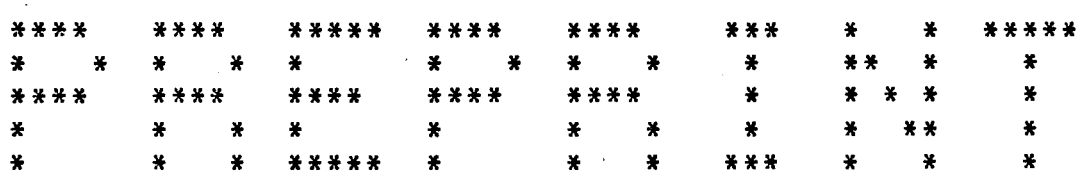
To: Distribution

From: T. H. Van Vleck

Date: February 5, 1981

Subject: Implementation of Security Concepts in a Large-Scale System

This is a copy of the paper presented by Charlie Clingen at a Cii-Honeywell Bull Security Symposium in Monaco on Jan. 28, 1981. It will appear in the proceedings of that conference. We have Cii-HB permission to reprint the paper here; however, please do not refer to this MTB in bibliographies, but instead refer to the conference proceedings.



Implementation of Security Concepts in a Large-Scale System

Charles T. Clingen and Thomas H. Van Vleck
Honeywell Information Systems
Cambridge, Massachusetts, USA

ABSTRACT. The Honeywell Multics system includes information security hardware and software features which continue to set the pace for commercial computer systems. The features represent the systematic implementation of three security models, each of which provides a set of valuable services. The resulting system is efficient, reliable, coherent, and easy to use.

This paper discusses the security features of Multics, a large-scale operating system which runs on Honeywell Level 68 computers. The security environment of Multics is the richest in function and the most studied of any commercially available system. The information protection features in Multics serve as the basis for similar features in several current operating systems. An understanding of the security features of Multics, and the design process which led to them, is therefore a valuable tool for describing and predicting directions in operating system security.

History

ORGANIZATIONS AND NEEDS

Multics was begun in 1965 as a joint project of the Massachusetts Institute of Technology, Bell Telephone Laboratories, and the General Electric Company. Each of the participants brought to the system a different approach to security problems, but all three organizations agreed that information protection was to be a major feature of the new system. Bell Laboratories and GE were concerned with protection of company assets in the form of information, and with protection of the company's ability to function, where computers were becoming increasingly important. In addition, both companies did contract work for the US Government, and so were concerned with satisfying growing government requirements for security of information.

MIT's concerns were aimed toward providing effective sharing of information, subject to individual control and responsibility. The experience provided on the Compatible Time-Sharing System at MIT's Project MAC in the years prior to 1965 showed that on-line sharing of programs and data held significant promise, permitting the exploration of new problems and new modes of man-computer interaction. Several years' operational experience with CTSS had shown, however, that academic computer users required flexible and dynamic control over what information was shared and how it was shared. In addition, several attempts by users to interfere with the operation of CTSS itself showed that the

system's integrity and accounting data bases needed the highest degree of protection, even in a relatively open environment.

DESIGN PROCESS

Security was therefore emphasized during the design of the hardware and software for Multics. All of the system implementers considered themselves responsible for security, and sought ways to enhance and assure the security of the system. Security was not considered an afterthought or an option; from the very beginning of the design process, it was intended that the system provide precise control over the sharing of information and protection of that information from unauthorized disclosure or damage.

DESIGN NOT SECRET

A key principle was established early in the design of Multics: namely, that the design of the security facilities must not be secret. The parameters of security, such as passwords, are assumed to be secret, but not the mechanisms which provide security. This meant that the designers could not count on the user's ignorance of, say, software auditing mechanisms or particular storage locations, but rather had to design a system secure even against knowledgeable programmers. Since Multics was intended to become a commercial operating system, the design had to be secure enough to protect against a would-be system infiltrator who owned a copy of the hardware and software he was trying to penetrate.

SECURITY RESEARCH

The Multics design has continued to be a hospitable framework for theoretical investigations of computer security as well. Research in various aspects of information protection has continued up to the present time at a multitude of locations, including Honeywell, MIT, the MITRE Corporation, Harvard University, and Stanford Research Institute. Some of the fruits of these research efforts will be described in this paper.

Overview of Multics

Before describing features of the Multics security mechanism, it is useful to review some of the basic concepts and terms used by Multics. (For more detailed information, see [ref].) System resources are provided to many simultaneous users. A user may be a person, logged in to the system from a remote terminal; or a user may be some generalization of this sort of use, such as the agency which writes user files to the printer on request. Each user has a process, which can be thought of as a virtual computer executing a sequence of programs. Each process has a segmented virtual memory which contains the programs and data operated on by the process. Supervisor programs and data are included in this virtual memory.

SEGMENTS

Two kinds of sharing go on in Multics: the invisible kind, like the rapid multiplexing of central processors among multiple processes demanding service, and visible sharing of resources, where the user process can observe the results of sharing. Segments are the primary entities shared in this second fashion. Segments are used for file storage for programs and data, and as directories which provide a tree-structured catalogue hierarchy. Each segment's catalogued security attributes control which users may access the segment and how they may access it.

Features of Multics Security

FLEXIBLE SHARING

It is easy in Multics to specify sharing of segments or other system resources by groups of users. The user group may be as small as a single user, or may include all users of the system, or may specify a long list of users by name.

Sharing is further controlled by specifying the mode of access for each user. Different modes are provided by the system depending on the legal operations on an object; for segments, the access modes permitted are Read, Write, Execute, and combinations of these. For directories, the modes permitted are Status, Modify, and Append.

Decentralized administration of the security attributes is another feature of the design. Each user may grant and revoke access to his own segments, within limits set by installation management. Accounting, identification, and resource management remain centralized.

The Multics protection mechanism is extensible, so that patterns of use beyond those provided in the basic system can be accommodated with appropriate patterns of protection. User subsystems which define new security rules, operations, and modes can be constructed by users, and these subsystems can themselves be shared. For example, the mail subsystem on Multics implements special objects called mailboxes, containing messages from many users. Access rights are calculated by the subsystem for each message; Multics ensures that only the mail subsystem can access a mailbox's contents.

EFFICIENT SECURITY

Throughout the implementation process, the builders of Multics were concerned with making the security implementation efficient, since inefficient operation would lead frustrated users and management to bypass security. The most important step toward efficiency of the security mechanism was to ensure that access control checks which are made frequently are made by the hardware, without requiring additional memory access. The tables which the hardware uses

to form virtual memory addresses also contain access control flags which are checked on every access. The supervisor computes correct values for the access control flags only when it needs to: several levels of hardware and software cache are employed to keep the cost of this calculation low.

RELIABLE SECURITY

The hardware which supports the Multics security implementation today combines the initial processor design with the results of an extremely thorough US Air Force study, which revealed several classes of problems; these were resolved in the design of the current hardware. In addition, hardware design changes were made to eliminate several software problems identified by the study, and to move the implementation of some security features from software to hardware.

Security Functions

Three kinds of security combine to provide the protection features the user sees in Multics. These three sets of features are integrated with auxiliary mechanisms into a system which is coherent and easy to use.

SUPERVISOR PROTECTION: RINGS

The first kind of security protects the supervisor from interference. Actions which could cause a system service interruption, or tamper with supervisor data bases such as accounting and security, are detected and prevented. If the supervisor cannot be protected, of course, no other protection mechanism can be trusted.

The Multics ring mechanism provides this intra-user protection function. The segments in each process's virtual memory are divided into groups called rings, and the processor contains a register which specifies the current ring number. Lower numbered rings are more privileged, with ring 0 used for the most privileged part of the supervisor. When the system is executing in a given ring, data in lower numbered rings is inaccessible, while that in the same or higher numbered rings is available. This arrangement is a straightforward generalization of the distinction between master mode and slave mode in traditional operating systems.

The ring mechanism not only protects the supervisor from the user, it protects one part of the supervisor from another and can protect one set of the user's programs from another. This extension supports modern layered definition of operating systems and allows the supervisor and user to build complex objects out of simpler types.

The Multics ring mechanism is actually somewhat more complicated than described above, since it includes provision for making procedures and data shareable between rings, provides automatic validation of pointers to protect inner rings from misuse of their privilege,

and has arrangements for detecting and managing changes in the processor's ring of execution. The interested reader may find more detail in [ref].

INTER-USER PROTECTION: ACLS

The controlled voluntary sharing associated with objects like segments and disk packs is provided by the second kind of security, which provides inter-user protection. Operations by processes on segments, directories, reels of tape, and so forth are partitioned into legal and illegal actions, and the illegal ones are prevented.

Each object protected in this fashion has an Access Control List (ACL) associated with it, which is maintained in the directory entry for the object. The ACL of a segment, for example, might look like this:

```
rw     Smith.FED
r      LJones.*
null   Brown.*
rew    *.MMPP
r      *.*
```

When the system is determining the appropriate mode of a segment with respect to a user, it searches the ACL from most specific to least. The resulting mode is inserted into the virtual memory control tables so that the hardware will check access on every operation.

CONTROL OF SHARING: AIM

The third kind of security applies when users are not completely trusted. It provides control over disclosure of information, which limits the discretion of the user. Military security, in which data classified as secret may not be given to someone lacking a clearance, is an example of the kind of policies the third kind of security implements.

The Access Isolation Mechanism (AIM) facility provides this third type of security by associating a classification with each object and an authorization with each user. The rules enforced by AIM prevent a user of a given level of authorization from reading data of higher classification, or writing data of lower classification, regardless of permission granted by ACL.

If this third type of security is not required at a Multics site, it need not be activated; in such a case it is not visible to the system's users and imposes no extra cost. Only a few of the current Multics sites use this facility.

All these types of security: rings, ACLs, and AIM, are simultaneously active for every segment reference. Only if all these mechanisms grant permission can a user process reference information.

EASE OF USE

This formidable collection of security features may give the impression that Multics users do little but worry about information security. In fact, care has been taken so that the security features are not obtrusive, and need concern

only those who benefit from them.

Considerable human engineering has gone into the identification interface, which the user first sees when connecting a terminal to the system. The system's registration files hold per-user default values for most parameters, so that logging in to Multics does not require lengthy parameter specification. The values of user attributes are controlled by a decentralized administration mechanism, which allows multiple administrators the ability to specify or delegate the ability to specify user attributes.

The user who wishes to use Multics without sharing information need not take any explicit action to protect his programs and data from other users. Default values for the security parameters of objects he creates provide full privacy (unless administrators have specified otherwise). As the user begins to interact with other users, sharing programs and data, he needs to learn more and more about the kinds of security Multics supports, and the commands which implement them. In this area also, care has been taken to make the application of the system to simple cases simple itself.

IDENTIFICATION AND ADMINISTRATION

The Multics security system provides a number of ancillary features which do not themselves improve security but which assist users and administrators in managing and dealing with the protection mechanisms.

Audit trails are preserved by the system so that security problems can be detected and analyzed. A selective facility allows the administrator to control what events and which users are monitored, and prepares daily reports on security threats. If a user attempted to find an exploitable weakness in the hardware security, he would have to try all the illegal actions to see which ones succeeded. This search would leave its traces in the software logs and alert installation management.

Passwords for each user are assigned on a per-person basis. A user may change his password at any time; the system will generate a password for the user if he does not wish to supply one. Control is also provided to require a user to change his password at designated intervals. If a user forgets his password, the system administrator cannot look it up for him, since only the result of a one-way encryption algorithm is stored. In a situation like this, the system administrator may force the user's password to a new value, which the user may then change at first login.

Finally, the user is warned of suspicious events, and given the opportunity to help with his own protection, by the provision of system messages telling him the date, time, and terminal identifier of his last login. Other messages warn the user that his password was given incorrectly

from a specified terminal, or that more than one instance of the user is logged in.

Modeling of Multics Security

The initial design of Multics security features was mostly empirical, and aimed at curing the problems of predecessor systems. Government and academic interest in Multics, and in the theory of computer security, led to several theoretical studies of the system's security; the result of these studies was not only a theoretical basis for understanding computer security, but also a practical understanding of the weaknesses of the prototype of Multics which was then available. Changes were made to the processor architecture and the operating system software to take advantage of these insights. The resulting system has been acknowledged to be the best available: a 1975 MITRE study described Multics as "... superior in both hardware and software."

The theoretical studies of security focused on a two-stage mapping, first from real-world policies, rules, or laws into a model, and then from the model into a computer system implementation. The first step, that of representing the required behavior within the model and of proving properties of the security implementation by mathematical deduction, has been thoroughly studied for Multics by SRI and MITRE. The second step, the verification that a security model has been correctly mapped into programs and hardware, is an area of current computer science research. This verification has not yet been performed for Multics; it awaits the availability of practical mechanizations of the verification process.

Benefits of Multics Security

The Multics security mechanisms we have described here, and their derivation from well-studied security models, provide the user and system administrator with several benefits.

First, Multics security provides insurance against several kinds of security threat. Malicious threat, in which a deliberate attempt to compromise the organization's security policy, is the most obvious. But accidents can also occur which might disclose confidential information or perform incorrect modification of data; the security system protects against many of these occurrences too. In addition, the security system prevents the propagation of system or machine error.

Second, it is easy to choose from the facilities provided by Multics in order to implement an organization's security policy; this match between the system's abilities and the organization's needs is important for effective use of large-scale computing.

Conclusion

It is important to understand what kind of security is

claimed for Multics. Nobody claims that the system, as implemented today, can be proven secure against all possible threats. However, we do believe that the design framework and current implementation of Multics security provide the best security features in a commercially available general purpose operating system.

Sixteen years' work by a dedicated and able team on the implementation of Multics, in parallel with sixteen years of progress in computer science research, have provided us with a coherent, rigorously based, and easy to use security implementation. Future progress in large operating systems will include the wider acceptance of the design techniques and system features now available in Multics, coupled with further steps toward rigorous proof of security properties and verification of their implementation.

REFERENCES

Anderson, J. P., Computer Security Technology Planning Study, Deputy for Command and Management Systems, HQ Electronics Systems Division, US Air Force, L. G. Hanscom Field, Bedford, MA (October 1972).

Bell, D. E., and L. J. LaPadula, Secure Computer Systems: Mathematical Foundations and Model, MITRE Corp., Bedford, MA (September 1974).

Bensoussan, A., C. T. Clingen, and R. C. Daley, The Multics Virtual Memory: Concepts and Design, Comm. ACM 15, 5 (May 1972) 308-318.

Biba, K. J., S. R. Ames, Jr., E. L. Burke, P. A. Karger, W. R. Price, R. R. Schell, W. L. Schiller, A Preliminary Specification of a Multics Security Kernel, WP-20119, MITRE Corporation, Bedford, MA (April 1975).

Daley, R. C., and J. B. Dennis, Virtual Memory, Processes, and Sharing in Multics, Comm. ACM 11, 5 (May 1968), 306-313.

Graham, R. M, Protection in an Information Processing Utiltiy, Comm. ACM 11, 5 (May 1968), 365-369.

Karger, P. A., and R. R. Schell, Multics Security Evaluation: Vulnerability Analysis, ESD-TR-74-193, Vol. 2, Electronic Systems Division, USAF (June 1974).

Lipner, S. B, Computer Security Research and Development Requirements, MTP-142, MITRE Corp., Bedford MA (February 1973).

Neumann, P. G., L. Robinson, K. N. Levitt, R. S. Boyer, and A. R. Saxena, A Provably Secure Operating System, Stanford Research Institute, Menlo Park, CA (June 1975).

Neumann, P. G., R. S. Fabry, K. N. Levitt, L. Robinson, and J. H. Wensley, On the Design of a Provably Secure Operating System, Proc. Workshop on Protection in Operating Systems, IRIA, Rocquencourt, France, pp. 161-175 (August 1974).

Popek, G. J., and C. Kline, The Design of a Verified Protection System, Proc. Workshop on Protection in Operating Systems, IRIA, Rocquencourt, France, pp. 183-196 (August 1974).

Saltzer, J. H., Protection and the Control of Information Sharing in Multics, Fourth ACM Symposium on Operating Systems Principles, Elmsford, NY (October 1973).

Schell, R. R., P. J. Downey, and G. J. Popek, Preliminary Notes on the Design of Secure Military Computer Systems, Computer Security Branch, Directorate of Information Systems Technology, Deputy for Command and Management Systems, HQ Electronics Systems Division, US Air Force, L. G. Hanscom Field, Bedford, MA (August 1972).

Schiller, W. L., K. J. Biba, and E. L. Burke, The Top Level Specification of a Multics Security Kernel, WP-20377, MITRE Corporation, Bedford, MA (August 1975).

Schroeder, M. D., and J. H. Saltzer, A Hardware Architecture for Implementing Protection Rings, Comm. ACM 15, 3 (March 1972), 157-170.

Schroeder, M. D, Engineering a Security Kernel for Multics, ACM SIGOPS Review 9, 5 (November 1975) 25-32.