Creating Dynamic Groups

SLAPO-DYNLIST - Dynamic List overlay to slapd

Description

The dynlist overlay to slapd(8) allows expansion of dynamic groups and more. Any time an entry with a specific objectClass (defined in the overlay configuration, e.g. `groupOfURLs`) is being returned, the LDAP URI-valued occurrences of a specific attribute (also defined in the overlay configuration, e.g. `memberURL`) are expanded into the corresponding entries, and the values of the attributes listed in the URI are added to the original entry. No recursion is allowed, to avoid potential infinite loops.

Since the resulting entry is dynamically constructed, it does not exist until it is constructed while being returned. As a consequence, dynamically added attributes do not participate in the filter matching phase of the search request handling. In other words, filtering for dynamically added attributes always fails.

The resulting entry must comply with the LDAP data model, so constraints are enforced. For example, if a SINGLE-VALUE attribute is listed, only the first value found during the list expansion appears in the final entry. The above described behavior is disabled when the manageDSAit control (RFC 3296) is used. In that case, the contents of the dynamic group entry is returned; namely, the URLs are returned instead of being expanded.

Configuration

The config directives that are specific to the dynlist overlay must be prefixed by dynlist-, to avoid potential conflicts with directives specific to the underlying database or to other stacked overlays.

`overlay dynlist`

This directive adds the dynlist overlay to the current database, or to the frontend, if used before any database instantiation; see slapd.conf(5) for details.

This following slapd.conf configuration option is defined for the dynlist overlay. It may have multiple occurrences that could appear after the overlay directive. However, it would require additional attributes be defined in the schema for the specific group memberships indicated.

`dynlist-attrset <group-oc> [<URI>] <URL-ad> [[<mapped-ad>:]<member-ad> ...]`

The value `group-oc` is the name of the objectClass that triggers the dynamic expansion of the data.

The optional `URI` restricts expansion only to entries matching the DN, the scope and the filter portions of the URI.

The value `URL-ad` is the name of the attributeDescription that contains the URI that is expanded by the overlay; if none is present, no expansion occurs. If

the intersection of the attributes requested by the search operation (or the asserted attribute for compares) and the attributes listed in the URI is empty, no expansion occurs for that specific URI. It must be a subtype of labeledURI.

The value `member-ad` is optional; if present, the overlay behaves as a dynamic group: this attribute will list the DN of the entries resulting from the internal search. In this case, the attrs portion of the URIs in the URL-ad attribute must be absent, and the DNs of all the entries resulting from the expansion of the URIs are listed as values of this attribute. Compares that assert the value of the member-ad attribute of entries with group-oc objectClass apply as if the DN of the entries resulting from the expansion of the URI were present in the group-oc entry as values of the member-ad attribute.

Alternatively, `mapped-ad` can be used to remap attributes obtained through expansion. member-ad attributes are not filled by expanded DN, but are remapped as mapped-ad attributes. Multiple mapping statements can be used.

The dynlist overlay may be used with any backend, but it is mainly intended for use with local storage backends. In case the URI expansion is very resource-intensive and occurs frequently with well-defined patterns, one should consider adding a proxy cache later on in the overlay stack.

Authorization

By default the expansions are performed using the identity of the current LDAP user. This identity may be overridden by setting the dgIdentity attribute in the group's entry to the DN of another LDAP user. In that case the dgIdentity will be used when expanding the URIs in the object. Setting the dgIdentity to a zero-length string will cause the expansions to be performed anonymously. Note that the dgIdentity attribute is defined in the dyngroup schema, and this schema must be loaded before the dgIdentity authorization feature may be used. If the dgAuthz attribute is also present in the group's entry, its values are used to determine what identities are authorized to use the dgIdentity to expand the group. Values of the dgAuthz attribute must conform to the (experimental) OpenLDAP authz syntax.

Examples

This example collects all the email addresses of a database into a single entry; first of all, make sure that slapd.conf contains the directives:

```
include /opt/symas/etc/openldap/schema/dyngroup.schema
# ...
moduleload dynlist.la
# ...
database <database>
# ...
overlay dynlist
dynlist-attrset groupOfURLs memberURL
```

The overlay indicates the objectClass to be used for creating dynamic groups is "groupOfURLs" and the attribute to be used is "memberURL", both of which are defined in the dyngroup.schema file that must be loaded.

The dynamically group needs to be defined in the database. The `memberURL` attribute can be generated by using the ldapurl command.

```
dn: cn=Dynamic List,ou=Groups,dc=example,dc=com
objectClass: groupOfURLs
cn: Dynamic List
memberURL:
ldap:///ou=People,dc=example,dc=com?mail?sub?(objectClass=person)
```

If no <attrs> are provided in the URI, all (non-operational) attributes are collected. However, multiple attributes can be defined in the memberURL by use of filters.

```
dn: cn=Dynamic List,ou=Groups,dc=example,dc=com
objectClass: groupOfURLs
cn: Dynamic List
memberURL:
ldap:///ou=people,dc=example,dc=com??sub?(&(title=manager)(location=main
office))
```

Where the first example queues off the "mail" attribute only, the second example would look for people with a title of "manager" who are located in the "main office".

This example implements the dynamic group feature on the "member" attribute:

```
include /path/to/dyngroup.schema
#...
moduleload dynlist.la
# ...
database <database>
# ...
overlay dynlist
dynlist-attrset groupOfURLs memberURL member
```

This causes group membership to be represented by the "member" attribute so it looks like a "normal" LDAP group. Otherwise it just returns the full entries of all matching members which is typically not a desired result.

(Currently Experimental)
This example creates a dynamic group with dgIdentity authorization. Such a group could only be expanded/evaluated by the specified dgIdentity member(s) and would be invisible to all other users.

```
dn: cn=Dynamic Group,ou=Groups,dc=example,dc=com
objectClass: groupOfURLs
objectClass: dgIdentityAux
cn: Dynamic Group
memberURL: ldap:///ou=People,dc=example,dc=com??sub?(objectClass=person)
dgIdentity: cn=Group Proxy,ou=Services,dc=example,dc=com
```

NOTE: In order to manage the dynamic database URL, you must enable manageDSAit before connecting to the database and browsing to the group entry. Otherwise, the LDAP browser will return the group members instead of the URL attribute.