SAML Metadata Extension for Query Requesters

Committee Draft 01, 14 March 2006

Document identifier:
sstc-saml-metadata-ext-query-cd-01

Location:
http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

Editors:
Tom Scavo (trscavo@gmail.com), Individual
Scott Cantor (cantor.2@osu.edu), Internet2

Contributors:
Tom Wisniewski, Entrust

Abstract:
This specification defines an extension to the SAML V2.0 metadata specification [SAML2Meta]. The extension defines role descriptor types that describe a standalone SAML V1.x or V2.0 query requester for each of the three predefined query types. Readers are advised to familiarize themselves with that specification before reading this one.

Status:
This is a **Committee Draft** approved by the Security Services Technical Committee on 14 March 2006.

Committee members should submit comments and potential errata to the security-services@lists.oasis-open.org list. Others should submit them by filling out the web form located at http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=security.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights web page for the Security Services TC (http://www.oasis-open.org/committees/security/ipr.php).

Table of Contents

1 Introduction

This specification defines an extension to the SAML V2.0 metadata specification. The extension defines a set of role descriptor types that describe a standalone SAML query requester for each of the three predefined query types. The profile addresses both SAML V1.x and SAML V2.0.

Unless specifically noted, nothing in this document should be taken to conflict with the SAML V2.0 metadata specification [SAML2Meta]. Readers are advised to familiarize themselves with that specification before reading this one.

1.1 Notation

This specification uses normative text to define an extension to the SAML V2.0 metadata specification.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC 2119]:

> …they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)…

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

```
Listings of XML schemas appear like this.
```

```
Example code listings appear like this.
```

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
`saml:`	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace [SAML2Core].
`md:`	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace [SAML2Meta].
`query:`	urn:oasis:names:tc:SAML:metadata:ext:query	This is the SAML V2.0 metadata query extension namespace, defined by this document and its accompanying schema [MDext-XSD].
`xsd:`	http://www.w3.org/2001/XMLSchema	This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this is the default namespace and no prefix is shown.
`xsi:`	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].
`ds:`	http://www.w3.org/2000/09/xmldsig#	This is the XML Signature namespace [XMLSig].

This specification uses the following typographical conventions in text: `<SAMLElement>`, `<ns:ForeignElement>`, `Attribute`, **Datatype**, `OtherKeyword`.

2 Query Metadata Extensions for SAML V2.0

This section defines new role descriptor types that support the requester role of the three predefined SAML query types, authentication, attribute, and authorization decision.

2.1 Namespaces

The SAML V2.0 metadata specification [SAML2Meta] and its accompanying schema [SAML2Meta-xsd] define the following namespace:

```
urn:oasis:names:tc:SAML:2.0:metadata
```

By convention, the namespace prefix `md:` is used to refer to the above namespace.

This specification defines a new namespace:

```
urn:oasis:names:tc:SAML:metadata:ext:query
```

The prefix `query:` is used here and in the accompanying schema [MDext-XSD] to refer to this new namespace. In what follows, any unqualified type is assumed to belong to this new namespace.

2.2 Element <md:RoleDescriptor>

The `<md:RoleDescriptor>` element defined in [SAML2Meta] is an abstract extension point that contains descriptive information common across various entity roles. New roles can be defined by extending its abstract **md:RoleDescriptorType** complex type, which is the approach taken here.

2.3 Abstract Complex Type QueryDescriptorType

Abstract complex type **QueryDescriptorType** extends complex type **md:RoleDescriptorType** with content generally applicable to query requesters. The type **QueryDescriptorType** contains the following additional attributes and elements:

`WantAssertionsSigned` [Optional]

> Optional attribute that indicates a requirement for assertions received by this requester to be signed. If omitted, the value is assumed to be `false`. This requirement is in addition to any requirement for signing derived from the use of a particular profile/binding combination.

`<md:NameIDFormat>` [Zero or More]

> Zero or more elements of type **xsd:anyURI** that enumerate the name identifier formats supported by this requester. See Section 8.3 of [SAML2Core] for some possible values of this element.

As an abstract type, this type serves as a basis for the additional types defined in the following sections and is not used in metadata instances directly.

The following schema fragment defines the **QueryDescriptorType** complex type:

```
<complexType name="QueryDescriptorType" abstract="true">
  <complexContent>
    <extension base="md:RoleDescriptorType">
      <sequence>
        <element ref="md:NameIDFormat" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="WantAssertionsSigned" type="boolean" use="optional"/>
    </extension>
```

```
    </complexContent>
  </complexType>
```

2.4 Complex Type AuthnQueryDescriptorType

Complex type **AuthnQueryDescriptorType** extends complex type **QueryDescriptorType** into a concrete type usable to represent authentication query requesters. It contains no additional elements or attributes.

Instances of **AuthnQueryDescriptorType** are declared using the `<md:RoleDescriptor>` element with an `xsi:type` of **AuthnQueryDescriptorType**.

See for specifics on the transformation and use of particular elements and attributes for use with SAML V1.x.

The following schema fragment defines the **AuthnQueryDescriptorType** complex type:

```
  <complexType name="AuthnQueryDescriptorType">
    <complexContent>
      <extension base="md:QueryDescriptorType"/>
    </complexContent>
  </complexType>
```

2.5 Complex Type AttributeQueryDescriptorType

Complex type **AttributeQueryDescriptorType** extends complex type **QueryDescriptorType** with content specific to attribute query requesters, that is, consumers of SAML attributes. The type **AttributeQueryDescriptorType** contains the following additional elements:

`<md:AttributeConsumingService>` [Zero or More]

> Zero or more elements that describe an application or service provided by this requester that requires or desires the use of SAML attributes. It is RECOMMENDED that deployers provide at least one such element to facilitate configuration of policy by attribute providers.

At most one `<md:AttributeConsumingService>` element can have the attribute `isDefault` set to `true`. When multiple elements are specified and none has the attribute `isDefault` set to `true`, then the first element whose `isDefault` attribute is not set to `false` is to be used as the default. If all elements have their `isDefault` attribute set to `false`, then the first element is considered the default.

Instances of **AttributeQueryDescriptorType** are declared using the `<md:RoleDescriptor>` element with an `xsi:type` of **AttributeQueryDescriptorType**. See the example in Section 2.7.

See for specifics on the transformation and use of particular elements and attributes for use with SAML V1.x.

The following schema fragment defines the **AttributeQueryDescriptorType** complex type:

```
  <complexType name="AttributeQueryDescriptorType">
    <complexContent>
      <extension base="md:QueryDescriptorType">
        <sequence>
          <element ref="md:AttributeConsumingService" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

2.6 Complex Type AuthzDecisionQueryDescriptorType

Complex type **AuthzDecisionQueryDescriptorType** extends complex type **QueryDescriptorType** with content specific to authorization decision query requesters, that is, policy enforcement points. The type **AuthzDecisionQueryDescriptorType** contains the following additional elements:

`<ActionNamespace>` [Zero or More]

Zero or more elements of type **xsd:anyURI** that enumerate the action namespaces supported by this requester. See Section 8.1 of [SAML2Core] for some possible values of this element.

Instances of **AuthzDecisionQueryDescriptorType** are declared using the `<md:RoleDescriptor>` element with an `xsi:type` of **AuthzDecisionQueryDescriptorType**.

See for specifics on the transformation and use of particular elements and attributes for use with SAML V1.x.

The following schema fragment defines the **AuthzDecisionQueryDescriptorType** complex type:

```
  <element name="ActionNamespace" type="anyURI"/>
  <complexType name="AuthzDecisionQueryDescriptorType">
    <complexContent>
      <extension base="md:QueryDescriptorType">
        <sequence>
          <element ref="query:ActionNamespace" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

The following schema fragment defines the `<ActionNamespace>` element:

```
  <element name="ActionNamespace" type="anyURI"/>
```

2.7 Example

Following is a metadata example for a SAML attribute query requester that supports both SAML V1.1 and SAML V2.0.

```
<md:EntityDescriptor
  xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  entityID="https://gs.org/gridshib">
  <!-- insert ds:Signature element here -->
  <md:RoleDescriptor
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:query="urn:oasis:names:tc:SAML:metadata:ext:query"
    xsi:type="query:AttributeQueryDescriptorType"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol
urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo>
        <ds:KeyName>Requester Key</ds:KeyName>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>
      urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
    </md:NameIDFormat>
    <md:AttributeConsumingService isDefault="true" index="0">
      <md:ServiceName xml:lang="en">
```

```
        Shibbolized Grid Service
      </md:ServiceName>
      <md:RequestedAttribute
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
        Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
        FriendlyName="eduPersonEntitlement">
        <saml:AttributeValue xsi:type="xsd:anyURI">
          https://gs.org/gridshib/entitlements/123456789
        </saml:AttributeValue>
      </md:RequestedAttribute>
      <md:RequestedAttribute
        NameFormat="urn:mace:shibboleth:1.0:attributeNamespace:uri"
        Name="urn:mace:dir:attribute-def:eduPersonEntitlement">
        <saml:AttributeValue xsi:type="xsd:anyURI">
          https://gs.org/gridshib/entitlements/123456789
        </saml:AttributeValue>
      </md:RequestedAttribute>
    </md:AttributeConsumingService>
  </md:RoleDescriptor>
  <md:Organization>
    <md:OrganizationName xml:lang="en">
      GridShib Service Provider
    </md:OrganizationName>
    <md:OrganizationDisplayName xml:lang="en">
      GridShib Service Provider @ Some Location
    </md:OrganizationDisplayName>
    <md:OrganizationURL xml:lang="en">
      http://www.gs.org/
    </md:OrganizationURL>
  </md:Organization>
  <md:ContactPerson contactType="technical">
    <md:SurName>GridShib Support</md:SurName>
    <md:EmailAddress>gridshib-support@gs.org</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

3 References

The following works are cited in the body of this specification.

3.1 Normative References

[RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[MDext-XSD] S. Cantor et al. SAML metadata extension schema. OASIS SSTC, February 2006. Document ID sstc-saml-metadata-ext-query.xsd. See http://www.oasis-open.org/committees/security/.

[SAML1xMeta] G. Whitehead and S. Cantor. *Metadata Profile for the OASIS Security Assertion Markup Language (SAML) V1.x.* OASIS, March 2005. Document ID sstc-saml1x-metadata-cd-01. See http://www.oasis-open.org/committees/security/.

[SAML2Core] S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-core-2.0-os. See http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[SAML2Meta] S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-metadata-2.0-os. See http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf.

[SAML2Meta-xsd] S. Cantor et al. SAML V2.0 metadata schema. OASIS Standard, March 2005. Document ID saml-schema-metadata-2.0. See http://www.oasis-open.org/committees/download.php/11903/saml-2.0-os-xsd.zip.

[Schema1] H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web Consortium Recommendation, May 2001. See http://www.w3.org/TR/xmlschema-1/.

[XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing*, World Wide Web Consortium, February 2002. See http://www.w3.org/TR/xmldsig-core/.

A. Acknowledgments

The editors would like to acknowledge the contributions of the OASIS Security Services Technical Committee, whose voting members at the time of publication were:

- Hal Lockhart, BEA Systems, Inc.
- Steve Anderson, BMC Software
- Rick Randall, Booz Allen Hamilton
- Nick Ragouzis, Enosis Group
- Thomas Wisniewski, Entrust
- Sharon Boeyen, Entrust
- Carolina Canales-Valenzuela, Ericsson
- Dana Kaufman, Forum Systems
- Ashish Patel, France Telecom
- Irving Reid, Hewlett-Packard Company
- Greg Whitehead, Hewlett-Packard Company
- Guy Denton, IBM
- Heather Hinton, IBM
- Anthony Nadalin, IBM
- Eric Tiffany, IEEE Industry Standards
- Prasanta Behera, Individual
- Scott Cantor, Internet2
- Bob Morgan, Internet2
- Jeff Hodges, NeuStar
- Frederick Hirsch, Nokia
- Paul Madsen, NTT Corporation
- Ari Kermaier, Oracle
- Prateek Mishra, Oracle
- Vamsi Motukuru, Oracle
- Brian Campbell, Ping Identity
- John Hughes, PA Consulting
- Rob Philpott, RSA Security
- Jahan Moreh, Sigaba
- Bhavna Bhatnagar, Sun Microsystems
- Eve Maler, Sun Microsystems
- David Staggs, Veteran's Health Admin

Appendix B. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.