Using WebLogic LDAP with Hudson

Hudson Role-based Authorization Strategy
This plugin enables authorization using a role-based strategy. There was an issue with the plugin accessing roles that were created in WebLogic LDAP.

WebLogic Group and User
The group, mygroup, and the user, myuser, were created using the WebLogic console using the DefaultAuthenticator based on the Embedded LDAP server.

Checking the LDAP group query
The following LDAP query was being sent to try to locate the roles a user was in when they logged in:

[31/Mar/2012 11:07:07 EDT] conn=35 op=35 SRCH
base="ou=groups,ou=myrealm,dc=base_domain" scope=2
filter="(|(member=uid=myuser,ou=people,ou=myrealm,dc=base_domain)(uniqueMember=uid=myuser,ou=people,ou=myrealm,dc=base_domain)(memberUid=myuser))"

The search was looking for the ID myuser, and it was using three different versions of the filter (member, uniqueMember, and memberUid) to try to find the groups the user was in. The WebLogic LDAP stores the group relationship using the more modern LDAP attribute memberURL. This meant the plugin could never find the user in the group. WebLogic supports the older method of specifying individual entries for each user within the group using the uniqueMember attribute, which is one of the ways the searching was being done.
Using WebLogic LDAP with Hudson

Using JXplorer you can see that when the group is initially created in the WebLogic console the group entry is created with the memberURL attribute, which uses a dynamic search to find members of the group.

The group entry was modified to add the uniqueMember attribute containing the DN for the myuser entry.
Role Plugin Configuration
To setup the role plugin you go to Manage roles. In the setup below a global role was setup with the privilege to create jobs. A project role was setup for the role developer to provide full privileges to configure and run jobs that begin with the prefix DEVJOB. The screenshot below shows the role configuration.

![Manage and Assign Roles](image)

Once the attribute entry uniqueMember was added to LDAP the Role plugin was able to find the groups. You need to prefix the group name with the word ROLE_, and you must put the group in all caps. LDAP does not care about the caps, but the plugin does. Now the plugin shows the entries for global groups and project groups with the correct group icon on the assign roles screen. In this setup mygroup, which is a group of developers, was assigned the global role to create jobs and the project role to configure and run jobs based on the pattern set in manage roles.

![Assign Roles](image)
Using WebLogic LDAP with Hudson

**Hudson Login**
When the user logs in as myuser, using LDAP credentials and the LDAP group setup, the user will have the New Job privilege and will only see the jobs that match the pattern DEVJOB.*:
Using WebLogic LDAP with Hudson

WebLogic LDAP connections
The network channel feature can be used to separate the traffic for the LDAP or LDAPS queries from the normal admin traffic. This will allow the admin traffic to use HTTPS on the standard port, which will be accessed via the public network. The LDAP or LDAPS traffic can be setup to listen on the internal network. Below is a sample setup of the WebLogic channel configuration:

<table>
<thead>
<tr>
<th>Name</th>
<th>Protocol</th>
<th>Enabled</th>
<th>Listen Address</th>
<th>Listen Port</th>
<th>Public Address</th>
<th>Public Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel-1</td>
<td>ldap</td>
<td>true</td>
<td>localhost</td>
<td>7309</td>
<td>localhost</td>
<td>7309</td>
</tr>
<tr>
<td>Channel-2</td>
<td>ldap</td>
<td>true</td>
<td>192.168.1.10</td>
<td>7309</td>
<td>192.168.1.10</td>
<td>7309</td>
</tr>
<tr>
<td>Channel-3</td>
<td>ldaps</td>
<td>true</td>
<td>192.168.1.10</td>
<td>7506</td>
<td>192.168.1.10</td>
<td>7506</td>
</tr>
</tbody>
</table>