IdPMetadataProviderExamples

Metadata Provider Examples

The following examples are simply that, examples. They do not illustrate all possible configuration properties or features. Refer to the documentation for metadata providers for this information.

- Load UK and Swiss Metadata
- Refresh InCommon Metadata

Load UK and Swiss Metadata

Contributed by: Chad La Joie, SWITCH, Switzerland

The following example demonstrates reading the UK and Swiss federation metadata from a URL, storing a back up copy locally, and ensuring that the metadata is properly signed and has a reasonable validity period.

Show Example

```
<!-- Chaining metadata provider defined in the default IdP relying-party
configuration file -->
<MetadataProvider id="ShibbolethMetadata"
xsi:type="ChainingMetadataProvider"
                  xmlns="urn:mace:shibboleth:2.0:metadata">

    <!-- Load the UK metadata -->
    <MetadataProvider id="URLMD"
xsi:type="FileBackedHTTPMetadataProvider"
                      xmlns="urn:mace:shibboleth:2.0:metadata"

metadataURL="http://metadata.ukfederation.org.uk/ukfederation-metadata.x
ml"

backingFile="/opt/shibboleth-idp/metadata/ukfederation-metadata.xml">

        <!-- Using chaining filter to allow us multiple filters to be
added -->
        <MetadataFilter xsi:type="ChainingFilter"
xmlns="urn:mace:shibboleth:2.0:metadata">

            <!-- Ensure the metadata has a reasonable (1 week) validity
period. -->
            <MetadataFilter xsi:type="RequiredValidUntil"
xmlns="urn:mace:shibboleth:2.0:metadata"
                            maxValidityInterval="P30D" />

            <!--
                Ensure metadata is signed and use the
'shibboleth.MetadataTrustEngine'
                to determine its trustworthiness
            -->
            <MetadataFilter xsi:type="SignatureValidation"
xmlns="urn:mace:shibboleth:2.0:metadata"
```

```
trustEngineRef="shibboleth.MetadataTrustEngine"
                            requireSignedMetadata="true" />

        </MetadataFilter>
    </MetadataProvider>

    <!-- Load the Swiss metadata -->
    <MetadataProvider id="URLMD"
xsi:type="FileBackedHTTPMetadataProvider"
                      xmlns="urn:mace:shibboleth:2.0:metadata"

metadataURL="http://metadata.aai.switch.ch/metadata.switchaai.xml"

backingFile="/opt/shibboleth-idp/metadata/metadata.switchaai.xml">

        <!-- Using chaining filter to allow us multiple filters to be
added -->
        <MetadataFilter xsi:type="ChainingFilter"
xmlns="urn:mace:shibboleth:2.0:metadata">

            <!-- Ensure the metadata has a reasonable (1 week) validity
period. -->
            <MetadataFilter xsi:type="RequiredValidUntil"
xmlns="urn:mace:shibboleth:2.0:metadata"
                            maxValidityInterval="604800" />

            <!--
                Ensure metadata is signed and use the
'shibboleth.MetadataTrustEngine'
                to determine its trustworthiness
            -->
            <MetadataFilter xsi:type="SignatureValidation"
xmlns="urn:mace:shibboleth:2.0:metadata"

trustEngineRef="shibboleth.MetadataTrustEngine"
                            requireSignedMetadata="true" />

        </MetadataFilter>
    </MetadataProvider>

</MetadataProvider>

<!-- Define the shibboleth.MetadataTrustEngine used to evaluate the
trustworthiness of metadata -->
<security:TrustEngine id="shibboleth.MetadataTrustEngine"
xsi:type="security:StaticExplicitKeySignature">

    <!-- Trust metadata signed by UK federation cert -->
    <security:Credential id="UKFederationCredential"
xsi:type="security:X509Filesystem">

<security:Certificate>/opt/shibboleth-idp/credentials/ukfederation.crt</
security:Certificate>
```

```
    </security:Credential>

    <!-- Trust metadata signed by Swiss federation cert -->
    <security:Credential id="CHFederationCredential"
xsi:type="security:X509Filesystem">

<security:Certificate>/opt/shibboleth-idp/credentials/chfederation.crt</
security:Certificate>
```

```
    </security:Credential>
</security:TrustEngine>
```

Refresh InCommon Metadata

Contributed by: Tom Scavo, Internet2

The following example demonstrates how to fetch the InCommon Federation production metadata aggregate from a URL, store a back up copy locally, and ensure that the metadata is properly signed and has a reasonable validity period. This process is repeated every hour.

Show Example

```
<!-- Chaining metadata provider defined in the default IdP relying-party
configuration file -->
<MetadataProvider id="ShibbolethMetadata"
xsi:type="ChainingMetadataProvider"
                  xmlns="urn:mace:shibboleth:2.0:metadata">

  <!--
    Refresh the InCommon production metadata aggregate every hour.

    Note: The defaults for minRefreshDelay, maxRefreshDelay, and
refreshDelayFactor
    are "PT5M", "PT4H", and "0.75", respectively. The default for
maxRefreshDelay
    has been changed below, so that the metadata is refreshed every hour
("PT1H").
    The other properties merely regurgitate their default values.
  -->
  <MetadataProvider id="ICMD" xsi:type="FileBackedHTTPMetadataProvider"
                    xmlns="urn:mace:shibboleth:2.0:metadata"

metadataURL="http://md.incommon.org/InCommon/InCommon-metadata.xml"

backingFile="/opt/shibboleth-idp/metadata/InCommon-metadata.xml"
                    minRefreshDelay="PT5M"
                    maxRefreshDelay="PT1H"
                    refreshDelayFactor="0.75">

    <!-- Use a chaining filter to allow multiple filters to be added -->
    <MetadataFilter xsi:type="ChainingFilter">

        <!--
          Require a validUntil XML attribute on the EntitiesDescriptor
element
          and make sure its value is no more than 14 days into the
future
        -->
        <MetadataFilter xsi:type="RequiredValidUntil"
maxValidityInterval="P14D" />

        <!--
```

```
          Require the metadata to be signed and use the trust engine
          labeled id="ICTrust" to determine its trustworthiness
        -->
        <MetadataFilter xsi:type="SignatureValidation"
                        trustEngineRef="ICTrust"
requireSignedMetadata="true" />

        <!-- Consume all SP metadata in the aggregate -->
        <MetadataFilter xsi:type="EntityRoleWhiteList">
          <RetainedRole>samlmd:SPSSODescriptor</RetainedRole>
        </MetadataFilter>

    </MetadataFilter>
  </MetadataProvider>

</MetadataProvider>

<!--
  This TrustEngine (beneath the Security Configuration section) is an
  implementation of the Explicit Key Trust Model
(https://spaces.internet2.edu/x/t43NAQ).

  To bootstrap the trust fabric of the federation, each relying party
  obtains and configures an authentic copy of the federation operator's
  Metadata Signing Certificate (https://spaces.internet2.edu/x/moHFAg).

  Fetch the InCommon metadata signing certificate and check its
integrity:

  $ /usr/bin/curl --silent http://md.incommon.org/certs/inc-md-cert.pem
\
       | /usr/bin/tee /opt/shibboleth-idp/credentials/inc-md-cert.pem \
       | /usr/bin/openssl x509 -sha1 -noout -fingerprint
  SHA1
Fingerprint=7D:B4:BB:28:D3:D5:C8:52:E0:80:B3:62:43:2A:AF:34:B2:A6:0E:DD
-->
<security:TrustEngine id="ICTrust"
xsi:type="security:StaticExplicitKeySignature">

  <security:Credential id="MyFederation1Credentials"
xsi:type="security:X509Filesystem">

<security:Certificate>/opt/shibboleth-idp/credentials/inc-md-cert.pem</s
ecurity:Certificate>
```

```
    </security:Credential>
  </security:TrustEngine>
```