OIDC support in RUCIO

Rizart Dona

CERN

January 19, 2021 - WP5 AAI Workshop

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

Overview

- OAuth2/OIDC Terminology
- Rucio Authentication: Accounts & Identities
- Rucio User Authentication with OIDC
- Rucio Token Management
- Current Status & Future Work
- References

Rizart.Dona@cern.ch
Funded by the European Union's Horizon 2020 - Grant N° 824064

OAuth2/OIDC Terminology (I)

- **OAuth2** is an open standard for **access delegation**, it essentially allows **access tokens** to be issued to third-party clients by an **authorization** server, with the approval of the **resource owner**. The third party then uses the access token to access the protected resources hosted by the resource server → avoids sharing passwords around!
  - **Scopes** is a mechanism to limit an application's access, an application can request one or more scopes, the access token issued to the application will be **limited** to the scopes granted. (OAuth2 does not define any particular values for scopes)
- **OpenID Connect** (**OIDC**) is an **authentication** layer built on top of OAuth 2.0
  - Adds login and profile information about the person who is logged in (**openid** scope)
  - The authorization server is also called an identity provider (**IdP**)
  - Enables scenarios such as single sign-on (**SSO**), a user could link various types of accounts (Google accounts, social media accounts, etc.)

Funded by the European Union's Horizon 2020 - Grant N° 824064

OAuth2/OIDC Terminology (II)

- OAuth **Grant Types (flows)**
  - The OAuth framework specifies several grant types for different use cases, as well as a framework for creating new grant types.
- Common OAuth grant types (which are also employed in Rucio) are listed below
  - **Authorization Code**: The Authorization Code grant type is used by confidential and public clients to exchange an authorization code for an access token. After the user returns to the client via the redirect URL, the application will get the authorization code from the URL and use it to request an access token.
  - **Client Credentials**: The Client Credentials grant type is used by clients to obtain an access token outside of the context of a user. This is typically used by clients to access resources about themselves rather than to access a user's resources.
  - **Refresh Token** (scope:**offline_access** is needed): The Refresh Token grant type is used by clients to exchange a refresh token for an access token when the access token has expired. This allows clients to continue to have a valid access token without further interaction with the user.

Funded by the European Union's Horizon 2020 - Grant N° 824064

Rucio Authentication: Accounts & Identities

- Each user has a Rucio account (username)
  - The account can be connected with multiple identities (N:M mapping) , those include:
    - username/password, X.509, SSH public keys, GSS/Kerberos
    - **OIDC**
  - For **OIDC** support, the user needs first to register to the IdP (**ESCAPE IAM** in this case)
    - IdP admin manually approves users (or automatic enrolment via other configurable **trusted** IdPs)
    - The Rucio admin can manually create the accounts and assign the identities
    - **Better solution**: A Rucio probe syncs current users in IdP and their identities
      - A Rucio system client with access token that has **scim:read** scope has access to the IdP users (**client credentials** flow)
      - For each user, this includes their email, the identity of the user called subject in OpenID (**sub**) and the issuer url (**iss**).

Funded by the European Union's Horizon 2020 - Grant N° 824064

Rucio User Authentication with OIDC (I)

- User has two options
  - Get an access token (AT) from external sources (e.g. oidc-agent) and present it to the Rucio Server (RS)
    - Via the **REST API** ('X-Rucio-Auth-Token' header)
    - Via the **CLI** ('auth_token_file_path' configuration on local rucio.cfg)
    - Basic assumptions are that the user presented a **valid** AT with the **scopes** and **audiences** Rucio requires and that the user's identity is already registered in Rucio DB
  - Request an access token from Rucio itself (**authorization code flow**)
    - Three Rucio Client (RC) CLI options to do this
      - RC waiting for fetch code (copy-pasting fetch code from Internet Browser)
      - RC polling RS for an access token (AT) after authentication
      - automatic, RC trusted with user's IdP credentials (**strongly discouraged**)

Funded by the European Union's Horizon 2020 - Grant N° 824064

Rucio User Authentication with OIDC (II)

```
(test_rucio_client_py2) [ridona@lxplus714 ~]$ rucio whoami
/afs/cern.ch/user/r/ridona/.virtualenvs/test_rucio_client_py2/lib/python2.7/site-packages/paramiko/transport.py:33: CryptographyDeprecati
onWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed
 in a future release.
  from cryptography.hazmat.backends import default_backend

Please use your internet browser, go to:

    https://rucio-doma-auth.cern.ch/auth/oidc_redirect?Mr5shi45E8fRHyPkULAU51t

and authenticate with your Identity Provider.
Copy paste the code from the browser to the terminal and press enter:
8ZC76NscgOKdpnuTvszaTgEpwKTZ5yhC30sDAsXnYAJOWGLGh2
status     : ACTIVE
account    : root
account_type : SERVICE
created_at : 2018-08-23T12:20:42
suspended_at : None
updated_at : 2018-08-23T12:20:42
deleted_at : None
email      : None
(test_rucio_client_py2) [ridona@lxplus714 ~]$ rucio whoami
/afs/cern.ch/user/r/ridona/.virtualenvs/test_rucio_client_py2/lib/python2.7/site-packages/paramiko/transport.py:33: CryptographyDeprecati
onWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed
 in a future release.
  from cryptography.hazmat.backends import default_backend
status     : ACTIVE
account    : root
account_type : SERVICE
created_at : 2018-08-23T12:20:42
suspended_at : None
updated_at : 2018-08-23T12:20:42
deleted_at : None
email      : None
(test_rucio_client_py2) [ridona@lxplus714 ~]$
```

RUCIO

SCIENTIFIC DATA MANAGEMENT

Rucio authorization server has been granted access to your information. Please copy-paste the following code to the open terminal session with Rucio Client in order to get your access token:

8ZC76NscgOKdpnuTvszaTgEpwKTZ5yhC30sDAsXnYAJOWGLGh2

Funded by the European Union's Horizon 2020 - Grant N° 824064

Rucio User Authentication with OIDC (III)

```
(test_rucio_client_py2) [ridona@lxplus714 ~]$ rucio-admin --oidc-polling account list
/afs/cern.ch/user/r/ridona/.virtualenvs/test_rucio_client_py2/lib/python2.7/site-packages/paramiko/transport.py:33: CryptographyDeprecati
onWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed
 in a future release.
  from cryptography.hazmat.backends import default_backend

Please use your internet browser, go to:

    https://rucio-doma-auth.cern.ch/auth/oidc_redirect?v54o3g9sXdFipBTtqN6uddu_polling

and authenticate with your Identity Provider.
In the next 3 minutes, Rucio Client will be polling
the Rucio authentication server for a token.
----------------------------------------
abrigandi
guenther
maricaantonacci
mlassnig
pmillar
ridona
root
vokac
wlcg_doma
```

RUCIO

SCIENTIFIC DATA MANAGEMENT

Rucio authorization server has been granted access to your information.
Rucio Client should now be able to fetch your token automatically.

Funded by the European Union's Horizon 2020 - Grant N° 824064

Rucio User Authentication with OIDC (IV)

- RC stores locally user's AT in a file

  (this can be configured in the local rucio.cfg)
- There is no need to re-authenticate, the RC can

  get a new AT from RS regularly (again given

  basic assumptions about the validity of both the

  refresh and the access token, scopes, audiences etc.)
- Client config:

```
[client]
rucio_host = https://rucio-doma.cern.ch:443
auth_host = https://rucio-doma-auth.cern.ch:443
auth_type = oidc
account = root
oidc_issuer = wlcg
```

```
--oidc-user OIDC_USERNAME
                      OIDC username
--oidc-password OIDC_PASSWORD
                      OIDC password
--oidc-scope OIDC_SCOPE
                      Defines which (OIDC) information user will share with
                      Rucio. Rucio requires at least -sc="openid profile".
                      To request refresh token for Rucio, scope must include
                      "openid offline_access" and there must be no active
                      access token saved on the side of the currently used
                      Rucio Client.
--oidc-audience OIDC_AUDIENCE
                      Defines which audience are tokens requested for.
--oidc-auto           If not specified, username and password credentials
                      are not required and users will be given a URL to use
                      in their browser. If specified, the users explicitly
                      trust Rucio with their IdP credentials.
--oidc-polling        If not specified, user will be asked to enter a code
                      returned by the browser to the command line. If
                      --polling is set, Rucio Client should get the token
                      without any further interaction of the user. This
                      option is active only if --auto is *not* specified.
--oidc-refresh-lifetime OIDC_REFRESH_LIFETIME
                      Max lifetime in hours for this an access token will be
                      refreshed by asynchronous Rucio daemon. If not
                      specified, refresh will be stopped after 4 days. This
                      option is effective only if --oidc-scope includes
                      offline_access scope for a refresh token to be granted
                      to Rucio.
--oidc-issuer OIDC_ISSUER
                      Defines which Identity Provider is goign to be used.
                      The issuer string must correspond to the keys
                      configured in the /etc/idpsecrets.json auth server
                      configuration file.
```


Funded by the European Union's Horizon 2020 - Grant N° 824064

Rucio Token Management (I)

- Rucio stores all tokens and the accounts associated with those in the DB
  - When user requests action that involves authentication Rucio checks the DB to see if the relevant token exists, if not it uses the refresh token to acquire a new one
- **oauthmanager** daemon running periodically in the backend
  - **Refreshes** access tokens via their refresh tokens (if access tokens have expired)
  - **Deletes** all expired tokens except tokens which have a refresh token that is still valid, in case there is a valid refresh token, expired access tokens will be kept until the refresh token expires as well
  - **Deletion** of expired OAuth session parameters (state, nonce, etc.)
- For **upload**/**download** operations the user must request (either via Rucio or externally) an access token that can be accepted by the relevant storage element (with correct scopes and audiences)

Funded by the European Union's Horizon 2020 - Grant N° 824064

Rucio Token Management (II)

- For rules that involve replica creation, FTS is getting employed and in that case a **token exchange** is taking place:
  - Rucio exchanges the current access token (for example with **scope**:"openid offline_access profile" and audience:**rucio**) with an FTS access token (for example with **scope**:fts-transfer and **audience**:fts.example)
  - Then it just **delegates** this token to FTS when the transfer job is initiated
  - FTS then must do the same for the involved storages
- All relevant scopes and audiences that are required in those steps can be configured on the Rucio server side

Funded by the European Union's Horizon 2020 - Grant N° 824064

Current Status & Future Work

- Testing is ongoing in the context of the DOMA Rucio testbed
- The ESCAPE Rucio instance has not been completely configured or tested yet to demonstrate these capabilities (work in progress)

- Next step would be to configure the ESCAPE instance and present an actual demo of how the interaction using tokens would work
  - Involves configuring the FTS instance we use to consume those tokens
  - Involves configuring some testbed endpoints that also can understand those tokens

- Dev and testing work remains to be done in Rucio
  - Is integration of a token management tool for the user needed (probably oidc-agent)?
  - More testing is needed of the code itself

Funded by the European Union's Horizon 2020 - Grant N° 824064

References (I)

- Token support in Rucio, Jaroslav Günther, DOMA-TPC Meeting 20/5/2020
  https://indico.cern.ch/event/918191/contributions/3859170/attachments/2042527/3421215/Rucio_DOMA_May2020.pdf
- DOMA TPC: Token-based AuthZ Testbed, Andrea Ceccanti, 20/5/2020
  https://indico.cern.ch/event/918191/contributions/3859170/attachments/2042527/3421225/DOMA-TPC-TokenBasedAuhtZTestbed200520.pdf
- An Illustrated Guide to OAuth and OpenID Connect
  https://developer.okta.com/blog/2019/10/21/illustrated-guide-to-oauth-and-oidc
- oidc-agent https://github.com/indigo-dc/oidc-agent
- Setting up Rucio Server with OIDC support
  https://rucio.readthedocs.io/en/latest/installing_server.html#server-configuration-for-open-id-connect-authn-z
- Setting up Rucio Client with OIDC support
  https://github.com/rucio/rucio/blob/master/doc/source/cli_examples.rst#open-id-connect-authentication-examples

Funded by the European Union's Horizon 2020 - Grant N° 824064

References (II)

- OAuth 2.0 https://oauth.net/2/
- OpenID Connect https://openid.net/connect/
- DOMA Rucio Wiki https://twiki.cern.ch/twiki/bin/view/LCG/DomaRucio
- ESCAPE IAM https://iam-escape.cloud.cnaf.infn.it/

Funded by the European Union's Horizon 2020 - Grant N° 824064

ESCAPE
European Science Cluster of Astronomy & Particle physics ESFRI research Infrastructures

Thank you!

Questions?

Rizart.Dona@cern.ch
Funded by the European Union's Horizon 2020 - Grant N° 824064