ShibMetaExt V1.0

Name	SAML 2.0 Metadata Extensions for Shibboleth
Version	1.0
Status	Stable
IPR	Licensed under Apache 2.0 (c) The Ohio State University

The Shibboleth Identity Provider and Service Provider products implement support for two SAML 2.0 Metadata schema extensions, as described below.

Schema Extensions

The schema extensions are defined in the namespace `urn:mace:shibboleth:metadata:1.0` by the following XML Schema document:

Shibboleth Metadata Extension Schema

```
<?xml version="1.0" encoding="US-ASCII"?>
<schema targetNamespace="urn:mace:shibboleth:metadata:1.0"
 xmlns="http://www.w3.org/2001/XMLSchema"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 elementFormDefault="unqualified"
 attributeFormDefault="unqualified"
 version="1.0">

 <import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="xmldsig-core-schema.xsd"/>

 <element name="Scope">
  <annotation>
   <documentation>
   SAML metadata extension used to regulate allowable attribute scopes.
   </documentation>
  </annotation>
  <complexType>
   <simpleContent>
    <extension base="string">
     <attribute name="regexp" type="boolean" use="optional"
default="false"/>
    </extension>
   </simpleContent>
  </complexType>
 </element>

 <element name="KeyAuthority">
  <annotation>
   <documentation>
   Binds keying authorities to the system entity/entities to which the
enclosing
   metadata element applies.
   </documentation>
  </annotation>
  <complexType>
   <sequence>
    <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
   </sequence>
   <attribute name="VerifyDepth" type="unsignedByte" use="optional"
default="1"/>
   <anyAttribute namespace="##other" processContents="lax"/>
  </complexType>
 </element>

</schema>
```

Relevant schema definition fragments are repeated in the descriptions of each extension below.

<shibmd:Scope>

Scope Extension Element

```
<element name="Scope">
    <annotation>
        <documentation>
            SAML metadata extension used to regulate allowable attribute
scopes.
        </documentation>
    </annotation>
    <complexType>
        <simpleContent>
            <extension base="string">
                <attribute name="regexp" type="boolean" use="optional"
default="false"/>
            </extension>
        </simpleContent>
    </complexType>
</element>
```

The `<shibmd:Scope>` element MUST appear within the `<md:Extensions>` element of an `<md:EntityDescriptor>` element or the `<md:Extensions>` element of a producer role descriptor element (such as `<md:IDPSSODescriptor>` or `<md:AttributeAuthorityDescriptor>`). The use of the `<shibmd:Scope>` element outside of these contexts is undefined.

When a `<shibmd:Scope>` element appears in the `<md:Extensions>` element of an `<md:EntityDescriptor>` element it applies to all descendant role descriptor elements. That is to say, this usage is equivalent to putting an identical `<shibmd:Scope>` on every descendant role descriptor.

Each `<shibmd:Scope>` element identifies a permissible attribute "scope" for the role. Scope is an attribute-specific concept used in Shibboleth to enhance the functionality of the attribute acceptance policy features. For each attribute designated in the Service Provider configuration as being a scoped attribute, the scope component of each attribute value received is compared against the collection of scopes designated as permissible for the asserting Identity Provider through `<shibmd:Scope>` elements in its metadata. Any attribute values whose scope is not permissible are discarded, thus ensuring that all scoped attribute values accepted by the Service Provider and passed to the application will have permissible scopes.

> The XML Schema definition of the `<shibmd:Scope>` element includes an explicit `default` value for the `regexp` attribute. One effect of this is that the meaning of an omitted `regexp` attribute will be different for a schema-valiting processor than for one which does not schema-validate. If a document containing a `<shibmd:Scope>` element with an omitted `regexp` attribute is digitally signed, the signature value will therefore depend on whether the signer schema-validates, and validation of such a signature will only succeed if the validator has chosen to take the same approach.
>
> To ensure interoperability between signers and validators no matter whether each schema validates or does not, it is therefore strongly RECOMMENDED that any `<shibmd:Scope>` element appearing in a metadata document that is to be digitally signed incorporate an explicit `regexp` attribute. `regexp="false"` SHOULD always be used instead of an omitted `regexp` attribute.

If `regexp="false"` or absent, the text content of the `<shibmd:Scope>` element is interpreted as the literal scope value. In this case, the scope component of each scoped attribute value processed by the service provider MUST exactly match the value of `<shibmd:Scope>` element.

If `regexp="true"`, the text content of the `<shibmd:Scope>` element is interpreted as specifying a regular expression. In this case, the scope component of each scoped attribute value processed by the service provider MUST match the regular expression.

> Great care should be taken in using `regexp="true"` as it is extremely easy to write regular expressions which match the desired patterns but also permit additional, sometimes surprising, matches. This can lead to the identity provider being permitted a wider range of scopes than intended. Common mistakes are not appropriately quoting meta-characters such as `"."`, and not appropriately anchoring the ends of the match. For example:
>
> ```
> <shibmd:Scope regexp="true">.*.example.edu</shibmd:Scope>
> ```

This regular expression looks like it matches scopes with the general form `something.example.edu`, and indeed it does. However, the unquoted `"."` meta-characters and lack of anchoring means that it will also match the following, among many other cases:

- `eexample.edu`
- `example2edu.example.com`

To avoid over-permissive matching of this kind, the example could be rewritten as follows:

```
<shibmd:Scope regexp="true">^.*\.example\.edu$</shibmd:Scope>
```

It is very common to use DNS domain names as scope values. Because scopes in metadata are matched exactly against the scope component of attribute values in a case-sensitive manner, it is RECOMMENDED that deployers adhere to a convention of representing such scope values as lower case.

Support in Shibboleth Products

This extension corresponds to the `<OriginSite>/<Domain>` element in legacy Shibboleth metadata.

The Shibboleth V1.3 Service Provider processes this element when found in the `<md:Extensions>` element of an attribute-supplying role descriptor (`<md:IDPSSODescriptor>` or `<md:AttributeAuthorityDescriptor>`).

The Shibboleth V2 (and above) Service Provider also processes this element when found in the `<md:Extensions>` element of the `<md:EntityDescriptor>`, interpreting it as applying to all roles.

<shibmd:KeyAuthority>

KeyAuthority Extension Element

```
<element name="KeyAuthority">
    <annotation>
        <documentation>
            Binds keying authorities to the system entity/entities to which
the enclosing
            metadata element applies.
        </documentation>
    </annotation>
    <complexType>
        <sequence>
            <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="VerifyDepth" type="unsignedByte" use="optional"
default="1"/>
        <anyAttribute namespace="##other" processContents="lax"/>
    </complexType>
</element>
```

This extension corresponds to the `<Trust>/<KeyAuthority>` element in legacy Shibboleth metadata.

This element is found in the `<md:Extensions>` element of the `<md:EntitiesDescriptor>` and `<md:EntityDescriptor>` elements.

Each element represents a set of inputs to a certificate path-building operation during transactions involving the roles or system entities contained within the parent element. Each `<ds:KeyInfo>` element represents a single trust anchor for such operations, generally an X.509 certificate.

The `VerifyDepth` attribute controls the maximum path length to allow, using the PKIX-specified definition of path length (i.e., one less than the actual length of the chain).