Published: 05/31/67

Identification

Personnel List
C. Marceau

Purpose

When a user logs in, the procedure which identifies him (user_who, see BQ.3.02) must have access to certain information about the person who is logging in, the project he wishes to work on, and the console from which he is logging in. For example, the procedure must ascertain that the person logging in is really the person he claims to be (he must give a secret password). Section BQ.4.00 contains an overview of the directory structure containing information about persons, projects, and consoles. This section (BQ.4.02) discusses the personnel list, which contains information needed to log in persons. (Note: administrative records on personnel are not directly associated with the personnel list, which is kept solely for reference at login time.)

Discussion

The reader of this section should be familiar with section BQ.2.03 on the User Control Process, which discusses the user_who procedure and the method it follows to identify the user logging in.

Briefly, in the course of identifying the user, user_who asks for the user's personal password and checks it against a password kept in the user's entry in the personnel list (his personal identification file). If the user did not specify a project id in his login input line, user_who finds a default in his personal identification file. Similarly, if one person (A) logs in as a proxy for another person (B), user_who checks in B's personal identification file to make sure that A is a valid proxy.

Access Control

Access to the personnel list and personal identification files is strictly controlled so that unauthorized users cannot tamper (either inadvertently or deliberately) with the information they contain. The file system access control mechanism is described in section BX.8.00; access control for the personnel list is described in section BQ.4.00. This section briefly summarizes those descriptions, and adds a description of access control on personal identification files.

It is possible to define the mode of access of a particular user (or even of a particular instance of that user, a process-group) to a directory or segment. It is further possible to limit that user's access so that he may only get at the directory or segment from a particular protection ring. Three protection rings concern us here: the hard core ring, which contains the access control mechanism and other equally sensitive modules; the administrative ring, which contains administrative modules and data bases; the user base ring, which is a relatively unprotected user area. (A more complete description of the purpose and philosophy of protection rings may be found in BD.9.00.) Note that the ring in which a segment resides depends on which user is trying to access it; i.e., user A may be able to access a segment from the user ring, while user B can only get at it from the hard core ring.

If a user has access to a segment or directory in some ring, his mode of access is defined by certain usage attributes. For example, he may read a directory to see what is in it (Read attribute), search the directory for a particular entry (Execute attribute), change or delete entries in the directory (Write attribute), or add an entry to the directory (Append attribute). For segments the attributes have slightly different meanings: a user may read the segment (Read attribute), execute a procedure segment (Execute attribute), change or delete part of the segment (Write attribute), add to the segment (Append attribute). A fifth attribute, Trap, does not concern us here (see BG.9.00).

The personnel list contains the names of all persons who may log in. To add or delete an entry in the personnel list directory is equivalent to adding or deleting one person from the set of those who may log in. Therefore, only the system administrator has the write or append attributes on for the personnel list directory. He may also read and search the directory. He has this access only in the administrative ring and only when he is working on project "system" - i.e., working in his capacity as system administrator.

Every user may search the personnel list (execute attribute) to find his own personal identification file. He has this access only in the hard core ring.

Similarly, when a user is logging in, the User Control Process which logs him in (see BQ.2.03) must be able to find his personal identification file. Hence the User Control Process may search (execute attribute) the Personnel List from the administrative ring.

Personal Identification Files

A user's personal identification file may be accessed only by himself and the User Control Process. He may write in his personal identification file in the hard core ring, and the User Control Process may read from his personal identification file in the administrative ring. The reason that a person's access to his own file is limited is this: a proxy may log in for the user and have the same access rights as the user. I.e., he may write in the user's personal identification file! Now a proxy logs in using his own password, and does not need to know the password of the user for whom he proxies.

We therefore require that to read or write in a user's personal identification file, it is necessary both to be logged in as that user, and to give the user's personal password. The hard core ring module that changes passwords always demands that the user give his password before he is allowed to change it. (See BX.3.02, the password command.)

Besides the person's password, his personal identification file also contains a default project id and a list of persons who are allowed to log in as proxies for this person. If a user often logs in on a certain project, he may specify this as his default project and omit the "project id" argument when he logs in. If he does not give a project id when logging in, the User Control Process looks up his default in his personal identification file. The list of proxies kept in the file is checked whenever some person claims to be logging in as a proxy for this person.

Implementation

As stated in BQ.4.00, the personnel list is immediately inferior to the login directory, which is immediately inferior to the root directory. The personal identification

file for each person bears the name of the person as he logs in (it must be less than 24 characters long). Thus the personal identification file for a person named John Doe might be the segment named John_Doe which is an entry in the personnel list directory. The segment would have path name

(root) > login_dir > personnel_list > John_Doe

The segment has the form of a controlled PL/I structure:

```
dcl 1 personal_file ctl (p),
    2 password char (8),
    2 project_id char (24),     /* default project id */
    2 nproxies fixed bin (17), /* number of proxies  */
    2 proxies (p->personal_file.nproxies)
      char (24);               /* array of proxy names */
```

Initialization

When the system administrator adds a person to the system, he issues an administrative command which adds a personal identification file for the user to the personnel list directory. He places in the personal identification file some arbitrary password, which he communicates to the user. He also enters the user's default project id and sets the number of proxies equal to zero. Finally he modifies the access to the personal identification file to correspond with that indicated above. Now the user may log in and change his password.

Should disaster strike, and the user forgets his password, he can appeal to the system locksmith for help. The locksmith is a trustworthy soul who has power to peek into anyone's personal identification file (leaving auditing trails behind) but does not do so unless the person asks him to. If there were no locksmith, forgetting your password would mean you could never log in again.

This leaves one person out in the cold - the proxy who is not also a user. That is, he may log in for other users but does not himself work on a project. Hence he cannot log in as himself to change his password. However, he can ask the system administrator to let some other user (for whom he proxies) have access to his personal file. Of course, the other user cannot read or change the password (unless he knows the password) but when the proxy logs in for that user he can change the password by issuing the appropriate command.

Finally, note that once a person is known to the system, he can be admitted to any number of projects by the administrators of the projects. No further intervention by the system administrator is necessary.