Grouper Bug GRP-928 - Grouper UI allows unauthorized users to view the privileges of other subjects

Grouper has a bug which allows unauthorized users to see the group and folder privileges (e.g. ADMIN/UPDATE/READ/etc on a group or STEM/CREATE on a folder) for any subject. So for example, if a user has UPDATE privileges on a group, typically only admins of the group would be able to see that. But with this bug, anybody that can access the Grouper UI can see that the user has UPDATE on the group. This bug does not expose the memberships of subjects.

This bug affects v1.4, v1.5, v1.6, v2.0, and v2.1 (up to 2.1.4). The bug is fixed in each of the branches and included in 2.1.5. The bug probably exists in v1.3 and before, but has not been confirmed or fixed in those old releases. This patch can be applied to v1.4.0+.

Reproduce the problem

```
subjectIdAuthenticating = "test.subject.0";  // you will authenticate to the Grouper UI as this subject
subjectIdCheckPrivileges = "test.subject.1"; // you will check privileges for this subject

grouperSession = GrouperSession.startRootSession();

subjectAuthenticating = SubjectFinder.findByIdOrIdentifier(subjectIdAuthenticating, true);
subjectCheckingPrivileges = SubjectFinder.findByIdOrIdentifier(subjectIdCheckPrivileges, true);

groupNameCanViewPrivs = "test:testUIPrivilegeIssue:groupCanViewPrivs";
groupNameCannotViewPrivs = "test:testUIPrivilegeIssue:groupCannotViewPrivs";

folderNameCanViewPrivs = "test:testUIPrivilegeIssue:folderCanViewPrivs";
folderNameCannotViewPrivs = "test:testUIPrivilegeIssue:folderCannotViewPrivs";

groupCanViewPrivs = new GroupSave(grouperSession).assignName(groupNameCanViewPrivs).
assignCreateParentStemsIfNotExist(true).save();
groupCannotViewPrivs = new GroupSave(grouperSession).assignName(groupNameCannotViewPrivs).
assignCreateParentStemsIfNotExist(true).save();

folderCanViewPrivs = new StemSave(grouperSession).assignName(folderNameCanViewPrivs).save();
folderCannotViewPrivs = new StemSave(grouperSession).assignName(folderNameCannotViewPrivs).save();

groupCanViewPrivs.grantPriv(subjectAuthenticating, AccessPrivilege.ADMIN);
groupCanViewPrivs.grantPriv(subjectCheckingPrivileges, AccessPrivilege.UPDATE);
groupCannotViewPrivs.grantPriv(subjectCheckingPrivileges, AccessPrivilege.UPDATE);

folderCanViewPrivs.grantPriv(subjectAuthenticating, NamingPrivilege.STEM);
folderCanViewPrivs.grantPriv(subjectCheckingPrivileges, NamingPrivilege.CREATE);
folderCannotViewPrivs.grantPriv(subjectCheckingPrivileges, NamingPrivilege.CREATE);
```

Now reproduce the issue using the Grouper UI.

1. Login to the Grouper UI using the subject test.subject.0.
2. Click the "Search" link on the left.
3. Search for test.subject.1 and select the subject.
4. Click on the radio box next to "Show all GROUPS where this entity has the privilege", select "UPDATE" in the drop-down list, and click on "Change display".
5. You should see results for test:testUIPrivilegeIssue:groupCanViewPrivs and test:testUIPrivilegeIssue:groupCannotViewPrivs. The bug is that you shouldn't be able to see the latter.
6. Click on the radio box next to "Show all FOLDERS where this entity has the privilege", select "Create Group" from the drop down list, and click "Change display".
7. You should see results for test:testUIPrivilegeIssue:folderCanViewPrivs and test:testUIPrivilegeIssue:folderCannotViewPrivs. Again, the bug is that you shouldn't be able to see the latter.

Patch the Grouper UI

You need to edit PopulateSubjectSummaryAction.java. Note, these instructions are tomcat-specific. Note, anytime you see /PATH_TO_GROUPER_UI_TOMCAT/ substitute that for the path to that tomcat, and anytime you see grouperUiAppName, substitute that for the app name you use for grouper UI, which is generally grouper.

Unzip the grouper-ui.jar file in a location where you can compile a classfile. These instructions are for linux, though could be used in windows if you change the path separators (colon to semicolon)

```
$ mkdir /tmp/grouperUiPrivPatch
$ cd /tmp/grouperUiPrivPatch/
$ mkdir temp
$ unzip /PATH_TO_GROUPER_UI_TOMCAT/webapps/grouperUiAppName/WEB-INF/lib/grouper-ui.jar -d temp
$ mkdir -p edu/internet2/middleware/grouper/ui/actions/
$ cp temp/edu/internet2/middleware/grouper/ui/actions/PopulateSubjectSummaryAction.java edu/internet2/middleware
/grouper/ui/actions/
$ rm -rf temp
$ vi edu/internet2/middleware/grouper/ui/actions/PopulateSubjectSummaryAction.java
```

There are two method calls to GrouperHelper.getGroupsOrStemsWhereMemberHasPriv(). We're going to filter the results to remove groups/folders that the person shouldn't be able to see.

First, replace the following:

```
subjectScopes = GrouperHelper.getGroupsOrStemsWhereMemberHasPriv(member,accessPriv);
```

with this:

```
subjectScopes = GrouperHelper.getGroupsOrStemsWhereMemberHasPriv(member,accessPriv);

// filter out groups where the subject can't see privs
removeObjectsNotAllowedToSeePrivs(subjectScopes);
```

Second, replace the following:

```
subjectScopes = GrouperHelper.getGroupsOrStemsWhereMemberHasPriv(member,namingPriv);
```

with this:

```
subjectScopes = GrouperHelper.getGroupsOrStemsWhereMemberHasPriv(member,namingPriv);

// filter out stems where the subject can't see privs
removeObjectsNotAllowedToSeePrivs(subjectScopes);
```

And third, add the following method at the end:

```
  /**
   * remove objects not allowed to see privileges on
   * @param groupsAndStems
   */
  public static void removeObjectsNotAllowedToSeePrivs(Set<?> groupsAndStems) {

    if (groupsAndStems == null) {
      return;
    }

    //subject who is making the query
    final Subject grouperSessionSubject = GrouperSession.staticGrouperSession().getSubject();

    java.util.Iterator<?> iterator = groupsAndStems.iterator();

    while (iterator.hasNext()) {
      Object groupOrStem = iterator.next();

      if (groupOrStem instanceof edu.internet2.middleware.grouper.Group) {

        edu.internet2.middleware.grouper.Group group = (edu.internet2.middleware.grouper.Group)groupOrStem;
        if (!group.hasAdmin(grouperSessionSubject)) {
          iterator.remove();
        }
      } else if (groupOrStem instanceof edu.internet2.middleware.grouper.Stem) {

        edu.internet2.middleware.grouper.Stem stem = (edu.internet2.middleware.grouper.Stem)groupOrStem;
        if (!stem.hasStem(grouperSessionSubject)) {
          iterator.remove();
        }

      } else {
        //this should never happen
        throw new RuntimeException("Not expecting object of type: " + groupOrStem.getClass() + ", " +
groupOrStem);
      }
    }
  }
```

Compile the patched file, note, if you are using Java previous to 1.6, then you need to list out the jar files, instead of the asterisk. Install, and bounce tomcat

```
$ javac -classpath "/PATH_TO_GROUPER_UI_TOMCAT/webapps/grouperUiAppName/WEB-INF/lib/*:/PATH_TO_GROUPER_UI_TOMCAT
/lib/*" -sourcepath . edu/internet2/middleware/grouper/ui/actions/PopulateSubjectSummaryAction.java
$
$ find . -name PopulateSubjectSummaryAction*
./edu/internet2/middleware/grouper/ui/actions/PopulateSubjectSummaryAction.class
./edu/internet2/middleware/grouper/ui/actions/PopulateSubjectSummaryAction.java
$
$ cp -R edu /PATH_TO_GROUPER_UI_TOMAT/webapps/grouperUiAppName/WEB-INF/classes/
$ find /PATH_TO_GROUPER_UI_TOMAT/webapps/grouperUiAppName/WEB-INF/classes/edu -name
PopulateSubjectSummaryAction*
/PATH_TO_GROUPER_UI_TOMAT/webapps/grouperUiAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ui/actions
/PopulateSubjectSummaryAction.class
/PATH_TO_GROUPER_UI_TOMAT/webapps/grouperUiAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ui/actions
/PopulateSubjectSummaryAction.java
```

Note, if you run on a cluster, you should zip up the files in the patch, and copy them to all servers that need them. Make sure the path is grouperUiAppName/WEB-INF/classes/edu/internet2... Also, you should probably copy the source file and the classfiles in case there are further tweaks.

Bounce tomcat.

Test that the patch worked.

1. Login to the Grouper UI using the subject test.subject.0.
2. Click the "Search" link on the left.
3. Search for test.subject.1 and select the subject.

4. Click on the radio box next to "Show all GROUPS where this entity has the privilege", select "UPDATE" in the drop-down list, and click on "Change display".
5. You should see results for test:testUIPrivilegeIssue:groupCanViewPrivs but **NOT** test:testUIPrivilegeIssue:groupCannotViewPrivs.
6. Click on the radio box next to "Show all FOLDERS where this entity has the privilege", select "Create Group" from the drop down list, and click "Change display".
7. You should see results for test:testUIPrivilegeIssue:folderCanViewPrivs but **NOT** test:testUIPrivilegeIssue:folderCannotViewPrivs.