Using a Header for Authentication

Using a header from a trusted server for authentication is a common pattern. Most usually the REMOTE_USER header is used. OA4MP allows for *any* header to be used to identify a user after a service has authenticated them. This implies a trust relation with that server. Note that this also means restricted access by that server.

Configuring OA4MP

This is relatively simple and requires the following element in your server XML configuration:

```
<authorizationServlet useHeader="true"
                      requireHeader="true"
                      headerFieldName="REMOTE_USER"/>
```

See the full reference for the authorizationServlet for more. In this case, headers are enabled and the REMOTE_USER will be used. (The **`requireHeader`** attribute is needed if this is the sole way to authenticate, which is the most common use pattern.)

There must, of course, be some way to limit access to the service since any request with the right header can impersonate a valid user. If using Tomcat (outlined below) this is done by Tomcat.

Tomcat and OA4MP

Tomcat is the standard container for OA4MP and one way to set up authentication is to use Tomcat itself. This means that logins for Tomcat are logins for OA4MP. The plus is that it is quite easy to set up, but the downside is that unless all of your user management will be done this way, it may require specific logins for OA4MP. However, it is extremely useful in setting up and testing OA4MP in various scenarios, or for small organizations.

There are many ways to configure Tomcat authentication. For the purpose of this documentation, we'll illustrate basic authentication, which is the simplest method. Basically, set up users then tell Tomcat that the authorize endpoint requires authentication. The REMOTE_USER header will be set and there is no other way to access that endpoint, hence it is secure.

Edit the **`$CATALINA_HOME/conf/tomcat-users.xml`** file by adding . An example is

```
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <role rolename="tomcat"/>
  <role rolename="oa4mp"/>

  <user username="jeff" password="ZDYoqJFT" roles="tomcat,oa4mp"/>
  <user username="bob" password="fl00fy" roles="tomcat,oa4mp"/>
</tomcat-users>
```

Now, we edit **`$CATALINA_HOME/webapps/oauth/WEB-INF/web.xml`** . (Not the same web.xml file as in the main Tomcat conf directory). Add the following:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>OAuth Security</web-resource-name>
    <url-pattern>/authorize</url-pattern>
    <http-method>POST</http-method>
    <http-method>GET</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>oa4mp</role-name>
  </auth-constraint>
  <user-data-constraint>
  <!-- transport-guarantee can be CONFIDENTIAL, INTEGRAL, or NONE -->
  <transport-guarantee>NONE</transport-guarantee>
  </user-data-constraint>
</security-constraint>
<login-config>
<auth-method>BASIC</auth-method>
</login-config>
```

Nota Bene: This enables remote user for the authorize endpoint. If you wanted to use this for the device endpoint, you would have another set the url-pattern accordingly to

```
    <url-pattern>/device</url-pattern>
```

If you look in the standard distribution in **`$CATALINA_HOME/webapps/oauth/WEB-INF/`** you will find a remote-user.xml configuration with all of this ready to go. Simply tweak it as needed, make surte it points to your server configuration, and use as your web.xml file.

Note that redeploying or updating OA4MP will overwrite the web.xml file, so we recommend backing it up somewhere secure. Redeploy OA4MP, then copy your configuration back and restart Tomcat. A quicker way to do this is to touch oauth2.war (which changes the date, prompting a reload of the app).