SHARING COMPUTER RESOURCES SECURELY

Michael J. Grohn, Project Manager
I.P. Sharp Associates Limited
Ottawa, Ontario
K1S 2E1

ABSTRACT

The concurrent sharing of computer resources by a community of users can result in security problems such as the unauthorized disclosure and the unauthorized modification of data processed by a system. Techniques for identifying and addressing these problems are described, and their applicability to general-purpose time-sharing systems is discussed. The emphasis is on internal (automatic) access controls, involving thorough user authentication, special user interfaces, and "front-end" and "back-end" processors.

LE PARTAGE DE RESSOURCES D'ORDINATEUR EN SECURITE

RESUME

Le partage commun de ressources d'ordinateur dans une communauté d'opérateur peut causé des problèmes de sécurité, par exemple divulgation de secret de reproduction. Nous analysons les techniques par lesquelles nous pouvons identifier ces problèmes et les corriger dans une situation ordinaire de partage de ressources. Nous voulons souligner l'importance d'un système d'accès interne et controlé qui comprend un système d'indentification authentique et un processus "front-end" et "back-end".

SHARING COMPUTER RESOURCES SECURELY

Michael J. Grohn, Project Manager
I.P. Sharp Associates Limited
Ottawa, Ontario
K1S 2E1

ABSTRACT

The concurrent sharing of computer resources by a community of users can result in security problems such as the unauthorized disclosure and the unauthorized modification of data processed by a system. Techniques for identifying and addressing these problems are described, and their applicability to general-purpose time-sharing systems is discussed. The emphasis is on internal (automatic) access controls, involving thorough user authentication, special user interfaces, and "front-end" and "back-end" processors.

1.0 INTRODUCTION

This paper describes some of the research work and analysis resulting from a computer security study contracted by the Department of National Defence and performed at I.P. Sharp's Ottawa office. Computer security is an important aspect of resource sharing since such issues address problems such as: guaranteeing that only the sharing which is desirable is possible; maintaining the privacy of computerized information; and ensuring the integrity (correctness) of automated information management.

Basically, computer security involves protection against certain undesirable events in the realm of computer technology. For the purposes of this paper such undesirable events can be categorized as unauthorized disclosures of information, or unauthorized modifications of information, or denial of service by the system.

Such events can result from accidental or deliberate causes, and can involve a variety of techniques (e.g. collusion to obtain non-shredded discarded documentation).

A number of orientations to providing protection from such threats is possible. In provable security, entire system designs are affected by security considerations. Mathematical proofs of security are possible when techniques such as "kernelization" (Schiller 1975) are used. In risk analysis, the probabilities and costs of certain events are determined, so that the most significant problems can be identified and solved. In the phylaxis (Grohn May 1978) approach, the effect on security of various enhancements is measured (i.e. analyzed).

1.1 Background

Information sharing by computerized means has certain definite characteristics in the military environment. Sets of data (files) are assigned classifications (e.g. SECRET, TOP-SECRET), and users are assigned clearances to access data of certain classifications. A well defined security policy (set of rules) exists which specifies exactly how classified information is to be shared. These rules state those requirements allowing the disclosure of information, those requirements allowing modification, those requirements for the accumulation of information, and requirements for extending and revoking authorization.

Since the information managed by military computers can relate to national security, ensuring that computer systems conform to these requirements is very important in the military environment.

1.2 The Basic Problem

The main problem that this paper will address has often been referred to as "the multi-level data sharing problem". It is the situation where data of a variety of security classifications ("multi-level") are all found on the same computer system. The classification of each set of data and the clearance of each user must be correctly maintained by the system internals throughout the life cycle of automated information management (i.e. from file creation to archiving). In order that internal (software) access control mechanisms allow only authorized access to data, they must be correctly implemented, tamperproof, and non-bypassable.

Another aspect to the problem is that a lack of user confidence in system security can result in the over-classification of much information. This loss of availability of certain information due to overly secure handling can be highly undesirable. Centralized control and monitoring of system security is required to ensure that all security activity is consistant, complete and effective. Also, certain threats of inappropriate information accummulation may not be recognizable without centralized monitoring.

2.0 APPROACH

In order to achieve useful results the study was organized into a number of steps. First, the relevant assets (Table 2.1) were identified. These are the things of value in themselves (e.g. data), or of value as "tools" (e.g. programs). Next, the threats (Table 2.2) were identified. These are undesirable events which might occur given certain circumstances. The scope of the effort which was reasonable for the resources available was established.

As the result of "brainstorming" the project team proposed a number of security enhancements. Although their security properties could not be mathematically proven (as with kernelization), the team's experience suggested that the mechanisms proposed would improve the secure nature of systems utilizing them. The general (system independent) nature of the security enhancements was specified rigorously by means of a formal specification language (Robinson 1977). An on-line interpreter was used to check the syntax of the specification.

Case studies were undertaken for some specific computer systems. For each case study the system dependent features of the various security enhancements were established, and the implications of their usage were analyzed. Their operation was simulated in APL (the interactive language of the I.P. Sharp time-sharing service). Claims that these security enhancements embody the military security policy, are reasonable and implementable, and increase system security were justified.

CATEGORY OF ASSET	SUB-CATEGORY
PERSONNEL	Users
	Application programmers
	System programmers
	Operations staff
	Non-users
HARDWARE	Terminals
	Communications lines
	CPU/Main memory
	Peripheral memory
	Peripheral devices
SOFTWARE	Communications protocol
	Operating System
	Time-sharing System
	Data Base Management System
	Applications programs
DATA	User and test data
	Backup/recovery data
SERVICE	CPU and channel time
(Availability of resources)	Main and peripheral memory space
DOCUMENTATION	

Table 2.1 A List of Assets

CATEGORY OF THREAT	SUB-CATEGORY
HARDWARE	Emanations Wire-tapping
SOFTWARE	"Bug" "Loop-holes" in design "Trojan horses" Covert channels (e.g. response time)
DATA	Aggregation Inference
PERSONNEL	Bribery Collusion
DOCUMENTATION	Manuals Micro-fiche Program listings Terminal hard-copy
ENVIRONMENT	Building (e.g. earthquake, fire) Electricity Water
PROCEDURES (the ways in which assets are used)	

Table 2.2 A List of Types of Threats

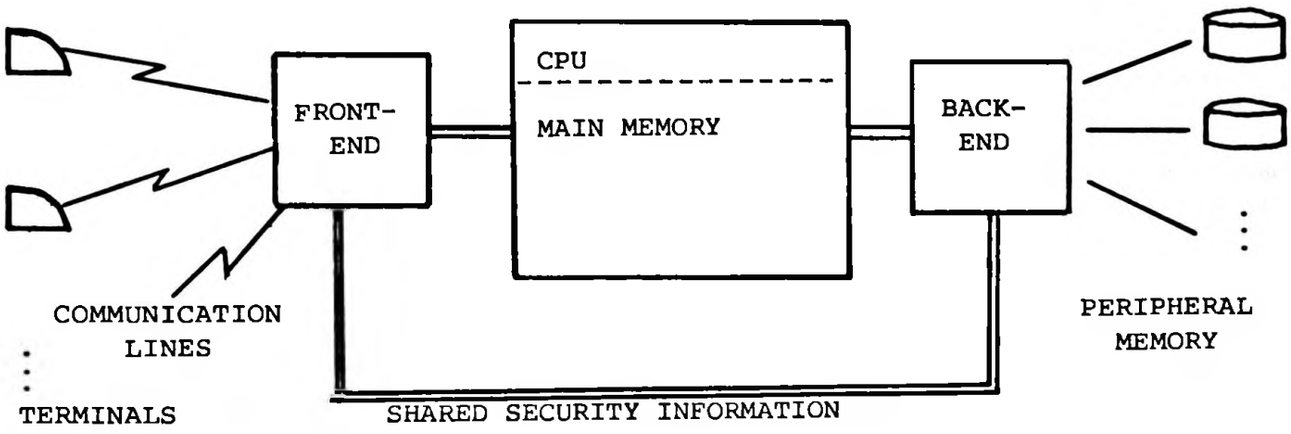

Figure 2.1 Front-end and Back-end Processors

2.1 The Scope of the Effort

Establishing a realistic scope for the study involved making a number of decisions. Consideration was limited to "general-purpose" time-sharing systems and single processor configurations of hardware. Therefore, the complications of networking and distributed processing were avoided. It was assumed that the assets are provided with physical security (e.g. a locked computer room). This includes encryption devices at each end of any communications line which may be used for classified data processing, yet is not shielded nor physically secure (e.g. public telephone lines).

No changes to the operating system software would be allowed, so as not to lose vendor support. The main objective was considered to be the exploration of all possible ways of enhancing secure processing by the system by means of: a "front-end" processor, implemented like a communications controller between the terminals and the computer; and a "back-end" processor, implemented in such a way as to allow it to monitor input/output commands to peripheral memory devices, prohibiting illegal ones.

This use of front-end and back-end processors is attractive because their implementation (Fig. 2.1) should be more manageable (when compared to modifying the operating system) and their potential for enhancing security appears great. Additionally, this technique is consistent with efforts to define standard I/O channel interfaces for computer systems (NBS 1976).

3.0 PROPOSED ENHANCEMENTS

3.1 The AUTHENTICATOR

The AUTHENTICATOR is the module in the front-end which verifies the identity of each user before he is allowed to sign on to the system. Additionally, this module scans user commands for SIGNOFF requests, and is made aware of unexpected disruptions of communications. Its main responsibility is to maintain, for each terminal, its "maximum" clearance, the identifier of its current user and his current session clearance.

Three techniques which may be used in terminal user identification (assuming appropriate devices are made available in each terminal) have been identified. The first is querying user "knowledge". An example of this is a password scheme. This information is readily changeable by the user, and should be easy for him to remember. However, there is a threat of disclosure when passwords are carelessly handled.

The second is requiring presentation of an item in the user's possession. An example of this is a magnetized identification card. An advantage is that it is visible to persons such as security guards. However, there is a threat of loss.

The third technique is obtaining a digitized representation of a user's personal characteristic. An example of this is an analysis of a user's signature. An advantage is that such a characteristic cannot be duplicated by another person. However, identification devices still present the threat of rejecting legitimate users (Forsen 1977).

3.2 User Interface

Since the time-sharing system presents the same interface of commands to unclassified users as to top-secret ones, it is desirable that the system security officer (see § 3.5) have the ability to restrict the commands (Fig. 3.1), the programs (Fig. 3.2), and the files (Fig. 3.3) which each user may use. These figures illustrate the structures involved in the internal (software) access control, whose contents are set by the system security officer. Figure 3.4 illustrates security classifications which define "implicit" access permission according to the (military) security policy.

command

user	Access	Append	Run	...
Joe	No	Yes	Yes	
Mary	Yes	No	Yes	

⋮

Figure 3.1 Table of user command restrictions

program

user	PROG1	PROG2	PROG3	...
Joe	No	No	Yes	
Mary	Yes	No	Yes	

Figure 3.2 Table of user program restrictions

files / users	FILE01	FILE02	...
Joe	READ	WRITE	
Mary	NONE	READ	

Figure 3.3 Table of explicit permissions

Users	Clearances
Joe	CONFIDENTIAL
Mary	TOP-SECRET
⋮	

Files	Classifications
FILE01	UNCLASSIFIED
FILE02	SECRET
FILE03	CONFIDENTIAL

Figure 3.4 Lists of assignments of security clearances and classifications

3.3 An Activity Log

The activity log is a sequential set of records each of which represents the occurrence of a certain event during the operation of the system. One important purpose of this log is to capture information suitable for later off-line analysis by the system security officer. A possible format for each record in the log is illustrated in Figure 3.5.

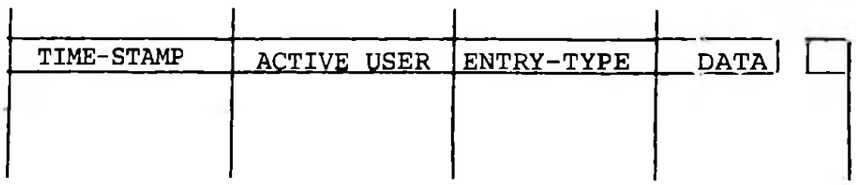

Figure 3.5 A possible format for the activity log

A function is provided which may be invoked at those points in the front-end and back-end software where the recording of activity is desired. Each use of the logging function should use a unique code for the "entry-type" parameter, to unambiguously identify the source of each entry in the log.

A complete activity log module should provide software/firmware for the secure recording of activity log records to some storage medium, and programs to analyze the off-line data. Examples

of actions which should be logged are authentication failures, submission of user commands, and attempts at unauthorized access.

3.4 Mediation

"Mediation" means checking the authorization of a given data transfer, and prohibiting it if it's illegal. The mediation of peripheral memory input/output commands is illustrated in Figure 3.6 and involves a number of highly technical considerations.

The back-end must be implemented in such a way that the I/O commands to each controller can be intercepted and suspended. The authorization of each suspended I/O command is then checked. Authorized I/O commands are released to their respective controllers. Authorized data transfers should occur from the storage device directly to main memory (bypassing the back-end). Any I/O error must be recognized.

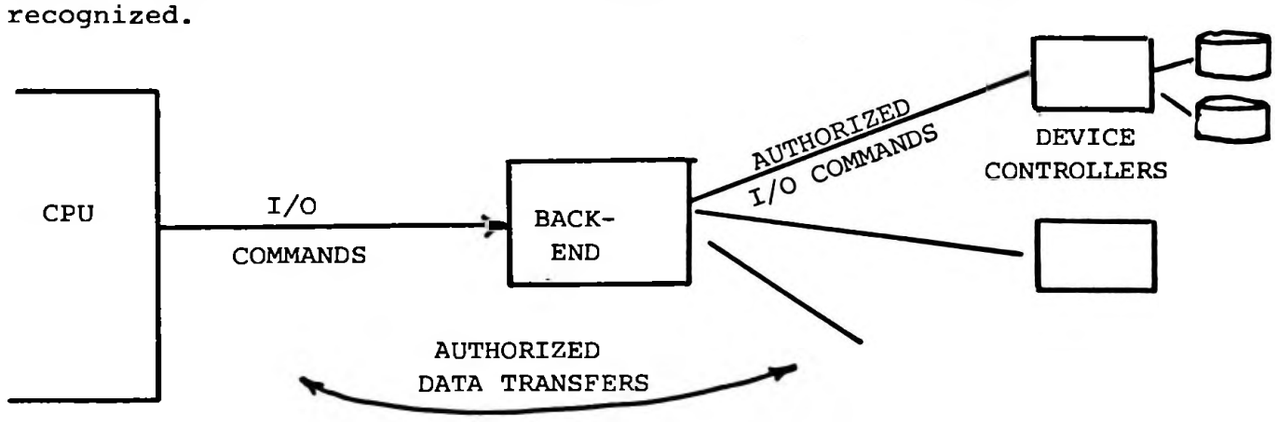

Figure 3.6 Mediation of I/O Commands

There are several benefits of I/O command mediation. A specific security policy (as opposed to the generalized file system policy) can be implemented to govern file access. Confidence in the operating system's file management should increase since the correctness of all I/O commands is checked. Logging and analysis of I/O activity is made possible.

3.5 Security Officer Functions

The security officer is responsible for all security-related activities and mechanisms which are found in the system. All authorizations to use commands, run programs and access files must be consistent, complete and appropriate at all times. User identifiers must be added and removed from the system from time to time. User and system activity must be monitored.

In certain systems where the value of the data and the threats to them are both high, the security officer may be required to preallocate all user files, extend and revoke all explicit access permissions to user files, and even exclude certain operating system modules from a SYSGEN (system generation).

All such activities will be performed from a special terminal (preferably with hard-copy capability) which is physically secure in a restricted area (e.g. computer room).

4.0 CONCLUSIONS

The security enhancement techniques which have been described in this paper provide a means of embodying military-standard security in an "off-the-shelf" computer system without modifying it. It should be possible to implement the front-end to be practically transparent to the users. The back-end processor is more involved since its implementation will depend on a computer's hardware architecture and type of I/O processing.

The reasonableness and implementability of the specifications were illustrated by the simulation of "normal" system interaction by a number of concurrent users. A simulation of the subsequent operation of the security enhancements, by means of an APL model (Grohn, Brans and Royds 1979) was done.

Several areas of difficulty for these security techniques were revealed. It could prove difficult for the front and back-ends to correctly interpret the results of actions in the host system based on host responses to terminals. Certain system features may have to be sacrificed when the front-end does not have enough information available to monitor their security. Executable command files and a batch subsystem could pose security risks.

An important by-product of this project was to produce expertise in the area of "internal" (software) system security, which will prove valuable in the construction of network and distributed processing systems.

REFERENCES

1. Forsen, G.E., Nelson, M.R., Staron, R.J., Personal Attributes Authentication Techniques, Pattern Analysis and Recognition Corp., RADC-TR-333, Rome, New York, October 1977.

2. Freeman, D.E. and Perry, O.R., I/O Design, Data Management in Operating Systems, Hayden Book Company, Inc., Rochelle Park, New Jersey, 1977.

3. Grohn, M.J. and Pase, W.J., Computer Protection Modelling, Report 3836-1, I.P. Sharp Associates Limited, Ottawa, Canada, May 1978.

4. Grohn, M.J. and Pase, W.J., Enhancing Computer Security, Report 3836-2, I.P. Sharp Associates Limited, Ottawa, Canada, September 1978.

5. Grohn, M.J., Brans, N.A., Royds, W.G., Designing and Specifying Computer Security Enhancements, Report 3836-3, I.P. Sharp Associates Limited, Ottawa, Canada, March 1979.

6. NBS, Draft Proposed American National Standard Specifications for I/O Channel Interface, X3T9/600 Rev. 2, Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, D.C. 20234, August 1976.

7. Neumann, P.G., et al, A Provably Secure Operating System: The System, Its Applications and Proofs, Final Report, Stanford Research Institute, Menlo Park, CA 94025, February 1977.

8. Robinson, L. and Roubine, O., SPECIAL - A Specification and Assertion Language, Technical Report CSL-46, Stanford Research Institute, Menlo Park, CA 94025, January 1977.

9. Schiller, W.L., The Design and Specification of a Security Kernel for the PDP-11/45, ESD-TR-76-69, A 011 712, The MITRE Corporation, Bedford, Massachusetts, March 1975.