Grouper Bug GRP-923 WS getGrouperPrivilegesLite can return more data than the user should be able to see

On July 24, 2013, it was reported that Grouper WS getGrouperPrivilegesLite() was returning more data than the user should see. Here is what the problem is:

1. Any valid WS user can call getGrouperPrivilegesLite WS operation specifying a subject and a group or stem, and if that subject is an admin, then the admin's privileges will be returned (e.g. if the admin can READ the group, ADMIN the group, etc)
2. Any valid WS user can call getGrouperPrivilegesLite WS operation with a subject and no group or stem, and the privileges will be returned on objects for which the subject is an admin
3. Any valid WS user can call getGrouperPrivilegesLite WS operation with a stem and all privileges will be returned

Note that if you have the grouper-ws.properties config option: ws.client.user.group.name, then the exposure is reduced since only valid WS users are connecting to the WS. All version of Grouper WS are affected by this problem: v1_4, v1_5, v1_6, v2_0, v2_1. This is fixed in all of the respective branches. If your release is older than July 24, 2013, then your version is affected.

Reproduce the problem

You need a group with an admin, and a web service user who is not an admin on the group (and not a wheel group member or sys admin). You might already have an example of this in Grouper, though if you dont you can customize this GSH script (change the strings at the top, with valid users and group) and run it. Note, on older versions, this script might not work, you can change the script or do this in the UI.

```
groupName = "test:testPrivilegeIssue";
subjectIdWithAdminPriv = "test.subject.0";
subjectIdWithUpdatePriv = "test.subject.1";
subjectIdNoPrivs = "test.subject.2";
subjectIdMember = "test.subject.3";
grouperSession = GrouperSession.startRootSession();
group = new GroupSave(grouperSession).assignName(groupName).assignCreateParentStemsIfNotExist(true).save();
subjectWithAdminPriv = SubjectFinder.findByIdOrIdentifier(subjectIdWithAdminPriv, true);
subjectWithUpdatePriv = SubjectFinder.findByIdOrIdentifier(subjectIdWithUpdatePriv, true);
subjectNoPrivs = SubjectFinder.findByIdOrIdentifier(subjectIdNoPrivs, true);
subjectMember = SubjectFinder.findByIdOrIdentifier(subjectIdMember, true);
group.grantPriv(subjectWithAdminPriv, AccessPrivilege.ADMIN);
group.grantPriv(subjectWithUpdatePriv, AccessPrivilege.UPDATE);
group.addMember(subjectMember);
```

This shows the problem, fast/medley.isc-seo.upenn.edu has no admin privileges on the group, mchyzer does, but the non-privileged user can see privileges on the group

```
[mchyzer@flash pennGroupsClient-test-2.0.0]$ grep webService.kerberosPrin grouper.client.properties
grouperClient.webService.kerberosPrincipal = fast/medley.isc-seo.upenn.edu
[mchyzer@flash pennGroupsClient-test-2.0.0]$ java -jar grouperClient.jar --operation=getGrouperPrivilegesLiteWs
--groupName=test:testGroup --subjectIdentifier=mchyzer
Index 0: success: T: code: SUCCESS: group: test:testGroup: subject: 10021368: access: admin
Index 1: success: T: code: SUCCESS: group: test:testGroup: subject: 10021368: access: read
Index 2: success: T: code: SUCCESS: group: test:testGroup: subject: 10021368: access: update
Index 3: success: T: code: SUCCESS: group: test:testGroup: subject: 10021368: access: view
[mchyzer@flash pennGroupsClient-test-2.0.0]$ java -jar grouperClient.jar --operation=getGrouperPrivilegesLiteWs
--groupName=test:testGroup --subjectId=fast/medley.isc-seo.upenn.edu
Index 0: success: T: code: SUCCESS: group: test:testGroup: subject: fast/medley.isc-seo.upenn.edu: access: read
Index 1: success: T: code: SUCCESS: group: test:testGroup: subject: fast/medley.isc-seo.upenn.edu: access: view
[mchyzer@flash pennGroupsClient-test-2.0.0]$
```

Patch the server

You need to edit GrouperServiceLogic.java. Note, these instructions are tomcat-specific. Note, anytime you see /PATH_TO_GROUPER_WS_TOMCAT/ substitute that for the path to that tomcat, and anytime you see grouperWsAppName, substitute that for the app name you use for grouper WS, which is generally grouper-ws or grouperWs.

Unzip the grouper-ws.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)

```
[appadmin@theprince lib]$ mkdir /tmp/grouperWsPatch
[appadmin@theprince lib]$ cd /tmp/grouperWsPatch
[appadmin@theprince grouperWsPatch]$ mkdir temp
[appadmin@theprince grouperWsPatch]$ cd temp
[appadmin@theprince temp]$ locate grouper-ws.jar
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/lib/grouper-ws.jar
[appadmin@theprince temp]$ unzip /PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/lib/grouper-ws.jar
[appadmin@theprince temp]$ cd ..
[appadmin@theprince grouperWsPatch]$ mkdir -p edu/internet2/middleware/grouper/ws
[appadmin@theprince grouperWsPatch]$ cp temp/edu/internet2/middleware/grouper/ws/GrouperServiceLogic.java edu
/internet2/middleware/grouper/ws
[appadmin@theprince grouperWsPatch]$ rm -rf temp/
[appadmin@theprince grouperWsPatch]$ emacs edu/internet2/middleware/grouper/ws/GrouperServiceLogic.java
```

Add this near the top of the file:

```
import java.util.HashMap;
import edu.internet2.middleware.grouper.GroupFinder;
import edu.internet2.middleware.grouper.exception.GrouperSessionException;
import edu.internet2.middleware.grouper.misc.GrouperSessionHandler;
```

Add one line to the method: getGrouperPrivilegesLite

FROM:

```
        //see if we need to remove, if specifying privs, and this doesnt match
        Iterator<? extends GrouperPrivilege> iterator = privileges.iterator();
        while (iterator.hasNext()) {
          GrouperPrivilege current = iterator.next();
          if (privilegeName != null && !StringUtils.equals(privilegeName.getName(), current.getName())){
            iterator.remove();
          }
        }
        WsGrouperPrivilegeResult[] privilegeResults = new WsGrouperPrivilegeResult[privileges.size()];
```

TO:

```
        //see if we need to remove, if specifying privs, and this doesnt match
        Iterator<? extends GrouperPrivilege> iterator = privileges.iterator();
        while (iterator.hasNext()) {
          GrouperPrivilege current = iterator.next();
          if (privilegeName != null && !StringUtils.equals(privilegeName.getName(), current.getName())){
            iterator.remove();
          }
        }

        //ADD THIS LINE: GRP-923 WS getGrouperPrivilegesLite can return more data should
        removePrivsNotAllowedToSee(privileges);

        WsGrouperPrivilegeResult[] privilegeResults = new WsGrouperPrivilegeResult[privileges.size()];
```

Add this method below that method:

```
  /**
   * remove privileges not allowed to see
```

```
   * @param privileges
   */
  public static void removePrivsNotAllowedToSee(TreeSet<GrouperPrivilege> privileges) {

    int originalNumberOfPrivileges = GrouperUtil.length(privileges);

    if (privileges != null) {

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

      //if this change breaks an app, and you need a quick fix, you can whitelist users
      final String groupNameOfUsersWhoCanCheckAllPrivileges = GrouperWsConfig.getPropertyString("ws.
groupNameOfUsersWhoCanCheckAllPrivileges");

      //if there is a whitelist to preserve old broken behavior
      if (!StringUtils.isBlank(groupNameOfUsersWhoCanCheckAllPrivileges)) {

        //do this as root since the user who is allowed might not be able to read the whitelist group...
        boolean done = (Boolean)GrouperSession.callbackGrouperSession(GrouperSession.staticGrouperSession().
internal_getRootSession(), new GrouperSessionHandler() {

          public Object callback(GrouperSession grouperSession1) throws GrouperSessionException {

            Group groupOfUsersWhoCanCheckAllPrivileges = GroupFinder.findByName(grouperSession1,
groupNameOfUsersWhoCanCheckAllPrivileges, false);

            if (groupOfUsersWhoCanCheckAllPrivileges != null) {

              //if the subject in the grouper session is in the whitelist group, then allow the query without
filtering privileges
              if (groupOfUsersWhoCanCheckAllPrivileges.hasMember(grouperSessionSubject)) {
                return true;
              }

            } else {

              //it is misconfigured, just keep going, but filter privileges based on calling user
              LOG.error("Why is ws.groupNameOfUsersWhoCanCheckAllPrivileges: " +
groupNameOfUsersWhoCanCheckAllPrivileges + ", not found????");
            }
            return false;
          }
        });

        //this means the calling user is in the whitelist for the old bad logic...
        if (done) {
          return;
        }
      }

      //map of group name to if the user is allowed to see privileges
      Map<String, Boolean> groupPrivilegeCache = new HashMap<String, Boolean>();

      //map of stem name to if the user is allowed to see privileges
      Map<String, Boolean> stemPrivilegeCache = new HashMap<String, Boolean>();

      Iterator<GrouperPrivilege> iterator = privileges.iterator();

      while (iterator.hasNext()) {
        GrouperPrivilege grouperPrivilege = iterator.next();

        GrouperAPI grouperApi = grouperPrivilege.getGrouperApi();
        if (grouperApi instanceof Group) {

          Group group = (Group)grouperApi;
          String groupName = group.getName();

          //check the cache
          Boolean allowed = groupPrivilegeCache.get(groupName);
          if (allowed == null) {
```

```
            //not in cache
            //see if allowed
            allowed = group.hasAdmin(grouperSessionSubject);

            //add back to cache
            groupPrivilegeCache.put(group.getName(), allowed);

          }

          if (!allowed) {
            iterator.remove();
          }

        } else if (grouperApi instanceof Stem) {

          Stem stem = (Stem)grouperApi;
          String stemName = stem.getName();

          //check the cache
          Boolean allowed = stemPrivilegeCache.get(stemName);
          if (allowed == null) {
            //not in cache
            //see if allowed
            allowed = stem.hasStem(grouperSessionSubject);

            //add back to cache
            stemPrivilegeCache.put(stem.getName(), allowed);

          }

          if (!allowed) {
            iterator.remove();
          }

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

      }
    }

    if (LOG.isDebugEnabled()) {
      LOG.debug("removePrivsNotAllowedToSee() from " + originalNumberOfPrivileges + " to " + GrouperUtil.length
(privileges) + " privileges");
    }

  }
```

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

```
[appadmin@theprince grouperWsPatch]$ javac -classpath "/PATH_TO_GROUPER_WS_TOMAT/webapps/grouperWsAppName/WEB-
INF/lib/*" -sourcepath . edu/internet2/middleware/grouper/ws/GrouperServiceLogic.java
[appadmin@theprince grouperWsPatch]$ find
.
./edu
./edu/internet2
./edu/internet2/middleware
./edu/internet2/middleware/grouper
./edu/internet2/middleware/grouper/ws
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$4$1.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$10.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$5.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$14.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$9.class
```

```
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$7.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$1$1.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$13.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$10$1.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$8.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$3.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$12$1.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$10$2.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$1.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$6.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$4.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$12.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$11.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic$2.class
./edu/internet2/middleware/grouper/ws/GrouperServiceLogic.java
[appadmin@theprince grouperWsPatch]$ cp -R edu /PATH_TO_GROUPER_WS_TOMAT/webapps/grouperWsAppName/WEB-INF
/classes/
[appadmin@theprince grouperWsPatch]$ find /PATH_TO_GROUPER_WS_TOMAT/webapps/grouperWsAppName/WEB-INF/classes/edu
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$4$1.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$10.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$5.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$14.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$9.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$7.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$1$1.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$13.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$10$1.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$8.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$3.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$12$1.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$10$2.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$1.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$6.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$4.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$12.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$11.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic$2.class
/PATH_TO_GROUPER_WS_TOMCAT/webapps/grouperWsAppName/WEB-INF/classes/edu/internet2/middleware/grouper/ws
/GrouperServiceLogic.java
[appadmin@theprince grouperWsPatch]$
```

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 grouperWsAppName/WEB-INF/classes/edu/insternet2... Also, you should probably copy the source file and the clsasfiles in case there are further tweaks.

Bounce tomcat.

Test that it works by seeing the privs not delivered

```
[mchyzer@flash pennGroupsClient-test-2.0.0]$ java -jar grouperClient.jar --operation=getGrouperPrivilegesLiteWs
--groupName=test:testGroup --subjectIdentifier=mchyzer
[mchyzer@flash pennGroupsClient-test-2.0.0]$
```

If you change the grouper.client.properties to a user who has admin, you should see stuff

```
[mchyzer@flash pennGroupsClient-test-2.0.0]$ java -jar grouperClient.jar --operation=getGrouperPrivilegesLiteWs
--groupName=test:testGroup --subjectId=fast/medley.isc-seo.upenn.edu
Index 0: success: T: code: SUCCESS: group: test:testGroup: subject: fast/medley.isc-seo.upenn.edu: access: read
Index 1: success: T: code: SUCCESS: group: test:testGroup: subject: fast/medley.isc-seo.upenn.edu: access: view
```

If this new fixed behavior breaks existing clients who depend on the broken behavior

Ideally you would grant whatever privileges these clients need so that they can make legitimate WS calls.

If it is not feasible to do that, a workaround is to grant those client access to the old behavior (hopefully temporarily until they can fix their code). You can do this by configuring a group in the grouper-ws.properties:

```
ws.groupNameOfUsersWhoCanCheckAllPrivileges = some:group:name
```

Then add the authenticated WS user to that group, and they will get the old behavior.