IBM – Technote

IBM Security Identity Governance and Intelligence

LDAP Adapter Configuration for Groups Names

Table of Contents

Issue

During a delta sync request from IGI, the Identity LDAP adapter for IGI returns all LDAP groups defined under the base point specified on the connector's configuration. However, the default behavior of the adapter is to use the RDN attribute value as the group name, which is presented as a permission on IGI. This behavior presents an issue for customers who create groups with the same RDN name in a different LDAP container under the same base point managed by the adapter. These groups will be ignored and only one group will be represented as a permission on IGI.

Solution

The LDAP adapter profile can be configured to use the full LDAP group DN as the group to resolve the issue described above. With this configuration, all groups under the base point managed by the adapter, will be represented as permissions in IGI including groups with same RDN value in different containers. Note that the IGI permission name will be the full group DN on LDAP.

Configuration Steps

The configuration involves modifying the LDAP adapter profile jar file prior to importing the profile.

- ### Prerequisite

This configuration is supported on

- IGI versions 5.2.5.1 or higher
- LDAP Adapter version 7.1.23 or higher

- ### Extract the Adapter profile content

Refer to the Identity LDAP Adapter knowledge center for instructions on how to extract the profile content

https://www.ibm.com/support/knowledgecenter/SSIGMP_1.0.0/isim70/ldap/install_config/t_aexpand.htm

- Modify the ***service.def*** file

1. *Locate the **ServiceGroups** definition within the file*

```
<ServiceGroups>
        <GroupDefinition profileName="LdapGroupProfile" className="erLdapGroupAccount"
                rdnAttribute="erLdapServiceGroup" accountAttribute="erLdapGroupName">
                <AttributeMap>
                        <Attribute name="erGroupId" value="erLdapGroupRDN"/>
                        <Attribute name="erGroupName" value="erLdapServiceGroup"/>
                        <Attribute name="erGroupDescription" value="erLdapGroupDescription"/>
                        </AttributeMap>
                <properties>
                        <property name = "Managed">
                                        <value>true</value>
                        </property>
                </properties>
                <form location="erLDAPGroupAccount.xml" />
        </GroupDefinition>
</ServiceGroups>
```

2. *Make the following changes*

Replace erLdapServiceGroup with erLdapGroupRDN as highlighted below.

```
<ServiceGroups>
        <GroupDefinition profileName="LdapGroupProfile" className="erLdapGroupAccount"
                rdnAttribute="erLdapGroupRDN" accountAttribute="erLdapGroupName">
                <AttributeMap>
                        <Attribute name="erGroupId" value="erLdapGroupRDN"/>
                        <Attribute name="erGroupName" value="erLdapGroupRDN"/>
                        <Attribute name="erGroupDescription" value="erLdapGroupDescription"/>
                        </AttributeMap>
                <properties>
                        <property name = "Managed">
                                        <value>true</value>
                        </property>
                </properties>
                <form location="erLDAPGroupAccount.xml" />
        </GroupDefinition>
</ServiceGroups>
```

3. *Save the **service.def** file*

- Modify the ***targetProfile.json*** file

- *Locate the **coreAttributes** within the **groupExtensions** section in the file*

```
"attributeMapping": {
    "objectProfile": "LdapGroupProfile",
   "commonAttributes": {
    "meta": {
   "created": "",
        "lastModified": ""
    }
},
"coreAttributes": {
    "displayName": "erLdapServiceGroup",
    "members": [
        {
            "value": "erLdapGroupName",
            "type": ""
        }
    ]
}
```

- *Make the following changes*

Replace erLdapServiceGroup with erLdapGroupRDN as highlighted below.

```
"attributeMapping": {
    "objectProfile": "LdapGroupProfile",
   "commonAttributes": {
    "meta": {
   "created": "",
        "lastModified": ""
    }
},
"coreAttributes": {
    "displayName": "erLdapGroupRDN",
    "members": [
        {
            "value": "erLdapGroupName",
            "type": ""
        }
    ]
}
```

- *Save the **targetProfile.json** file*

- Reconstruct the LDAP Adapter profile

Refer to the Identity LDAP Adapter knowledge center for instructions on how to recreate the adapter profile.

https://www.ibm.com/support/knowledgecenter/SSIGMP_1.0.0/isim70/ldap/install_config/t_aexpand.htm

- Import the LDAP Adapter profile

It's highly recommended to restart the "IBM Security Identity Governance and Intelligence Server" from the local management interface (LMI) of the virtual appliance. This is done to ensure that updated profile will be used since adapters profiles are cached in IGI for performance.

Observations

- ## IGI permissions names

With this configuration of the adapter, the LDAP groups permissions will be represented in IGI by the full LDAP group DN. An example of before and after:

Sample Groups Hierarchy in LDAP

Groups are indicated by ***cn=value***

ou=WebSphere, ou=Applications, dc=domain, dc=com
- ou=ApplicationA
  - ou=Development
    - ***cn=WAS Monitor***
    - ***cn=WAS Administrators***
  - ou=Testing
    - **cn=WAS Monitor**
    - **cn=WAS Administrators**
- ou=ApplicationB
  - ou=Development
    - ***cn=WAS Monitor***
    - ***cn=WAS Administrators***
  - ou=Testing
    - **cn=WAS Monitor**
    - **cn=WAS Administrators**

IGI Permissions names before the configurations described in the technote

- WAS Monitor
- WAS Administrators

IGI Permissions names after the configurations described in the technote

- cn=WAS Monitor, ou=Development, ou=ApplicationA, ou=WebSphere, ou=Applications, dc=domain, dc=com
- cn=WAS Administrators, ou=Development, ou=ApplicationA, ou=WebSphere, ou=Applications, dc=domain, dc=com
- cn=WAS Monitor, ou=Testing, ou=ApplicationA, ou=WebSphere, ou=Applications, dc=domain, dc=com
- cn=WAS Administrators, ou= Testing, ou=ApplicationA, ou=WebSphere, ou=Applications, dc=domain, dc=com
- cn=WAS Monitor, ou=Development, ou=ApplicationB, ou=WebSphere, ou=Applications, dc=domain, dc=com
- cn=WAS Administrators, ou=Development, ou=ApplicationB, ou=WebSphere, ou=Applications, dc=domain, dc=com
- cn=WAS Monitor, ou=Testing, ou=ApplicationB, ou=WebSphere, ou=Applications, dc=domain, dc=com
- cn=WAS Administrators, ou= Testing, ou=ApplicationB, ou=WebSphere, ou=Applications, dc=domain, dc=com

- ## Using the same LDAP connector

If you defined an LDAP connector in IGI prior to making the configuration changes described in this technote, then imported the new profile, you must perform the following steps:

- Issue a “Change Log Synch” on the connector
- Make sure the connector in running
- Wait until all the inbound events are completed

At this stage, there will be no permissions in IGI. You must do the following:

- *“Reset” the connector’s cache* **OR** *remove all entries from the itimuser.ib_re_gropus table for this connector (indicated by the target_id column)*
- Issue a “Change Log Synch” on the connector