SAML V2.0 Attribute Predicate Profile Version 1.0

Working Draft 01

17 May 2011

Abstract:

This profile provides a mechanism to allow a SAML authority to certify that a given Boolean predicate holds over one or more of a subject's attribute values, without revealing the exact values of these attributes. The profile further defines a query format for attribute predicates.

Status:

This Working Draft (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or approved as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document Approval Process begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

Table of Contents

1 Introduction

SAML attribute assertions enable a SAML authority, sometimes also referred to as an issuer, to certify to a relying party that a subject is associated with a specified set of attribute values. There are many circumstances, however, where the relying party is not interested in the exact attribute values, but merely needs to be assured that a certain condition is satisfied. For example, a SAML authority may keep its subjects' dates of birth on record, but to control access to a teenage chat room, the relying party merely needs to ensure that the requesting subject belongs to the correct age group. Not only may users be reluctant to disclose such identifying information to the chat room host, but also the host itself may prefer not to know the exact date to avoid liability claims in case a data breach occurs.

For small numbers of predicates this problem can obviously be overcome by defining a dedicated Boolean attribute for each possible predicate, which the issuer authenticates by means of a SAML attribute statement. While this may work for common age restrictions such as being younger or older than eighteen, it is impractical when the space of relevant predicates is large. For example, it is rather unlikely that a dedicated attribute will be globally agreed upon if the minimum age to sign up for a theoretical driving test is seventeen years and nine months.

As another example where attribute predicates can be useful, consider an online opinion poll hosted on a city's website and a SAML authority that certifies subjects' ZIP codes. The city may want to ensure that only legal residents of the metropolitan area can participate in the poll, but for privacy reasons may not want to keep track of the voting statistics of separate neighborhoods. Other websites may want to ensure that the subject has an email address within a particular domain (e.g., youremployer.com), but to guarantee the users' privacy, they have no interest in the exact address.

Predicates may also involve relation among multiple attributes known to the SAML authority. For example, the signup form for an obesity summer camp may restrict access to participants whose body-mass index, computed as the weight divided by the square of the height, is above a certain threshold, but does not want to force participants to disclose their exact weight, height, or body-mass index.

[All text is normative unless otherwise labeled]

[**Administrative note to TC**, to be removed at CSD01 publication time. Expected URI pattern: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/csd01/sstc-saml-attr-predicate-v1.0-csd01.doc]

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **[RFC2119]**.

1.2 Normative References

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

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

[XACML] eXtensible Access Control Markup Language (XACML), product of the OASIS XACML TC. See http://www.oasis-open.org/committees/xacml.

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

1.3 Non-Normative References

[Reference] [Full reference citation]

2 Profile Overview

This profile defines how to use SAML 2.0 to query a SAML authority for the truth value of a Boolean predicate over one ore more of a subject's attribute values. In particular, this profile defines the following types and elements:

- `<AttributePredicate>`: An element describing a Boolean predicate over one or more of a subject's attribute values.
- `<AttributePredicateQuery>`: An element used to query a SAML authority for the truth value of a particular subject's attribute predicate.
- `<AttributePredicateStatementType>`: A SAML statement type describing a statement asserting that a subject's attribute predicate has a particular truth value.

3 Profile Description

The following sections provide detailed definitions of the defined types and elements.

3.1 Element <AttributePredicate>

The <AttributePredicate> element describes a Boolean predicate over one or more of a subject's attribute values. The predicate is expressed by means of an <xacml:Apply> element that denotes application of a specified function (identified by a URI in its FunctionId attribute) to one or more arguments of the <xacml:Expression> element substitution group. The return data-type of the <xacml:Apply> element MUST be "http://www.w3.org/2001/XMLSchema#boolean". Arbitrarily complex predicates can be formulated by nesting multiple <xacml:Apply> elements.

```
<xs:element name="AttributePredicate" type="AttributePredicateType"/>
<xs:complexType name="AttributePredicateType">
  <xs:sequence>
    <xs:element ref="xacml:Apply"/>
  </xs:sequence>
  <xs:attribute name="FriendlyDescription" type="xs:string" use="optional"/>
</xs:complexType>
```

The following four members of the <xacml:Expression> element substitution group may occur in the attribute predicate:

<xacml:AttributeValue>

<xacml:AttributeDesignator>

Any <xacml:AttributeDesignator> element used MUST be of category urn:oasis:names:tc:xacml:1.0:subject-category:access-subject.

The Issuer attribute in the <xacml:AttributeDesignator> element, if present, MUST have the same value as specified in the <saml:Issuer> element of the enclosing query or assertion.

<xacml:Apply>

The attribute predicate expressed as an XACML expression. All function URIs marked as mandatory in XACML version 3.0 [XACML] MUST be supported as value for the FunctionId XML attribute by any implementation of this profile. Functions marked as optional MAY be supported, as well as any additional self-defined functions, as long as the semantics are understood by all of the involved parties. The return data type of the outermost <xacml:Apply> element MUST be http://www.w3.org/2001/XMLSchema#boolean.

<xacml:Function>

The <xacml:AttributeSelector> and <xacml:VariableReference> elements MUST not occur in an attribute predicate.

3.2 Element <AttributePredicateQuery>

The <AttributePredicateQuery> element is used to make the query "Does the specified Boolean predicate hold over this subject's attribute value(s)?". The query additionally specifies whether the response should contain an assertion that explicitly repeats the queried predicate in an attribute predicate statement. A successful response states that the queried predicate is true.

This element is of type **AttributePredicateQueryType**, which extends **SubjectQueryAbstractType** with the addition of the following element and attribute:

IncludePredicateInResponse [Optional]

A Boolean value. If “true”, the queried attribute predicate MUST be included in an attribute predicate statement in the response.

<AttributePredicate> [Required]

The queried Boolean predicate over one or more of the subject’s attribute values.

A SAML authority responds to an attribute predicate query according to the conventions specified in Section 4.

The following schema fragment defines the <AttributePredicateQuery> element and its **AttributePredicateQueryType** complex type:

```
<xs:element name="AttributePredicateQuery"
type="AttributePredicateQueryType"/>
<xs:complexType name="AttributePredicateQueryType">
  <xs:complexContent>
    <xs:extension base="samlp:SubjectQueryAbstractType">
      <xs:sequence>
        <xs:element ref="AttributePredicate"/>
      </xs:sequence>
      <xs:attribute name="IncludePredicateInResponse" type="xs:boolean"
use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

3.3 Type <AttributePredicateStatementType>

A <saml:Statement> element of type **AttributePredicateStatementType** describes a statement by the SAML authority asserting that a Boolean predicate over one or more of the assertion subject’s attribute values is true. In this profile, statements of this type are referred to as *attribute predicate statements*. Attribute predicate statements MUST contain a <Subject> element.

The type **AttributePredicateStatementType** extends **StatementAbstractType** with the addition of the following element:

<AttributePredicate> [Required]

The <AttributePredicate> element specifies a Boolean predicate over one or more of the assertion subject’s attribute values.

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

```
<xs:complexType name=”AttributePredicateStatementType”>
  <xs:complexContent>
    <xs:extension base="saml:StatementAbstractType">
      <xs:sequence>
        <xs:element ref="AttributePredicate"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

4 Predicate Evaluation

A SAML authority that receives an incoming <AttributePredicateQuery> evaluates the truth of the queried <AttributePredicate> in a way that is semantically equivalent to evaluating an XACML request [XACML] containing all known attributes of the concerned subject against an XACML policy – with rule-combining algorithm "Permit-overrides" – that contains a single applicable rule – with effect "Permit" – whose condition solely contains the queried predicate expressed as <xacml:Apply> element. An XACML Policy Decision Point (PDP) evaluating such request will return one of the following three evaluation decisions:

"Permit"

In case the XACML PDP returns decision "Permit", the SAML authority MAY return a SAML response with status code urn:oasis:names:tc:SAML:2.0:status:Success. If the IncludePredicateInResponse XML attribute of the corresponding query is present and "true", an assertion with an attribute predicate statement containing an attribute predicate that is equal to the queried attribute predicate MUST be included in the response. Here, equality is defined either as string equality (i.e., including whitespaces), or, in case the assertion is signed, as equality under the XML canonicalization method used to compute the XML Signature (see [XMLSig] for more information on canonicalization methods). Note that there may be situations where the SAML authority MAY not return an assertion containing an attribute predicate statement, even though the XACML PDP returns decision "Permit". For example, when additional policies apply or when the subject does not give permission to return the queried assertion.

"NotApplicable"

In case the XACML PDP returns decision "NotApplicable", which arises when the SAML authority knows all required attributes but they do not satisfy the predicate, the SAML authority MAY return a response with top-level status code urn:oasis:names:tc:SAML:2.0:status:Responder and second-level status code urn:oasis:names:tc:SAML:2.0:status:PredicateFalse.

"Indeterminate"

In case the XACML PDP returns decision "Indeterminate", the SAML authority MUST return a response with top-level status code urn:oasis:names:tc:SAML:2.0:status:Responder and second-level status code urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile. In particular, this situation arises when an <xacml:AttributeDesignator> element that must be present is not known by the SAML authority with respect to the concerned subject.

If the XACML PDP returns any decision other than "Permit", then the SAML authority MUST NOT return a response with status code urn:oasis:names:tc:SAML:2.0:status:Success.

In the following situations a SAML authority MUST return a response with top-level status code urn:oasis:names:tc:SAML:2.0:status:Requester and second-level status code urn:oasis:names:tc:SAML:2.0:status:InvalidPredicate:

- an attribute predicate contains an <xacml:AttributeDesignator> element with a category different from urn:oasis:names:tc:xacml:1.0:subject-category:access-subject,
- an attribute predicate contains an <xacml:AttributeDesignator> element with the Issuer value different from the one specified in the <saml:Issuer> element of the enclosing query, or
- an attribute predicate contains an <xacml:AttributeSelector> or <xacml:VariableReference> element.

Table 1 summarizes the prescribed status codes and their interpretations.

184 *Table 1 Status codes and interpretations*

Evaluation Decision	Top-level `<StatusCode>`	Second-level `<StatusCode>`	Interpretation
“Permit”	“Success”	Don’t care	Attribute predicate is true
“Permit”	“Requester”	Appropriate status code	Interpretation according to second-level status code For example, additional policies apply, subject does not give permission, etc.)
Different from “Permit”	MUST NOT be “Success”	Appropriate status code	Interpretation according to top-level and second-level status code
“NotApplicable”	“Responder”	“PredicateFalse”	Attribute predicate is false
“Indeterminate”	“Requester”	“UnknownAttrProfile”	Attribute contained in predicate not known by SAML authority
Don’t care	“Requester”	“InvalidPredicate”	Malformed attribute predicate

5 Examples (non-normative)

The following example query requests from SAML authority `idp.example.com` an assertion that the subject's date of birth is before 1 January 1993.

```
<AttributePredicateQuery
  Version="2.0"
  ID="query23a0821cf186ea0a22e3818750a809b6cb3b4cda"
  IssueInstant="2011-02-28T23:59:58"
  IncludePredicateInResponse="true">

  <samla:Issuer>idp.example.com</samla:Issuer>
  <samla:Subject>
    <samla:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
      pseudonym123456
    </samla:NameID>
  </samla:Subject>

  <AttributePredicate FriendlyDescription="The requestor is over 18 years of age.">
    <xacml:Apply
      FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal">
      <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
        <xacml:AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#date"
          MustBePresent="true"
          Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
          AttributeId="urn:example:identity:birthdate"/>
      </xacml:Apply>
      <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">
        1993-01-01
      </xacml:AttributeValue>
    </xacml:Apply>
  </AttributePredicate>

</AttributePredicateQuery>
```

If the subject's birth date is indeed before 1 January 1993, the response of the SAML authority could be the following. (Note that the signature is not valid and cannot be successfully verified.)

```
<samlp:Response
  Version="2.0"
  ID="responsebd6996d271d23129dc2068c497ea3415f00b8b73"
  InResponseTo="query23a0821cf186ea0a22e3818750a809b6cb3b4cda"
  IssueInstant="2011-02-28T23:59:59">

  <samla:Issuer>idp.example.com</samla:Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>

  <samla:Assertion
    IssueInstant="2011-02-28T23:59:59"
    ID="assertion116fcc68e6a3feb788e5c89520b5f4eadf5ebbcf"
    Version="2.0">
    <samla:Issuer>idp.example.com</samla:Issuer>
    <ds:Signature>
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference URI="#assertion116fcc68e6a3feb788e5c89520b5f4eadf5ebbcf">
          <ds:Transforms>
            <ds:Transform
              Algorithm="http://www.w3.org/2000/09/xmldsig#envelopedsignature"/>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <InclusiveNamespaces PrefixList="#default xacml samla samlp ds xsi"
              xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transform>
          </ds:Transforms>
```

```
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>192d3b53f6c4fda041ff36899a91422e9e9a8a2b</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
        hq4zk+ZknjggCQgZm7ea8fI79gJEsRy3E8LHDpYXWQIgZpkJN9CMLG8ENR4Nrw+n
        7iyzixBvKXX8P53BTCT4VghPBWhFYSt9tHWu/AtJfOTh6qaAsNdeCyG86jmtp3TD
        MwuL/cBUj2OtBZOQMFn7jQ9YB7klIz3RqVL+wNmeWI4=
      </ds:SignatureValue>
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>
            MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>

    <samla:Subject>
      <samla:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
        pseudonym123456
      </samla:NameID>
    </samla:Subject>

    <samla:Statement xsi:type="ap:AttributePredicateStatementType">
      <AttributePredicate FriendlyDescription="The requestor is over 18 years of age.">
        <xacml:Apply
          FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal">
          <xacml:Apply
            FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
            <xacml:AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#date"
              MustBePresent="true"
              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              AttributeId="urn:example:identity:birthdate"/>
          </xacml:Apply>
          <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">
            1993-01-01
          </xacml:AttributeValue>
        </xacml:Apply>
      </AttributePredicate>
    </samla:Statement>

  </samla:Assertion>

</samlp:Response>
```

6 Conformance

A conforming application implements all of the mandatory features of the normative sections.

A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

[Participant Name, Affiliation | Individual Member]

[Participant Name, Affiliation | Individual Member]

B. Revision History

Revision	Date	Editor	Changes Made
[Rev number]	[Rev Date]	[Modified By]	[Summary of Changes]