Grouper rules pattern - Veto permission if not eligible due to group

Grouper rules

If a user is not an employee, do not allow to have permissions assigned

Add this rule to the permission definition of the permission that is added.

Java example

```
    //add a rule on stem:a saying if not in stem:b, then dont allow add to stem:a
    AttributeAssign attributeAssign = permissionDef
      .getAttributeDelegate().addAttribute(RuleUtils.ruleAttributeDefName()).getAttributeAssign();

    AttributeValueDelegate attributeValueDelegate = attributeAssign.getAttributeValueDelegate();

    attributeValueDelegate.assignValue(
        RuleUtils.ruleActAsSubjectSourceIdName(), "g:isa");
    attributeValueDelegate.assignValue(
        RuleUtils.ruleActAsSubjectIdName(), "GrouperSystem");
    attributeValueDelegate.assignValue(
        RuleUtils.ruleCheckTypeName(), RuleCheckType.permissionAssignToSubject.name());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleIfConditionEnumName(), RuleIfConditionEnum.groupHasNoImmediateEnabledMembership.name());
    attributeValueDelegate.assignValue(
        RuleUtils.ruleIfOwnerNameName(), "stem:employee");
    attributeValueDelegate.assignValue(
        RuleUtils.ruleThenEnumName(), RuleThenEnum.veto.name());

    //key which would be used in UI messages file if applicable
    attributeValueDelegate.assignValue(
        RuleUtils.ruleThenEnumArg0Name(), "rule.entity.must.be.an.employee");

    //error message (if key in UI messages file not there)
    attributeValueDelegate.assignValue(
        RuleUtils.ruleThenEnumArg1Name(), "Entity cannot be assigned these permissions unless they are an
employee");

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

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

GSH shorthand method

```
RuleApi.vetoPermissionIfNotInGroup(SubjectFinder.findRootSubject(), permissionDef, groupEmployee, "rule.entity.
must.be.an.employee", "Entity cannot be assigned these permissions unless they are an employee");
```

GSH test case

```
gsh 0% grouperSession = GrouperSession.startRootSession();
edu.internet2.middleware.grouper.GrouperSession: a448462e34e243c5b28228c9e218c86a,'GrouperSystem','application'

//permission definition
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=e9c7fed59c464274b830d6a2abd8c6e2]
gsh 2% permissionDef.setAssignToEffMembership(true);
gsh 3% permissionDef.setAssignToGroup(true);
gsh 4% permissionDef.store();

//employee group that users must be in to get permissions
gsh 5% groupEmployee = new GroupSave(grouperSession).assignName("stem:employee").
assignCreateParentStemsIfNotExist(true).save();
group: name='stem:employee' displayName='stem:employee' uuid='6edfef8dd57444fba6f74c199230132d'

//role for 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='d2d23e4438a04150801f5e190c36bc5f'
gsh 7% 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 8% payrollUser.addMember(subject0, false);
true

//permission resource
gsh 9% 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=716818177e144bd9a244ede9a01612bd]

//assign the rule
gsh 10% RuleApi.vetoPermissionIfNotInGroup(SubjectFinder.findRootSubject(), permissionDef, groupEmployee, "rule.
entity.must.be.an.employee", "Entity cannot be assigned these permissions unless they are an employee");

//cant get assigned since not an employee
gsh 11% payrollUser.getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, subject0);
// Error: unable to evaluate command: Sourced file: inline evaluation of: ``payrollUser.
getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, su . . . '' : Method Invocation
assignSubjectRolePermission
// See error log for full stacktrace
// caused by: edu.internet2.middleware.grouper.rules.RuleVeto:
// rule.entity.must.be.an.employee: Entity cannot be assigned these permissions unless they are an employee

//does not have permission yet
gsh 12% member0 = MemberFinder.findBySubject(grouperSession, subject0, false);
member: id='test.subject.0' type='person' source='jdbc' uuid='24f532b13e1a4e4a8d16204b65dfd0ee'
gsh 13% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid());
gsh 14% permissions.size()
0

//add to employee group, assign permissions again
gsh 15% groupEmployee.addMember(subject0);
gsh 16% payrollUser.getPermissionRoleDelegate().assignSubjectRolePermission(canLogin, subject0);
edu.internet2.middleware.grouper.attr.assign.AttributeAssignResult: edu.internet2.middleware.grouper.attr.
assign.AttributeAssignResult@c0cd7d

//see that they have the permission now
gsh 17% 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=-1,action_depth=0,attrDef_depth=0,
perm_type=role_subject]
gsh 18% permissions.size()
1
gsh 19% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin
gsh 20%
```

GSH daemon test case

Run the above commands, then continue below

```
gsh 20% groupEmployee.deleteMember(subject0);
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=-1,action_depth=0,attrDef_depth=0,
perm_type=role_subject]
gsh 22% permissions.iterator().next().getAttributeDefNameName()
apps:payroll:permissions:canLogin
gsh 23% status = GrouperLoader.runOnceByJobName(grouperSession, GrouperLoaderType.GROUPER_RULES);
loader ran successfully: Ran rules daemon, changed 0 records
gsh 24% permissions = GrouperDAOFactory.getFactory().getPermissionEntry().findByMemberId(member0.getUuid());
gsh 25% permissions.size()
0
gsh 26% payrollUser.hasMember(subject0);
true
```

sdf