GROUPER DEPROVISIONING

PRESENTED BY: Chris Hyzer, Penn
Agenda

- Explain deprovisioning strategy and issues
- Current situation at Penn
- Problem statement
- Making deprovisioning easier with Grouper legacy features
- Improvements at Penn
- Grouper deprovisioning
Deprovisioning strategy and issues
Deprovisioning

- Removing access for the following reasons
  - No longer Penn affiliated
  - No longer employee
  - No longer IT department related
  - Switches positions in the IT department
  - Not working on a project anymore
Current deprovisioning procedure

- Email to a listserv from IT dept HR
- Deprovision this stuff for this person

  - Box
  - Listserv
  - Phone
  - Voicemail
  - LAN
  - Data Warehouse
  - Mainframe
  - Email/O365 Acct
  - Databases
  - Applications and Servers

  - Flash
  - FAST Apps
  - Atlassian
  - Remedy
  - Org Chart
  - Building Access
  - PlanView
  - LVS
  - ISC Web Site
  - CVS
Issues with current procedure

• Is that the full list?
  • How many services do we have? (hundreds)
• What if someone misses the email?
• What if no email sent?
• What if IdM status is not right
  • (e.g. still getting vacation pay after leaving, out for disability)
Issues with procedure (continued)

- I get an email I don’t recognize the person
- I assume they don’t have access, I don’t check, but wait the situation worsens
- One example is our framework
  - Each of the 100 apps has 3+ envs (dev/prod/test)
  - Each app has local security
  - Need to log in to each to see
- Tedious to check, turnover from contractors…
Deprovisioning current options
Current option #1

- App automatically deprovisions
- via Grouper or IdM or entitlement
Current option #2

- Composite group

  - If someone in the ad hoc group
  - Is not in the staff group
  - They will not be in the overall group
  - Composite intersection
Current option #3

- Rule unassigning

- If someone is removed from the staff group
- And they are in the ad hoc group
- They will be removed from the ad hoc group
Current option #4

- Rule to notify me
  - If someone is removed from the staff group
  - And they are in the ad hoc group
  - Send an email to the admins of the ad hoc group to review
Current option #5

• Grouper (not automatically)

- When I get an email about deprovisioning
- I try to look in Grouper and do all the unassigning
Current option #6

- Grouper (automatically)

  - “Loaded group”
  - Faculty, Students, or All Staff
  - Or an org like ISC staff
Current option #7

• Attestation

- Configure at Group or Folder level
- Email reminders to review group, don’t ignore
- Admins review memberships
- Mark the group as reviewed
Current option #8

- Combine some of those
  - Auto remove
  - Notify admins
  - Attest periodically
The problem

- Why not drive all access from Grouper and rules?
- App not compatible
- Existed before Grouper implementation
- Too much effort or perceived effort to integrate
- Prefer to assign privileges with the app’s UI
Improving deprovisioning with current Grouper features
Making deprovisioning easier

• Auto-feed entitlements to Grouper read only
• Grouper can notify you to remove access
  • When the email goes around
  • When someone’s org changes
• You can take advantage of other features
  • Attestation
  • Reports against active employees at in org
  • View your apps privilege history
  • More accurate picture of what someone has access to
  • More easily clone access onboarding someone new
Analyze memberships in Grouper

- Composite <minus> the group of app users with active community members
- Shows users suggested to be removed
Custom web application example

- Make a SQL view of memberships from framework
- Load that into Grouper (read only)
- When someone is deprovisioned email the admin of that application
- Send monthly reports or on demand
- Attest the access periodically
Step 1: make the view

```
CREATE OR REPLACE FORCE VIEW FAST_USER_GROUP_V
(
    PENNID,
    GROUP_NAME_SYSTEM,
    PENNKEY,
    EMAIL_ADDRESS,
    NAME
)
BEQUEATH DEFINER
AS
    SELECT pennid,
           group_name_system,
           pennkey,
           email_address,
           name
    FROM fast_user fu, fast_group fg, fast_user_group fug
    WHERE fu.USER_ID = fug.USER_ID
         AND fg.GROUP_ID = fug.GROUP_ID
         AND (fug.DISABLED_DATE IS NULL OR fug.DISABLED_DATE > SYSDATE)
         AND (fg.DISABLED_DATE IS NULL OR fg.DISABLED_DATE > SYSDATE)
         AND (fu.DISABLED_DATE IS NULL OR fu.DISABLED_DATE > SYSDATE);
```
Step 2: grant the view to Grouper for loader

```sql
GRANT SELECT ON FAST_USER_GROUP_V TO AUTHZADM WITH GRANT OPTION;
```
Step 3: see the groups

<table>
<thead>
<tr>
<th>PENNID</th>
<th>GROUP_NAME_SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10012528</td>
<td>FAST_AUTHENTICATED</td>
</tr>
<tr>
<td>10021368</td>
<td>FAST_ADMIN</td>
</tr>
<tr>
<td>10021368</td>
<td>eraAdmin</td>
</tr>
<tr>
<td>10064187</td>
<td>FAST_ADMIN</td>
</tr>
<tr>
<td>10198457</td>
<td>FAST_ADMIN</td>
</tr>
<tr>
<td>69795056</td>
<td>FAST_ADMIN</td>
</tr>
<tr>
<td>10754302</td>
<td>Change Management Admin</td>
</tr>
<tr>
<td>10754302</td>
<td>FAST_ADMIN</td>
</tr>
<tr>
<td>10120893</td>
<td>eraAdmin</td>
</tr>
<tr>
<td>10021294</td>
<td>eraAdmin</td>
</tr>
</tbody>
</table>
Step 4: see the grouper view

<table>
<thead>
<tr>
<th>SUBJECT_ID</th>
<th>SUBJECT_SOURCE_ID</th>
<th>GROUP_NAME</th>
</tr>
</thead>
</table>
Step 5: It’s on the UI too

![Image of a UI with a file named 'groups' and options for 'Folder contents', 'Privileges', and 'More'.]
Step 5: It’s on the UI too
Step 5: It’s on the UI too

Home > Root > penn > isc > ait > apps > fast > groups > fastPdfService > prod

prod

More

Filter for: Folder, group, or attribute name

Folder contents Privileges More

Name

Up one folder

Administrative Information Technology and Data Admin
Application Services
Change Management Admin
Change Management Approver
Client Services
Communications
Computer Operations
FAST_ADMIN

28
Step 5: It’s on the UI too

![Client Services UI screenshot]

- The following table lists all entities which are members of this group.

<table>
<thead>
<tr>
<th>Entity name</th>
<th>Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>Direct</td>
</tr>
<tr>
<td>Ann</td>
<td>Direct</td>
</tr>
<tr>
<td>Ian</td>
<td>Direct</td>
</tr>
</tbody>
</table>

**Uh...**
Step 6: Attestation

No attestation is configured on this folder or parent folder

Attestation
Yes, does have attestation directly assigned
If this folder has attestation configured on it directly, not inherited from ancestor folder

Has attestation
Yes, attestation is enabled
If configured to be attested. It is possible that attestation is configured to be off.
## Step 6: Attestation

<table>
<thead>
<tr>
<th>Send email</th>
<th>Yes, send email alerts (recommended)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If email alerts should be sent out to people who need to review the membership</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Email group managers</th>
<th>Yes, email the group admins and updaters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If the group managers (can ADMIN or can READ and UPDATE) should be emailed when group attestation is due</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use default recertify days</th>
<th>Yes, use default recertify days (180 days)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Use the system wide default of 180 days for recertification</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Folder scope</th>
<th>All groups in this folder or subfolders</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Do these attestation settings affect groups in this folder and all subfolders (default) or just this folder</td>
</tr>
</tbody>
</table>
Step 7: Monthly report

From: penngroups-noreply@isc.upenn.edu

Subject: Monthly report of FAST users not in ISC

To: Hyzer, Chris

Attached is your monthly report of FAST users not in ISC.
Please review and mark the groups as attested in penngroups.

Thanks
## Step 8: LOTS OF WORK TO DO

<table>
<thead>
<tr>
<th>B</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>fService:test:eraAdmin</td>
<td>Mol</td>
</tr>
<tr>
<td>fService:prod:Change_Management_ApiManager</td>
<td>Hae</td>
</tr>
<tr>
<td>f.test:FAST_ADMIN</td>
<td>Ric</td>
</tr>
<tr>
<td>fService:prod:eraAdmin</td>
<td>Pen</td>
</tr>
<tr>
<td>fService:prod:eraAdmin</td>
<td>Cam</td>
</tr>
<tr>
<td>fService:test:eraAdmin</td>
<td>Tod</td>
</tr>
<tr>
<td>fService:prod:eraAdmin</td>
<td>Rob</td>
</tr>
<tr>
<td>fService:prod:eraAdmin</td>
<td>Vve</td>
</tr>
<tr>
<td>fService:test:dbPasswordAdmin</td>
<td>Julia</td>
</tr>
<tr>
<td>fService:prod:FAST_ADMIN</td>
<td>Julia</td>
</tr>
<tr>
<td>fService:prod:eraAdmin</td>
<td>Ani</td>
</tr>
<tr>
<td>fService:prod:Change_Management_ApiManager</td>
<td>Craig</td>
</tr>
<tr>
<td>fService:prod:FAST_ADMIN</td>
<td>Craig</td>
</tr>
<tr>
<td>fService:prod:Network_Engineering_andMaintenance</td>
<td>Jan</td>
</tr>
<tr>
<td>fService:test:eraAdmin</td>
<td>Matt</td>
</tr>
<tr>
<td>fService:prod:Computer_Operations</td>
<td>Matt</td>
</tr>
<tr>
<td>f.dev:FAST_ADMIN</td>
<td>Ric</td>
</tr>
<tr>
<td>fService:test:dbPasswordAdmin</td>
<td>Toby</td>
</tr>
</tbody>
</table>
Oracle schema accounts

- Add all the schemas
- Make sure not already disabled
- Make sure they are a person and not an application
Oracle schema accounts
Oracle schema accounts

Filter for: Folder, group, or attribute name

Name

- Up one folder
- pcom
- tcom

Show: 100
Oracle schema accounts POC

The following table lists all entities which are members of this group.

<table>
<thead>
<tr>
<th>Entity name</th>
<th>Membership</th>
<th>Choose action</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td>Paul</td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td>Alice</td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td>Bob</td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td>Carol</td>
<td>Direct</td>
<td>Actions</td>
</tr>
</tbody>
</table>
109 accounts in test idm database non-staff!!!!!!!

The following table lists all entities which are members of this group.

Note: this group is a composite owner: 🚹 tcom_nonstaff is a composite of 🚹 tcom minus 🚹 staff
The following table lists all entities which are members of this group:

<table>
<thead>
<tr>
<th>Entity name</th>
<th>Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>Direct</td>
</tr>
<tr>
<td>N</td>
<td>Direct</td>
</tr>
<tr>
<td>T</td>
<td>Direct</td>
</tr>
<tr>
<td>A</td>
<td>Direct</td>
</tr>
<tr>
<td>S</td>
<td>Direct</td>
</tr>
</tbody>
</table>

Filter for:
- All members
- Member name
The following table lists all entities which are members of this group.

Note: this group is a composite owner: isc_o365_users_non_isc is a composite of isc_o365_users minus isc_staff

<table>
<thead>
<tr>
<th>Entity name</th>
<th>Membership</th>
<th>Choose action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Indirect</td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>Indirect</td>
<td>Actions</td>
</tr>
</tbody>
</table>
|             | Indirect   | Actions       |-
The following table lists all entities which are members of this group.

<table>
<thead>
<tr>
<th>Entity name</th>
<th>Membership</th>
<th>Choose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>Direct</td>
<td>Actions</td>
</tr>
<tr>
<td></td>
<td>Direct</td>
<td>Actions</td>
</tr>
</tbody>
</table>
The following table lists all entities which are members of this group.

Note: this group is a composite owner: prod_kite_nonstaff is a composite of prod_kite_enabled.
Will do in future

- Unix accounts (via script and WS)
- Sudo entries (via script and WS)
- Door access (feed into database and SQL loader)
- All other deprovisioning applications
Grouper deprovisioning
Grouper deprovisioning

- Soon to be released in 2.3 patch and 2.4
- [https://spaces.internet2.edu/display/Grouper/Grouper+deprovisioning](https://spaces.internet2.edu/display/Grouper/Grouper+deprovisioning)
- Allow deprovisioning administrators to deprovision access
  - Could be HR
  - For “red button”
Grouper deprovisioning enabled

- This feature is enabled by default
- You can disable it if you like in grouper.properties

```bash
# Deprovisioning

# if deprovisioning should be enabled
deprovisioning.enable = true
```
Grouper deprovisioning realms

- Identify affiliations, depts, cohorts, to deprovision

```python
# comma separated realms for deprovisioning e.g. employee, student, etc
# these need to be alphanumeric suitable for properties keys for further c:
# deprovisioning.realms =
```
Grouper deprovisioning act as admin

• If the deprovisioning admin is an inherited admin in a folder, then you don’t need this
• If you have an HR person deprovisioning employees, this might be useful

```bash
# users in this group who are admins of a realm but who are not Grouper SysAdmins, will be # able to deprovision from all grouper groups/objects, not just groups they have access to UPDATE/ADMIN
deprovisioning.admin.group = $$deprovisioning.systemFolder$$:deprovisioningAdmins
```
Grouper deprovisioning built-in groups

- Managers who can deprovision each realm

**Deprovisioning managers**

Identify the deprovisioning managers and add them to the managers group. e.g. "employee", then the group would be:

```
etc:deprovisioning:managersWhoCanDeprovision_employee
```
Grouper deprovisioning built-in groups

- Deprovisioned users
- etc:deprovisioning:usersWhoHaveBeenDeprovisioned_employee
- Internal group that Grouper uses to add users who have been deprovisioned
- Sets an end date according to grouper.properties
- Defaults to 2 weeks

```
# number of days in deprovisioning group. Should be the amount of time for
# systems of record to catch up and
# for people to change external systems of record in manual processes
deprovisioning.defaultNumberOfDaysInDeprovisioningGroup = 14
```
Configure folders and groups to be deprovisioned

- Identify a folder or group to have deprovisioning configuration
- Do this in the UI as the deprovisioning realm admin
- This configuration is realm specific
Deprovision users

• Once deprovisioning and realms are configured, start deprovisioning

Miscellaneous
Functions across the repository

- Inherited privileges
- Subject API diagnostics
- Instrumentation
- Loader jobs
- Attestation
- Deprovisioning
Deprovision users

• Select realm to be deprovisioned
Deprovision users

• See recently deprovisioned users
  • check up on access removal
Deprovisioning users

• Deprovisioning menu
Deprovision users

- Deprovision a user in a realm
See access, remove it

Below are the directly assigned memberships and privileges. Objects which are restricted from deprovisioning are not shown by default.

<table>
<thead>
<tr>
<th>Folder</th>
<th>Object</th>
<th>Object type</th>
<th>Group Member?</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>etc : grouperUI</td>
<td>grouperUIUserData</td>
<td>Group</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>testC</td>
<td>groupC</td>
<td>Group</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>testD</td>
<td>groupD</td>
<td>Group</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>testA</td>
<td>groupA</td>
<td>Group</td>
<td>false</td>
<td>Admin, Read, Update</td>
</tr>
<tr>
<td>testB</td>
<td>groupB</td>
<td>Group</td>
<td>false</td>
<td>Admin</td>
</tr>
<tr>
<td>testC</td>
<td>groupC</td>
<td>Group</td>
<td>false</td>
<td>Admin, Optout</td>
</tr>
<tr>
<td>testD</td>
<td>groupD</td>
<td>Group</td>
<td>false</td>
<td>Attribute read, Attribute update</td>
</tr>
<tr>
<td>testE</td>
<td>groupE</td>
<td>Group</td>
<td>false</td>
<td>Admin</td>
</tr>
<tr>
<td>Root</td>
<td>testA</td>
<td>Folder</td>
<td>NA</td>
<td>Create, Attribute read</td>
</tr>
<tr>
<td>testA</td>
<td>stemA</td>
<td>Folder</td>
<td>NA</td>
<td>Admin</td>
</tr>
<tr>
<td>testA</td>
<td>testAttributeDef</td>
<td>Attribute</td>
<td>NA</td>
<td>Attribute read, Update, View</td>
</tr>
</tbody>
</table>
Configure deprovisioning enabled

• true|false,
  • true to deprovision (default to true)
  • false to not deprovision
• Note, if this is set on a daemon job, then it will not deprovision any group in the loader job

• If you do/don’t want people deprovisioned
• If it is (un)related to the deprovisioning realm
• Do not show a checkbox on the deprovisioning screen
• Do not restrict assignments during the 2-week deprovisioning period
Deprovisioning folder scope

- Sub (Default)
  - Applies to objects in this folder and subfolders
- One
  - Only applies to this folder
De-provisioning configuration: email

- Send? 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
- Default email template in grouper.properties
- Custom email template configured on folder or group
  - Includes email, subject, email list to send to, or to admins of group
  - Similar to
Configuration: allow adds while deprovisioned

- False (default)
  - If a user tries to add this user to a group while in the 2-week deprovisioned window, give an error. But allow the user to override it
- True
  - If you want users added to the group or folder while deprovisioned in this realm
Configuration: auto change loader

- Default in the grouper.properties
  - Depends on your overall strategy
  - Useful when the systems of record do not update real-time when the red button is pressed
  - Useful if you don’t have a red button group excluded from your policy groups
- True (default)
  - If this is a loader job, if being in a deprovisioned group means the user should not be in the loaded group.
- False
  - Let the external SQL / LDAP be the system of record
Configuration: auto select for removal

• True (default): if the checkbox should be checked by default on the deprovisioning screen for a group/folder

• False: if the checkbox should not be checked. Allow the deprovisioning admin to check the checkbox based on circumstance
Configuration: show for removal

- True (default)
  - if the checkbox should be shown on the deprovisioning screen for a group/folder

- False: if the checkbox should not be shown.
  - If the group isn’t related to the deprovisioning realm or if the system of record is not in grouper, then do not allow the deprovisioning administrator to remove the user
Configuration: realm group

- Might have conflicting realms
- Example: VPN used by employees and students
- Deprovisionable by both employees and students
- In grouper.properties

```bash
# Group name of the group that identifies generally if an entity is
# in this realm. So if a group is deprovisioned
# by various realms, then only deprovision if the entity in the group
# is not in any realm eligible group.
# e.g. VPN is deprovisioned by realms employee and student. If the person
# is no longer an employee, but is still
# a student, then dont deprovision.
# deprovisioning.realm_<realmName>.groupNameMeansInRealm = a:b:c
# deprovisioning.realm_employee.groupNameMeansInRealm = community:employee
```
Conflicting realms

• If a group has conflicting realms (e.g. apps:vpn:vpnUser_includes)
  • Deprovisionable by employee and student
• If an employee is deprovisioned, and not a student, check the checkbox
• If a student is deprovisioned, and not an employee, check the checkbox
• If has both realms, and one is removed, do not check the checkbox
UI effects

- Do not allow assignments of deprovisioned users to deprovisionable groups by realm
  - Screen will prompt if user wants to override
Loader effects

- Do not allow assignments of deprovisioned users to deprovisionable groups by realm
  - Useful if your system of record is not immediately up to date
  - Can remove all loader jobs from deprovisioning users in grouper.properties
  - Can remove each individual loader job from deprovisioning by realm
  - If there are issues, then everything will sync up after 14 days
WS effects

- Do not allow assignments of deprovisioned users to deprovisionable groups by realm
  - Can allow this in grouper.properties
- Allow an override param
  - allowAssignDeprovisionedUser=true
How to set this up

- See which departments / affiliations want to participate
- Create your realms
  - Start with employee?
- Carefully deprovision at first
  - Note which groups shouldn’t be deprovisioned
  - Configure this realm and / or other realms
  - See which loader jobs are not real time if not all
- Get more applications to use Grouper as system of record
- Get more applications to send entitlements to Grouper read-only
- Use attestation / reports
Thanks

Grouper deprovisioning

PRESENTED BY: Chris Hyzer, Penn

© 2018 Internet2