How do permissions work?

In an ideal world, all permissions are managed directly in Keycloak and propagated down to downstream systems automatically. We mostly live in that world. We would also like more parts of profiles to propagate from Keycloak into downstream systems (see changing names document).

First, let's enumerate the access that we *have available* to grant.

Available access

Roles that exist in Keycloak

"sticky" is referring to whether later-removed permissions get stuck in the downstream system if they are removed upstream

- Wiki roles (not sticky)
    - Editor
    - ModerationBook
    - Admin
- Buildbot (not sticky)
    - can-perform-mutations (currently just a gate on all access)
- Forgejo (administrator is probably sticky, however others are not)
    - Administrator permissions
    - lix-project/nixos issue triage team
    - lix-project owners team
- Keycloak (not sticky)
    - lix-project-user-admin realm role grants administration on keycloak
- Grafana (?)
    - Admin
    - Editor
- Pad (not sticky)
    - access
- Mattermost (not sticky?)
    - access

Roles that we wish existed in Keycloak

These can't happen due to current technical limitations.

- Gerrit: https://git.lix.systems/lix-project/web-services/issues/100
    - Lix team

    - Possibly other rights like tvix-fork team or others?

Structure of access

Keycloak appears to want its structure to work like:

Group -> Composite Realm Role -> Client Role

We don't have that many groups or client roles to assign to make composite realm roles make much sense at the time of this writing.

When creating new clients, make their roles *client roles*, which we can then assign to other objects inside Keycloak so we can do role-based access control at a later time without having to mess with the services.

Groups

- lix core team -> grants administrative permission across a lot of infrastructure:
    - Wiki
        - ModerationBook
        - Admin
        - Editor
    - Forgejo
        - Administrator
        - lix-project owners team
    - Grafana
        - Editor
        - Admin
    - Mattermost
        - access
    - Pad
        - access
    - Buildbot
        - can-perform-mutations
- trusted-contributors
    - Forgejo
        - lix-project/nixos issue triage team
    - Wiki
        - Editor
    - Buildbot
        - can-perform-mutations

Policy on who goes in groups

- lix core team: *only* members of the Lix core team (this group will probably be rearranged later on)
- trusted-contributors: anyone who is reasonably competent and trustworthy: these are basically the rights we hand to every single package maintainer in nixpkgs, and we want to do similarly in Lix.
  Especially assign this if someone should be able to have issues assigned to them (since it is a prerequisite); this means this role should be handed to anyone who is going to contribute code so we can use the assignment feature of Forgejo to say who is taking an issue.
  - /private beta: this group has been made to indicate who was in trusted-contributors already due to our simply dumping everyone from the private beta into it

---

Revision #2
Created 10 May 2024 05:39:37 by jade
Updated 10 May 2024 21:59:28 by jade