Selective Group Exclusion When Provisioning to LDAP

NYU's use case involving composite groups recently came about due to piloting a project. The behavior that was seen was that if someone is a member of one of these composite groups then such a member would also obtain additional memberships that end in "systemOfRecords", "includes", "excludes", "includes and systemOfRecords" where these values are not going to be consumed by any application, just the resulting group. When this pilot makes its way to production, at the NYU scale, there will be a significant use of such composite groups involving large number of users and it has been determined that we need to add selective group exclusion when provisioning to LDAP. Below is the code used to implement such functionality:

```
package edu.nyu;

import edu.internet2.middleware.grouper.Group;
import edu.internet2.middleware.grouper.GroupType;
import edu.internet2.middleware.grouper.GroupTypeFinder;
import edu.internet2.middleware.grouper.GrouperSession;
import edu.internet2.middleware.grouper.cfg.GrouperConfig;
import edu.internet2.middleware.grouper.exception.GrouperSessionException;
import edu.internet2.middleware.grouper.hooks.GroupHooks;
import edu.internet2.middleware.grouper.hooks.beans.HooksContext;
import edu.internet2.middleware.grouper.hooks.beans.HooksGroupBean;
import edu.internet2.middleware.grouper.misc.GrouperSessionHandler;

/**
 * hook to prevent ldap provisioning by setting an attribute on the group.
*/

public class LDAPProvisioningHook extends GroupHooks {

/**
* @see edu.internet2.middleware.grouper.hooks.GroupHooks#groupPostInsert(edu.internet2.middleware.grouper.
hooks.beans.HooksContext, edu.internet2.middleware.grouper.hooks.beans.HooksGroupBean)
*/

  @SuppressWarnings("deprecation")

  @Override

  public void groupPostInsert(HooksContext hooksContext, HooksGroupBean postInsertBean) {

    final Group group = postInsertBean.getGroup();

    String name = group.getName();

    //System.out.println(name);

    boolean excludeMatches = false;

    int count = 0;

    while (true) {

      String property = "LDAPProvisioningHook.exclude.regex." + count;

      String regex = GrouperConfig.retrieveConfig().propertyValueString(property);

      if (regex == null) {

        break;

      }

      if (name.matches(regex)) {

        excludeMatches = true;

        break;

      }

      count++;

    }


    if (excludeMatches) {

      //System.out.println("Should be excluding: " + name);
```

```
      GrouperSession.callbackGrouperSession(GrouperSession.staticGrouperSession().internal_getRootSession(),
 new GrouperSessionHandler() {

        @Override

        public Object callback(GrouperSession grouperSession) throws GrouperSessionException {

          GroupType groupType = GroupTypeFinder.find("LDAPProvisioning", false);

          if (groupType == null) {

            groupType = GroupType.createType(grouperSession, "LDAPProvisioning");

            groupType.addAttribute(grouperSession, "LDAPProvisioningExclude");

          }

          group.addType(groupType);

          group.setAttribute("LDAPProvisioningExclude", "true");

          return null;

        }

      });

    }

  }

}
```

1. Use the code above and run maven to produce a jar.
2. Copy the jar to all locations (UI/WS/GSH)
3. Edit the grouper.properties in GSH and add the following

```
hooks.group.class=edu.nyu.LDAPProvisioningHook

LDAPProvisioningHook.exclude.regex.0=.*_excludes$
LDAPProvisioningHook.exclude.regex.1=.*_includes$
LDAPProvisioningHook.exclude.regex.2=.*_systemOfRecord$
LDAPProvisioningHook.exclude.regex.3=.*_systemOfRecordAndIncludes$
LDAPProvisioningHook.exclude.regex.4=^test:group.*$
```

4. Make sure that this config update is also added UI/WS grouper.properties

5. Make the PSP changes for the LDAP exclusion. In psp-resolver.xml, in each section that returns groups, you basically want to subtract the following:

```
<grouper:Filter xsi:type="grouper:GroupExactAttribute" name="LDAPProvisioningExclude" value="true" />
```

There are a few of these. So for example, this:

```
   <grouper:Filter xsi:type="grouper:MINUS">
      <!-- The GroupInStem filter matches groups which are children of the given stem. -->
      <grouper:Filter
        xsi:type="grouper:GroupInStem"
        name="${edu.internet2.middleware.psp.baseStem}"
        scope="SUB" />
      <grouper:Filter
        xsi:type="grouper:GroupInStem"
        name="etc"
        scope="SUB" />
    </grouper:Filter>
```

Would become the following:

```
<grouper:Filter xsi:type="grouper:MINUS">
    <grouper:Filter xsi:type="grouper:MINUS">
      <!-- The GroupInStem filter matches groups which are children of the given stem. -->
      <grouper:Filter
        xsi:type="grouper:GroupInStem"
        name="${edu.internet2.middleware.psp.baseStem}"
        scope="SUB" />
      <grouper:Filter
        xsi:type="grouper:GroupInStem"
        name="etc"
        scope="SUB" />
    </grouper:Filter>
    <grouper:Filter xsi:type="grouper:GroupExactAttribute" name="LDAPProvisioningExclude" value="true" />
 </grouper:Filter>
```