Federating Confluence with mod_auth_mellon

This is probably my longest standing action item in TERENA 😉: implement a federated version of the Atlassian Confluence wiki.

Below is the recipe for getting this to work with Ubuntu 14.04, Confluence 5.6, Apache 2.4, and mod_auth_mellon.

I choose mod_auth_mellon because it seemed like a cleaner solution than mod_shib, requiring no additional daemons and much simpler configuration.

The wiki will be open to the public, with only SAML logins (i.e. no local accounts). New users will have their account automatically created, and are put in the confluence-users group.

Prerequisites

Before you start, make sure you have these bits:

- A correctly configured apache web server that is able to serve an HTTPS web site (https://example.com).
- A SAML Identity Provider (IdP).
- An account on that IdP.
- An attribute that can be used as username in Confluence (for example eduPersonPrincipalName). Attributes for full name and e-mail are optional but recommended. In this case we assume 'mail' and 'displayName' can be used.
- The user name of the to-be administrator account. So, if you choose eduPersonPrincipalName as the attribute for username, you need to know your own value (for instance 'dvisser@surfnet.nl').

PostgreSQL

```
apt-get install postgresql
```

Create a dedicated database user, and a database:

```
sudo su - postgres
createuser -S -d -r -P -E confuser
createdb -O confuser confluence
```

Confluence - part 1

This is a default install of Confluence, which has only local account and no federated logins - that comes later in part 2.

Install OpenJDK:

```
apt-get --no-install-recommends install openjdk-7-jdk
```

Download the source http://www.atlassian.com/software/confluence/downloads/binary/atlassian-confluence-5.6.3.tar.gz and unpack it to `/opt/confluence`. All relative paths mentioned below are relative to this directory.

Create a home directory for Confluence (`/home/confluence`).

Edit `confluence/WEB-INF/classes/confluence-init.properties` and configure `confluence.home=/home/confluence`.

Upstart script for Confluence

Ubuntu uses the new upstart init scripts, which we should use.

Create the upstart script `/etc/init/confluence.conf`:

```
# Upstart script for confluence
description     "Atlassian Confluence"
start on runlevel [2345]
stop on runlevel [!2345]
kill timeout 30
env RUN_AS_USER=root
env BASEDIR=/opt/confluence
script
    LOGFILE=$BASEDIR/logs/catalina.out
    exec su - $RUN_AS_USER -c "$BASEDIR/bin/catalina.sh run" >> $LOGFILE 2>&1
end script
```

Once this script is here, issue "start confluence" to get going, and watch the log file `/opt/confluence/log/catalina.out`. After some time you should see something like this:

```
INFO: Starting Coyote HTTP/1.1 on http-8090
Apr 09, 2013 5:14:43 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 65971 ms
```

By this time you can point your browser to http://example.com:8090, and it should come up with a configuration wizard that will ask for a license key, database credentials, a local admin account, etc. Once that is all done, things should be working, but nothing federated yet, only local accounts.

Make sure that anonymous users can look at the content. Go to https://example.com/admin/permissions/globalpermissions.action

At this point you need to do some preparation so that stuff will work properly later on through Apache:

1. Create a new admin account with the correct federated username. For instance, if you have decided on using eduPersonPrincipalName as the username, and the value of that attribute for your federated account is 'dvisser@surfnet.nl', create an account with that exactly that username.
2. Make sure this newly created account is a member of "confluence-administrators".
3. Configure the **Server Base Url** to be the (HTTPS) URL that Apache will listen to. This is done in Confluence Admin -> General Configuration -> Edit.
4. Configure the global permissions to allow anonymous use (https://example.com/admin/permissions/globalpermissions.action).

Once this is complete, shut down Confluence by issuing "stop confluence".

mod_auth_mellon

mod_auth_mellon is an Apache module. Ubuntu 14.04 and later contain the correct package. To get things working for Ubuntu 12.04 I recompiled the Debian source packages from the University of Tilburg and made them available in our own APT repository.

The needed packages can be installed like:

```
apt-get install libapache2-mod-auth-mellon apache2
# other needed modules
a2enmod proxy proxy_http auth_mellon rewrite ssl headers
```

Create a directory `/etc/apache/mellon`, and store the Identity Provider metadata in XML format to a file called `idp.xml`.

Create the cryptographic material for the mellon SP:

```
openssl req -new -newkey rsa:4096 -days 3650 -nodes -x509 -keyout sp.key -out sp.crt
```

Now add this to the configuration of the vhost (note that this is not the entire config - you should have the HTTPS stuff etc already configured):

```
ServerName example.com

ProxyRequests Off
<Proxy http://ip6-localhost:8090>
    Require All Granted
</Proxy>


ProxyPass /mellon/ !
ProxyPass / http://ip6-localhost:8090/
ProxyPassReverse / http://ip6-localhost:8090/


# Mobile theme does not honour new seraph values for login URL, so we have to redirect that
RewriteEngine on
RewriteCond     %{QUERY_STRING} ^originalUrl=(.*)$      [NC]
Rewriterule     ^/plugins/servlet/mobile/login          /mellon/login?ReturnTo=%1 [R,NE]

# After upgrading to 5.6 the mobile login links were changed.
# New redirects needed:
RewriteCond     %{QUERY_STRING} ^os_destination=%2Fplugins%2Fservlet%2Fmobile%3F%23content%2Fview%2F(.*)$ [NC]
RewriteRule     ^/login.action  /mellon/login?ReturnTo=/pages/viewpage.action?pageId=%1 [R,NE]


# Remove the jsessionid from the URL, to prevent 404 errors when
# unauthenticated visitors try to access a protected resource.
ReWriteRule     ^(.*);jsessionid=[A-Za-z0-9]+(.*)$ $1$2 [R,NE]


<Location />
        MellonEnable "info"
        MellonSecureCookie On
        MellonSessionDump Off
        MellonUser "eduPersonPrincipalName"
        MellonSamlResponseDump Off
        MellonEndpointPath "/mellon"
        MellonSPPrivateKeyFile /etc/apache2/mellon/sp.key
        MellonSPCertFile /etc/apache2/mellon/sp.crt
        MellonIdPMetadataFile /etc/apache2/mellon/idp.xml

        RequestHeader unset CONF_FULL_NAME
        RequestHeader set CONF_FULL_NAME "%{MELLON_displayName}e" env=MELLON_displayName

        RequestHeader unset CONF_EMAIL
        RequestHeader set CONF_EMAIL "%{MELLON_mail}e" env=MELLON_mail
</Location>
```

By this time, you should be able to download the Service Provider metadata from https://example.com/mellon/metadata, and use it to add it to your IdP, thereby creating a trust relationship.

And once that is done, you should be able to use federated authentication by going to https://example.com/mellon/login?ReturnTo=%2F

Confluence - part 2

Now everything is in place to federate Confluence. Make sure that Confluence isn't running any more.

1. Download the right version of remoteUserAuth.jar (I used 2.5.0) from https://github.com/chauth/confluence_http_authenticator/tree/master/releases, and store it in `confluence/WEB-INF/lib.` Make sure you're actually downloading the JAR file and not the HTML page.
2. Download https://github.com/chauth/confluence_http_authenticator/blob/master/conf/remoteUserAuthenticator.properties and save it as `confluence/WEB-INF/classes/remoteUserAuthenticator.properties`. The defaults were almost OK, the only thing I needed to change was convert.to.utf8=true.

3. Edit `confluence/WEB-INF/classes/seraph-config.xml` and change these values:

```
       <init-param>
            <param-name>login.url</param-name>
            <param-value>/login.action?os_destination=${originalurl}</param-value>
        </init-param>
        <init-param>
            <param-name>link.login.url</param-name>
            <param-value>/login.action</param-value>
        </init-param>
```

   To these:

```
        <init-param>
            <param-name>login.url</param-name>
            <param-value>/mellon/login?ReturnTo=${originalurl}</param-value>
        </init-param>
        <init-param>
            <param-name>link.login.url</param-name>
            <param-value>/mellon/login?ReturnTo=${originalurl}</param-value>
        </init-param>
```

   Also, change the authenticator from this:

```
<authenticator class="com.atlassian.confluence.user.ConfluenceAuthenticator"/>
```

   to this:

```
<authenticator class="shibauth.confluence.authentication.shibboleth.RemoteUserAuthenticator"/>
```

You should now be able to use federated logins.

If for some reason your account isn't an administrator, there is no way to fix this. You should disable the changes from step 3 and restart Confluence so that it doesn't use federated authentication any more. Then go back in and fix the permissions, then change back.

Confluence - mobile theme

The new Confluence feature a dedicated theme for use on mobile devices. This is great, but unfortunately both the login and logout buttons in that theme do not work - they still point to the 'old' static login/logout links.

Login button

I couldn't find any way to do this in Confluence, so I ended up rewriting it in Apache. See the snippet in the Apache config above.

Logout button & Invitation link

The logout button link can be configured in Confluence, but the configuration file is located inside a JAR file (Java ARchive), so it's a little bit of work. You need to extract the JAR and copy/edit an XML file to WEB-INF/classes - see https://confluence.atlassian.com/confkb/how-to-edit-files-in-confluence-jar-files-103711179.html.

Then do:

- `mkdir /tmp/jar`
- `unzip /opt/confluence/confluence/WEB-INF/lib/confluence-5.5.3.jar -d /tmp/jar`

Now /tmp/jar should contain the contents of the jar. Copy the file `xwork.xml` to /opt/confluence/confluence/WEB-INF/classes and change this part:

```
        <action name="logout" class="com.atlassian.confluence.user.actions.LogoutAction">
            <interceptor-ref name="defaultStack"/>
            <result name="error" type="velocity">/logout.vm</result>
            <result name="success" type="redirect">/login.action?logout=true</result>
        </action>
```

to this:

```
        <action name="logout" class="com.atlassian.confluence.user.actions.LogoutAction">
            <interceptor-ref name="defaultStack"/>
            <result name="error" type="velocity">/logout.vm</result>
            <result name="success" type="redirect">/mellon/logout?ReturnTo=%2Fdashboard.action</result>
        </action>
```

While you're at it, you can also change the **Invite Users** link on /admin/users/browseusers.action. This will take you a non-federated invitation page, which is not what you want. If your Confluence admins don't know how they should add people (by asking new users to simply log in), then chances are that they will use this option, ending up with a 'wrong' account.

Copy plugins/user-management.xml to /opt/confluence/confluence/WEB-INF/classes/plugins/ (you have to create the plugins dir) and change this line:

```
<link linkId="invite-tab-link">/admin/users/inviteuser.action</link>
```

so that it points to a custom page that you created with the proper instructions.

The page I'm using has a clickable mailto link so that everything is precooked.

Restart Confluence. You should now also be able to use federated logins on your iPad/etc.

jsessionid errors

If unauthenticated users try to access content that is protected, Confluence tries to set jsessionid as part of the URL. This leads to 404 errors like this:

```
NOT FOUND
The requested URL /mellon/login;jsessionid=8A736F43779F96249F6C3DC41067BB98 was not found on this server.
```

Since the jsessionid part isn't needed, it can be removed uses a rewrite statement (see apache config above).

Logging

You might want to change the default apache log file configuration to include the federated user name. While you're at it, add milliseconds to the timestamp, and change it to something that is not a nightmare to sort later on:

```
#LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
# Sortable log format, with proper federated username. DV 2016-04-05
LogFormat "%v:%p %{%F %T}t.%{msec_frac}t %h %{MELLON_CONF_USER}e \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}
i\"" vhost_combined
```

This will yield useful stuff like:

```
wiki.geant.org:443 2016-04-05 14:23:10.714 2001:610:148:dead:49be:5225:a8a0:4b1f federated-user-3 "GET /rest
/mywork/latest/status/notification/count HTTP/1.1" 200 944 "https://wiki.geant.org/dashboard.action" "Mozilla/5.
0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.110 Safari/537.36"
```

Limit access to the unprotected TCP port

Confluence by default listens to TCP port 8090 on all interface. Since Apache will be the internet facing application, there is no need for Confluence to listen on all interfaces. Even worse, if you do let it listen on the internet then it is trivial to add a REMOTE_USER header and spoof any account. Of course it is good practice to use a firewall to protect this port, but you can limit this in Confluence as well. Since Apache is configured to only connect to the (IPv6) localhost address, this is what you should configure Confluence to use as listening address. As per Tomcat docs, you should add an "address" attribute to the Connector, which is located in `conf/server.xml`:

<Connector className="org.apache.coyote.tomcat4.CoyoteConnector" port="8090" address="::1" minProcessors="5"