Windows Enterprise Design
FERPA-Compliant Class Lists as Windows Security Groups

Steven L. Kunz
Windows Enterprise Administration
Information Technology Services

Base Version: July 25, 2006
Last Update: May 18, 2012

At Iowa State University we have a "single forest, single domain" Active Directory model with about 56,000 user accounts and 25,000 Windows systems. Since the early 1990s we have had an "MIT Athena based" UNIX system provisioning user accounts for the entire university. In 2000 we began synchronizing the UNIX accounts into Active Directory with a locally written interface between the Unix KDCs and the domain controllers. Usernames and passwords are always identical in both the Unix and Windows environments (user provisioning pushdown from Unix, bi-directional password synch).

The existing Unix system handles "non-disclosure" of personal information by allowing students to indicate (via a form they fill out) that they do not want name, address, phone, etc. published in "online directories". This means the master LDAP and "paper phone book" has "user suppressed" supplied for all such fields. When we started populating Active Directory from the UNIX "master sources" (the MIT Kerberos KDCs, the moira database, and the public LDAP servers) the user objects in Active Directory also began to say "user suppressed" for those people who wanted personal information suppressed. Active Directory contains only minimal information about the user - mainly enough to identify the person in the GAL and provide a few other attributes. It is not a "primary directory repository" for a lot of university information.

The existing UNIX system already had "lists" of usernames for "departments", "colleges", "majors", and "classes" automatically generated from official university sources. We wanted to push those down to Windows Active Directory as Global Security Groups in a FERPA-compliant fashion. Class lists were particularly problematic (you cannot expose what classes a student is taking) so we needed to do something about Windows "group membership" if we wanted groups based off class lists.

We extended our "account synch" process to include a "list synch" function, pushing down list memberships to synch them as Global Security Groups. Our solution was a "two tier" approach first suggested by Ross Wilper of Stanford. Matt Kramer (Boston University) also provided some valuable input into ACL techniques for what we wanted to do. What we do is create a security group based on the existing list name of the class (such as "f2005.com_s.103.1", meaning "Fall 2005 COM S 103 Section 1"). However, we do NOT make the membership of this group be the user accounts of the members of the class. Instead, at the time we create the "f2005.com_s.103.1" security group we

create a second group named with a random-number name like "hsg-39472649". We then populate the second group with the members of the class, make the second group a member of the first. The final (important) step is tweaking the ACLs of the groups so the average user sees an "empty" membership of "f2005.com_s.103.1" (indeed, they cannot see any of the "hsg-<n>" groups in the OU container they live in). The instructor of record can see the membership of their classes only (this is done by also having a "f2005.com_s.103.1-instructors" group created at the same time the class group is created, with appropriate ACLs set up).

The end result is you look at a user object for "astudent" and click on the "Member Of" tab and you see "Domain Users", some "hsg-<n>" groups (for some classes they are taking) and whatever other groups people have made them a member of. Click on one of the "hsg-<n>" groups and the membership is hidden. There is no indication of what class the "hsg-<n>" group represents. Resource-Kit tools like "ifmember.exe" work just fine in logon scripts to control mounts based on "class", "college", "department" or "major" membership. Anybody can use the security group for a class or any other list-based group (you can see the name) for access control without being able to see the membership.

Adds/Drops (for classes) and faculty/staff departmental hiring changes are automatically populated into the official university lists, and pushed down via the software interface to Active Directory. These updates are "incremental" (there is not a total rebuild of the group each night based on the entire list). Since there is an exact match of accounts on the Unix and Windows side list/group membership actions (adds/deletes) are simple.

Maintenance of the "official university" groups (for "departments", "colleges", "majors", or "classes") by anyone is not allowed either within Active Directory or Unix - it is all automated from the official university data feed (from the Registrar's and Personnel offices). A nightly process pushes down updates (including list changes and personal attribute changes) since that is when the data is processed from the master sources.

A general document on official university lists we provide for the user community is available here:

http://tech.its.iastate.edu/windows/admin/ListSync.pdf

Again, this is all locally written and based off a rather extensive existing Unix system that few other people will have. Some of the ideas may be useful to others so I present them here.