SAML v2.0 Protocol Extension for Requesting Attributes per Request

Version 1.0

Working Draft 01

23 November 2016

Technical Committee:
OASIS Security Services (SAML) TC

Chairs:
Thomas Hardjono (hardjono@mit.edu), MIT
Nate Klingenstein (ndk@internet2.edu), Internet2

Editors:
Madalina Sultan (madalina.sultan@connectis.nl), Connectis
Mert Aybat (mert.aybat@connectis.nl), Connectis
Robert van Herk (robert.van.herk@connectis.nl), Connectis
Martijn Kaag (martijn.kaag@connectis.nl), Connectis

Additional artifacts:
This prose specification is one component of a Work Product that also includes:
XSD schema: sstc-req-attr-ext.xsd

Related work:
This specification is related to:
Security Assertion Markup Language (SAML) v2.0, OASIS Standard.

Abstract:
This specification defines an extension to SAML 2.0 protocol specification SAMLCore. The extension provides a more flexible structure for expressing which combination of Attributes are requested by service providers in comparison to the existing mechanisms. This is achieved by allowing a wrapper element req-attr:RequestedAttributes over a list of md:RequestedAttribute elements in samlp:AuthnRequest, which is an alternative to specifying these elements in SAML metadata. The extension thereby allows service providers to specify attributes per request. The expectation is that the extension is used to limit the set of Attributes returned to the service provider, thereby supporting (new) privacy regulations that require data minimization.

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.

Any machine-readable content (aka Computer Language Definitions) declared Normative for this Work Product must also be provided in separate plain text files. In the event of a discrepancy between such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

URI patterns:
Initial publication URI:
http://docs.oasis-open.org/security/saml-protoc-req-attr-req/v1.0/csd01/saml-protoc-req-attr-req-v1.0 -csd01.odt
Permanent "Latest version" URI:
http://docs.oasis-open.org/security/saml-protoc-req-attr-req/v1.0/saml-protoc-req-attr-req-v1.0.odt
(Managed by OASIS TC Administration; please don’t modify.)
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>Terminology</td>
<td>4</td>
</tr>
<tr>
<td>Normative References</td>
<td>4</td>
</tr>
<tr>
<td>Non-Normative References</td>
<td>4</td>
</tr>
<tr>
<td>SAML V2.0 Protocol Extension For Requesting Attributes Per Request</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Element <a href="">req-attr:RequestedAttributes</a></td>
<td>5</td>
</tr>
<tr>
<td>2.2 Example</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Processing Rules</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Metadata Considerations</td>
<td>6</td>
</tr>
<tr>
<td>2.4.1 Metadata Example</td>
<td>7</td>
</tr>
<tr>
<td>2.4.2 Metadata Processing Rules</td>
<td>7</td>
</tr>
<tr>
<td>2.5 Security Considerations</td>
<td>7</td>
</tr>
<tr>
<td>Conformance</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Implementation Conformance</td>
<td>8</td>
</tr>
<tr>
<td>3.2 Conformance as a Service Provider</td>
<td>8</td>
</tr>
<tr>
<td>3.3 Conformance as an Identity Provider</td>
<td>8</td>
</tr>
<tr>
<td>Appendix A: Acknowledgments</td>
<td>9</td>
</tr>
<tr>
<td>Appendix B: Revision History</td>
<td>9</td>
</tr>
<tr>
<td>Appendix C: sstc-req-attr-ext.xsd</td>
<td>10</td>
</tr>
</tbody>
</table>
1. Introduction

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

1.3 Non-Normative References
2. SAML V2.0 Protocol Extension For Requesting Attributes Per Request

This specification defines an extension to SAML 2.0 protocol specification SAMLCore that provides a more flexible structure for expressing combinations of Attributes for Web Single Sign On than do existing mechanisms.

Existing mechanisms for indicating the requested attributes depend on md:RequestedAttribute elements in metadata and samlp:AttributeConsumingServiceIndex in the samlp:AuthnRequest. This approach has two limitations. First, all possible combinations of attributes should be known and exchanged beforehand. Second, the number of possible combination of attributes is limited because of AttributeConsumingServiceIndex is of type short. In federations with many different attributes and where data minimization is required, the number of possible combinations easily exceeds the maximum number of 32767. It only takes a set of 15 distinct attributes, and the sum of all their possible combinations is 2^15= 32768.

This specification provides service providers a more flexible way of requesting Attributes by allowing them to specify a req-attr:RequestedAttributes element, wrapping a list of md:RequestedAttribute inside the extensions part of a samlp:AuthnRequest, instead of specifying them in their metadata.

The extension thereby allows service providers to specify attributes per request.

Unless specifically noted, nothing in this document should be taken to conflict with the SAML 2.0 protocol specification SAMLCore. Readers are advised to familiarize themselves with that specification first.

2.1 Element <req-attr:RequestedAttributes>

The element <req-attr:RequestedAttributes>, of complex type req-attr:RequestedAttributesType, wraps a list of <md:RequestedAttribute> elements. In this way, the Service Provider specifies its desire that the resultant assertion contains a list of AttributeStatements expressing the values of the queried attribute.

The following schema fragment defines the <req-attr:RequestedAttribute> element:

```xml
<element name="RequestedAttributes" type="req-attr:RequestedAttributesType"/>
<complexType name="RequestedAttributeType">
  <sequence>
    <element ref="md:RequestedAttribute" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

2.2 Example

The following is an example of a <samlp:Extensions> element in <samlp:AuthnRequest> where the SP is expressing that it desires the resultant assertions to contain a <saml:AttributeStatement> that contains the LastName and FirstName, optionally includes the Email and includes the Roles of the user that match 'End User' or 'Administrator'.
2.3 Processing Rules

A `<samlp:Extensions>` element is included in an `AuthnRequest` message by placing it in the optional `<req-attr:RequestedAttributes>` element. Due to existing processing requirements, all extensions are explicitly deemed optional. Therefore, senders SHOULD only include this extension when they can be reasonably confident that the extension will be understood by the recipient. The SAML V2.0 metadata extension defined in section 2.6 MAY be used for this purpose.

Each `<md:RequestedAttribute>` describes a SAML attribute the requester desires or requires to be supplied by the identity provider in the `Response` message. The identity provider MAY use this information to populate one or more `<saml:AttributeStatement>` elements in the assertion(s) it returns.

The `<md:RequestedAttribute>` elements marked as required are just informational, expressing that they are mandatory for the ServiceProvider. The Identity Provider MAY choose to ignore this flag and do not include these attributes in the response, if it cannot provide them.

If a Service Provider uses the `<req-attr:RequestedAttributes>` extensions, then it MUST NOT use the `<samlp:AttributeConsumingServiceIndex>` index as well.

If an Identity Provider receives from a non-compliant Service Provider an AuthenticationRequest containing both a `<samlp:AttributeConsumingServiceIndex>` and a `<req-attr:RequestedAttributes>` section inside extensions, then the Identity Provider receiving this message MUST ignore the extension and send back the list of attributes associated to the AttributeConsumingService index.

In order to keep backwards compatibility, the extension proposed by this document MUST be processed by Identity Providers only if there is no AttributeConsumingServiceIndex in the message.

2.4 Metadata Considerations

SAML Metadata MAY be used to indicate support for this protocol extension at particular protocol endpoints, using the extensions capabilities of metadata schema.

Support for this extension is expressed in SAMLV2.0 metadata [Saml2Meta] by adding a boolean-typed XML attribute to an element derived from `<md:EndpointType>` complex type, indicating that SAML request messages sent to that endpoint MAY include this extension.

The following schema fragment defines the `req-attr:suppportsRequestedAttributes` attribute:
2.4.1 Metadata Example

The example below shows a fragment of an `<md:SingleSignOnService>` element that advertises support for this extension. The namespace declaration must be in scope, but the prefix is arbitrary.

```xml
    req-attr:supportsRequestedAttributes="true" .../>
```

2.4.2 Metadata Processing Rules

If the Identity Provider metadata uses the `req-attr:supportsRequestedAttributes` flag, then the Service Provider MAY send the `<req-attr:RequestedAttributes>` element inside its AuthnRequest extensions.

If the Identity Provider metadata declares both the `req-attr:supportsRequestedAttributes` flag (set on true) and it also defines AttributeConsumingService indices with sets of associated attributes, the Service Provider MUST use the AttributeConsumingServiceIndex element if it can find a predefined set of attributes matching his needs, and do not use the `<req-attr:RequestedAttributes>` extension.

If there is no predefined set of attributes defined in the metadata of the IdentityProvider matching the ServiceProvider needs, then the Service Provider MAY choose to request the attributes through the `<req-attr:RequestedAttributes>` extension. But in this case, an AttributeConsumingServiceIndex MUST NOT be included in the request (see section 2.3)

2.5 Security Considerations

The identity provider MAY choose to ignore these attributes and populate the response with more or fewer attributes than provided. The identity provider MAY also ignore the `isRequired` attribute and continue processing if a user does not possess a specific attribute.

The service provider should therefore always inspect the returned attributes and should not rely on the identity provider for authorization.
3. Conformance

3.1 Implementation Conformance
An implementation is a conforming Service Provider if it meets the conditions in section 3.2. An implementation is a conforming Identity Provider if it meets the conditions in section 3.3.

3.2 Conformance as a Service Provider
A Service Provider implementation conforms to this specification if it meets the following conditions:
1. Supports sending the <req-attr:RequestedAttributes> element inside the <samlp:Extensions> of a <samlp:AuthnRequest>, as specified in section 2.1
2. Conforms to the Processing Rules for Service Providers defined in section 2.3

3.3 Conformance as an Identity Provider
An Identity Provider implementation conforms to this specification if it meets the following conditions:
1. Supports receiving and processing a <req-attr:RequestedAttributes> element, as specified in section 2.1
2. Conforms to the Processing Rules for Identity Providers defined in section 2.3
3. Optionally supports the metadata flag and its corresponding processing rules defined in section 2.4
Appendix A: Acknowledgments

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]

Appendix B: Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Editor</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01-09-2015</td>
<td>Mert Aybat</td>
<td>First Draft</td>
</tr>
<tr>
<td>2</td>
<td>19-01-2016</td>
<td>Mert Aybat</td>
<td>Adding in xsd and wrapping RequestedAttribute elements with RequestedAttributes element.</td>
</tr>
<tr>
<td>3</td>
<td>13-10-216</td>
<td>Madalina Sultan</td>
<td>Defined a new namespace for RequestedAttribute. Added supportsRequestedAttributes metadata flag</td>
</tr>
<tr>
<td>4</td>
<td>21-10-2016</td>
<td>Madalina Sultan</td>
<td>Added the req:RequestedAttributes element, as a wrapper for a list of md:RequestedAttribute elementss</td>
</tr>
<tr>
<td>5</td>
<td>21-11-2016</td>
<td>Madalina Sultan</td>
<td>Cosmetic changes</td>
</tr>
<tr>
<td>6</td>
<td>02-12-2016</td>
<td>Madalina Sultan</td>
<td>Added Conformance section and other changes based on feedback received during SSTC meeting</td>
</tr>
</tbody>
</table>
Appendix C: sstc-req-attr-ext.xsd

<?xml version="1.0" encoding="UTF-8"?>

    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified" attributeFormDefault="unqualified" blockDefault="substitution"
    version="1.0">
  <import namespace="urn:oasis:names:tc:SAML:2.0:metadata"
      schemaLocation="saml-schema-metadata-2.0.xsd"/>
  <annotation>
    <documentation>
      Document title: SAML V2.0 Protocol Extension For Requesting Attributes Per Request
      Document identifier: sstc-req-attr-ext
      Location: http://docs.oasis-open.org/security/saml/Post2.0/
      Revision history: V4.0 (October 2016): Document with integrated feedback
    </documentation>
  </annotation>

  <element name="RequestedAttributes" type="req-attr:RequestedAttributesType"/>

  <complexType name="RequestedAttributesType">
    <sequence>
      <element ref="md:RequestedAttribute" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <attribute name="supportsRequestedAttributes" type="boolean"/>
</schema>