SCOPE/Hustler's security facilities were crude. But if used carefully, they could result in a fairly secure system.

Memory Security

CPU code could access only its own memory ("field length"), as controlled by the computer's RA and FL registers. Any given piece of code (system or user) either had complete read/write/execute access to any given address, or no access to that address.

User jobs, and system control point jobs (equivalent to services under Microsoft Windows, or non-inetd daemons under UNIX) had RA and FL set so that they could not access system memory or the memory belonging to other jobs. This meant that user jobs could not access or corrupt other jobs, or the OS.

However, CPU code in what would now be called the OS kernel had RA and FL set so that it could access any memory location. And PP code could always access any central memory location. This meant that these parts of the operating system had to be coded very carefully. It also led to the requirement that users could not load their own PP code.

File Security

SCOPE/Hustler protected permanent files only with passwords rather than by user or group as done by most modern operating systems. Furthermore, by default, permanent files were created without passwords, meaning that anyone could read, modify, or even delete them. In practice, this was not much of a problem. There were few malicious users back then, and many users did not mind if others saw their data. Users who did care were careful to use passwords when creating ("cataloging") permanent files.

Even a systems programmer could not easily gain access to a permanent file without knowing its passwords. However, with some difficulty it could be done at the console.

The essential programs on the operating system did not reside in the filesystem. Instead, they were loaded into memory, or to special regions on disk outside the filesystem, from tape at deadstart. Thus, there was little chance for a malicious user to hack the operating system by modifying files.

Tape Security

Any user could read any magnetic tape, except for tape whose Visual Reel Numbers started with certain characters. For example, tapes with VRNs starting with VIM or SYS were restricted to systems programmers.
A user could write to a tape only if the tape was unlabelled, or if the label on the tape matched their Problem Number.

These restrictions could be overridden by an operator. One could place a HAL, PAUSE, msg to operator. control statement in a job to send a message to the operator to accommodate special tape requests. The operator had the responsibility to ensure that this capability was not being used to circumvent security.

When files were backed up ("dumped") to tape, their passwords and ownership information were also dumped. The passwords were not encrypted. Furthermore, any user could dump any file to a tape that they owned. Because of this, the OS had complex restrictions on how dump tapes could be used.

The system dump programs could write only to tape with certain VRNs. (Curiously, there were two dump programs: SELDUMP, which was used by operators to dump all files on the system, and PFDUMP, which was used by users to dump their own files.) For example, when run by non-systems programmers, PFDUMP could write only to tapes with VRNs starting with UP. And programs other than system dump and restore programs were forbidden from reading these tapes. Finally, tapes with these special VRNs were restricted from ever being taken out of the machine room. This prevented users from being able to learn the passwords for a file by dumping it to tape, and then reading the tape.

I know of no theoretical flaws in this scheme, nor do I know of any incident in which permanent file security was subverted by this or any other mechanism. However, it seems like a good avenue for attack.

**User Security**

To run a job, a user had to provide a Problem Number (PN; more about that later), userid, and password. There were many settings for each user, but most of them were for resource control. Such settings included the maximum number of CPU seconds per job, maximum number of pages printed per job, and which card readers and line printers the user was permitted to use. Only a few were security-related. In particular, the “systems bit”, assigned only to systems programmers, allowed the user to make certain system requests that were forbidden to ordinary users.

There was no equivalent to a UNIX root user: the systems bit did not give its owner powers equivalent to that of root. If a systems programmer needed the equivalent of root powers, s/he had to go into the machine room to use the system console. There were essentially no restrictions to what could be done at the console, but the commands were low-level and required considerable expertise for any “hacking”.

Users submitting batch jobs via card reader had to provide a special Problem Number Card (PNC) as the second card of deck. The PNC was a special binary card, issued by the Computer Laboratory, that in addition to the Problem Number had a “seed” punched into it. I do not recall why this special card was necessary: it seems redundant, since the user still had to provide a userid and password. At any rate, the PNC was unnecessary for interactive access.
User Management

Most current operating systems have a flat namespace of usernames for any given computer, and any administrative user on that computer can manage any user. SCOPE/Hustler provided a more sophisticated hierarchy of users. Any given user belonged to a “problem number”. A problem number was a group of related users that were all billed to the same account, and were all controlled by the same manager. Typically, each University course or research project had its own PN, but some small departments used one PN for the entire department. Userids were unique only within a problem number; thus, there could be many different users with the userid FRED, each in a different PN.

Each problem number had one PN manager—a user within that PN who had the power to create and otherwise manager users within that PN.

The Computer Laboratory Main Office had a special account that could be used to create and manage problem numbers and their associated PN managers. Administration of accounts within the PN was delegated to the individual PN managers.

Written 30 Oct 2004; last updated 30 October 2004 12:12:57 PM