Sharing Accounts

Matt Bishop
mab@ riacs.edu
Research Institute for Advanced Computer Science
NASA Ames Research Center; Moffett Field, CA 94035

(Work reported here was supported by NASA under contract NCC2-387.)

The Numerical Aerodynamic Simulator project runs a variety of UNIX based operating system, on its computers (a Cray 2, 2 Amdahl 5840s, 4 VAX-11/780s and 25 IRIS 3500 workstations.) Users work on and off site, using a variety of networks not all of which are under NASA's control. Off site installations can be as close as a different building on the base or as distant as ICASE (on the East Coast.) In our environment, sharing accounts is very common; it provides a very quick and easy way to enable many people to work together, and allows many people to share responsibility for a particular task. For example, the account *nasops* is used to maintain a database of users (among other functions.) So, many people need access to that account. Unfortunately, this poses some problems.

First is the question of accountability. If someone logs in on the account *nasops* and compromises the database, how can the offending user be traced? Password management is the second problem; how can the site administrator force 40 or so people to keep the password secret, particularly since these users need at least two passwords (one for their own account and one for the *nasops* account)? When someone changes *nasops*' password, how does he or she communicate that change to the other users in a timely manner?

A group account is an account meant to be shared. No-one can log into a group account, but an *su*-like program called *lsu* overlays the login identity with the group identity (just as *su* overlays the login identity with a new user's identity.) The user must type his own password when switching to the group identity. *Lsu* checks an access file to ensure that the user can access the group account at the given time and from the given terminal, and then checks the password. If access is allowed and the user types his own password correctly, the group identity is pushed over the user's identity; if access is denied or the password is incorrect, *lsu* simply informs the user permission is denied.

Because of the sensitive nature of the program, we took several steps to prevent compromise. While it seems redundant to require the user to type his password (didn't he type it to log in?), experience shows that people do leave their terminals unattended, so checking the password provides some assurance the user is the person running *lsu*. The password must always be typed, even when *lsu* has already determined the user has no right to *lsu* to the group account. The location of the access and log files are constructed in memory when *lsu* runs, and are erased immediately after being used. Both files must be owned by ***root*** and must be mode 600; if not, the *lsu* fails and mail is sent to the *lsu* administrators. Of course, when access is denied, the user is not told why access is denied.

We also intend to provide the same control for overlaying user accounts. The program *nsu* functions like *lsu* but requires the new account's password as well as meeting any conditions in the access file. (If the account is not listed in the access file, anyone can *nsu* to it.) In the event the access file is trashed, *nsu* can only be used to access *root*.

Lsu and *nsu* use the startup file of the new (group or user) identity, not that of the user running the program; the environment variables **USER** and **HOME** are also changed. This provides uniformity among users who may have wildly different environments in their private accounts. A third program, called *su*, provides the usual *su* environment but uses the *nsu* access file to check permission.

Currently *lsu* and *nsu* run on twelve different systems (Berkeley's 4.2 and 4.3 BSD, Sequent's DYNIX 2.0, the NAS' NPSN 3, Ridge's ROS 3.3, Silicon Graphics' SGI 2.3, 3.4, and 3.5, Sun's 4.2, AT&T's System V, Amdahl's UTS, and Cray's UNICOS.) User reaction to *lsu* has been very favorable, largely because of the consistent environment and the relief from remembering multiple passwords. *Nsu* and *su* have just recently been made available, so we do not yet know how the user community will accept them.