BoxIDG

This documentation will help you integrate your identity services with Box through Internet2's NET+ program. Associated portions of the NET+ Identity Guidance for Services are noted to by section.

- Discovery and Authentication
- Attributes
    - Privileges
- Provisioning
    - Deprovisioning
- Logout
- Implementation
    - Metadata Support
    - SAML Assertions
- Non-Browser Access
- Other Notes
- Example Configuration for SAML Implementations

Discovery and Authentication

Box performs IdP discovery using the email address entered on the Box email/password interface(1.2.2). Customized discovery URL's can also be set up for your organization by Box(1.1.1, example).

Attributes

Box can receive a number of user attributes with configurable mappings between internal Box attributes and external attributes.

Only email address is mandatory, as it is the primary user identifier. As such, EPPN is a good candidate for use as "email address". However, this should also be a routable email address because Box sends a variety of important messages via email to this address.

Box is able to map custom attribute names to the conceptual attributes listed below, but the preference is to rely on the recommended SAML attribute names.

Box Attribute	Recommended SAML Attribute Name	Optional
First Name	urn:oid:2.5.4.42	Yes
Last Name	urn:oid:2.5.4.4	Yes
Email Address	urn:oid:1.3.6.1.4.1.5923.1.1.1.6	No

Users are able to change their displayed first name and last name in their Box account.

Privileges

Box manages access control and privileges separately from federated identity, as in 2.5.3.

Box supports groups by representing them within Box. Groups may be provisioned to Box using the attribute name `http://schemas.xmlsoap.org/claims/Group`. When a resource is shared with a group in Box, the individuals in the group are listed as having privileges. Updates to group membership are not automatically reflected for existing resources.

Box does not support an explicit eligibility attribute. Ineligibility can be expressed indirectly by suppressing the release of all attributes to Box when an ineligible user authenticates.

Provisioning

Users may be provisioned to Box in two ways. You can use front channel provisioning (3.1) and provision users that can successfully authenticate at your IdP for this SP. You may also use the back channel (3.2) using a custom provisioning API.

Once a user has been provisioned, changes in attributes received via the front channel will not automatically result in changes to their Box representation.

Deprovisioning

Users can be deprovisioned in Box in a variety of ways. Special considerations will need to be made when deprovisioning students using the API since students are eligible to maintain their Box account after leaving your organization.

Logout

Box performs local logout (5.1.2) or can redirect users upon logout to any URL determined by the campus. This URL could provide information about the SSO session at the IdP, or terminate it, or allow for other options at the school's discretion.

Implementation

Box uses Ping Federate as its SAML solution.

A high level SSO Overview.

Metadata Support

Box requires manual metadata provisioning. You will need to supply Box with information about your provider. This is accomplished by completing and submitting a Federation Questionnaire hosted within Box.

Box's Metadata is available through InCommon with the entityID `https://services.box.com/sp`.

SAML Assertions

By default, Box expects SAML assertions to be unencrypted. Optionally, they will support encrypted assertions (such as Shibboleth), however, this support must be specifically requested as part of their SSO configuration. Under this configuration, Box uses the same certificate for both signing and encryption

Non-Browser Access

Box has desktop and mobile device access that is provided using a custom protocol over HTTPS with a custom persistent token. This token is bootstrapped from the SSO transaction.

Other Notes

Example Configuration for SAML Implementations

Download and copy Box's Metadata as a flat file to your IdP's metadata directory. By default, Shibboleth expects to do encrypted assertions which Box optionally supports. You will need to request this support specifically from Box if you plan to use the encrypted assertions Metadata. Otherwise, you will need to take additional steps (not covered here) to ensure your relying-party.xml is configured use the plain assertions Metadata.

Ensure you have the appropriate attribute definitions for eppn, sn, and givenName in `attribute-resolver.xml`:

```
  <!--  *********   eduPersonPrincipalName   ********** -->
  <resolver:AttributeDefinition xmlns="urn:mace:shibboleth:2.0:resolver:ad" id="eduPersonPrincipalName" xsi:
type="Scoped" scope="uncg.edu" sourceAttributeID="cn">
    <resolver:Dependency ref="myAD"/>
    <resolver:AttributeEncoder xmlns="urn:mace:shibboleth:2.0:attribute:encoder" xsi:type="SAML1ScopedString"
name="urn:mace:dir:attribute-def:eduPersonPrincipalName"/>
    <resolver:AttributeEncoder xmlns="urn:mace:shibboleth:2.0:attribute:encoder" xsi:type="SAML2ScopedString"
name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" friendlyName="eduPersonPrincipalName"/>
  </resolver:AttributeDefinition>

  <!--  *********   sn   ********** -->
  <resolver:AttributeDefinition xmlns="urn:mace:shibboleth:2.0:resolver:ad" id="sn" xsi:type="Simple"
sourceAttributeID="sn">
    <resolver:Dependency ref="myAD"/>
    <resolver:AttributeEncoder xmlns="urn:mace:shibboleth:2.0:attribute:encoder" xsi:type="SAML1String" name="
urn:mace:dir:attribute-def:sn"/>
    <resolver:AttributeEncoder xmlns="urn:mace:shibboleth:2.0:attribute:encoder" xsi:type="SAML2String" name="
urn:oid:2.5.4.4" friendlyName="sn"/>
  </resolver:AttributeDefinition>

  <!--  *********   givenName   ********** -->
  <resolver:AttributeDefinition xmlns="urn:mace:shibboleth:2.0:resolver:ad" id="givenName" xsi:type="Simple"
sourceAttributeID="givenName">
    <resolver:Dependency ref="myAD"/>
    <resolver:AttributeEncoder xmlns="urn:mace:shibboleth:2.0:attribute:encoder" xsi:type="SAML1String" name="
urn:mace:dir:attribute-def:givenName"/>
    <resolver:AttributeEncoder xmlns="urn:mace:shibboleth:2.0:attribute:encoder" xsi:type="SAML2String" name="
urn:oid:2.5.4.42" friendlyName="givenName"/>
  </resolver:AttributeDefinition>
```

Add a new `AttributeFilterPolicy` to `attribute-filter.xml`:

```
<!-- Policy that releases attributes to Box -->
  <AttributeFilterPolicy id="BoxSSO">
    <PolicyRequirementRule xsi:type="basic:AttributeRequesterString" value="https://services.box.com/sp"/>
    <AttributeRule attributeID="eduPersonPrincipalName">
      <PermitValueRule xsi:type="basic:ANY"/>
    </AttributeRule>
    <AttributeRule attributeID="sn">
      <PermitValueRule xsi:type="basic:ANY"/>
    </AttributeRule>
    <AttributeRule attributeID="givenName">
      <PermitValueRule xsi:type="basic:ANY"/>
    </AttributeRule>
  </AttributeFilterPolicy>
```

Box has written some general instructions for a standard ADFS configuration for Box available at https://cloud.box.com/s/wgcatqg9s5cqsa9u7yo1. These are not NET+ specific, but they are still generally applicable.