Table of Contents

1. Introduction
   1.1 Notation
2. SAML Protocol Extension For Requesting Attributes Per Request
   2.1 Required Information
   2.2 Profile Overview
   2.3 Element <req-attr:RequestedAttribute>
   2.4 Example
   2.5 Processing Rules
   2.6 Metadata Considerations
      2.6.1 Metadata Example
   2.7 Security Considerations
3. References
   3.1 Normative References
Appendix A. Acknowledgments
Appendix B. Revision History
1. Introduction

SAML protocol extensions consist of elements defined for inclusion in the `<samlp:Extensions>` element that modify the behavior of SAML requesters and responders when processing such extended messages.

This specification defines an extension to the SAML 2.0 protocol specification that can be used to request specific Attributes to be returned to the service provider for Web Single Sign On. The extension allows service providers to express per request which specific Attributes may be returned in the response, which attributes are required and what values the service provider is interested in.

1.1 Notation

This specification uses normative text.

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.*

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:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>XML Namespace</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>saml:</td>
<td>urn:oasis:names:tc:SAML:2.0:assertion</td>
<td>This is the SAML V2.0 assertion namespace SAMLCore.</td>
</tr>
<tr>
<td>samlp:</td>
<td>urn:oasis:names:tc:SAML:2.0:protocol</td>
<td>This is the SAML V2.0 protocol namespace SAMLCore</td>
</tr>
<tr>
<td>md:</td>
<td>urn:oasis:names:tc:SAML:2.0:metadata</td>
<td>This is the SAML V2.0 metadata namespace. SAMLMeta</td>
</tr>
<tr>
<td>req-attr:</td>
<td>urn:oasis:names:tc:SAML:protocol:ext:req-attr</td>
<td>This is the namespace defined by this document and its accompanying schema [ReqAttrExt-xsd]</td>
</tr>
<tr>
<td>xsd:</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>This namespace is defined in the W3C XML Schema specification Schema1. In schema listings, this is the default namespace and no prefix is shown.</td>
</tr>
</tbody>
</table>

This specification uses the following typographical conventions in text:
<SAMLElement>, <ns:ForeignElement>Attribute, Datatype, OtherCode.
2. SAML V2.0 Protocol Extension For Requesting Attributes Per Request

2.1 Required Information

**Identification**: urn:oasis:names:tc:SAML:protocol:ext:req-attr  
**Contact Information**: security-services-comment@lists.oasis-open.org  
**Description**: Given below  
**Updates**: None

2.2 Profile Overview

This specification defines an extension to SAML 2.0 protocol specification [SAMLCore](https://docs.oasis-open.org/security/saml/protocol/v2.0/samlcore-2.0-os.pdf) 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 the `req-attr:RequestedAttribute` elements 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](https://docs.oasis-open.org/security/saml/protocol/v2.0/samlcore-2.0-os.pdf). Readers are advised to familiarize themselves with that specification first.

2.3 Element `<req-attr:RequestedAttribute>`

The element `<req-attr:RequestedAttribute>`, with complex type `md:RequestedAttributeType` specifies the SP desire that the resultant assertion contains an AttributeStatement expressing the value of the queried attribute.

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

```xml
<element name="RequestedAttribute" type="req-attr:RequestedAttributeType"/>
<complexType name="RequestedAttributeType">
  <sequence>
    <element ref="req-attr:AttributeValue" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Name" type="string" use="required"/>
  <attribute name="NameFormat" type="anyURI" use="optional"/>
  <attribute name="FriendlyName" type="string" use="optional"/>
  <attribute name="isRequired" type="boolean" use="optional"/>
</complexType>
```
2.4 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 an <AttributeStatement> that contains the LastName and FirstName, optionally includes the Email and includes the Roles of the user that match ‘End User’ or ‘Administrator’.

```xml
<samlp:Extensions>
  <req-attr:RequestedAttribute isRequired="true" Name="LastName"/>
  <req-attr:RequestedAttribute isRequired="true" Name="FirstName"/>
  <req-attr:RequestedAttribute Name="Email"/>
  <req-attr:RequestedAttribute Name="Role">
    <req-attr:AttributeValue>End User</req-attr:AttributeValue>
    <req-attr:AttributeValue>Administrator</req-attr:AttributeValue>
  </req-attr:RequestedAttribute>
</samlp:Extensions>
```

2.5 Processing Rules

A list of RequestedAttribute is included in an AuthnRequest message by placing it in the optional <samlp:Exensions> 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 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 RequestedAttributes marked as required are just informational, expressing that they are mandatory for the ServiceProvider. The IdentityProvider MAY choose to ignore this flag and do not include these attributes in the response, if it cannot provide them.

2.6 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:

```xml
<attribute name="supportsRequestedAttributes" type="boolean"/>
```
2.6.1 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
<md:SingleSignOnService
    req-attr:supportsRequestedAttributes="true" .../>
```

2.7 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 SAML V2.0 Protocol Extension for RequestedAttributes per Request

An asserting party can claim to support the RequestedAttribute extension by simply including this element inside the `<samlp:Extensions>` element of a `<samlp:AuthnRequest>`.

A relying party can claim to support the RequestedAttribute extension by including the `supportsRequestedAttributes` flag in its exposed metadata and by demonstrating the ability to return the values of the requested attribute(s). Successful processing MAY consist of no changes to a relying party's behavior.
4. References
The following works are referenced in the body of this specification.

4.1 Normative References

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-req-atr-ext.xsd">http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-req-atr-ext.xsd</a></td>
</tr>
</tbody>
</table>
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</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added supportsRequestedAttributes metadata flag</td>
</tr>
</tbody>
</table>
Appendix C. sstc-req-attr-ext.xsd

When submitting the proposal to Oasis SAML TC, the following XSD should be attached and uploaded to http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-req-attr-ext.xsd

```xml
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="1.0">
    <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: V1.0 (October 2016): Initial version.
        </documentation>
    </annotation>
    <element name="RequestedAttribute" type="req-attr:RequestedAttributeType"/>
    <complexType name="RequestedAttributeType">
        <sequence>
            <element ref="req-attr:AttributeValue" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="Name" type="string" use="required"/>
        <attribute name="NameFormat" type="anyURI" use="optional"/>
        <attribute name="FriendlyName" type="string" use="optional"/>
        <attribute name="isRequired" type="boolean" use="optional"/>
    </complexType>
    <element name="AttributeValue" type="anyType" nillable="true"/>
    <attribute name="supportsRequestedAttributes" type="boolean"/>
</schema>
```