Configuring an IdP for eduGAIN

After meeting the organisational requirements for joining eduGAIN, the IdP needs to meet the following technical requirements:

- Run up-to-date software
- Declare support for R&S
- Declare SIRTFI and provide SIRTFI contact
- Provide a logo representing the organisation
- Load eduGAIN metadata
- Configure attribute release
- Configure page layout for Service Provider logos (optional)

The final step is to send the Request to Join eduGAIN to Tuakiri/REANNZ.

This page provides further details on these technical requirements. For the details of the full process of joining eduGAIN, please see the eduGAIN Information Pack.

Running up-to-date IdP software

The IdP MUST run Shibboleth IdP 3.x, if possible the latest version. Please follow the instructions at Upgrading a Shibboleth 3.x IdP.

Declaring support for R&S

Please see the eduGAIN Information Pack and the REFEDS Research and Scholarship Category definition for full details.

In essence:

- R&S Service Providers (SPs) are SPs that support Research and Scholarship activities and request only a small selection of attributes (see the above link for details)
- IdPs declare support for R&S by agreeing to release these attributes.

To meet the requirements of R&S, an IdP MUST load the `rns-attribute-filter.xml` as per instructions below.

The actual declaration of support for R&S will be part of the joining request sent to Tuakiri as per the instructions in the eduGAIN Information Pack.

SIRTFI

SIRTFI (Security Incident Response Trust Framework for Federated Identity) is a lightweight framework to request and provide security incident response assistance, publish security incident contact information and review security incident capability.

For IdPs, SIRTFI is a requirement for joining eduGAIN. Organisations self-assess against the SIRTFI criteria and then self-assert SIRTFI. They also have to provide a security contact - which should be a role-based email address, not personal.

- The security contact should be provided by adding the contact as a `security` contact of the IdP in the Tuakiri Federation Registry.
- The step of self-asserting SIRTFI is included in the request to join eduGAIN as per the instructions in the eduGAIN Information Pack.

For more information, please see the eduGAIN Information Pack and the REFEDS SIRTFI documentation.

Logo

An IdP should provide a logo to help users with identifying the right IdP at the Discovery Service.

To provide a logo, please include an **HTTPS** link to your logo in your request to join eduGAIN. The Tuakiri team will take care of including the logo in the metadata sent to eduGAIN.

Loading eduGAIN metadata

Tuakiri provides a version of the eduGAIN metadata tailored for consumption by Tuakiri members.

The eduGAIN metadata is signed with the same certificate as the Tuakiri metadata - and is published separately for the Production and TEST federations.

Note that there is no TEST eduGAIN federation, so test IdPs cannot do full end-to-end eduGAIN testing, but partial tests can still confirm the configuration has been applied to the IdP correctly, and that's what the eduGAIN metadata stream for TEST is for.

The metadata URL and signing certificate are:

Federation name	Tuakiri	Tuakiri TEST
Metadata name	**tuakiri.ac.nz/edugain-verified**	**test.tuakiri.ac.nz/edugain-verified**
Metadata distribution point	https://directory.tuakiri.ac.nz/metadata/tuakiri-edugain-verified.xml	https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-edugain-verified.xml
Metadata signing certificate	https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-cert.pem	https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-metadata-cert.pem

To load the metadata, add the following snippet in `/opt/shibboleth-idp/conf/metadata-providers.xml` just **BELOW** the snippet loading the Tuakiri metadata

Tuakiri (PRODUCTION): load eduGAIN metadata

```
<MetadataProvider id="TuakirieduGAINMetadata"
        xsi:type="FileBackedHTTPMetadataProvider"
        refreshDelayFactor="0.125"
        maxRefreshDelay="PT2H"
        backingFile="%{idp.home}/metadata/tuakiri-edugain-metadata.xml"
        metadataURL="https://directory.tuakiri.ac.nz/metadata/tuakiri-edugain-verified.xml">

  <MetadataFilter xsi:type="SignatureValidation"
          certificateFile="${idp.home}/credentials/tuakiri-metadata-cert.pem"
          requireSignedRoot="true">
  </MetadataFilter>
  <MetadataFilter xsi:type="EntityRoleWhiteList">
          <RetainedRole>md:SPSSODescriptor</RetainedRole>
  </MetadataFilter>
</MetadataProvider>
```

Tuakiri-TEST: load eduGAIN metadata

```
<MetadataProvider id="TuakiriTESTeduGAINMetadata"
        xsi:type="FileBackedHTTPMetadataProvider"
        refreshDelayFactor="0.125"
        maxRefreshDelay="PT2H"
        backingFile="%{idp.home}/metadata/tuakiri-test-edugain-metadata.xml"
        metadataURL="https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-edugain-verified.xml">

  <MetadataFilter xsi:type="SignatureValidation"
          certificateFile="${idp.home}/credentials/tuakiri-test-metadata-cert.pem"
          requireSignedRoot="true">
  </MetadataFilter>
  <MetadataFilter xsi:type="EntityRoleWhiteList">
          <RetainedRole>md:SPSSODescriptor</RetainedRole>
  </MetadataFilter>
</MetadataProvider>
```

Configuring Attribute Release

In the initial IdP deployment for Tuakiri, IdPs were configured to automatically release attributes according to an attribute filter constructed by the Tuakiri Federation Registry, based on attribute request information gathered from the Service Providers (SPs). This approach would not scale for eduGAIN.

Attribute release for eduGAIN is driven by two separate mechanisms:

- Research and Scholarship (R&S) entity category: SPs in the R&S entity category expect attributes from a small predefined set to be automatically released by the IdP. IdPs supporting the R&S category MUST release these attributes. Tuakiri requires that IdPs joining eduGAIN declare support for R&S and automatically release the R&S attributes to R&S SPs.
- Metadata-based attribute release: SP requests for attributes can also be recorded in the metadata. We strongly recommend to configure the IdP to release attributes as requested in the metadata - these requests would have been vetted by the home federation of the SP, and the attribute release is crucial for the login to succeed.

To configure the attribute release:

- download `rns-attribute-filter.xml` and `metadata-based-attribute-filter.xml` from https://github.com/REANNZ/Tuakiri-public/tree/master/shibboleth-idp/idp into your `/opt/shibboleth-idp/conf` :

```
wget -O /opt/shibboleth-idp/conf/rns-attribute-filter.xml https://github.com/REANNZ/Tuakiri-public/raw/master/shibboleth-idp/idp/rns-attribute-filter.xml
wget -O /opt/shibboleth-idp/conf/metadata-based-attribute-filter.xml https://github.com/REANNZ/Tuakiri-public/raw/master/shibboleth-idp/idp/metadata-based-attribute-filter.xml
```

> If deploying these rules on a TEST IdP connected to Tuakiri-TEST, edit the downloaded `metadata-based-attribute-filter.xml` and change the `PoliceRequirementRule` at the top of the file as per the instructions in the file.

- add the files to the list of attribute filters in `/opt/shibboleth-idp/conf/services.xml` stored in the `shibboleth.AttributeFilterResources` list:

```
    <util:list id ="shibboleth.AttributeFilterResources">
        <value>%{idp.home}/conf/attribute-filter.xml</value>
        <value>%{idp.home}/conf/rns-attribute-filter.xml</value>
        <value>%{idp.home}/conf/metadata-based-attribute-filter.xml</value>
        ...
```

> With the `metadata-based-attribute-filter.xml` filter in place, the original attribute filter generated by the Federation Registry (loaded by the IdP in the same section) can be removed - it would be only duplicating the metadata-based attribute filter for Tuakiri SPs. (The only exception are value-specific rules with regular expressions - please contact Tuakiri at tuakiri@reannz.co.nz if your IdP uses these).
>
> After removing the remotely loaded attribute filter, we recommend to reduce the reload interval of attribute filter files from 15m to 5s (now that all files are local). In `/opt/shibboleth-idp/conf/services.properties` , change `idp.service.attribute.filter.checkInterval` to `PT5S` :
>
> ```
> idp.service.attribute.filter.checkInterval = PT5S
> ```
>
> Note that if the Tuakiri-generated attribute filter was loaded from a local file refreshed by an external script (cron job), when no longer loaded this file, the cron job can be (and should be) removed.

- and restart the IdP to pick up the change:

```
service tomcat restart
```

Configuring page layout for Service Provider logo display

The IdP login and consent pages would display a logo of the SP if it's present in the metadata. However, Tuakiri has not been publishing logos in the metadata, so this layout would be untested - and might render in undesirable ways. In particular, if the CSS stylesheet for the consent page has been modified as recommended in the Customization and Branding section of the IdP install manual (width increased from 600px to 800px), we recommend removing the `width: 50%;` style from the `.organization_logo` class - which is also typically done to the `.federation_logo` class (which however applies to the IdP logo, not really matching the name of the class).

```
--- /opt/shibboleth-idp/edit-webapp/css/consent.css.dist        2019-08-16 09:30:59.206412350 +1200
+++ /opt/shibboleth-idp/edit-webapp/css/consent.css        2019-09-12 14:48:35.224742616 +1200
@@ -113,14 +113,14 @@

 .federation_logo
 {
-        width: 50%;
+        /* width: 50%; */
     float: left;
     padding-top: 35px;
     border: 0;
 }
 .organization_logo
 {
-        width: 50%;
+/*        width: 50%; */
     float: right;
     border: 0;
 }
```

Applying Changes - Restart

Changes to `$IDP_HOME/conf/services.xml` and `$IDP_HOME/conf/services.properties` require restarting the IdP (reloading individual IdP components will not be sufficient here).

Changes to CSS files (or other static files served by the application) rquires also rebuilding the WAR file.

If no files included in the WAR File were changed, run just `service tomcat restart`

Otherwise, to restart tomcat, rebuild the WAR file (and fix file permissions on the fly), run:

```
service tomcat stop ; /opt/shibboleth-idp/bin/build.sh < /dev/null ; chown -R tomcat.tomcat /opt/shibboleth-idp; service tomcat start
```