NRC·CNRC

NRC.CANADA.CA

Tokens for Non-Browser Clients Updates

Group Membership Service (GMS) Updates

Brian Major, Pat Dowler, Canadian Astronomy Data Centre
IVOA Interop, May 2021

Agenda

1. Non-browser Authentication Updates
2. GMS Updates

Non-browser Authentication Updates

A Brief History of Tokens

- Auth Lessons, Markus Demleitner, October 2019
  https://wiki.ivoa.net/internal/IVOA/InterOpOct2019DAL/authlessons.pdf

- Gaia Authentication, Juan Gonzalez, October 2019
  https://wiki.ivoa.net/internal/IVOA/InterOpOct2019DAL/GACS_Auth_jgonzalez.pdf

- Auth Strawman, Patrick Dowler, May 2020
  https://wiki.ivoa.net/internal/IVOA/InterOpMay2020GWS/auth-reqs-strawman.pdf

- OAuth2 Lessons & GMS, Sonia Zorba, November 2020
  https://wiki.ivoa.net/internal/IVOA/InterOpNov2020GWS/GMS-OAuth2-update.pdf

- Non-browser OAuth2, Brian Major, November 2020
  https://wiki.ivoa.net/internal/IVOA/InterOpNov2020GWS/TokensForCLI.pdf

- Authentication Implementation Report, Mark Taylor, November 2020
  https://wiki.ivoa.net/internal/IVOA/InterOpNov2020GWS/auth.pdf

May 2021: Progress! (sorry if I missed anyone)

Unauthenticated call to TAP

```
curl https://ws-cadc.canfar.net/youcat/tables

RESPONSE HEADERS:

www-authenticate: ivoa standard_id="ivo://ivoa.net/sso#tls-with-password",
access_url="https://ws-cadc.canfar.net/ac/login"

www-authenticate: ivoa standard_id="ivo://ivoa.net/sso#OAuth",
access_url="https://ws-cadc.canfar.net/ac/authorize"

www-authenticate: ivoa standard_id="ivo://ivoa.net/sso#tls-with-certificate"

www-authenticate: Bearer
```

The **'what'**, **'how'**, and **'where'** information set in www-authenticate

(Auth Strawman, Pat Dowler: https://wiki.ivoa.net/internal/IVOA/InterOpMay2020GWS/auth-reqs-strawman.pdf)

These are set on *all* unauthenticated calls to all places (successful or not)

Authenticated call to TAP

```
curl -H "Authorization: ivoa <token>" https://ws-cadc.canfar.net/youcat/tables

RESPONSE HEADERS:

x-vo-authenticated: majorb
```

Successful call to #tls-with-password

```
curl -d "username=majorb" -d "password=pwd" https://ws-cadc.canfar.net/ac/login

RESPONSE HEADERS:

x-vo-authenticated: majorb
```

Token in response body

Failed call to #tls-with-password

```
curl -d "username=majorb" -d "password=wrong" https://ws-cadc.canfar.net/ac/login

RESPONSE HEADERS:

www-authenticate:
    ivoa standard_id="ivo://ivoa.net/sso#tls-with-password",
    access_url="https://ws-cadc.canfar.net/ac/login",
    error="invalid_request",
    error_description="Invalid password"
```

Format of attribute names taken from OAuth2 specification

SSO Update: StandardID Clarifications

Separate standard IDs for Security Methods into two groups:

// Methods of obtaining credentials
`ivo://ivoa.net/sso#tls-with-password`
`ivo://ivoa.net/sso#OAuth`
`ivo://ivoa.net/sso#OpenID`

These give you tokens

// Ways credentials are accepted
`ivo://ivoa.net/sso#anon`
`ivo://ivoa.net/sso#tls-with-certificate`
`ivo://ivoa.net/sso#cookie`
`ivo://ivoa.net/sso#BasicAA`
`ivo://ivoa.net/sso#token`

Information on header use and token types

- Add HTTP examples for all combinations of authentication

Final matters

- 'authcheck' endpoint (eg `https://ws-cadc.canfar.net/youcat/authcheck`)
  - a sibling to the capabilities endpoint

- What should clients use as an index for user identification? Should the value of x-vo-authenticated be deterministic?

- Client implementations
  - TopCat
  - CADC networking libraries (Java)
  - Python/PyVO next?

- Tokens in Credential Delegation Protocol (CDP)

GMS Updates

Now 2 GMS implementations!

INAF & CADC

Two changes resulted from the implementation experience:

1. Change REST API to use param instead of path to identify groups
2. Allow multiple group params

1 - Group identified by param, not path

For check single group membership, was:

```
HTTP GET to /gms/search/{groupName}
```

Now:

```
HTTP GET to /gms/search?group={groupName}
```

Reasons:

- Can't implement with UWS because the path after 'search' is used by UWS (sync, async, etc..)
- Allows extensions of GMS to be extendible (by future GMS versions or custom extensions).

2 - Allow multiple group params

After change #1 the API is:

A) `HTTP GET to /gms/search` - get all user's membership
B) `HTTP GET to /gms/search?group={groupName}` - checks membership in only {groupName}.

Change #2 is to allow multiple group parameters, so the API becomes:

`HTTP GET to /gms/search` with 0..n `group={groupName}` parameters.

- If no group parameters present the scope of the search is all groups in the GMS server
- If 1 or more group parameters are present the scope of the search is limited to the groups identified in the parameters.

Reason:

- It allows the client to optimize (reduce the number of calls to GMS) based on its knowledge of how many group membership checks may be necessary.

GMS RFC Imminent

- Draft Specification with these two changes nearly complete
- Next version a PR

THANK YOU

brian.major@nrc-cnrc.gc.ca