Grouper rules pattern - Remove invalid permissions due to group

Grouper rules

If an entity is no longer a member of the employee group, remove them from the role for application X which has certain permissions assigned; also unassign any direct permissions to the user.

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());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleCheckOwnerIdName(), mustBeInGroup.getId());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleCheckTypeName(),
        RuleCheckType.membershipRemove.name());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleIfConditionEnumName(),
        RuleIfConditionEnum.thisPermissionDefHasAssignment.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.permissionGroupIntersection(SubjectFinder.findRootSubject(), permissionDef, groupEmployee);
```

GSH test case

```
gsh 0% grouperSession = GrouperSession.startRootSession();
edu.internet2.middleware.grouper.GrouperSession: 9351993908a847b190da1e428033c579,'GrouperSystem','application'

//make a permission definition, which is assignable to roles or users in the context of a role
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=dde36f66d60841448c217dc009d4eac8]
gsh 2% permissionDef.setAssignToEffMembership(true);
```

```
gsh 3% permissionDef.setAssignToGroup(true);
gsh 4% permissionDef.store();

//employee group is the group that if the user falls out of, remove the permissions assignments
gsh 5% groupEmployee = new GroupSave(grouperSession).assignName("stem:employee").
assignCreateParentStemsIfNotExist(true).save();
group: name='stem:employee' displayName='stem:employee' uuid='7934203d09bf407bbf813f6768759ea9'

//these are roles for the application
gsh 6% 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='663d14fda9aa45d48caaab84a4df11dc'
gsh 7% 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='bcdce97b557e46c59b663bb3c7db7126'

//subject0 is in payrollUser, subject1 is a payroll guest
gsh 10% addMember("apps:payroll:roles:payrollUser", "test.subject.0");
true
gsh 11% addMember("apps:payroll:roles:payrollGuest", "test.subject.1");
true

//canLogin is a permissions resource in the permission definition
gsh 12% 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=83107cf902544375a46376c8a4210051]

//for subject0, this permission is assigned to the role, and subject0 gets it from being a member of the role
gsh 13% payrollUser.getPermissionRoleDelegate().assignRolePermission(canLogin);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.AttributeAssignResult@1d74bb1
gsh 16% subject1 = SubjectFinder.findById("test.subject.1", true);
subject: id='test.subject.1' type='person' source='jdbc' name='my name is test.subject.1'
gsh 17% subject0 = SubjectFinder.findById("test.subject.0", true);
subject: id='test.subject.0' type='person' source='jdbc' name='my name is test.subject.0'

//subject1 is a guest, and gets the permission as directly assigned in the context of role payrollGuest
gsh 18% payrollGuest.getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, subject1);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.AttributeAssignResult@c5dbb
gsh 19% member0 = MemberFinder.findBySubject(grouperSession, subject0, false);
gsh 20% member1 = MemberFinder.findBySubject(grouperSession, subject1, false);
member: id='test.subject.1' type='person' source='jdbc' uuid='bc39785bb650459585043d9b4cffc051'

//subject0 and subject1 each have one permission
gsh 21% 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 22% permissions.size()
1
gsh 23% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin
gsh 24% 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 25% permissions.size()
1
gsh 26%  permissions.iterator().next().getAttributeDefNameName();
apps:payroll:permissions:canLogin

//add the rule
gsh 27% RuleApi.permissionGroupIntersection(SubjectFinder.findRootSubject(), permissionDef, groupEmployee);

//add users to employee group
gsh 29% groupEmployee.addMember(subject0);
```

```
gsh 30% groupEmployee.addMember(subject1);

//remove the first user
gsh 33% groupEmployee.deleteMember(subject0);

//since the permission was due to role, the user should not be in role anymore, or the permissions
gsh 34% payrollUser.hasMember(subject0);
false
gsh 35% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid());
gsh 36% permissions.size();
0

//subject1 still have permissions in the context of the role
gsh 37% 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 38% permissions.size()
1
gsh 39% permissions.iterator().next().getAttributeDefNameName();
apps:payroll:permissions:canLogin

//remove subject1 from employee group
gsh 40% groupEmployee.deleteMember(subject1);

//subject1 is still a guest since guest does not have login ability...
gsh 43% payrollGuest.hasMember(subject1);
true

//however, the direct assignment to user in the context of the role is removed
gsh 44% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member1.getUuid());
gsh 45% permissions.size()
0
gsh 46%
```

GSH daemon test case

Run the above test case, and continue below

```
gsh 48% addMember("apps:payroll:roles:payrollUser", "test.subject.0");
true
gsh 49% payrollGuest.getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, subject1);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.Attribu
teAssignResult@c50443
gsh 50% status = GrouperLoader.runOnceByJobName(grouperSession, GrouperLoaderType.GROUPER_RULES);
loader ran successfully: Ran rules daemon, changed 0 records
gsh 51% payrollUser.hasMember(subject0);
false
gsh 52% payrollGuest.hasMember(subject1);
true
gsh 53% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member1.getUuid());
gsh 54% permissions.size()
0
gsh 55% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid());
gsh 56% permissions.size()
0
gsh 57%
```