Contribution ID: **11** Type: **Full presentation: long (25 mins.)**

oidc-agent: Your OpenID Connect tokens on the command line

Tuesday, 3 November 2020 11:15 (20 minutes)

Background

OpenID Connect is widely used in modern Authentication and Authorization Infrastructures including the infrastructures of multiple EU projects like the European Open Science Cloud and also EGI. Due to their nature, OpenID Connect Access Tokens were not straightforward to use from the command line. They have a high character count and are short lived, so they cannot be learnt by heard like a password. Copying the access token from a web service whenever needed is clearly suboptimal in a command line based process. However, retrieving an access token on the command line without oidc-agent requires substantial effort that is both, time consuming and cannot be expected from the average user.

Considering this insufficient usability from the command line, our goal was to overcome this by developing a tool that manages OpenID Connect tokens. It should allow a user to obtain access tokens on the command line as easy as possible, so it can be integrated in his workflow.

OIDC-AGENT

Oidc-agent is the swiss-army-knife tools for OpenID Connect in any non-web environment.

The design of oidc-agent is oriented at the ssh-agent, providing the user a familiar way to handle OIDC tokens. Essentially oidc-agent supports several flows to obtain the Refresh Token, which it uses whenever an Access Token is required. All credentials are stored in encrypted ways (both on Disk and in RAM).

In summary, oidc-agent supports a wide range of features:
- Handle all communicate with OpenID Provider
- Register OIDC client and initialize configuration
- Store ecnrypted configurations
- Provide Access Tokens to
- command line usage (syntax allows easy integration)
- other applications
- Easy to use, hidden complexity
- Libaries for various languages so other applications can directly obtain tokens from the agent:
- C
- Go
- Python
- Integrated with Xsession to autostart at startup and availabilty throughout a session
- Agent forwarding to obtain tokens on remote computers
- Tested to work with many OIDC providers
- EGI-Checkin, IAM, B2Access, Keycloak, Human-Brain, ...
- Support of restricted access tokens:
- scope
- audience
- Privacy and security focused design.
- Privilege separation
- Strong cryptography

- Memory obfuscation
- Local application run by the user on his own machine
- No data collection and calling home
- Open source code under the MIT license.
- Available for different platforms:
- Debian/Ubuntu via PPA
- Process started to include oidc-agent in the official debian package repository.
- CentOS as prebuilt package
- Gentoo
- Fedora and EPEL is planned
- MacOS via homebrew

Primary authors: ZACHMANN, Gabriel (Karlsruhe Institute of Technology); Dr HARDT, Marcus (Karlsruhe Institute of Technology); STEVANOVIC, Uros (KIT-G)

Presenter: Dr HARDT, Marcus (Karlsruhe Institute of Technology)

Session Classification: Authentication-Authorisation solutions - Part 1