LDAP Provisioning Plugin Mark 2

Background

The LDAP Provisioning Plugin has become increasingly complicated over time. Some of this is unavoidable, due to the nature of LDAP and the use cases supported by the Plugin. This document discusses proposals for simplifying the Plugin.

Drop Support for Externally Managed Attributes

Externally Managed Attributes are those not managed by the LDAP Provisioning Plugin. The primary use case for Externally Managed Attributes is Grouper integration, where COmanage provisions person attributes and Grouper provisions group attributes. There are other use cases as well.

Supporting Externally Managed Attributes requires two features: *Unconfigured Attribute Mode* and *Additional ObjectClasses*. Unconfigured Attribute Mode must typically be set to *Ignore*, which creates issues when attributes are deconfigured or objectclasses are removed (as discussed in the documentation). It becomes difficult or impossible to remove attributes from the LDAP record via COmanage, requiring administrative operations on the LDAP server.

Alternate Approach

The proposed new approach would leverage *meta directories*. The LDAP Provisioning Plugin would assume it always has full control over the primary LDAP record constructed for the CO Person. This is equivalent to the current Unconfigured Attribute Mode *Remove*.

The Plugin would also, if configured, construct a skeletal record in a second location (another tree on the same server or an a different server). This minimal record would consist only of the search attribute used by Grouper and whatever other minimal attributes are required by the minimal possible set of objectclasses. The Plugin would simply create this skeletal record if it does not exist, and delete it when the appropriate deprovisioning conditions are met. All other management would be delegated to the appropriate applications.

(This might actually be implemented by having a "minimal attribute mode", so that a deployment using Grouper would have two LdapProvisioner configurations, one regular and one minimal.)

Finally, deployers would set up a meta directory, combining the results into a single LDAP record for clients.

Impact

Existing deployments that are not using *Additional ObjectClasses* would not be affected.

Other deployments would need to set up two new LDAP instances. This increases the server setup and operational burdens, though in exchange would experience fewer edge case difficulties in managing operational records.

Move All Object Classes to Plugins

LDAP Schema Plugins were introduced in v2.0.0 to allow deployers to add local ObjectClasses to their LDAP server via the LDAP Provisioning Plugin. Core ObjectClasses should be moved to a CoreSchema plugin to unify schema handling and simplify the LdapProvisioner implementation.

See Also

- CFM-75 LDAP Provisioning Plugin