

File - AF/ESD
~~proposal study~~
AF/ESD
Saltzer: Prop-
001

RECEIVED
APR 6 1973
J. H. SALTZER

2 April 1973

Prof. Jerome H. Saltzer
Project MAC
Massachusetts Institute of Technology
Cambridge, Mass 02139

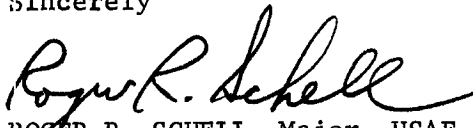
Dear Jerry

I appreciate the opportunity to review the draft proposal that you provided in your 28 December 1973 letter. My comments and thoughts are attached (Attachment 1).

We are proceeding with efforts in the area of computer security, and I have also attached additional documentation that may be of interest to you.

I also wish to thank you for the RFC documents distributed to us, and I am looking forward to continuing an open exchange of technical information.

Sincerely


ROGER R. SCHELL, Major, USAF

- 5 Atchs
- 1. Comments on Proposal
- 2. Multics Spec
- 3. Multics LTD
- 4. Case University SOW
- 5. MTP-142

Cy w/Atch 1
MITRE/Steve Lipner

Comments on Draft Proposal

In evaluating the Project MAC ARPA proposal for Research in the Certification of Computer Systems, provided by your letter of December 28, 1973 several points of interest were noted relating to the computer security work here in ESD. The proposal's justification for research into the area of security kernels as part of large operating systems is very interesting and relevant. Spread throughout the computer industry are design errors which can cause catastrophic system failures (system "crashes") or can compromise the security or integrity of stored files (both on and off line). Such design errors cannot be found through normal testing, since one is testing for the lack of a deficiency, not the presence of a well-defined function. The second area (compromise to stored files) is of particular interest to the Air Force due to the problem of the "malicious user," a user of the system who may be surreptitiously attempting to gain access to sensitive information. The "malicious user" may have large resources at his disposal and may even have cooperating agents within a software contractor's own employee structure.

The proposal identifies three distinct strategies for obtaining a reliable and secure operating system:

1. Use top-down structured programming.
2. Subject existing operating system to a proof of correctness.
3. Modify an existing system to permit line-by-line human auditing, by producing a security kernel.

While we agree with the proposal's evaluations of these alternatives -- as far as they go, none of the three strategies necessarily meets the requirement of protecting against the "malicious user" in any practical sense. Rather, a fourth strategy (which is, albeit, a modification of the third) is necessary and realizable to provide security against the active penetration agent.

The first approach, top-down programming, can be rejected for reasons described in the proposal -- complexity and excessive cost; however, some of the programming techniques associated with this approach can be very useful for implementation. The second approach, certification of an existing system, can be rejected as

far beyond the current state of the art, due to the size and complexity of existing systems.

The third approach, that of modifying an existing system to form a small and isolated security kernel, is a step in the right direction. By reducing the size of the security relevant portions of an operating system, one can more easily apply line-by-line audits to catch many errors. However, this type of auditing, while useful, may well not withstand the attack of a "malicious user." An ad-hoc design of a kernel coupled with ad-hoc audits can only say with reliability that this list of bugs have been found and fixed. It only remains for the clever "malicious user" to find the next bug and penetrate the system. An ad-hoc system of design and audits cannot assure that all possible penetration routes have been stopped. Only one penetration route is necessary for an agent to establish himself within the system permanently, even if his original entry is later found and repaired.

This brings us to the above mentioned fourth approach, that of a "certified" security kernel. As mentioned in the proposal, proofs of correctness cannot yet be applied to very large programs such as complete contemporary operating systems. However, the proposed

reduction in size of the security kernel brings the kernel into the range in which its implementation correctness with respect to well-defined functions (but not with respect to security) can be established. Thus, if the kernel is designed, not on an ad-hoc basis, but with a mathematical model of a set of primitive functions that are proven to be sufficient to provide security for all access to information, then the system can be "certified" to be secure. Security can be precisely defined in terms of user and information access attributes and individual authorization, at least as far as national security information is concerned. If the security primitives are directly implemented, then one can develop an operating system which can withstand the attacks of a "malicious user." Such a system would be demonstrated to be secure against unauthorized access to information. (We assume that the system certifiers are not agents of the "malicious user," since otherwise we have a recursively unsolvable problem.)

We believe that the technology for such a certified security kernel exists today. A soon to be published ESD working paper, "Preliminary Notes on the Design of Security Military Computer Systems" and a MITRE Technical

Report, MTR-2547, detail alternative designs for the abstract model of the security kernel. These abstract models are compatible with goal of not restricting the present functional capabilities of a system like Multics in any important way. Work is underway at the MITRE Corporation to develop a full operational security kernel for a PDP 11/45, acting as a communications processor. Work is also underway at Case Western Reserve University to develop a methodology to apply these abstract models to real systems, with Multics as a primary instance of a real system (a statement of work for this effort is attached for your information). Areas of Multics that seem to be affected include the need for system control over all stored information (viz., demountable segments) and over all operation (viz., I/O).

Thus, we feel that by using a proven mathematical model of security, one can design a security kernel for a system such as Multics, within the constraints outlined in the proposal to ARPA, which can be certified to correctly protect the security of stored files. The attached briefing summary (MTP-142) outlines some of our own activities and plans. In addition, potential improvements to Multics security are of significant interest to Air

Force users currently planning to operate a Multics system; their additional technical interests include tools for securely managing very large files and an intense interest in a capability to securely run any GCOS user program -- perhaps by extending the Multics distributed supervisor to include an efficient "GCOS supervisor" for selected users. Some of these requirements are reflected in the attached Multics procurement specification (Attachments II & III) -- please do not widely distribute these.