MappedAttributeInMetadataConfiguration

Overview

The MappedAttributeInMetadata type is a Matcher which filters results based on `<md:RequestedAttribute>` elements within the request-indicated `<md:AttributeConsumingService>` in the SP's metadata.

In contrast to the `saml:SAMLAttributeInMetadata` type, the attributes are compared as native IdP attribute objects and values. That is, when the SAML metadata is loaded, the `<md:RequestedAttribute>` elements are examined and the attribute encoding rules defined in the AttributeResolverConfiguration are reversed to dereference the SAML names back into internal IdP-assigned attribute IDs.

- Any Attribute Value type can be compared (the comparison is delegated to the implementation of the AttributeValue).
- The cost of applying the mapping from `<md:RequestedAttribute>` elements is encountered once, when the metadata is loaded (and that in a background thread) as opposed to being done every time the Matcher is evaluated.
- All potential ID to Name mappings are available for comparison

The parameterization controls:

- The behavior when the metadata contains no `<md:RequestedAttribute>` elements (via matchIfMetadataSilent)
- The behavior with respect to the isRequired XML attribute
- Whether this is a Matcher or a PolicyRule (via attributeID)

AttributeInMetadata or MappedAttributeInMetadata?

The AttributeInMetadata and MappedAttributeInMetadata matchers have significant overlap. In practice, use AttributeInMetadata if you need to base the rule on the SAML attribute naming, and MappedAttributeInMetadata if you need to compare non-string values or are concerned about the extra costs of constantly performing the indirection between the SAML naming in the metadata and the internal encoding rules in the IdP.

Schema Type and Location

The MappedAttributeInMetadata type is defined in the urn:mace:shibboleth:2.0:afp namespace, the schema for which can be located at http://shibboleth.net/schema/idp/shibboleth-afp.xsd

The deprecated `basic:AttributeRequesterRegex` type is defined in the urn:mace:shibboleth:2.0:afp:mf:basic namespace, the schema for which can be located at http://shibboleth.net/schema/idp/shibboleth-afp-mf-basic.xsd

Reference

Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>matchIfMetadataSilent</td>
<td>Boolean</td>
<td>false</td>
<td>If true then all input values are returned if and only if the metadata contains no <code>&lt;md:RequestedAttribute&gt;</code> information.</td>
</tr>
<tr>
<td>onlyIfRequired</td>
<td>Boolean</td>
<td>true</td>
<td>If this is true and the corresponding <code>&lt;md:RequestedAttribute&gt;</code> element does not specify <code>isRequired=&quot;true&quot;</code>, then no values are matched.</td>
</tr>
<tr>
<td>attributeID</td>
<td>String</td>
<td></td>
<td>If this is present, then this is a PolicyRule returning true if the Matcher, when applied to the attribute with this ID, matches any values. See AttributeValueString for an example of how attributeID changes the meaning of a Matcher in a slightly less daunting scenario.</td>
</tr>
<tr>
<td>objectStrategyRef</td>
<td>String</td>
<td></td>
<td>Reference to Function&lt; SAMLMetadataContext,XMLObject&gt; that allows the mapped attributes to be pulled from somewhere other than <code>&lt;md:RequestedAttribute&gt;</code> elements</td>
</tr>
</tbody>
</table>

Child Elements

None

Example

The example presumes that an attribute internal to the IdP called "eduPersonPrincipalName" is encoded to a SAML name that may be reflected in an SP's metadata. Thus, you can see that the attribute rule makes no reference to the SAML name at all, but relies on the mapping machinery within the IdP to figure out what the corresponding SAML name happens to be, and any/all attributes that will be encoded to a matching SAML name are going to match for the purposes of this rule.
<AttributeRule attributeID="eduPersonPrincipalName">
  <PermitValueRule xsi:type="MappedAttributeInMetadata" onlyIfRequired="true" matchIfMetadataSilent="true"/>
</AttributeRule>