Grouper rules pattern - Remove invalid permissions due to folder

Grouper rules

If an entity falls out of any group in the IT organization groups (meaning not a central IT employee anymore), then remove permissions from a permission definition or remove from roles which have assignments to the permission definition

Assign this to the permission definition of the permission to be removed.

Java example

```
    //add a rule on stem:permission saying if you are out of stem:employee,
    //then remove assignments to permission, or from roles which have the permission
    AttributeAssign attributeAssign = permissionToAssignRule
      .getAttributeDelegate().addAttribute(RuleUtils.ruleAttributeDefName()).getAttributeAssign();


    AttributeValueDelegate attributeValueDelegate = attributeAssign.getAttributeValueDelegate();

    attributeValueDelegate.assignValue(
        RuleUtils.ruleActAsSubjectSourceIdName(), actAs.getSourceId());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleActAsSubjectIdName(), actAs.getId());

    //folder where membership was removed
    attributeValueDelegate.assignValue(
        RuleUtils.ruleCheckOwnerIdName(), mustBeInGroupInFolder.getUuid());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleCheckTypeName(),
        RuleCheckType.membershipRemoveInFolder.name());

    //SUB for all descendants, ONE for just children
    attributeValueDelegate.assignValue(
        RuleUtils.ruleCheckStemScopeName(),
        stemScope.name());

    //if there is no more membership in the folder, and there is a membership in the group
    attributeValueDelegate.assignValue(
        RuleUtils.ruleIfConditionEnumName(),
        RuleIfConditionEnum.thisPermissionDefHasAssignmentAndNotFolder.name());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleThenEnumName(),
        RuleThenEnum.removeMemberFromOwnerPermissionDefAssignments.name());

    //should be valid
    String isValidString = attributeValueDelegate.retrieveValueString(
        RuleUtils.ruleValidName());

    if (!StringUtils.equals("T", isValidString)) {
      throw new RuntimeException(isValidString);
    }
```

GSH shorthand method

```
RuleApi.permissionFolderIntersection(actAsSubject, permissionDef, itEmployeeStem, Stem.Scope.SUB);
```

GSH test case

```
gsh 0% grouperSession = GrouperSession.startRootSession();
edu.internet2.middleware.grouper.GrouperSession: a22fcbc1abb749b6bf3afd5f441896ca,'GrouperSystem','application'

//definition for permission
gsh 1% permissionDef = new AttributeDefSave(grouperSession).assignName("stem:permissionDef").
assignCreateParentStemsIfNotExist(true).assignAttributeDefType(AttributeDefType.perm).save();
edu.internet2.middleware.grouper.attr.AttributeDef: AttributeDef[name=stem:permissionDef,
uuid=a20cf95b75154a2da7b817d19a37cf73]
gsh 2% permissionDef.setAssignToEffMembership(true);
gsh 3%  permissionDef.setAssignToGroup(true);
gsh 4% permissionDef.store();

//two groups in the org chart for the IT department
gsh 5% groupProgrammers = new GroupSave(grouperSession).assignName("stem:orgs:itEmployee:programmers").
assignCreateParentStemsIfNotExist(true).save();
group: name='stem:orgs:itEmployee:programmers' displayName='stem:orgs:itEmployee:programmers'
uuid='e9c49da6801446538372ef6f583b7df2'
gsh 6% groupSysadmins = new GroupSave(grouperSession).assignName("stem:orgs:itEmployee:sysadmins").
assignCreateParentStemsIfNotExist(true).save();
group: name='stem:orgs:itEmployee:sysadmins' displayName='stem:orgs:itEmployee:sysadmins'
uuid='14728c7b48fd4ecc82cbf692ab2aba13'

//folder for IT employee
gsh 7% itEmployee = StemFinder.findByName(grouperSession, "stem:orgs:itEmployee", true);
stem: name='stem:orgs:itEmployee' displayName='stem:orgs:itEmployee' uuid='3d55c81499ce4b059c8e1f2a147c71ae'

//two roles for the application
gsh 8% payrollUser = new GroupSave(grouperSession).assignName("apps:payroll:roles:payrollUser").
assignTypeOfGroup(TypeOfGroup.role).assignCreateParentStemsIfNotExist(true).save();
group: name='apps:payroll:roles:payrollUser' displayName='apps:payroll:roles:payrollUser'
uuid='0e93b9d5802c475f8d98350226679313'
gsh 9% payrollGuest = new GroupSave(grouperSession).assignName("apps:payroll:roles:payrollGuest").
assignTypeOfGroup(TypeOfGroup.role).assignCreateParentStemsIfNotExist(true).save();
group: name='apps:payroll:roles:payrollGuest' displayName='apps:payroll:roles:payrollGuest'
uuid='11efd3897df241e2a51e57742296aa08'

gsh 10% subject0 = SubjectFinder.findByIdAndSource("test.subject.0", "jdbc", true);
subject: id='test.subject.0' type='person' source='jdbc' name='my name is test.subject.0'
gsh 11% subject1 = SubjectFinder.findByIdAndSource("test.subject.1", "jdbc", true);
subject: id='test.subject.1' type='person' source='jdbc' name='my name is test.subject.1'
gsh 12% subject2 = SubjectFinder.findByIdAndSource("test.subject.2", "jdbc", true);
subject: id='test.subject.2' type='person' source='jdbc' name='my name is test.subject.2'

//subject0 is assigned to payrollUser role, and that role has the permission (RBAC)
gsh 13% payrollUser.addMember(subject0, false);
true

//subject1 is a guest, and has the permission directly assigned
gsh 14% payrollGuest.addMember(subject1, false);
true

//this is the permission resource
gsh 15% canLogin = new AttributeDefNameSave(grouperSession, permissionDef).assignName("apps:payroll:permissions:
canLogin").assignCreateParentStemsIfNotExist(true).save();
edu.internet2.middleware.grouper.attr.AttributeDefName: AttributeDefName[name=apps:payroll:permissions:canLogin,
uuid=bc135affbeb84c069cf53a89833c0cca]
gsh 16% payrollUser.getPermissionRoleDelegate().assignRolePermission(canLogin);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.AttributeAssignResult@1dd66fd
gsh 17% payrollGuest.getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, subject1);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.AttributeAssignResult@e5d155

gsh 18% member0 = MemberFinder.findBySubject(grouperSession, subject0, false);
member: id='test.subject.0' type='person' source='jdbc' uuid='d65c59dac1494a84940c45190dd44f3e'
gsh 19% member1 = MemberFinder.findBySubject(grouperSession, subject1, false);
member: id='test.subject.1' type='person' source='jdbc' uuid='94a1f7bbc08f4c0c962b4c19b1dbecbe'
```

```
//subject0 and subject1 both have the permission
gsh 20% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid());
edu.internet2.middleware.grouper.permissions.PermissionEntry: PermissionEntry[roleName=apps:payroll:roles:
payrollUser,attributeDefNameName=apps:payroll:permissions:canLogin,action=assign,sourceId=jdbc,subjectId=test.
subject.0,imm_mem=true,imm_perm=true,mem_depth=0,role_depth=0,action_depth=0,attrDef_depth=0,perm_type=role]
gsh 21% permissions.size()
1
gsh 22% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin
gsh 23% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member1.getUuid());
edu.internet2.middleware.grouper.permissions.PermissionEntry: PermissionEntry[roleName=apps:payroll:roles:
payrollGuest,attributeDefNameName=apps:payroll:permissions:canLogin,action=assign,sourceId=jdbc,subjectId=test.
subject.1,imm_mem=true,imm_perm=true,mem_depth=0,role_depth=-1,action_depth=0,attrDef_depth=0,
perm_type=role_subject]
gsh 24% permissions.size()
1
gsh 25% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin

//configure the rule
gsh 26% RuleApi.permissionFolderIntersection(SubjectFinder.findRootSubject(), permissionDef, itEmployee, Stem.
Scope.SUB);

gsh 27% groupProgrammers.addMember(subject0, false);
gsh 28% groupSysadmins.addMember(subject0, false);
true
gsh 29% groupProgrammers.addMember(subject1, false);
true
gsh 30% groupSysadmins.addMember(subject1, false);
true
gsh 31% groupProgrammers.addMember(subject2, false);
true
gsh 32% groupSysadmins.addMember(subject2, false);
true

//if subject2 is removed, nothing should happen (subject2 didnt have permissions)
gsh 33% groupProgrammers.deleteMember(subject2);
gsh 34% groupSysadmins.deleteMember(subject2);

//remove subject0 from one group, should still have permissions
gsh 35% groupProgrammers.deleteMember(subject0);
gsh 36% GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid()).size();
1

//remove from the other org group, and the permissions should be gone, should not be in the role anymore
gsh 37% groupSysadmins.deleteMember(subject0);
gsh 38% GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid()).size();
0

//subject1 still have permission
gsh 39% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member1.getUuid());
edu.internet2.middleware.grouper.permissions.PermissionEntry: PermissionEntry[roleName=apps:payroll:roles:
payrollGuest,attributeDefNameName=apps:payroll:permissions:canLogin,action=assign,sourceId=jdbc,subjectId=test.
subject.1,imm_mem=true,imm_perm=true,mem_depth=0,role_depth=-1,action_depth=0,attrDef_depth=0,
perm_type=role_subject]
gsh 40% permissions.size()
1
gsh 41% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin

//remove subject1 from one org, should still have permission
gsh 42% groupSysadmins.deleteMember(subject1);
gsh 43% GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member1.getUuid()).size();
1

//remove from other and loses permission
gsh 44% groupProgrammers.deleteMember(subject1);
gsh 45% payrollGuest.hasMember(subject1)
true
gsh 46% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member1.getUuid())
```

```
gsh 47% permissions.size();
0
```

GSH daemon test case

Run the above GSH commands, then continue below:

```
gsh 48% payrollUser.addMember(subject0, false);
true
gsh 49% payrollGuest.getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, subject1);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.AttributeAssignResult@692a88
gsh 50% payrollUser.addMember(subject2, false);
true
gsh 51% subject3 = SubjectFinder.findByIdAndSource("test.subject.3", "jdbc", true);
subject: id='test.subject.3' type='person' source='jdbc' name='my name is test.subject.3'
gsh 52% payrollGuest.addMember(subject3, false);
true
gsh 53% payrollGuest.getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, subject3);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.AttributeAssignResult@1d6e77
gsh 54% groupProgrammers.addMember(subject2, false);
true
gsh 55% groupProgrammers.addMember(subject3, false);
true
gsh 56% status = GrouperLoader.runOnceByJobName(grouperSession, GrouperLoaderType.GROUPER_RULES);
loader ran successfully: Ran rules daemon, changed 0 records
gsh 57% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid());
gsh 58% permissions.size();
0
gsh 59% payrollUser.hasMember(subject0);
false
gsh 60% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member1.getUuid());
gsh 61% permissions.size();
0
gsh 62% payrollGuest.hasMember(subject1);
true
gsh 65% member2 = MemberFinder.findBySubject(grouperSession, subject2, false);
member: id='test.subject.2' type='person' source='jdbc' uuid='2ccf68d9fe4241888822be1a0546c8e5'
gsh 66% member3 = MemberFinder.findBySubject(grouperSession, subject3, false);
member: id='test.subject.3' type='person' source='jdbc' uuid='efcec181c7e34907abafa6ba1fa1143f'
gsh 67% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member2.getUuid());
edu.internet2.middleware.grouper.permissions.PermissionEntry: PermissionEntry[roleName=apps:payroll:roles:
payrollUser,attributeDefNameName=apps:payroll:permissions:canLogin,action=assign,sourceId=jdbc,subjectId=test.
subject.2,imm_mem=true,imm_perm=true,mem_depth=0,role_depth=0,action_depth=0,attrDef_depth=0,perm_type=role]
gsh 68% permissions.size();
1
gsh 69% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin
gsh 70% payrollUser.hasMember(subject2);
true
gsh 71% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member3.getUuid());
edu.internet2.middleware.grouper.permissions.PermissionEntry: PermissionEntry[roleName=apps:payroll:roles:
payrollGuest,attributeDefNameName=apps:payroll:permissions:canLogin,action=assign,sourceId=jdbc,subjectId=test.
subject.3,imm_mem=true,imm_perm=true,mem_depth=0,role_depth=-1,action_depth=0,attrDef_depth=0,
perm_type=role_subject]
gsh 72% permissions.size();
1
gsh 73% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin
gsh 74% payrollGuest.hasMember(subject3);
true
gsh 75%
```

sdaf