SAML 2.0 Profile of SPML 2.0 Submission

Submitting Organizations

AOL
BMC Software
HP
Intel
Neustar
Sun Microsystems
Tripod Technology Group

Introduction

This document describes a submission to the OASIS Security Services Technical Committee for a new SAML 2.0 Profile of SPML 2.0 for federated provisioning. The Federated Provisioning Profile is designed to support the "Bulk Provisioning" use case where an Identity Management Lifecycle exists between the IdP and SP.

The proposed profile will use the OASIS Service Provisioning Markup Language (SPML) 2.0 standard as the provisioning protocol with elements from the SAML 2.0 Assertion schema as the provisioning data.

For the purposes of this submission, the examples are given in terms of the IdP making provisioning requests to the SP. There are valid use cases for the SP to make provisioning requests to the IdP, but there are no examples of this in this submission, as they would be redundant and would add no additional insight.

Overview

Provisioning Data

Object Identifiers

All objects used in SPML 2.0 must have an identifier that is unique within the namespace of the provisioning service. This is known as the Provisioning Service Object ID (PSO ID). The format of the PSO ID is profile-specific. For the SAML 2.0 Federated Provisioning Profile, the SAML 2.0 NameIdentifier element is used.

Provisioning Object Identifier Example:

```
<spml:pso xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0">
  <spml:psoID>
```

```
    <saml:NameIdentifier
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">cn=John
Doe,dc=acme,dc=com</saml:NameIdentifier>
  </spml:psoID>
</spml:pso>
```

Object Data

Each Provisioning Service Object (PSO) contains an identifier and data. For the SAML 2.0 Federated Provisioning Profile, the PSO Data consists of a set of SAML 2.0 Attribute elements.

Provisioning Object Data Example:

```
<spml:pso xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0">
  <spml:psoID>
    <saml:NameIdentifier
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">cn=John
Doe,dc=acme,dc=com</saml:NameIdentifier>
  </spml:psoID>
  <spml:data>
    <saml:Attribute Name="Email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
  <saml:AttributeValue>jdoe@acme.com</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute Name="cn"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
      <saml:AttributeValue>John Doe</saml:AttributeValue>
    </saml:Attribute>
  </spml:data>
</spml:pso>
```

Schema

SPML 2.0 supports a provisioning schema (metadata) that allows for a provisioning service to publish definitions of supported object types. The schema definition language is profile-specific. As SAML 2.0 defines no such schema definition language, this proposed Federated Provisioning Profile defines one.

For the purpose of this proposal, two elements, samlprov:objectDef and samlprov:attributeDef are used to define the supported object classes that a service can provision and the attributes that are associated with the object classes. The namespace samlprov is defined in the namespace urn:oasis:names:tc:SAML:2.0:provision.

Provisioning Schema Example:

```
<spml:schema xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:samlprov="urn:oasis:names:tc:SAML:2:0:provision">
  <samlprov:schema>
    <samlprov:objectClassDefinition
name="urn:summittrust:account">
      <samlprov:attributeDefinition name="uid"
required="true"
nameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic" />
      <samlprov:attributeDefinition name="email"
nameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic" />
    </samlprov:objectClassDefinition>
  <samlprov:schema>
</spml:schema>
```

Core Capability

The SPML 2.0 protocol defines a required core capability with the following operations

- Add – add a new object
- Modify – modify an existing object
- Delete – delete an existing object
- Lookup – lookup an existing object
- List targets – list all targets and the provisioning schema for each target

Add Request

The SPML 2.0 Add Request provisions a new object. There are two main flavors of Add Request; the IdP may specify the new object identifier along with the object data, or it may specify the data alone and rely on the SP to define the new identifier.

Add Request Example 1 - IdP defines the new PSO ID

```
Request:

<spml:addRequest
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:samlprov="urn:oasis:names:tc:SAML:2:0:provision"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
```

```
  <spml:psoID>
    <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
  </spml:psoID>
  <spml:data>
    <samlprov:objectDef name="urn:summittrust:account" />
    <saml:Attribute Name="uid"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
      <saml:AttributeValue>jdoe</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute Name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
  <saml:AttributeValue>jdoe@acme.com</saml:AttributeValue>
    </saml:Attribute>
  </spml:data>
</spml:addRequest>

Response:

<spml:addResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0">
```

If the IdP does not supply the PSO ID, the SP must determine the PSO ID and return it in the response.

Add Request Example 2- SP defines the new PSO ID

```
Request:

<spml:addRequest
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:samlprov="urn:oasis:names:tc:SAML:2:0:provision"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <spml:data>
    <samlprov:objectDef name="urn:summittrust:account">
    <saml:Attribute Name="uid"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
      <saml:AttributeValue>jdoe</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute Name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
```

```
  <saml:AttributeValue>jdoe@acme.com</saml:AttributeValue>
    </saml:Attribute>
  </spml:data>
</spml:addRequest>

Response:

<spml:addResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
<spml:psoID>
    <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
  </spml:psoID>
</spml:addResponse>
```

Modify Request

The SPML 2.0 Modify Request modifies an existing object. The SPML 2.0 modification element defines the attribute modification type (add, replace, delete).

Modify Request Example - IdP modifies an existing PSO

```
Request:

<spml:modifyRequest
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:saml=" urn:oasis:names:tc:SAML:2.0:assertion">
  <spml:psoID>
    <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
  </spml:psoID>
  <spml:modification modificationMode="spml:replace" >
    <saml:Attribute Name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
  <saml:AttributeValue>jane_doe@acme.com</saml:AttributeVal
ue>
    </saml:Attribute>
  </spml:modification>
</spml:modifyRequest>

Response:
```

```
<spml:modifyResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0">
```

Delete Request

The SPML 2.0 Delete Request deletes an existing object.

Delete Request Example - IdP deletes an existing PSO

```
Request:

<spml:deleteRequest
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:saml=" urn:oasis:names:tc:SAML:2.0:assertion">
  <spml:psoID>
    <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
  </spml:psoID>
</spml:deleteRequest>

Response:

<spml:deleteResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0">
```

Lookup Request

The Lookup request returns a single PSO. The Lookup request may optionally specify that no data be returned, in which case the lookup request is an existence test.

Lookup Request Example - IdP looks up an existing PSO

```
Request:

<spml:lookupRequest
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <spml:psoID>
    <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
  </spml:psoID>
```

```
</spml:lookupRequest>

Response:

<spml:lookupResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
<spml:pso>
    <spml:psoID>
      <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
    </spml:psoID>
    <spml:data>
      <samlprov:objectDef name="urn:summittrust:account" />
      <saml:Attribute Name="uid"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
      <saml:AttributeValue>jdoe </saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">

  <saml:AttributeValue>jdoe@acme.com</saml:AttributeValue>
      </saml:Attribute>
    </spml:data>
  </spml:pso>
</spml:lookupResponse>
```

List Targets Request

The List Targets Request returns a list of provisioning targets (namespaces) that the provisioning service supports. In addition to the target names, the supported provisioning schema and capabilities for each target are returned in the response.

List Targets Request Example - IdP lists targets for a service

```
Request:

<spml:listTargetsRequest
xmlns:spml="urn:oasis:names:tc:SPML:2:0" />

Response:
```

```
<spml:listTargetsResponse
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:samlprov="urn:oasis:names:tc:SPML:2:0:provision"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <spml:target  targetID="urn:acme:sp1">
    <spml:schema>
      <samlprov:schema>
        <samlprov:objectDef name="urn:summittrust:account">
          <samlprov:attributes>
            <samlprov:attributeDef  name="uid"
required="true"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic" />
            <samlprov:attributeDef  name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic" />
          </samlprov:attributes>
        </spmldsml:objectDef>
      <samlprov:schema>
    </spml:schema>
  </spml:target>
</spml:listTargetsResponse>
```

Search Capability

The SPML 2.0 Search Capability allows the IdP to search the SP for a list of existing PSOs. This is used to support reconciliation of accounts and other object types. The search capability supports the use of filters to scope which PSOs should be returned and also to select a subset of the data to return.

Both the search filtering and attribute selection mechanisms are profile-specific. The attribute definition elements defined in the urn:oasis:names:tc:SAML:2:0:provision schema can be used to define attribute selection criteria. A filtering mechanism will also need to be defined.

Search Request Example - IdP searches the SP for existing PSOs

```
Request:
```

```
<spmlsearch:searchRequest
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:spmlsearch="urn:oasis:names:tc:SPML:2:0:search"
xmlns:samlprov="urn:oasis:names:tc:SAML:2:0:provision"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <spmlsearch:query>
    <spml:basePSOID>
      <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">ou=accounting,
o=acme.com</saml:NameID>
    </spml:basePSOID >
    <samlprov:attributes>
      <samlprov:attributeDef  name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic" />
    </samlprov:attributes>
    <spmlsearch:and>
      <samlprov:equalityMatch name="email">
        <samlprov:value>jdoe@acme.com</samlprov:value>
      </samlprov:equalityMatch>
    </spmlsearch:and>
  </spmlsearch:query>
</spmlsearch:searchRequest>
```

Response:

```
<spmlsearch:searchResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:spmlsearch="urn:oasis:names:tc:SPML:2:0:search"
xmlns:saml="urn:oasis:names:tc:SPML:2:0"
xmlns:samlprov="urn:oasis:names:tc:SAML:2:0:provision" >
  <spml:pso>
    <spml:psoID>
      <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
    </spml:psoID>
    <spml:data>
      <samlprov:objectDef name="urn:summittrust:account" />
      <saml:Attribute Name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">

  <saml:AttributeValue>jdoe@acme.com</saml:AttributeValue>
      </saml:Attribute>
```

```
    </spml:data>
  </spml:pso>
...
</spmlsearch:searchResponse>
```

The Search Capability also defines an iterator for handling large data sets. If the SP determines that there are too many search results to return in one response, it may return a subset of the results along with an iterator that the IdP may use to retrieve additional search results in a subsequent call.

Response with iterator:

```
<spmlsearch:searchResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:spmlsearch="urn:oasis:names:tc:SPML:2:0:search"
xmlns:saml="urn:oasis:names:tc:SPML:2:0"
xmlns:samlprov="urn:oasis:names:tc:SAML:2:0:provision" >
  <spml:pso>
    <spml:psoID>
      <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
    </spml:psoID>
    <spml:data>
      <samlprov:objectDef name="urn:summittrust:account" />
      <saml:Attribute Name="email"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">

  <saml:AttributeValue>jdoe@acme.com</saml:AttributeValue>
      </saml:Attribute>
    </spml:data>
  </spml:pso>
...
  <spmlsearch:iterator ID="DF56HJ8" />
</spmlsearch:searchResponse>
```

Updates Capability

The SPML 2.0 Updates Capability allows the IdP to query the SP for changes that have occurred since a specific point in time. By asking for deltas since a specific point in time, the IdP can perform reconciliation without needing to search for all PSOs on the IdP.

Updates Request Example - IdP queries the SP for modified PSOs

```
Request:

<spmlupdates:updatesRequest updatedSince= "2006-04-
25T18:48:54Z"
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:spmlupdates="urn:oasis:names:tc:SPML:2:0:updates"/>

Response:

<spmlupdates:updatesResponse status="spml:success"
xmlns:spml="urn:oasis:names:tc:SPML:2:0"
xmlns:spmlupdates="urn:oasis:names:tc:SPML:2:0:updates"
xmlns:saml="urn:oasis:names:tc:SPML:2:0">
  <spmlupdates:update timestamp="2006-04-25T18:48:54Z"
updateKind="spml:modify">
    <spml:psoID>
      <saml:NameID
Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">uid=jdoe, o=acme.com</saml:NameID>
      </spml:psoID>
  </spmlupdates:update>
...
</spmlupdates:updatesResponse>
```

The Updates Capability also defines an iterator for handling large data sets. If the SP determines that there are too many updates results to return in one response, it may return a subset of the results along with an iterator that the IdP may use to retrieve additional updates results in a subsequent call.

Profile XSD

The SAML 2.0 Profile of SPML 2.0 requires normatively defined elements for defining the SPML 2.0 Provisioning Schema and Search Filters. The following XSD defines those elements.

```
<!--************************************************-->
<!--  sstc_saml2_prov.xsd                             -->
<!--                                                  -->
<!--  Schema for the SAML 2.0 Profile for SPML 2.0    -->
<!--************************************************-->

<schema
targetNamespace="urn:oasis:names:tc:SAML:2:0:provision"
```

```
  xmlns:samlprov="urn:oasis:names:tc:SAML:2:0:provision"
  xmlns:spml="urn:oasis:names:tc:SPML:2:0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace='urn:oasis:names:tc:SPML:2:0'/>

  <complexType name="AttributeDefinitionType">
    <complexContent>
      <extension base="spml:ExtensibleType">
        <attribute name="name" type="string" use="required"
/>
        <attribute name="nameFormat" type="anyURI"
          use="optional" />
        <attribute name="required" type="boolean"
use="optional"
          default="false" />
        <attribute name="multivalued" type="boolean"
          use="optional" default="false" />
        <attribute name="type" type="QName" use="optional"
/>
        <attribute name="friendlyName" type="string"
          use="optional" />
        <attribute name="description" type="string"
          use="optional" />
      </extension>
    </complexContent>
  </complexType>

  <complexType name="ObjectClassDefinitionType">
    <complexContent>
      <extension base="spml:ExtensibleType">
        <sequence>
          <element name="attributeDefinition"
            type="samlprov:AttributeDefinitionType"
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="description" type="string"
          use="optional" />
```

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

  <complexType name="SchemaType">
    <complexContent>
      <extension base="spml:ExtensibleType">
        <sequence>
          <element name="objectClassDefinition"
            type="samlprov:ObjectClassDefinitionType"
minOccurs="0"
            maxOccurs="unbounded" />
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="AttributeValueAssertion">
    <complexContent>
      <extension base="spml:QueryClauseType">
        <sequence>
          <element name="value" type="anyType" />
        </sequence>
        <attribute name="name" type="string" use="required"
/>
        <attribute name="nameFormat" type="anyURI"
          use="optional" />
      </extension>
    </complexContent>
  </complexType>
  <complexType name="AttributeDescription">
    <complexContent>
      <extension base="spml:QueryClauseType">
        <attribute name="name" type="string" use="required"
/>
        <attribute name="nameFormat" type="anyURI"
          use="optional" />
      </extension>
    </complexContent>
  </complexType>
  <complexType name="SubstringFilter">
    <complexContent>
      <extension base="spml:QueryClauseType">
        <sequence>
          <element name="initial" type="string"
minOccurs="0" />
          <element name="any" type="string" minOccurs="0"
```

```
            maxOccurs="unbounded" />
          <element name="final" type="string" minOccurs="0"
/>
        </sequence>
        <attribute name="name" type="anyURI" use="required"
/>
        <attribute name="nameFormat" type="anyURI"
          use="optional" />
      </extension>
    </complexContent>
  </complexType>

  <element name="schema" type="samlprov:SchemaType" />
  <element name="equalityMatch"
    type="samlprov:AttributeValueAssertion" />
  <element name="substrings"
type="samlprov:SubstringFilter" />
  <element name="greaterOrEqual"
    type="samlprov:AttributeValueAssertion" />
  <element name="lessOrEqual"
type="samlprov:AttributeValueAssertion" />
  <element name="present"
type="samlprov:AttributeDescription" />
  <element name="approxMatch"
type="samlprov:AttributeValueAssertion" />
</schema>
```