RESTful Subject Source for Grouper to Complement the LDAP and SQL Protocols Already Supported

Grouper uses the Internet2 “Subject API” which used to be used by Signet and Grouper when signet was a thing. I have used it in a couple of other projects, but in general it is a grouper thing at this point. Basically it allows for various subject sources.

Subject sources have:

- sourceId (identifies the source)
- other attributes like friendly name

Subjects managed by the source are identified by:

- sourceId
- subjectId: which should not change (i.e. probably opaque)

It used to be that to identity a subject you needed the sourceId, the subjectId, and the type (e.g. person, group, system, etc), but we migrated away from that and now subjects are identified by sourceId and subjectId. I don’t think we have a standard way of determining if a subject is a person or a system or group except by looking at which source it originated from, and we don’t really care anyways [1].

Grouper can manage what we call “local entities” which are things on the grouper namespace that don’t have members (like groups do), but can we assign as members of groups or have permissions. Or you could have a simple SQL query or LDAP filter to manage your external entities or non person entities.

A little more info… subjects can be looked up by “subjectIdentifier” which is something that uniquely identifies the subject but which could change. i.e. don’t store this in your database since it can change.

At Penn we have these subject sources:

- g:gsa: standard grouper subject source returns groups, the uuid is the subjectId and the system name is a subjectIdentifier
- pennperson: for people at penn, the pennid is the subjectId (e.g. 12345678), the netId is a subjectIdentifier (e.g. jsmith), and the eppn is a subjectIdentifier (e.g. jsmith@upenn.edu)
- servPrinc: this has Kerberos service principals (how we do our WS and LDAP authn from applications). The service principal is the subjectId. Note that these are just stored in a SQL table and managed by a simple webapp
- goruperEntities: These are what I described above. Anyone who can create objects in a folder can create these. I use them to represent schemas who need access to a VPD/FGAC oracle schema. But someone could use them to represent whatever needs to be used as an entity
- grouperExternal: Grouper has a way to invite and manage external users (e.g. for incommon federated systems or social authn systems). It gives them a uuid which is the subjectId and the eppn is a subjectIdentifier (since the eppn could change)

Thanks,

Chris

[1] i.e. if the authentication system returned an entity (person or system), and we go to grouper and ask if that entity has access to something, we don’t care what type of entity it is. .