Proposal for SAML Attribute Changes

Proposal 04, 3 May 2004

Document identifier:

sstc-maler-w28a-attribute-draft-04

Location:

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

Previous draft:

http://www.oasis-open.org/apps/org/workgroup/security/download.php/5946/sstc-maler-w28a-attribute-draft-03-diff.pdf

Author:

Eve Maler, Sun Microsystems (eve.maler@sun.com)

Contributor:

Rebekah Lepro, NASA Ames Research Center

Abstract:

This document proposes a set of solutions that meet the requirements and goals expressed in Rebekah Lepro's **Attribute Representation in SAML 2.0** document [AttribRep]. Portions of that document have been reproduced here in order to give the full context for attribute-related requirements and changes in SAML V2.0. As the SAML V2.0 design effort runs to completion, this document has been updated so that it can serve as a historical record of decisions made in this area and their rationales.

Status:

Please send comments to the author.

Rev 01: 6 Feb: Initial draft.

Rev 02: 21 Feb: Includes relevant requirements and proposals from Rebekah's paper.

Rev 03: 15 Mar: Includes unofficial results from focus group telecon on 9 Mar.

Rev 04: 3 May: Brings the proposal up to date with the latest SSTC decisions.

Table of Contents

1 V1.1 Schema for Attributes....3
2 Goals and Requirements....4
2.1 Allow for Alignment with Existing Attribute Representations....4
2.2 Identify Consistent Attribute Datatypes....4
2.3 Standardize Semantics of Attribute Naming and Metadata....4
2.4 Cleanly Handle Null-Valued and Multi-Valued Attributes....4
2.5 Make Attribute Complexity Match Power....5
3 Proposed Changes....6
3.1 Handling Datatypes that Map to Other Systems....6
3.2 Clarifying Naming and Adding Metadata....6
3.3 Handling Null-Valued and Multi-Valued Attributes....7
3.4 Keeping the Changes Simple....7
4 References....8

1 V1.1 Schema for Attributes

Following is the SAML V1.1 assertion schema snippet related to attributes, for reference:

```
<element name="AttributeDesignator" type="saml:AttributeDesignatorType"/>
<complexType name="AttributeDesignatorType">
    <attribute name="AttributeName" type="string" use="required"/>
    <attribute name="AttributeNamespace" type="anyURI" use="required"/>
</complexType>
<element name="Attribute" type="saml:AttributeType"/>
<complexType name="AttributeType">
    <complexContent>
        <extension base="saml:AttributeDesignatorType">
            <sequence>
                <element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element name="AttributeValue" type="anyType"/>
```

This results in instances like this in a query:

```
<AttributeDesignator
  AttributeName="any-name"
  AttributeNamespace="URI-representing-set-of-att-names"/>
```

And in instances like this in an assertion sent in response:

```
<Attribute
  AttributeName="any-name"
  AttributeNamespace="URI-representing-set-of-att-names">
any-string-or-structured-value
</Attribute>
```

2 Goals and Requirements

Note: In this proposal, the word “attribute” always refers to a SAML attribute or similar piece of information about a subject, rather than the so-named XML construct. When XML attribute markup is meant, it is called a “field” or an “XML attribute”.

The earlier attribute proposal [AttribRep] and the TC's discussion on 5 February 2004 [F2FMinutes] highlighted the following goals.

2.1 Allow for Alignment with Existing Attribute Representations

SAML needs to prepare for alignment with LDAP and with XACML's attribute handling, but in a way that doesn't massively inconvenience existing SAML attribute statement users who have no desire to do this mapping.

One critical use case of SAML attribute exchange is the provision of attributes to a policy evaluation process, such as XACML defines. Often, a policy can be represented directly in terms of attribute designators, such as the XACML policy representation. This requires that all information needed to represent that policy in terms of attributes must be available.

Following are the basic differences between SAML's and XACML's attribute representations:

- SAML has two fields that contribute to a unique attribute name, `AttributeName` and `AttributeNamespace`. XACML has a single URI-based field.
- SAML allows specification of an attribute's datatype only through XSD means. XACML has a field for a URI-based datatype identifier.
- SAML supplies issuer information only at the assertion level. XACML supplies it per attribute.

Following are the basic differences between SAML's and X.500/LDAP's attribute representations (refer to Bob Morgan's proposal [MorganX500] for suggested conventions on how to map SAML attributes to X.500/LDAP):

- The X.500/LDAP concept of OIDs and LDAP's ability to provide an attribute's short name has only a loose resemblance to SAML's ability to represent typed attribute namespaces and names.
- X.500 and LDAP have a native ability to represent attribute schemas that themselves have OIDs, whereas SAML relies entirely on XSD (or on out-of-band means) for schemas and constraints.

2.2 Identify Consistent Attribute Datatypes

SAML needs to provide the ability to determine the expected type of an attribute value whether or not an attribute value is present. Several consumers of attribute statements require such datatype information, as a function of the attribute rather than the attribute value. There is currently no way to exchange this information in-band within a SAML assertion that does not contain an attribute value or a SAML query.

Also, SAML needs to allow for datatype information to be supplied consistently for all the values (if there are more than one) of an attribute.

2.3 Standardize Semantics of Attribute Naming and Metadata

Currently, SAML's `AttributeNamespace` field is used in several inconsistent ways. One way is to provide scoping, administrative domain, or sourcing information about the attribute, which may be general needs.

2.4 Cleanly Handle Null-Valued and Multi-Valued Attributes

SAML needs to provide the clear, interoperable ability to represent null-valued and multi-valued attributes within a single XML element.

2.5 Make Attribute Complexity Match Power

SAML needs to ensure that any new (and existing) attribute features provide only enough schema complexity to match the power gained therefrom.

3 Proposed Changes

Following are proposed changes, taking into account the goals, requirements, and other proposals made to date.

3.1 Handling Datatypes that Map to Other Systems

To satisfy goals 2.1 and 2.2 regarding datatype mapping, it should be possible for SAML extensions/profiles to add a URI-based `ValueType` field or other similar field for holding the attribute value's type. It could be added either to just **AttributeType** so as to affect only returned attributes, or to **AttributeDesignatorType** so that it is picked up by both `<AttributeDesignator>` (used in attribute queries) and `<Attribute>` (used in attribute statements). The Baseline Attributes specification should describe how this can be achieved with maximum interoperability.

3.2 Clarifying Naming and Adding Metadata

To satisfy goal 2.1 regarding mapping of attribute names and goal 2.3, the `AttributeNamespace` field should be renamed to `NameFormat` (and the `AttributeName` field should be renamed to `Name` to follow suit). The schema change is as follows:

```
<complexType name="AttributeDesignatorType">
    <attribute name="Name" type="string" use="required"/>
    <attribute name="NameFormat" type="anyURI" use="required"/>
</complexType>
```

The `NameFormat` field should be defined in the spec as:

> *A URI reference representing the classification of the attribute name for purposes of interpreting the name. See Section X.X for some URI references that MAY be used as the value of the `NameFormat` attribute and their associated descriptions and processing rules. If no `NameFormat` value is provided, the identifier `urn:oasis:names:tc:SAML:2.0:attname-format:unspecified` (see Section X.X.X) is in effect.*

This choice of field names provides brevity, and also consistency with the rest of SAML when it comes to "format" fields. In addition, a new subsection of the core spec's Identifiers section should define the following URI-based name formats:

> *`urn:oasis:names:tc:SAML:2.0:attname-format:unspecified`*
> *The interpretation of the attribute name is left to individual implementations.*
>
> *`urn:oasis:names:tc:SAML:2.0:attname-format:uri`*
> *The attribute name follows the convention for URI references [BIBREF], for example as used in XACML [BIBREF] attribute identifiers. The interpretation of the URI content or naming scheme is application-specific.*

(It was contemplated to add a new optional `Source` field to **AttributeType**, defined in the spec as:

> *The source location or database from which the attribute came. Interpretation of the source information is application-specific.*

However, the TC decided that this was too vaguely defined and shouldn't be added because it wouldn't increase interoperability.)

Finally, an `<xs:anyAttribute>` wildcard should also be added to **AttributeDesignatorType**, to allow the arbitrary addition of global XML attributes onto the `<AttributeDesignator>` and `<Attribute>` elements. The schema change is as follows:

```
<complexType name="AttributeDesignatorType">
            ...
            <anyAttribute/>
</complexType>
```

This will permit the addition of various kinds of scope data and other context necessary to interpret the attribute value or to constrain the retrieval of particular attributes based on their values, without prematurely forcing all SAML users to use a long list of predefined fields that may not meet their needs. The explanation of this feature should include the following prose limitation:

> *SAML extensions MUST NOT add local (non-namespace-qualified) XML attributes to the* ***AttributeType*** *complex type or to any element bound to this type or a derivation of it; such attributes are reserved for future maintenance and enhancement of SAML itself.*

3.3 Handling Null-Valued and Multi-Valued Attributes

The SAML V2.0 core spec, rev 05 [SAMLCore2.0], already includes wording that addresses goal 1.4:

> `<AttributeValue>` *[Any number]: The value of the attribute. If an attribute contains more than one discrete value, it is RECOMMENDED that each value appear in its own* `<AttributeValue>` *element. If the attribute exists but has no value, then the* `<AttributeValue>` *element MUST be omitted.*

3.4 Keeping the Changes Simple

In keeping with goal 2.5, the changes proposed are structurally not very invasive. Following is a summary of the proposed schema changes in previous sections:

```
<element name="AttributeDesignator" type="saml:AttributeDesignatorType"/>
<complexType name="AttributeDesignatorType">
    <attribute name="Name" type="string" use="required"/>
    <attribute name="NameFormat" type="anyURI" use="required"/>
    <anyAttribute>
</complexType>
<element name="Attribute" type="saml:AttributeType"/>
<complexType name="AttributeType">
    <complexContent>
        <extension base="saml:AttributeDesignatorType">
            <sequence>
                <element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element name="AttributeValue" type="anyType"/>
```

This results in instances like this in a query:

```
<AttributeDesignator
  Name="any-name-here"
  NameFormat="URI-indicating-how-to-interpret-name"
  foreign-ns:context="application-specific-context-string"
/>
```

And in instances like these in an assertion sent in response (where `Source` is optional):

```
<Attribute
  Name="any-name-here"
  NameFormat="URI-indicating-how-to-interpret-name"
  foreign-ns:context="application-specific-context-string"/>
any-string-or-structured-value-here
</Attribute>
```

The changes do not affect the basic type hierarchy: **AttributeDesignatorType**>**AttributeType**. The new fields are optional (with carefully specified semantics for the case of their absence) in order to avoid adding new types and elements for the present vs. absent options.

4 References

[AttribRep] R. Lepro, “Attribute Representation in SAML v2.0”, proposal to OASIS SSTC, document identifier draft-sstc-attribute-02, 31 December 2003. http://www.oasis-open.org/committees/download.php/4884/draft-sstc-attribute-02.pdf.

[F2FMinutes] Minutes of the OASIS SSTC, 3-5 February 2004. http://lists.oasis-open.org/archives/security-services/200402/msg00123.html.

[MorganX500] R.L. “Bob” Morgan, “Conventions for Use of X.500/LDAP Attribute Types in SAML”, proposal to OASIS SSTC, document identifier draft-morgan-saml-attr-x500-00, 5 November 2003. http://www.oasis-open.org/committees/download.php/4124/draft-morgan-saml-attr-x500-00.pdf.

[SAMLCore2.0] E. Maler et al., Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, revision 05, document identifier sstc-saml-core-2.0-draft-05, 17 February 2004. http://www.oasis-open.org/committees/download.php/5519/sstc-saml-core-2.0-draft-05-diff.pdf.

A. 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.