DATE: November 6, 1973

TO: Distribution

FROM: R. F. Mabee

SUBJECT: Metering tools and security

Problem statement

Hardcore data segments contain many items which are useful for measuring various aspects of system performance. They also contain some information which should be protected, such as I/O buffers. Currently all metering tools obtain their data by calling the ring-zero entry phcs_$ring_0_peek. This loophole is too general: it can be readily misused to read any ring-zero segment. This problem could be ignored when only a trusted few had access to call phcs_, but we now have on the order of a hundred people with good reasons for needing access.

The solution to this problem should make it possible to let any user read selected system data without endangering specific protected data. The set of users who are permitted to use the meters should be controllable (by an ACL) to suit individual installations.

Analysis

Due to haphazard design, some existing data segments contain both private and public data. These data should be separated into different segments because the segment is the fundamental unit of hardware access control. However, hardcore programs which use these segments may run in any user's process, so (without per-user ring brackets) the ACL's on the data segments cannot be used to discriminate among users. Therefore, access to metering data has to be controlled by an inner-ring filter program. There are two alternatives at this point:

(1) Rearrange data according to protection requirements. Let the filter program decide per-segment whether or not to permit the access.

(2) Leave data intermixed. Create a more complicated filter program which knows what parts of which segments to protect.

It seems clear that the hardcore data should be grouped according to the access restrictions to be placed on it, so that finer distinctions can be drawn between public access, metering access, system-programmer access, and highly-privileged access. Such a global rearrangement would take too long (several months) to meet the immediate need, but I strongly recommend that any new or redesigned data bases be organized in this way.

For interim use I will choose (2) above, but how should the filter program decide what to protect? Well, several people have pointed out to me that the most critical hardcore data segments (typically I/O buffers) have fixed-size headers (essentially public) followed by some variable structure of per-device or per-user blocks (essentially private). Therefore, I propose to associate with each hardcore segment number Sn a limiter L_{Sn} such that word i of the segment can be read in outer rings if and only if $0 \leq i < L_{Sn}$.

Proposed solution

I will write a procedure called real_ring_zero_peek_ to implement the filtering algorithm described above. It will be installed in ring one in the on-line libraries so that hardcore installations will not be required in order to update it. It will be accessed through a new gate into ring one named ring_zero_peek_. Ring-one programs will access ring-zero segments through a new gate into ring zero, which will function exactly as does the old ring_0_peek. All the gates will be on the system tape, until some facility for on-line installation of gates is provided.

The list of limiter values has to be in a symbolic file in the directory containing real_ring_zero_peek_ itself (>tools). It is an ASCII file containing reference names and corresponding numeric read limiters, one per line. The first time in a process that real_ring_zero_peek_ is invoked, it compiles the symbolic form into an internal static vector indexed by segment number. Any segment not mentioned in this list is not accessible.

A suggested list of limiters is attached (Appendix A). This list restricts access to the minimum required for the installed metering tools to be usable.

To avoid having to change the dozens of programs that currently call phcs_, I propose to make phcs_ a user-ring transfer vector that will reference ring_zero_peek_ or a new ring-zero gate called privileged_gate_ as appropriate. This is not an essential change, as the existing metering commands can be fixed over a period of time.

The following installed tools will cease to work, given the suggested limiter list, and will need to be modified:

copy_out, ring_zero_dump, pre_page_meters,
patch_ring_zero, lpatch, copy_salvager_output,

print_aste_ptp

Schedule

Although the proposal above is not very detailed, it covers the necessary external design well enough so that implementation can begin immediately. Coding and preliminary checkout should take two weeks, or about twenty hours of connect time. Two or three short development runs will also be needed. A reasonable target for final submission is one month from the date of publication of this bulletin.

Please feel free to make comments or ask questions about this proposal, on line to Mabee.CompSys or by phone at 253-6004.

Appendix A - suggested limits file.

```
" Only data named in this list can be read.  The limit is
" given as 262144 if some program copies the entire segment.

" Name of segment              Limit     " Reason

tc_data                        262144    " For traffic_control_queue.
sst                               512    " Used by most of the tools.
dseg                              256    " Used by most of the tools.
config_deck                    262144    " For print_configuration_deck.
active_hardcore_data               74    " Only for system_link_meters.
hcs_                           262144    " For spg_ring_0_info_ and meter_gat
hcs_.link                      262144
privileged_gate_               262144    " For meter_gate.
privileged_gate_.link          262144
hphcs_                         262144
hphcs_.link                    262144
xray_communications.link           54    " For link_meters.
pds                              2540    " For link_meters.
dsu_170_seg                       256    " For device_meters.
dsu_270_seg                       256
dsu_181_seg                       256
dsu_190_seg                       256
bulk_store_mailbox                256
fsdct                            4096    " Can't get exact figure.
tty_buf                           112    " For DClark's uninstalled tool.

" The limit on tc_data could be lowered to 256, to protect per-user
" time used figures.  tcq and print_apt_entry would then stop working.
```