Agenda

- 211 Privileges introduction
- 211.1 Folder privileges
- 211.2 Group privileges
- 211.3 Attribute privileges
- 211.4 Grouper system admin
- 211.5 Inherited privileges
- 211.6 Grouper security groups
Privilege definition

- Grouper privilege: Grant that allows a subject to do something in Grouper
- Privilege == entitlement == right == permission == grant
- We call it privilege in this case
- We call “permission” an externalized one of these things from an application stored in Grouper
- Entitlement is usually saved for SAML
Types of Grouper privileges

• Generally we talk about Group, Folder, Attribute definition privileges
• There are other “privileges”
  • System administrator
  • Global READ of memberships
  • Who can execute Web Service calls
  • Who can do attestation
  • List is very long
Example “privilege”

- From grouper.base.properties
Default privileges

• If a subject creates an object (group, folder, attribute), then that subject will automatically be an ADMIN of the object

• Unless they are in an inherited ADMIN assigned group

• Unless they are a sysadmin
Default “all” privileges

- If the subject “EveryEntity” is assigned then every subject effectively has the privilege
- You can default EveryEntity to various privileges in the grouper.properties
  - E.g. VIEW or READ
  - Perhaps not recommended
Sometimes a privilege implies another privilege

• Some privileges are superset of other privileges

• E.g. if you can ADMIN a folder, you can CREATE objects in the folder

• E.g. if you can UPDATE a group’s memberships, you can OPTIN yourself to the group
Privileges inform the UI

- The UI will appear differently to different users depending on their privileges
- E.g. If you can’t ADMIN a group, you can’t see the privilege tab
Hands on: Open two browser windows

• Lets have two windows open
• Since cookies are shared, if you use the same browser, use an incognito window. Otherwise separated browsers
• One as banderson (who is a sysadmin)
• One as jsmith (commoner)
Hands on: See jsmith

- Look as jsmith, can’t see much
Folder privileges

Grouper in Action - 211
Folder privileges

• ADMIN: Can do anything including rename, edit description, delete the folder, create objects in folder, assign attributes, etc

• CREATE: Can create objects in folder

• ATTR_READ: Can see attribute assignments (of allowed attributes)

• ATTR_UPDATE: Can assign attributes (of allowed attributes)
Hands on: Folder privileges

- Click on the test folder privilege tab
- Add jsmith with CREATE
Hands on: Folder privileges (continued)

- Look as jsmith, refresh
- Create a group: test:jsmithGroup
Hands on: Folder privileges (continued)

- Note that jsmith is an ADMIN of folder
- Note that banderson can see it since sysadmin
Group privileges

- **ADMIN**: Can do anything including rename, edit description, delete the group, assign members, assign attributes, etc.
- **UPDATE**: Can add/remove memberships
- **READ**: Can see memberships
- **VIEW**: Can that the group exists, and name and description
Group privileges

- **OPTIN**: Can add self to group
- **OPTOUT**: Can remove self from group
- **ATTR_READ**: Can see allowed attribute assignments
- **ATTR_UPDATE**: Can add or remove allowed attribute assignments
Hands on: Group privileges

- As banderson, create testGroupPriv group
Hands on: Group privileges (continued)

- As banderson, add GrouperSysAdmin as member to test:testGroupPriv
Hands on: Group privileges (continued)

- As banderson, click on privileges tab, assign READ to jsmith
Hands on: Group privileges (continued)

- As jsmith, see group, do you see differences?
Hands on: Group privileges (continued)

• Look as banderson and compare
Attribute privileges

Grouper in Action - 211
Attribute privileges

• Note: these are only on the attribute definition

• Attribute names use their definition’s privileges

• E.g. if you can VIEW a definition, you can VIEW the attribute name
Attribute privileges

- **ADMIN**: Can do anything including rename, edit description, delete the attribute, assign it (if allowed on target), assign attributes to the def, etc
- **UPDATE**: Can add/remove attribute assignments (if allowed on target)
- **READ**: Can see assignments (if allowed on target)
- **VIEW**: Can that the attribute exists, and name and description
Attribute privileges

- OPTIN: Can assign attribute to self
- OPTOUT: Can assign attribute to self
- ATTR_READ: Can see allowed attribute assignments
- ATTR_UPDATE: Can add or remove allowed attribute assignments
What do I need to READ attribute assignments???

• Two things
  • 1. READ or ADMIN on the attribute definition assigned
  • 2. Privileges on the assignment target:
    • A. ATTR_READ or ADMIN on group
    • B. ATTR_READ or ADMIN on folder
    • C. ATTR_READ of ADMIN on attribute def
• Note: if you can see an attribute assignment, you can see the value(s)
Hands on: Exercise for another time

- As banderson, create an attribute def, assignable to folders
- Assign READ priv to jsmith
- Create an attribute name
- Assign the attribute to the test folder
- See that jsmith cannot see attribute assignment
- Assign ATTR_READ to jsmith on the test folder
- See that jsmith can see attr assignment
Grouper system admin

- There is a GrouperSysAdmin user which is all powerful
- This is not used in WS or UI since SSO is usually configured and that principal cannot login
- There is a sys admin group configured in grouper.properties
- Members of this group are equivalent to the GrouperSysAdmin subject
Grouper system admin (continued)

- See grouper.base.properties

```properties
# A wheel group allows you to enable non-GrouperSystem subjects to act like a root user when interacting with the registry.
groups.wheel.use = false

# Set to the name of the group you want to treat as the wheel group. The members of this group will be treated as root-like users.
groups.wheel.group = $$grouper.rootStemForBuiltInObjects$$:sysadmingroup

# A viewonly wheel group allows you to enable non-GrouperSystem subjects to act like a root user when viewing the registry.
groups.wheel.viewonly.use = false

# Set to the name of the group you want to treat as the viewonly wheel group. The members of this group will be treated as root-like users when viewing objects.
groups.wheel.viewonly.group = $$grouper.rootStemForBuiltInObjects$$:sysadminViewersGroup

# A readonly wheel group allows you to enable non-GrouperSystem subjects to act like a root user when reading the registry.
groups.wheel.readonly.use = false

# Set to the name of the group you want to treat as the readonly wheel group. The members of this group will be treated as root-like users when reading objects.
groups.wheel.readonly.group = $$grouper.rootStemForBuiltInObjects$$:sysadminReadersGroup
```
Hands on: Sysadmin

- See banderson’s groups and find the sysadmin group

The following table lists all groups in which Bob Anderson is a member.

<table>
<thead>
<tr>
<th>Folder</th>
<th>Group name</th>
<th>Membership</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>banner_user</td>
<td>Indirect</td>
<td></td>
</tr>
<tr>
<td></td>
<td>banner_user_allow</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>grouperUiUserData</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sysadminUserGroup</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fulltime</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bulkLoad</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parentGroup</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>qualtrics_excludes</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>qualtrics_system_of_record</td>
<td>Indirect</td>
<td></td>
</tr>
<tr>
<td></td>
<td>qualtrics_system_of_record_and_excludes</td>
<td>Indirect</td>
<td></td>
</tr>
<tr>
<td></td>
<td>twoStepExcludes</td>
<td>Direct</td>
<td></td>
</tr>
</tbody>
</table>
Internet2 Techex 2018

Inherited privileges

Grouper in Action - 211
Folders can inherit privileges to objects

- Can inherit group, folder, or attribute privileges
- Can be for just that one folder, or for that folder and subfolders
- Will automatically assign for newly create objects
- If you remove the inherited assignment, it will remove all assignments of objects
Hands on: inherited group privileges

• As banderson: Go to test folder, click tab: More -> Privileges inherited to objects in folder

There are currently no inherited privileges related to this folder.
Hands on: inherited group privileges

- Assign VIEW to jsmith on all groups
Hands on: inherited privs (continued)

- As jsmith, see groups
Privileges to individuals have limitations

• Can’t do composites on activeEmployee
• Can’t easily clone a “role” for another user
• Too many individual assignments
Privileges to individuals have limitations - solution

- Try to generally assign privileges to groups, not individuals
- Try to use inherited privileges to the group from a folder
- Generally projects have 4 roles:
  - Admin: can do everything
  - Manager: can READ / UPDATE
  - Reader: can READ
  - Viewer: can see objects exist
Hands on: Privileges to groups

- As banderson: create the group test:testManagers
- Add jsmith to group
Hands on: Privs to groups (continued)

- On the test folder, assign inherited READ/UPDATE privs to testManagers