Toehold -- A Workstation Login Facility

Preliminary Design Draft

Jim Aspnes

Abstract

Toehold is intended to be a comprehensive system for managing user login sessions on Athena workstations. This will include management of Kerberos authentication tickets, management of system and personal Remote Virtual Disks, and creation of temporary accounts for workstation users. The system will also provide information to the User Locator server, and will provide a hook for an automatic workstation software update facility.

Table of Contents

1 Purpose

The workstation environment differs significantly from the current Athena timesharing environment. Differences important to Toehold tend to fall into three classes:

- Software management. Unlike timesharing machines, workstations will need to support some sort of automatic software distribution system. Thus it will be necessary for workstations to spin down their /srvd and /urvd RVD's periodically, allowing their contents to be updated from a central location. It will also be necessary for workstations to periodically compare their software installation with a centralized distribution and update locally resident software if needed.

- User account management. It is unfeasible for each of many thousands of workstations to maintain /etc/passwd and /etc/rvdtab entries for all Athena users. Similarly, the existing Unix tools for user location and notification, designed for environments with one or only a few closely-connected machines, are unsuited for the Athena workstation environment. It will thus be necessary for the Toehold system to interact with centralized database and user locator services to dynamically create accounts at login and to notify the user locator server of this fact.

- Security. Since workstations will not be physically secure, it will be necessary to have a secure software-based authentication protocol. This facility is provided by Kerberos; Toehold will need to support it.

2 Design Issues

Several other issues come up in designing a login/logout system for the Athena workstation environment. Most have come up as questions regarding the ability to carry over certain desirable features of the timesharing user environment to the workstation environment.

2.1 Multi-User Workstations

There is some controversy as to whether more than one user should be logged into a workstation at one time. On the one hand, we would like to guarantee that the user on the workstation console should be able to treat the workstation as a private machine; but on the other hand, we should not preclude multiple users if it is reasonable to have them. Our design provides mechanisms for providing additional privileges to the primary user, and does not assume that only a single user will be present. This allows restrictions on multi-user use to be enforced optionally by the rlogin software, and for other programs (for example, su, o_kill, or o_renice) to be made usable only by the

primary user.

2.2 Kerberos and /etc/passwd

In the long run, all but a few system-specific users should be authenticated by Kerberos. Even when this is the case, though, there will be some need for users to be authenticated by the local /etc/passwd file even though they are not recognized by Kerberos. Thus our design allows authentication through /etc/passwd, although this will only be tried if Kerberos fails and will prevent automatic RVD spinup or account creation.

2.3 Background Processes

Several users have pointed out the utility of being able to leave batch-mode jobs running as background processes after logout. Unfortunately, most background processes left behind will not have much justfication for their existence, since neither the user's Kerberos tickets, RVD locker, nor /etc/passwd entry will survive his or her logout, and thus most background processes should be terminated on logout. However, this is not the case for a user who is authenticated by /etc/passwd rather than Kerberos, so batch-mode processes for such users could reasonably be left running.

3 Proposed Design

Changes to the current system will be made by altering /bin/login and by implementing a new program, /etc/toehold, which will run under init on the console. /bin/login will be responsible for those parts of the toehold system which are user-dependent; /etc/toehold will be, for the most part, responsible for system-wide operations.

3.1 Modifications to /bin/login

/bin/login will be modified to perform the following additional actions on login:

1. Spinup of system RVDs. This allows, for example, a user to be logged in remotely even if no one is on the console.
2. Kerberos Authentication.
3. Notification of User Locator.
4. Temporary account creation (from information in the user database).
5. Spinup of personal RVD locker, which is mounted on the temporary home directory.

Each login process will remain running during its login session, in contrast to the present system under which the user's shell is exec'd in place of /bin/login. This will allow it to send periodic updates to the User Locator service; it will also allow it to perform the following actions on logout, if this is the last login session remaining for this user. (This cleanup will not be necessary for passwd-authenticated users).

1. Termination of all process owned by the user.
2. Destruction of Kerberos authentication tickets.
3. Spindown of personal RVD(s) if possible.
4. Elimination of temporary account.
5. Notification of User Locator.

This has the effect of restoring the workstation to its state before login was run for the current user, except that system RVDs are not spun down. This is to allow other users to continue to use the system; in the event that there are no other users, system RVDs will be spun down by /etc/toehold.

3.2 /etc/toehold

/etc/toehold is principally responsible for system-wide operations. These include managing the X server and the system RVDs, and performing cleanup operations when there are no users on the system. Under normal conditions, /etc/toehold will wait for a keystroke on the console, which will tell it to bring up the system RVDs and start X; during this time it will periodically check to see if cleanup is necessary.

Cleanup operations will include:

- Stopping X.
- Forcible spindown of system RVDs, terminating all processes which depend on same.
- Rebuilding the /etc/passwd dbm database from the /etc/passwd file, thus eliminating any temporary entries which were accidentally not removed.
- Checking the workstation for software consistency and updating if necessary. (The method by which updates will be performed is at present somewhat vague.)

The cleanup procedure will be separated from the rest of

/etc/toehold as much as possible; it will be run when the console user logs out, if no other user is logged in. It will also be run periodically if no user is logged in, so that login windows will time out if they are not used.

4 User-visible differences

Most of the functions provided by the Toehold system can be performed invisibly, without requiring changes to the user interface presently provided by /etc/getty and /bin/login. Certain differences, however, are unavoidable.

4.1 X Startup

In order to meet the goal of minimizing the amount of software stored locally to the workstation, X will not be running on an unused workstation. Instead, the user will be prompted on the console to hit a key to start X; only when this is done will X be started and a login window created. A quick test of this feature determined that this would take about 30 seconds, not including additional time to spin up system RVDs if the workstation is in a totally dormant state.

4.2 RVD selection

If a user is associated with more than one RVD locker, he may need to specify which RVD lockers, if any, the login program should spin up. I have been assuming that each user will have at least one designated RVD locker which will replace his or her home directory and into which other lockers will be mounted. The exact behavior of this feature is dependent on the interface available to the user database, which has not yet been fully specified.

4.3 Background process termination

Some users might find it annoying that their background processes are terminated on logout, although it is my hope that most such processes are unnecessary and will not be missed. It might be worthwhile to establish a standard, no-password guest account which is not Kerberos-verified to allow users to run background processes which survive between login sessions if this is truly a desirable feature.

5 Implementation Issues

The design is set up in such a way that a framework for it could be constructed quickly, with most of the features missing; this would include:

- /etc/toehold with X startup, system RVD spinup, and cleanup facilities.

- /bin/login, with Kerberos authentication and temporary account creation/deletion, but without automatic RVD spinup.

Missing would be features dependent on being able to obtain RVD information from the user database, and notification of the User Locator Service, since neither of these facilities have been documented at present. These additional features could be added without much effort once the initial framework was in place.