TO: System Programmers

FROM: J. H. Saltzer, R. R. Fenichel

SUBJ: System Programming Security Measures.

DATE: 8/28/65

It appears that with the advent of the new file system it is now possible to make CTSS as secure against unauthorized tampering as is felt administratively desirable, while maintaining considerable freedom of action for system programmers wishing to make changes to the system. Herein is proposed a set of administrative procedures for use by system programmers which is intended to meet the following objectives:

1. Any authorized system programmer can work easily on the system with a minumum of hindrance from security measures.

2. The knowledge of a single system programmer's password is not usually, by itself, enough to allow an unauthorized user to tamper with the system or access confidential information.

3. Full knowledge of security procedures and all details of the system should be generally available, yet of no help to a programmer with malevolent intent.

Four changes are assumed to be made to the supervisor and file system:

1. Read-only and Write-only files cannot be deleted without first changing their mode. These files become the equivalent of the "read-only, class 1" files of the old system.

2. The 6.36 entries are only available to a privileged command, and then only to a programmer with the 6.36 permission bit.

3. The SELECT entry in the protection section is disabled.

4. The ability to read a private file not belonging to the author is restricted to:

   a. Privileged commands.

   b. Daemon-dump programs.

   c. User with private file privilege <u>if key 22 is down</u>.

With the above discussion in mind, we may now list one possible set of administrative policies to make the system secure:

1. A 7094 system programmer has the following privileges:

   a. Access to common files.

   b. Ability to patch core A if key 22 is down.

   c. Permission to use 6.36 command.

   d. Ability to use privileged file system entries (optional.)

2. All "staff maintained" commands are stored in common file two in Read-only mode, and any system programmer can change them.

3. The current working system is combined into a single bss file. This file and its associated load list is Protected and read only. Except for test sessions, the operating staff will never load an unprotected system.

4. Privileged commands are considered to be an extension of the core-A supervisor, and are stored in common file two as protected, read-only files.

5. The author of all protected system files is a single system programmer; updating of these critical files can therefore only be done with his knowledge and password.

6. All work on the A-core system and privileged commands should be carefully audited for unauthorized "midnight" changes. The following suggestions are typical techniques which are good practice anyway:

   a. Check the new length of a reassembled routine to see if it changed by the expected amount.

   b. Check the size of a newly loaded command.

   c. Glance through listings of newly translated modules for possible spurious code.

   d. Pre-load a new core-A system and check amount of core memory remaining.