Connecting a Shibboleth SP to eduTEAMS

This guide describes how the Shibboleth v3 SP can be configured as a SAML Service Provider for eduTEAMS.

Shibboleth (pronounced "Shibboleth") is the reference implementation of the OASIS SAML standard.

Installing and setting up the Shibboleth Service Provider in full is beyond the scope of this document. Many resources are available, such as the Shibboleth Wiki (https://wiki.shibboleth.net) and the installation instructions supplied and maintained by SWITCH (https://www.switch.ch/aai/guides/sp/installation/).

It is assumed in the following that you are using Shibboleth's v3 SP alongside the Apache webserver. If you are using a different webserver, the configuration of the SP should remain the same, with any differences being a requirement of your chosen web server.

1. Shibboleth configuration

Locate the file **shibboleth2.xml**. For example, using the switch.ch instructions (see above) for an Ubuntu Linux system, you will find the file at **/etc/shibboleth/shibboleth2.xml**.

We show to the right, sections of the file in which you will need to enter information relevant to your installation.

Enter your **[your_sp_entity_id]**:

eg **https://example.com/shibboleth**

```
<!-- By default, in-memory StorageService, ReplayCache, ArtifactMap, and
SessionCache are used. See example-shibboleth2.xml for samples of
explicitly configuring them. -->
<!-- The ApplicationDefaults element is where most of Shibboleth's SAML
bits are defined. -->
<ApplicationDefaults entityID="[your_sp_entity_id]"
    REMOTE_USER="eduPersonUniqueId"
    cipherSuites="DEFAULT:!EXP:!LOW:!aNULL:!eNULL:!DES:!IDEA:!SEED:!RC4:!
3DES:!kRSA:!SSLv2:!SSLv3:
                 !TLSv1:!TLSv1.1"
    >
```

Make sure **handlerSSL="true" cookieProps="https"** exist as follows:

```
        <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
                  checkAddress="false" handlerSSL="true" cookieProps="
https">
```

We would also encourage you to consider redirectLimit="exact" in the above <Sessions .... > section. You might find the following useful : https://wiki.shibboleth.net/confluence/display/SP3/Sessions

Enter your **[vo_entity_id]**:

eg **https://proxy.eduteams.org/proxy**

```
<!--
Configures SSO for a default IdP. To properly allow for >1 IdP, remove
entityID property and adjust discoveryURL to point to discovery service.
You can also override entityID on /Login query string, or in RequestMap
/htaccess.
-->
<SSO entityID="https://proxy.eduteams.org/proxy">
SAML2
</SSO>
```

Enter **[your_support_email_address]**:

eg **support@example.com**

```
<!--
        Allows overriding of error template information/filenames. You can
        also add your own attributes with values that can be plugged into
the
        templates, e.g., helpLocation below.
        -->
        <Errors supportContact="[your_support_email_address]"
            helpLocation="/about.html"
            styleSheet="/shibboleth-sp/main.css"/>
```

Add the filename of the local copy of the eduTEAMS proxy's metadata

```
       <!-- Example of locally maintained metadata. -->
        <!--
        <MetadataProvider type="XML" validate="true" path="partner-
metadata.xml"/>
        -->
       <!-- Metadata for the eduTEAMS proxy -->
       <MetadataProvider type="XML" validate="true" path="proxy.eduteams.
org-frontend.xml" />
```

Now save a copy of the eduTEAMS proxy metadata to the file **/etc/shibboleth/proxy.eduteams.org-frontend.xml**

eg

```
wget "https://<your-proxy-endpoint>/metadata/frontend.xml" -O /etc/shibboleth
/proxy.eduteams.org-frontend.xml
```

Finally, set up the signing and encryption certificates.
Please note the filenames used for the **signing** and **encryption certificates**. See note below the following snippet for help with these files if required.

```
        <!-- Simple file-based resolvers for separate signing/encryption
keys. -->
        <CredentialResolver type="File" use="signing"
            key="sp-key.pem" certificate="sp-cert.pem"/>
        <CredentialResolver type="File" use="encryption"
            key="sp-key.pem" certificate="sp-cert.pem"/>
```

You can use the command shib-keygen to create the signing and encryption pairs in the correct directory:

shib-keygen -h **[your_sp_domain]**

eg

cd /etc/shibboleth/

shib-keygen -h example.com

Next, edit the file **/etc/shibboleth/attribute-map.xml**

Add the following lines before the closing "</Attributes>" :

The eduTEAMS service provides a number of attributes for users who have authenticated. The attribute "eduPersonUniqueId" (see Attributes available to Relying Parties#eduTEAMSIdentifier) is the preferred attribute for identifying a remote user, and works alongside the REMOTE_USER=" eduPersonUniqueId" edit you made in the **shibboleth2.xml** file.

The other "Attribute name=" lines refer to the remaining attributes nominally made available by the eduTEAMS service.

```
   <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.13" id="
eduPersonUniqueId">
      <AttributeDecoder xsi:type="StringAttributeDecoder" caseSensitive="
false"/>
   </Attribute>

   <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/>
   <Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>
   <Attribute name="urn:oid:2.5.4.4" id="surname"/>
   <Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="email"/>
   <Attribute name="urn:oid:1.3.6.1.4.1.25178.4.1.11" id="
voPesonExternalAffiliation"/>
   <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.11" id="
eduPersonAssurance"/>
   <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.16" id="eduPersonOrcid"
/>
```

2. Apache configuration

You will need to configure Apache to recognize Shibboleth as an authorization "gatekeeper".

You can do this at the VirtualHost level, for example. Here we show a very basic example. Other recipes are available. See for instance the Shibboleth Wiki at https://wiki.shibboleth.net/confluence/display/SP3/Apache

```
<VirtuaHost example.com:443>

   ...

   ...

   <Location />

      AuthType shibboleth

      <IfVersion < 2.3>

         ShibCompatWith24 On

      </IfVersion>

      ShibRequestSetting requireSession true

      ShibUseEnvironment On

      require shibboleth

   </Location>
   ...
</VirtualHost>
```

2a. Entitlements (authorization)

An authenticated user will have a number of "entitlements" associated with their account.

These entitlements are presented to your SP in the form of the following:

> **Attribute FriendlyName="eduPersonEntitlement" Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"**

- within the SAML assertion.

The apache webserver populates the server environment with the variable "entitlements" and populates it accordingly.

See the example to the right.

```
           <ns1:Attribute FriendlyName="eduPersonEntitlement"
                          Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
                          NameFormat="urn:oasis:names:tc:SAML:2.0:
attrname-format:uri"
                          >
                <ns1:AttributeValue xmlns:xs="http://www.w3.org/2001
/XMLSchema"
                                    xsi:type="xs:string"
                                    >urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS#acc.eduteams.org</ns1:AttributeValue>
                <ns1:AttributeValue xmlns:xs="http://www.w3.org/2001
/XMLSchema"
                                    xsi:type="xs:string"
                                    >urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS:gitlab#acc.eduteams.org</ns1:AttributeValue>
                <ns1:AttributeValue xmlns:xs="http://www.w3.org/2001
/XMLSchema"
                                    xsi:type="xs:string"
                                    >urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS:gitlab:admin#acc.eduteams.org</ns1:
AttributeValue>
                <ns1:AttributeValue xmlns:xs="http://www.w3.org/2001
/XMLSchema"
                                    xsi:type="xs:string"
                                    >urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS:Developers#acc.eduteams.org</ns1:
AttributeValue>
                <ns1:AttributeValue xmlns:xs="http://www.w3.org/2001
/XMLSchema"
                                    xsi:type="xs:string"
                                    >urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS:gitlab:audit#acc.eduteams.org</ns1:
AttributeValue>
            </ns1:Attribute>
```

.... Is presented to your Apache instance as:

```
[entitlement] => urn:geant:eduteams.org:service:eduteams-acc:group:
eduTEAMS#acc.eduteams.org;urn:geant:eduteams.org:service:eduteams-acc:group:
eduTEAMS:gitlab#acc.eduteams.org;urn:geant:eduteams.org:service:eduteams-acc:
group:eduTEAMS:gitlab:admin#acc.eduteams.org;urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS:Developers#acc.eduteams.org;urn:geant:eduteams.
org:service:eduteams-acc:group:eduTEAMS:gitlab:audit#acc.eduteams.org
```

- that is, a colon (":") separated list.

Apache also allows server and .htaccess - level directives to control access.

eg

```
<Location /login>
   AuthType shibboleth
   ShibRequestSetting requireSession 1
   ShibUseEnvironment On
   <RequireAny>
      Require shib-attr entitlement "urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS:Developers#acc.eduteams.org"
                                    "urn:geant:eduteams.org:service:
eduteams-acc:group:eduTEAMS:gitlab:audit#acc.eduteams.org"
   </RequireAny>
</Location>
```

3. Restarting the shibd and apache2 services

systemctl restart apache2 && systemctl restart shibd

You can run the command

shibd -t

to check for errors, and also look inside the log file

/var/log/shibboleth/shibd.log

in case things do not work first time (for example, file permissions on the local copy of the proxy's metadata)

4. Conclusion

You should now have a working integration of Apache and Shibboleth v3 SP services on your machine.