To: Distribution

From: J.C.Whitmore

Date: May 17, 1974

Subject: Security Access Controls - Potential Problems

I. Introduction

Over the next few months, the Multics standard product will be modified to provide an additional access control mechanism. To some extent, the new access controls will limit a programmer's flexibility in designing new subsystems and will cause some existing commands and subsystems to stop working under certain conditions.

The purpose of this MTB is to identify some design pitfalls which are lurking around the corner and to provide a "first cut" set of design guidelines. These problems are of particular interest to designers of system functions. We would like all system functions to work equally well no matter how the new access controls are used. So now is the time to begin coping with the problems outlined below.

II. The New Access Controls

Before describing the problems, it is useful to review just what the new access control mechanism is and why it is being added to the standard product.

A. Purpose

To date, the approach to controlled sharing of information has been to allow any user (who has permission to change an ACL) to take full responsibility in specifying the mode of access for each ACL entry. This does not allow an administrator to have the system enforce his desire to restrict access on sensitive segments to a certain group of persons. He must rely on those who can set the ACL to do the right thing.

The new access control mechanism will identify all processes with a set of attributes. Likewise, segments and other objects, which provide communication paths between processes, will have a set of attributes. These attributes are defined system wide and are the basis for restricting access to some object by a process. This

---

will guarantee that system wide access definitions cannot be changed by an individual user setting an access control list.

The obvious application of this mechanism is the military security system, but the attributes could equally apply to divisions within a company, task groups within a project or group of projects, or even competitive companies using a single service bureau.

The control of these attributes rests with a single administrator (called the System Security Officer for lack of a better term).

B. New Rules for Access Control

A quick review of MTB 047 shows that the process attributes are called a clearance and the segment (object) attributes are called a classification. There is no change in the operation of the system when the clearance is equal to the classification for all objects a process attempts to access.

However, if the process clearance is "greater than" the object's classification, the process will not be able to pass information to the object (e.g., can't write, modify, send, wakeup, etc.).

If the process clearance is "less than" the objects classification, the process will have no access.

The access controls will have the effect of limiting the users' ability to specify the access mode on an ACL entry. That is, even though the listacl command shows

```
rew       *.SysDaemon.*
rew       *.*.*
```

on the ACL of a segment, the SDW will contain only "re" when a process of a higher clearance initiates the segment. If a process of a lower clearance than the segment attempts to initiate the segment, it will not be added to the process address space.

More succinctly, the new access control mechanism is attempting to prohibit <u>any</u> information transfer between processes which would violate these rules.

III. <u>The Problem of Shared Data Among Processes</u>

A. All system segments in ring 0 are exempt from the new access restrictions. Ring 0 is prelinked and has a static address space which is the same for all processes. Thus, ring 0 is "trusted."

B. In outer rings (1 through 7), all segments which need read-execute access for all users should have the lowest classification in the system. This will allow processes of all clearances to read and/or execute the segment (as will be the case for all system object segments).

Examples:

who table
installation parms

C. In the outer rings, all segments which are intended to have "rw *.*.*" access will cause access violations when a process of a higher clearance attempts to write in the segment. Thus, it is simply not possible for a single segment to serve as a writeable shared data base outside ring 0 for processes of all clearances. If such a data base is needed, then it will require one segment per possible process clearance.

Examples:

mailbox
user.con_msgs
command usage monitoring
ring 1 message segments - locking on read
Volume descriptor segments - if modified by a user process
ring 1 system tape data segments - if modified by a user process

D. The same problem described in "C" above will befall the user who must operate at more than one clearance using the new login options. Several "per user" segments will no longer be writeable when working at a clearance higher than the segments classification.

Examples: (these are assumed to be system low classification)

mailbox - command attempts to lock
user.con_msgs - can't store event channel
user.pmotd - can't update
user.profile - can't add abbreviations
user.breaks - can't modify

E. It is planned that each user will be able to create a process at the lowest possible clearance to perform common functions such as:

mail and message communication
program debugging
updating user.xxx segments
project and system administration

F. Exceptions to the security access restrictions will be provided for certain processes which must perform functions on behalf of other (user) processes. They are:

Initializer
I/O coordinator
Backup
Reloader/Retriever
System Security Officer

IV. The Problem of the Storage System Structure

Analysis of the security storage system model has shown that each segment of classification "N" must be contained in a directory of classification "N" and that the classification of a directory must be equal to or higher than its parent. (This means, of course, that the root and all system directories must be at the lowest possible level.) This additional restriction on the structure of the storage system introduces two potential problems.

A. The ACL mode restrictions described in section II.B above apply to directories as well. An entry of "sma *.*.*" will provide only "s" for a process with a clearance higher than the directory. Programs which attempt to verify "sma" access by looking at the ACL mode bits rather than the effective mode may be in error. This, of course, will not apply to the privileged processes (see III.F above).

When multiple classifications are in use, a program like "sweep" will not operate in a process without special privilege, since it attempts to give "sma" access to the process. In this environment, users who rely on "walk_subtree" to set access in sub-directories will fail. This will not be the case if only one classification describes the entire subtree.

B. If a program is written to provide separate "rw" data segments for each classification to avoid the shared data problem, the programmer should be aware that these segments must be in different directories!

Attempts to calculate the pathname of an "rw" data segment based on the clearance of the process must consider the directories as well or place enough links in a directory to find the segment by a calculated entryname.

V. The Problem of IPC Restrictions

Calls to hcs_$wakeup will not be successful in sending a wakeup to a process with a lower clearance.

Example:

send_message
(The write-down restriction will affect this as well)

IPC within a process will not be affected nor will any wakeups originating in ring 1.

Exceptions will be made for the initializer and the I/O coordinator so the following functions will still work:

logout
new_proc
dprint
enter_abs_request
install

VI. Design Guidlines

This section is divided into two parts since the guidelines for system functions is conceptually a little different than for user subsystems, though they will both be subject to the same access control considerations.

A. System Functions

1. Each process should be "an island unto itself." Passing information among processes other than through ring 0 data segments may create big trouble (i.e., will not work unless the clearances are equal).

2. Message segments have caused real headaches in the context of security access controls. This is because the implementation requires rw access for all processes while the ring 1 procedures provide their own access control mechanism. Similar problems may occur in ring 1 tape managment and/or vds segments for demountable disk unless we are careful in the design.

   The most basic tenet of the security design is that ring 0 is the access control kernel: future extensions to access control must be implemented in the kernel in order to work properly in a multilevel environment. The present conflict (message segments) has arisen because the access control decisions are made in ring 1, which is not trusted, nor a portion of the kernel.

This last statement may appear to conflict with the basic purpose of the ring concept (i.e., hardware enforced program controlled access). In fact, the ring concept is still valid in that an inner ring program will still be able to mediate what ever mode of access the hardware supports for the process in that ring. The only conflict occurs when outer rings are used for system functions which must span all classifications and are thus in violation of the new access control rules. This problem will require further study and we hope to stimulate a good deal of discussion and suggestions of how to solve it and still adhere to the purpose of the new access controls.

3. All system functions which are written to provide a basic capability which will be needed by all user processes must be carefully structured so that: 1) user writeable data can be segregated by process clearance, or 2) user writeable data is stored and written by ring 0 procedures only. Any other design approach will result in a basic system function being unavailable to some set of users: which could reduce the value of the system to a potential customer.

3. User Subsystem Design

The main difference between system functions and user subsystems is that the system programmer must consider a design which will be available to all processes on the system: whereas the writer of a user subsystem can be free to choose if he wants to restrict his design to a single clearance level. If a single clearance design is acceptable, he will not have to worry about the new access controls at all (short of the correct initial classification of all segments.) If a multiple clearance design is called for, the considerations for system functions will apply.