LDAP PosixAccount and PosixGroup Management

Basic Idea

This scenario documentation is in progress.

This scenario helps to understand how midPoint can create both standard LDAP groups (groupOfNames) and posixGroup LDAP groups as projections of midPoint roles. The roles can be assigned to users to provision either LDAP accounts (inetOrgPerson) with LDAP groups membership, or to extend the standard LDAP account with posixAccount auxiliary object class and make them members of posixGroups. One possible usage of posixAccounts and posixGroups can be central Linux access management (using LDAP accounts) and limiting access to specific Linux servers using specific LDAP groups with PAM.

In this scenario, the posixGroups will be created for this purpose (that's why the "Unix" is used in the metarole name), but the actual Linux server configuration is outside of the scope of this story.

This sample assumes OpenLDAP installation.

The directory tree structure in OpenLDAP is pretty simple:

- dc=example,dc=com
    - ou=people
        - uid=jsmith
        - uid=jdoe
        - ...
    - ou=groups
        - cn=wiki-users
        - cn=bughunt-users
        - ...
    - ou=unixGroups
        - cn=server1-users
        - cn=server2-users
        - ...

The standard LDAP groups will be created in ou=groups container while the posixGroups will be created in ou=unixGroups container. Account will be created in ou=people (flat, no further structure). All these containers are assumed to exist.

Scenario Details

The roles created in midPoint can be provisioned as either LDAP groupOfNames or posixGroup objects, based on the metarole assigned to the role. Creating midPoint role with "LDAP Group Metarole" metarole assigned will create new group in "ou=groups" container. Creating midPoint role with "LDAP Unix Group Metarole" assigned will create new group in "ou=unixgroups" container. In both cases, the name of the group (cn) will be set to the value of "identifier" role attribute. This means that "name" of the role can be anything, only "identifier" will be used for provisioning the group.

The metaroles contain also higher order inducements so that assigning the newly created roles to the users will create LDAP account and put it to the corresponding group. The "LDAP Unix Group Metarole" will additionally extend the LDAP account with "posixAccount" auxiliary object class (and its mandatory attributes). This means that we can have either standard LDAP accounts with standard group memberships, or extend the standard LDAP accounts with auxiliary objectClass "posixAccount" just by assigning a midPoint role which has "LDAP Unix Group Metarole" assigned! We can also reverse the operation and remove the auxiliary objectClass "posixAccount" (and all its attributes) from the account by unassigning the midPoint role which has "LDAP Unix Group Metarole" assigned. At any time, midPoint user has only one projection - LDAP account; just the objectClass and the attributes differ.

You can create any number of roles, but the whole logic is always in the two metaroles. Nowhere else. This means, the solution is perfectly managable; and the metaroles work as "role templates".

Another interesting feature is how the uidNumber and gidNumber attributes are generated when creating posixGroup object and posixAccount object in LDAP. These attributes are required and must be unique. So midPoint Sequences are used.

TODO more here with fragments?

Copyright © 2019 Evolveum. All rights reserved.

The Files

OpenLDAP ACI	https://github.com/Evolveum/midpoint/blob/master/samples/stories/unix-ldap/aci.ldif	ACI for OpenLDAP user management for this sample. Update as you wish.
Sequences	https://github.com/Evolveum/midpoint/tree/master/samples/stories/unix-ldap/other	Sequences to generate unique UID/GID numbers.
Resources	https://github.com/Evolveum/midpoint/blob/master/samples/stories/unix-ldap/resources/ldap-posix.xml	Configuration for source and target systems. Connection properties, schema handling and synchronization configuration.
Roles	https://github.com/Evolveum/midpoint/tree/master/samples/stories/unix-ldap/roles	Meta roles for creating standard LDAP groups and posixGroups.

Resources

Resource	Type	Definition	Description
OpenLDAP posix	LDAP		Target Resource

OpenLDAP posix

Target resource. Organizational structure allows separate containers for each customer accounts and groups.

Resource Objects	kind	intent	Description
user accounts	account	default	Standard inetOrgPerson LDAP accounts (stored in ou=people,dc=example,dc=com)
LDAP groups	entitlement	ldapGroup	Standard groupOfNames LDAP groups (stored in ou=groups,dc=example,dc=com)
LDAP groups	entitlement	unixGroup	posixGroup LDAP groups (stored in ou=unixgroups,dc=example,dc=com)

Setup

Before testing, install OpenLDAP, setup the ACI using the aci.ldif file and import all the configuration from the files above to midPoint:

1. sequences
2. resource
3. roles

Scenarios

The following sections describe scenarios prepared for this sample.

New role with standard LDAP group projection

1. administrators logs in using midPoint GUI
2. administrators clicks Roles, then New role
3. administrator fills in the following attributes:
    a. `name`, e.g. "LDAP Group Wiki Users"
    b. `identifier` (this will become the group's `cn` attribute), e.g. "wiki-users"
4. administrator assigns the metarole "LDAP Group Metarole" (Assignments, not Inducements tab!)
5. administrator saves the form
6. midPoint will create a new LDAP group in OpenLDAP:
    a. dn: cn=identifier,ou=groups,dc=example,dc=com, e.g. cn=wiki-users,ou=groups,dc=example,dc=com
    b. objectClass: groupOfNames

New role with posixGroup LDAP group projection

1. administrators logs in using midPoint GUI
2. administrators clicks Roles, then New role
3. administrator fills in the following attributes:
    a. `name`, e.g. "LDAP Unix Group - Access to Athena"

Copyright © 2019 Evolveum. All rights reserved.

   b. `identifier` (this will become the group's `cn` attribute), e.g. "athena-users"
4. administrator assigns the metarole "LDAP Unix Group Metarole" (Assignments, not Inducements tab!)
5. administrator saves the form
6. midPoint will generate unique GID number using Sequences and store it in role's `extension/gidNumber` attribute. This is done in the metarole's focus mapping named "sequenceGID".
7. midPoint will create a new LDAP group in OpenLDAP:
   a. dn: cn=identifier,ou=unixgroups,dc=example,dc=com, e.g. "cn=athena-users,ou=unixgroups,dc=example,dc=com"
   b. gidNumber: the value generated by midPoint sequence, which is present in role's `extension/gidNumber` attribute.
   c. objectClass: posixGroup

New user with standard LDAP account with standard LDAP group membership

1. administrator logs in using midPoint GUI
2. administrators clicks Users, then New user
3. administrator fills in the following attributes:
   a. `name`, e.g. jsmith
   b. `givenName`, e.g. John
   c. `familyName`, e.g. Smith
   d. `password`
4. administrators assigns the previously created role for LDAP group (not metarole!), e.g. "LDAP Group Wiki Users"
5. administrator saves the form
6. midPoint will create a new LDAP account in OpenLDAP, e.g.:
   a. dn: uid=jsmith,ou=people,dc=example,dc=com
   b. objectClass: inetOrgPerson
   c. cn: John Smith
   d. sn: Smith
   e. givenName: John
   f. uid: jsmith
7. midPoint will make the new account member of the group created previously by the assigned role, e.g. "cn=wiki-users,ou=groups,dc=example,dc=com".

Everything for provisioning standard LDAP accounts is contained in the metarole "LDAP Group Metarole". So all you need is to do is create roles which will have projections - groupOfNames; and to create users and assign them the newly created roles.

Extending standard LDAP account with posixAccount objectClass and posixGroup membership

1. administrator logs in using midPoint GUI
2. administrators clicks Users, then List users and edits the user, e.g. "jsmith"
3. administrators assigns the previously created role for LDAP Unix group (not metarole!), e.g. "LDAP Unix Group - Access to Athena"
4. administrator saves the form
5. midPoint will generate unique UID number using Sequences and store it in user's `extension/uidNumber` attribute.This is done in the metarole's focus mapping named "sequenceUID" in the higher order inducement (so it will apply to the User, not Role).
6. midPoint will create a new LDAP account in OpenLDAP if it does not exist yet and posixAccount auxiliary objectClass and its attribute will be computed:
   a. dn: uid=jsmith,ou=people,dc=example,dc=com
   b. objectClass:
      i. inetOrgPerson
      ii. posixAccount (auxiliary)
   c. cn: John Smith
   d. sn: Smith
   e. givenName: John
   f. uid: jsmith
   g. gidNumber: the value generated by midPoint sequence, which is present in user's `extension/uidNumber` attribute. This is intentionally the same value as `uidNumber` (primary user group).
   h. homeDirectory: /home/jsmith
   i. uidNumber: the value generated by midPoint sequence, which is present in user's `extension/uidNumber` attribute.
   j. gecos: John Smith
7. midPoint will make the new account member of the posixGroup created previously by the assigned role, e.g. "cn=athena-users,ou=unixgroups,dc=example,dc=com". (The membership will use "memberUid" attribute of the group). This is done in metarole's association in the higher order inducement (so it will apply to the User, not Role).

Everything for provisioning posixAccount accounts is contained in the metarole "LDAP Unix Group Metarole". So all you need is to do is create roles which will have projections - posixGroups; and to create users and assign them the newly created roles.

See Also

- Configuration Samples

Copyright © 2019 Evolveum. All rights reserved.