

Attribute based access control (ABAC) with scripted groups

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

Overview

To implement access policies, it has often been necessary to set up intermediate groups, include/exclude, requirement groups, and [allow/deny manual groups](#). Grouper has features to help in this area including: [rules](#), [hooks](#), templates, move/copy, import/export, and GSH scripts.

The ABAC with scripted groups feature is designed to offer increased efficiency in implementing access policies. It's important for the common groups and policy language to be well documented and people to be properly trained.

JEXL loaded groups

In Grouper v2.6.6+ there is a first pass at JEXL loaded groups using memberships of groups only. In v5+ scripted groups can also be based on [entity data fields](#). It is basic and can be built on. Note: this is subject to change as we see a working solution and discuss the optimal path forward.



See the blog!

For more info, see the February 2022 blog on [Attribute Based Access Control with Grouper](#).

Video

Expression language (JEXL) scripts facilitate implementing the part of ABAC that defines who is included in a policy based on attributes of those users. Other parts of ABAC such as resource attributes or environment attributes can be taken into consideration with Grouper permissions or by the service which has protected resources.

We want to be able to craft policies by an expression instead of creating loaders or tons of reference groups based on cartesian products of basis/ref groups.

Individual groups can be configured to automatically have their membership managed with individual subjects (or in future groups as members)

Why do we need this feature?

- Reduces pre-loaded rollups that might not be used
- You don't need a loader job for each one of these groups
- Any Grouper user could edit the policies if they can READ underlying groups. The expressions are secure (future state)
- The memberships of the ABAC groups are near real time based on an intelligent change log consumer (future state)
- You can have a UI to help build it and give good error messages
- Could visualize the policies. Perhaps could be integrated into existing visualization (future state)
- This solves the issue of composites with any number of factors

1.1. UI to configure

+ Create new group

Quick links

- My groups
- My folders
- My favorites
- My services
- My activity
- Miscellaneous

Browse folders

- Root
 - etc
 - test
 - testGroup0
 - testGroup1
 - testGroup2
 - testGroup3
 - testGroup4
 - testGroup5
 - testGroup6
 - testGroup7
 - testGroup8
 - testGroup9
 - testScript

Home > Root > test > testScript

testScript

Group actions

Show details

MembersPrivilegesMore

Loader settings

Loader actions

This loader group contains members who are the result of a JEXL script

Entity JEXL script

```
{ entity.memberOf('test:testGroup0') && entity.memberOf('test:testGroup1') && entity.memberOf('test:testGroup2') }
```

Enter a JEXL expression that controls the group membership (generally this is users or people). The variable 'entity' is an instance of class: edu.internet2.middleware.grouper.abac.GrouperAbacEntity. You can use entity.memberOf('full group id path') exactly like that to see if user is in a group or not. Here is an example of a three part intersection: S(entity.memberOf('ref:staff') && entity.memberOf('ref:payroll:fullTime') && entity.memberOf('ref:mfaEnrolled')) Here is an example of an example policy: S((entity.memberOf('ref:employee') || entity.memberOf('ref:student') || (entity.memberOf('ref:guests') && entity.memberOf('app:vpn:vpnManualOverrides'))) && entity.memberOf('ref:globalLockout') && entity.memberOf('app:vpn:vpnManualLockout')) That means users who are not in globalLockout and not in vpnManualLockout and in an eligible population which is faculty, students, or guests who are in the manual app override group

Include internal subject sources

No, only include institution defined subject sources (default)
If we should include internal subject sources in the entity script results. e.g. g:gsa (groups), g:isa (e.g. GrouperSystem, GrouperAll), grouperExternal, grouperEntities . Default: No

Quick links

My groups
My folders
My favorites
My services
My activity
Miscellaneous

Browse folders

Root

etc

test

testGroup0

testGroup1

testGroup2

testGroup3

testGroup4

testGroup5

testGroup6

testGroup7

testGroup8

testGroup9

testScript

Group actions

Show details

Members

Privileges

More

Edit loader settings

Loader actions

Loader

Yes, has loader configuration

If this group has loader configuration

Source type

JEXL script

Pull the members from a JEXL script

Entity JEXL script

```

${ entity.memberOf('test:testGroup0') && entity.memberOf('test:testGroup1') &&
entity.memberOf('test:testGroup2') }

```

Enter a JEXL expression that controls the group membership (generally this is users or people).
The variable 'entity' is an instance of class: edu.internet2.middleware.grouper.abac.GrouperAbacEntity
You can use entity.memberOf('full.group.id:path') exactly like that to see if user is in a group or not.
Here is an example of a three part intersection:

```

${ entity.memberOf('ref:staff') && entity.memberOf('ref:payroll:fullTime') && entity.memberOf('ref:mfaEnrolled') }

```

Here is an example of an example policy:

```

${ ( entity.memberOf('ref:employee') || entity.memberOf('ref:student') || (entity.memberOf('ref:guests') &&
entity.memberOf('app:vpn:vpnManualOverrides'))) && !entity.memberOf('ref:globalLockout') &&
!entity.memberOf('app:vpn:vpnManualLockout') }

```

That means users who are not in globalLockout and not in vpnManualLockout and in an eligible population which is
faculty, students, or guests who are in the manual app override group

Include internal subject sources

If we should include internal subject sources in the entity script results. e.g. g:gsa (groups), g:isa (e.g. GrouperSystem, GrouperAll), grouperExternal, grouperEntities . Default: No

Save

Cancel

1.2. Daemon screen

Note in Grouper v2.6.6 you need to wait an hour after changing a script, or run the JEXL script loader full job. In v5+ an incremental job will adjust the members quicker. Note: there is one full daemon and one incremental daemon that handles all of the JEXL script ABAC groups. You do not add this, it is built-in

Home > Miscellaneous > All daemon jobs

All daemon jobs

Daemon actions ▾

Filter for:

Common filters ▾

☐ Show extended results?
 ☐ Show only errors?

Job name	State	Overall status	Last run status	Actions	Schedule
OTHER_JOB_grouperLoaderJexlScriptFullSync	ENABLED	SUCCESS	SUCCESS	Job actions ▾	CRON: 31 19 * * * ? At 31 seconds past the minute, at 19 minutes past the hour

Show:

Showing 1-1 of 1 · [First](#) | [Prev](#) | [Next](#) | [Last](#)

Home > Miscellaneous > All daemon jobs > Daemon logs

Daemon logs

Filter for: OTHER_JOB_grouperLoaderJexlScriptFullSync

Start time between:
End time between:
Last updated between:

☐ Show subjobs
 Status: ☐ Success ☐ Error ☐ Started ☐ Running ☐ Warning ☐ Config error ☐ Subject problems

Number of rows:

5 logs found for job name: OTHER_JOB_grouperLoaderJexlScriptFullSync

Status	Loaded group	Job type	Start time	End time	Millis get data	Millis load data	Total count	Add count	Update count	Delete count	Unresolvable count	Log ID	Last updated	Host	Job message
Success	N/A	overall	2022-02-01 05:57:21.0	2022-02-01 05:57:21.0	136		0	1	0	0	0	8a9c97817eb4ed720176b4ee925b0015	2022-02-01 05:57:21.0	ISC20-0637-WL	jexlScriptGroups: 1, groupsWithInvalidScripts: 0, distinctGroupsInScripts: 3, inserts: 1, deletes: 0, errors: 0

1.3. Scripts

The script can only be written by people who can READ groups in the script and UPDATE the owner group. Since this is actually a JEXL script (not a JEXL expression), so you could have multiple lines, variables, conditionals, etc

In an entity script, the variable 'entity' is an instance of class: edu.internet2.middleware.grouper.abac.GrouperAbacEntity

You can use entity.memberOf('full:group:id:path') exactly like that to see if user is in a group or not.

Expression	Description
<pre> \${ entity.memberOf('ref:staff') && entity.memberOf('ref:payroll:fullTime') && entity.memberOf('ref:mfaEnrolled') } </pre>	Three part intersection. Full time staff in MFA

<pre> \${ (entity.memberOf('ref:employee') entity.memberOf('ref:student') // employees or students (entity.memberOf('ref:guests') && entity.memberOf('app:vpn:vpnManualOverrides'))) // or guests who are in manual allow && !entity.memberOf('ref:globalLockout') && !entity.memberOf('app:vpn:vpnManualLockout') } // and not in either lockout group </pre>	Example policy That means users who are not in globalLockout and not in vpnManualLockout and in an eligible population which is faculty, students, or guests who are in the manual app override group
<pre> \${ entity.memberOf('app:vpn:users') != entity.memberOf('ref:mfaEnrolled') } </pre>	Exclusive OR This is VPN users not in MFA and MFA users not in VPN:

1.4. How it works in v5+

The script is parsed and converted to SQL. The results represent the members of the group. The diffs will be added or removed from the group.

1.5. Analyze policy

To confirm a policy is correct, a long form translation of the policy can be displayed along with group names and group counts

Group actions ▾

Group types: [ref.](#)
Show details ▾

MembersPrivilegesMore ▾

Edit loader settings

Loader actions ▾

Member name or ID:
Enter an entity name or ID, or [search for an entity.](#)

Analyze

The overall JEXL analysis is the first row.

Population count	ABAC script description
43	Has row 'cp_user' with attribute 'cp_active' and not with attribute 'cp_blocked' and with attribute 'cp_known' and with attribute 'cp_org' value 'Perelman School of Medicine' and (not member of group 'penn:ref:member' or member of group 'penn:ref:lockout') and has attribute 'cp_role' value 'desktop-user'
1672	Has row 'cp_user' with attribute 'cp_active'
283	Has row 'cp_user' with attribute 'cp_blocked'
1717	Has row 'cp_user' not with attribute 'cp_blocked'
1449	Has row 'cp_user' with attribute 'cp_active' and not with attribute 'cp_blocked'
1741	Has row 'cp_user' with attribute 'cp_known'
1261	Has row 'cp_user' with attribute 'cp_active' and not with attribute 'cp_blocked' and with attribute 'cp_known'
232	Has row 'cp_user' with attribute 'cp_org' value 'Perelman School of Medicine'
151	Has row 'cp_user' with attribute 'cp_active' and not with attribute 'cp_blocked' and with attribute 'cp_known' and with attribute 'cp_org' value 'Perelman School of Medicine'
1509	Member of group 'penn:ref:member'
816	Not member of group 'penn:ref:member'
127	Member of group 'penn:ref:lockout'
908	(not member of group 'penn:ref:member' or member of group 'penn:ref:lockout')
44	Has row 'cp_user' with attribute 'cp_active' and not with attribute 'cp_blocked' and with attribute 'cp_known' and with attribute 'cp_org' value 'Perelman School of Medicine' and (not member of group 'penn:ref:member' or member of group 'penn:ref:lockout')
1940	Has attribute 'cp_role' value 'desktop-user'

Save

Cancel

1.6. Policy patterns

Your institution can make a GSH template that will help users setup policies

TODO document this

See Also

[Access Management Features Overview](#)