of an expiration date. The only way to validate the expiration date is to parse the XML.

Signature Verification

Federation metadata is signed for integrity and authenticity. Participants are strongly encouraged to verify the XML signature on the metadata file before use; failure to do so will seriously compromise the security of your SAML deployment.

> **Verify the XML signature on InCommon metadata!**
>
> A trusted metadata process MUST verify the XML signature on InCommon metadata. It is **not** sufficient to request the metadata via a TLS-protected HTTP connection, which is why the sample process shown below does not rely on TLS.

The InCommon Federation is based on the *Explicit Key Trust Model*, one of several possible metadata trust models. To bootstrap the trust fabric of the Federation, participants download and configure an authentic copy of the Metadata Signing Certificate into their metadata refresh process. The certificate must be obtained securely since all subsequent operations depend on it.

Once the certificate file is locally installed, you can use it to verify the signature on the metadata file. For example, you could use the XmlSecTool (or some similar 3rd-party tool) to verify the signature:

```
$ MD_LOCATION=http://md.incommon.org/InCommon/InCommon-metadata.xml
$ MD_PATH=/tmp/InCommon-metadata.xml
$ /usr/bin/curl --silent $MD_LOCATION > $MD_PATH
$ ./xmlsectool.sh --verifySignature --signatureRequired \
    --certificate $MD_CERT_PATH --inFile $MD_PATH
INFO  XmlSecTool - Reading XML document from file '/tmp/InCommon-metadata.xml'
INFO  XmlSecTool - XML document parsed and is well-formed.
INFO  XmlSecTool - XML document signature verified.
```

You may also want to schema validate the metadata:

```
$ ./xmlsectool.sh --validateSchema \
    --schemaDirectory $SCHEMA_DIR --inFile $MD_PATH
INFO  XmlSecTool - Reading XML document from file '/tmp/InCommon-metadata.xml'
INFO  XmlSecTool - XML document parsed and is well-formed.
INFO  XmlSecTool - XML document is schema valid
```

For convenience, we provide a set of (suitably modified) schema files that permit offline schema validation.

Firewall Configuration

Depending on your environment, you may have to poke a hole in an outbound firewall to allow your metadata client to reach the metadata server. In that case, you will actually want to poke **two** holes in that firewall since there are two physical servers as described on the Metadata Server wiki page.

For More Information

- http://www.incommon.org/federation/metadata.html
- https://wiki.shibboleth.net/confluence/display/CONCEPT/MetadataCorrectness
- Metadata Interoperability Profile (OASIS)