Grouper deprovisioning development notes

Attributes

The deprovisioning attribute is assignable to memberships, groups, and folders. This is a single-assign marker attribute. The rest are assigned on that attribute assignment. Note: not all attributes are used for each type of owner (group/folder/membership)

Attribute name	Description
deprovisioning	Marker on group/folder
deprovisioningAffiliation	Affiliation configured in the grouper.properties
deprovisioningDeprovision	true|false, true to deprovision, false to not deprovision (default to true). Note, if this is set on a daemon job, then it will not deprovision any group in the loader job (they will be marked as such)
deprovisioningStemScope	one|sub, if in folder only or in folder and all subfolders (default to sub)
deprovisioningSendEmail	true|false, default to false. Set this to true for objects where the system of record is outside of grouper or where manual removal is preferred
deprovisioningEmailSubject	custom subject for emails, if blank use the default configured subject. Note there are template variables $$name$$ $$netId$$ $$userSubjectId$$ $$userEmailAddress$$ $$userDescription$$
deprovisioningEmailBody	custom email body for emails, if blank use the default configured body. Note there are template variables $$name$$ $$netId$$ $$userSubjectId$$ $$userEmailAddress$$ $$userDescription$$
deprovisioningAllowAddsWhileDeprovisioned	If allows adds to group of people who are deprovisioned can be: blank, true, or false. If blank, then will not allow adds unless auto change loader is false
deprovisioningAutoChangeLoader	If this is a loader job, if being in a deprovisioned group means the user should not be in the loaded group. can be: blank (true), or false (false)
deprovisioningAutoselectForRemoval	If the deprovisioning screen should autoselect this object as an object to deprovision can be: blank, true, or false. If blank, then will autoselect unless deprovisioningAutoChangeLoader is false
deprovisioningDirectAssignment	If deprovisioning configuration is directly assigned to the group or folder or inherited from parent. true for direct, false for inherited, blank for not assigned
deprovisioningEmailAddresses	Email addresses to send deprovisioning messages. If blank, then send to group managers, or comma separated email addresses (mutually exclusive with deprovisioningMailToGroup)
deprovisioningMailToGroup	Group ID which holds people to email members of that group to send deprovisioning messages (mutually exclusive with deprovisioningEmailAddresses)
deprovisioningSendEmail	If this is true, then send an email about the deprovisioning event. If the assignments were removed, then give a description of the action. If assignments were not removed, then remind the managers to unassign. Can be <blank>, true, or false. Defaults to false unless the assignments were not removed.
deprovisioningShowForRemoval	If the deprovisioning screen should show this object if the user as an assignment. can be: blank, true, or false. If blank, will default to true unless auto change loader is false.
deprovisioningInheritedFromFolderId	Stem ID of the folder where the configuration is inherited from. This is blank if this is a direct assignment and not inherited
deprovisioningLastEmailedDate	yyyy/mm/dd date that this was last emailed so multiple emails dont go out on same day
deprovisioningCertifiedMillis	(String) number of millis since 1970 that this group was certified for deprovisioning. i.e. the group managers indicate that the deprovisioned users are ok being in the group and do not send email reminders about it anymore until there are newly deprovisioned entities

TO DO

- DONE (Chris) Assignment of configuration screen
  - DONE API to propagate configuration down to objects
- (Chris) Add in a last reviewed date similar to attestation. Only send emails if the date the user was added to the deprovisioned group is after the last reviewed date
  - If user leaves "employee", someone deprovisions a user. Select most checkboxes. Member of deprovision_employees. Email is sent to 3rd party system. Email is just like an attestation. Batched. Link to "review deprovioned users" screen. Screen will show users deprovisioned after the last reviewed date. Check checkboxes (optionally), revmoe users, click button this group has been reviewed.
  - If they didnt go to the screen, or not remove users, or not click as reviewed, they would get an email every day for 2 weeks about those deprovisioned users.
  - Reviewed sets a date in the attribute for that group
- Done (Vivek) Daemon to send emails out
  - Loop through all affiliations for deprovisioning, loop through all users in those deprovisioned groups, see what memberships/privileges they have, and on those objects see which ones need emails
  - If the users deprovisioned membership date is after the last reviewed date for the object where they have a membership or a privilege, and that group/folder/attribute is set to send emails, then an email needs to be sent
- Done (Vivek) Also send emails out at the time someone is deprovisioned
  - API method to send emails at the time someone is deprovisioned
- Done (Vivek) Email to be sent out
  - If the user still has a membership or privilege in the object, include a link to the "remove access and mark as reviewed" screen
  - If the user was already removed, but an email should notify, the email should just say the user removed (the daemon emails would never send this)
  - Email can be sent to:
    - "ADMIN" or "UPDATE/READ" for a group/attributeDef, (or "ADMIN" for a folder since there is not READ/UPDATE) (like attestation)
    - comma separated list (like attestation)
    - members of a group (different from attestation)
- (Chris) Need a UI screen for group/folder/attribute to show deprovisioned users with memberships/privileges on that group/folder/attribute
  - Checkboxes to easily unassign
  - Button that allows "mark as reviewed"
- Done (Vivek) Loader needs to decide if user should be added to loader
  - API method to decide this
  - Loader integration to use this
    - if the attribute: deprovisioningAutoChangeLoader is false, then do not remove from loader job
- (Vivek) Dont deprovision the deprovision group itself... startup? add settings?
- Done (Vivek) API method all groups/folders/attributeDefs of access (move from UI code, add attributes)
- (LATER) On the UI if someone adds a member, or assigns a privilege, check to see if that user shouldnt be allowed, and prompt the UI person if they are sure
  - Should cache members of affiliation deprovisioned groups
- (LATER) On WS, throw an error if a deprovisioned user is added to a group which shouldnt have it based on configuration
  - addMember
  - assignPrivilege
  - Option to deny deprovisioned adds in grouper-ws.properties, default to on, allow users to disable this
  - Should cache members of affiliation deprovisioned groups
  - If there is a param in the WS call "overrideDeprovisionedUser" = "true" then allow the assignment
- (LATER) If removed from a role, make sure individual permissions to that user are unassigned as well
- (LATER) Global screen that shows all immediate configuration assignments (like attestation)
- (LATER) If an application owner, they can go to a folder, pull up a user and their access for that folder and easily remove
  - Not a global deprovision
  - Local to that service
- (LATER) add notes and keep track of what was deprovisioned
- (LATER) put message on group membership screen if there are (cached) entries in deprovisioning report
- (LATER) use the "in affiliation" group on deprovisioning reports. or wider group? active?

Do not allow assignments by WS of deprovisioned users to deprovisionable objects by affiliation. Allow a param to override this

Allow global deprovision across affiliations or if no affiliation specified. Or document how to do this

Notes

1. Users of this screen would need to be in a certain group. Grouper admins would also be allowed to use this page
   a. Note: users of this screen would effectively have a lot of access in grouper. They can pull up any subjects and see what they have. They can remove most things. But they do not have to be Grouper admins. This screen could be used by an HR person.
2. This screen could be disabled if an institution does not want it.
3. The screen would have a subject lookup for someone to be deprovisioned
4. When submitting that combobox, all the assignments in grouper would display, as well as deprovisioned status
5. A button "Deprovision user and remove access" adds the user to a built-in group for people who will be deprovisioned.
   a. This group has a membership expiry for a certain configured amount of time (2 weeks is the default)
   b. This group can be used in "exclude" groups or rules in grouper for lockouts
   c. Note, some institutions might already have this "lockout" group
6. Assignments on screen will include direct memberships, privileges, and attribute assignments
   a. Note, permissions are assigned on roles or memebrships in roles so those would not be shown but they would be removed
7. The screen will have checkbox about assignments to deprovision

8. There could be a way to see effective as well as immediate assignments, though it will default to immediate (ones you can deprovision)
9. There is a "check all" and "uncheck all" button
10. An "unassign" button will remove all those assignments
    a. Also adds the user to the deprovisioning group with end date on membership of 2 weeks
    b. Assignments are in point in time so they can be restored later or migrated to another user
11. Groups and folders have attributes related to deprovisioning
    a. Mark a group or folder as ineligible for deprovisioning (e.g. the lockout group)
    b. If Grouper is not the system of record for a group, mark a group or folder with attributes so that emails are sent out to application owners to deprovision that user. This would not remove the assignment in grouper because in this case grouper is not the source of the assignment but instead reflects it in another system. The receiver of the email would need to unassign the user and that data would flow back to grouper ater the next load
        i. e.g. an attribute to say "deprovision_notify_app_owner", an attribute "deprovision_notify_app_owner_email", attribute "deprovision_notify_app_owner_email_subject", "deprovision_notify_app_owner_email_body"
        ii. Attribute keep track of when last emailed so users dont get emailed more than once a day
12. There is feature in loader jobs to not load deprovisioned users (without having to adjust the query). Of course loader jobs could be exempt from this if they need deprovisioned users inside. The default would be to not include them
13. There is an overall audit and then keep individual audits
14. Daemon will send emails to application owners on users to deprovision. Will send one email with batch of users to deprovision
15. Screen for app owners to see who they should look at
16. There is a report of deprovisioned users and assignments they still have access to so that followups can be made after a week or two to make sure everything is removed for that user that should be
17. There could be a report of inactive users and things they are still assigned to to clean out users who left the institution long ago
18. If messaging queues are configured, messages will be sent to deprovision a user
19. If someone adds a deprovioned (for 2 weeks) user to a group or privilege or permission, then they will get a warning that the user has been deprovisioned...
20. Application managers could run membership reports to see which users have been deprovisioned (ever), or which users are not active, not active staff, etc

Future enhancements

- If the entitlements arent known in grouper, have it make a group, rule that adds a user to another group, which notifies someone to see entitlements need to be removed, and remove the user from the group

Comments

1. **Tom Jordan**

   I like the idea of having an attribute on a group that references a deprovision group. We've talked about having a standard set of attributes for Access Policy groups that correspond to the 'helper' groups or objects that usually come along for the ride. Some examples that we've talked about with Access Policy groups include references to manual includes and exclude groups, references to groups defining who's eligible vs. active, etc.

   Notifying an app owner and/or firing a message into a queue on deprovision would probably do a lot to help integrate other deprovisioning workflow.

2. **GETTES**

   Will this address the request for enhancement I submitted to have specify a number of days where a subject is unresolved before an USDU run will actually delete the subject from all groups? If so, then I am all for this solution. Being able to automate USDU (a daily run) and have subjects removed after a specified period is very important - otherwise it's possible to deprovision subjects too soon - especially in cases of a "disaster" whereby many subjects are mistakenly removed from the subject source.