Second Token-based AuthN/Z Hackathon

Andrea Ceccanti
INFN CNAF

September, 16th 2020

Agenda & useful resources

https://indico.cern.ch/event/953075/

Zoom room

Github Repo

Shared Google doc

CERN Mattermost channel

DOMA OIDC testbed docs: https://wlcg-authz-wg.github.io/wlcg-authz-docs/

Pre-requisites

https://github.com/WLCG-AuthZ-WG/hackathon/#pre-requisites

Basically:

- All devs registered in the WLCG VO can get tokens out of IAM
  - See docs https://wlcg-authz-wg.github.io/wlcg-authz-docs/token-based-authorization/
- Services configured to trust the WLCG VO and token issuer
  - with support for WLCG JWT scope based authorization (i.e., storage.read:/ storage.modify:/ scopes supported) and group-based authorization (see the repo for details)
- **NEW** Services configured to trust the `https://tf.cloud.cnaf.infn.it` token issuer
  - This is to enable JWT compliance testing, more on this later

Objectives

To be defined/prioritised based on attendance

A list of High-level objectives is in the agenda

- DOMA HTTP TPC smoke tests supporting JWT authN/Z
- Group-based authorization flows
- WLCG JWT profile compliance
  - Audience restriction…
- Discussion on local user mapping
- …

A possible list of fine-grained objective is https://github.com/WLCG-AuthZ-WG/hackathon/blob/master/objectives.md, feel free to add anything you'd like to see covered

DOMA HTTP TPC Smoke tests using JWT

https://github.com/paulmillar/http-tpc-utils

Extend current test suite to support WLCG issuer JWT tokens

WLCG JWT profile conformance test suite

A JWT profile compliance test-suite to assess conformance with the profile

- Check that issuer checks, signature checks, temporal validity, audience restrictions, path constraints, … are honoured by the implementations

The **token factory** (a mock token issuer) has been deployed at

https://tf.cloud.cnaf.infn.it

This gives the ability to create potentially malformed/expired tokens and will be used by the test suite to check JWT profile compliance (tokens are issued only to authenticated clients)

An endpoint for each service that trust the token factory is needed.

What does it mean supporting the WLCG profile?

As an **OAuth resource server** (RS):

- Ability to extract an access token from an incoming HTTP request
- Ability to parse and validate the incoming access token
  - identify if it has been issue by a trusted and recognized authorization server
  - verify temporal validity
  - verify signature, following OAuth/OIDC conventions
- Ability to honour access token audience restrictions
  - the RS needs the ability to identity itself with (one or multiple) audience labels and honour audience restrictions in access tokens
- Ability to map defined scopes to local authZ
  - e.g., storage.read:/folder on a storage area grants read access to the /folder part of the namespace (including sub-directories)
- Ability to map group-based to local authZ
  - e.g., /cms group membership grants read access to the /cms namespace

What does it mean supporting the WLCG profile?

As an **OAuth resource server** (RS):

- Ability to extract an access token from an incoming HTTP request
- Ability to parse and validate the incoming access token
  - identify if it has been issue by a trusted and recognized authorization server
  - verify temporal validity
  - verify signature, following OAuth/OIDC conventions
- Ability to honour access token audience restrictions
  - the RS needs the ability to identity itself with (one or multiple) audience labels and honour audience restrictions in access tokens

This is typically sorted out by OAuth/OIDC libraries

- Ability to map defined scopes to local authZ
  - e.g., storage.read:/folder on a storage area grants read access to the /folder part of the namespace (including sub-directories)
- Ability to map group-based to local authZ
  - e.g., /cms group membership grants read access to the /cms namespace

WLCG JWT profile conformance test suite

First incarnation lives at:

https://github.com/indigo-iam/wlcg-jwt-compliance-tests

Doesn't do much yet, but can already:

- get a token out of WLCG IAM with the requested scopes and audiences (via oidc-agent)
- submit GET/PUT/DELETE/COPY requests against SEs trusting the WLCG VO

Objectives for the hackathon:

- Integrate the token factory
- Add basic JWT profile compliance (required claims in tokens, audience restrictions, signature checks (also using different token issuer keys)
- Add basic scope-based authz tests (path contraints handling in storage.* scope, storage.create vs storage.modify handling… etc.)
- Add basic group-based authz tests

Group-based authorization flow

The scope-based authZ scenario for RUCIO-managed HTTP transfers has been already defined at the end of last year

- and AFAIK is currently implemented in RUCIO and FTS
- More detail in this slide deck

For the group-based authorization flow I've drafted a proposal here to bootstrap discussion:

https://github.com/WLCG-AuthZ-WG/hackathon/blob/master/authorization-flows.md

Local user mapping

We have established mapping mechanisms (i.e., LCMAPS, Argus) to map X509 certificates and VOMS attribute information to local UNIX accounts.

Do we need a similar mapping mechanism also for tokens? (there are already some implementations (e.g., scitokens library, dCache, EOS?)

What are the requirements that have emerged up to now?

Can we have a common approach?

A remote hackathon…

We have communication channel in place and hopefully clear objectives and topics to discuss but… how do we productively handle this?

Suggestions more than welcome!

Thanks for your attention.
Questions?