consume-metadata-best-practice

Query instead of "download" | Verify connection to the Metadata Service | Configure software to consume metadata | Refresh metadata at least daily when using the aggregate

This article describes useful configuration best practices when using InCommon metadata. To learn how to retrieve metadata, visit the InCommon Metadata Service Wiki.

Query instead of "download"

The new InCommon Metadata Distribution Service (MDQ Service) is based on the Metadata Query (MDQ) protocol. A metadata consumer can retrieve needed metadata by making a web query in real time. It eliminates the need to download the entire metadata aggregate, thus significantly reduces system resource overhead and reduces start up time.

To retrieve metadata using the MDQ-based Metadata Service, visit the new InCommon Metadata Service Wiki.

Verify connection to the Metadata Service

To consume InCommon metadata, your IdP or SP must must be able to perform HTTPS requests against mdq.incommon.org. The Metadata Service is hosted on cloud infrastructure with dynamic IP assignments. The IP addresses for these services will change over time. If your organization has egress firewall rules restricting access to outside resources, see Configuring Firewalls to work with the InCommon Metadata Service.

Configure software to consume metadata

See InCommon Metadata Service Wiki.

Refresh metadata at least daily when using the aggregate

While not recommended, the InCommon Metadata Distribution Service does support downloading metadata as an aggregate. If you do use the aggregate, we **strongly recommend** that all service providers and identity providers **refresh and verify metadata** at least daily. Daily refresh ensures your service will have access to the most up-to-date keys and other registered metadata information.

What happens if I don't refresh metadata frequently?

If you don't refresh metadata regularly, your service will likely fail at some point. An IdP or SP operator may update its metadata at any time. If your service does not have the latest information, it may not be able to connect to the services you integrate with. Further, metadata published through InCommon carries an expiration date (`validUntil`) set to expire published metadata approximately every two weeks. NOTE: we advise against rely on the `validUntil` value to determine refresh interval (ref: IdP or SP may update its metadata anytime).

Better yet, use the MDQ query to retrieve metadata on-demand, in real time, to eliminate any issues caused by over caching.

Validate downloaded metadata

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.

In this section

- Metadata signing certificate
- Configure ADFS to consume InCommon metadata
- Configure Shibboleth to consume InCommon metadata
- Configure SimpleSAMLphp to consume InCommon metadata
- Configure firewalls to work with the InCommon Metadata Service

Related content

- Signaling Encryption Method Support for a Service Provider
- Signing and Encryption Keys
- Working with SAML metadata
- Tagging an entity with Research and Scholarship entity attribute
- Introduction to Federation Manager
- Federation references

Get help

Can't find what you are looking for?

help Ask the community

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.

Verify signature

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.

Further Reading

- InCommon Metadata Service Wiki
- https://wiki.shibboleth.net/confluence/display/CONCEPT/MetadataCorrectness
- Metadata Interoperability Profile (OASIS)