DATA SECURITY AND DATA PROCESSING

Professor Stuart E. Madnick
Project Supervisor

REPORT CISR-8
SLOAN WP-771-75
March 3, 1975
DATA SECURITY AND DATA PROCESSING

Professor Stuart E. Madnick
Project Supervisor

ABSTRACT

In the 1972 the Center for Information Systems Research (CISR), in association with the MIT Information Processing Center (IPC), became one of four study sites participating in the IBM Data Security Study. CISR's primary area of investigation was user requirements for security. The study was completed in early 1974 and the findings of all of the study sites were published as a series of reports from IBM (G320-1370 through G320-1376) in June 1974.

CISR faculty and staff contributed nine of the twelve papers which constitute "Volume 4, Study Results: Massachusetts Institute of Technology (G320-1374)." Seven of these papers are reproduced in this report.

CONTENTS

<table>
<thead>
<tr>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER REQUIREMENTS SURVEY</td>
<td>5-23</td>
</tr>
<tr>
<td>DATA SECURITY IN UNIVERSITY ENVIRONMENTS</td>
<td>35-70</td>
</tr>
<tr>
<td>DATA SECURITY IN THE SERVICE BUREAU INDUSTRY</td>
<td>71-90</td>
</tr>
<tr>
<td>DATA SECURITY IN THE MEDICAL COMMUNITY</td>
<td>91-112</td>
</tr>
<tr>
<td>DATA SECURITY AND THE FINANCIAL COMMUNITY</td>
<td>113-133</td>
</tr>
<tr>
<td>APPLICATION AND ANALYSIS OF THE VIRTUAL MACHINE</td>
<td>*</td>
</tr>
<tr>
<td>APPROACH TO INFORMATION SYSTEM SECURITY AND ISOLATION</td>
<td></td>
</tr>
<tr>
<td>THE STATE OF PRACTICE OF COMPUTER SECURITY</td>
<td>163-178</td>
</tr>
<tr>
<td>THE SECURITY DECISION</td>
<td>197-221</td>
</tr>
<tr>
<td>COMPUTER AND DATA SECURITY: A COMPREHENSIVE ANNOTATED BIBLIOGRAPHY</td>
<td>**</td>
</tr>
</tbody>
</table>

* This paper has been previously published as CISR Report CISR-2.
**This paper is available as a separate report from CISR or Project MAC (as Report MAC-TR-122).
# USER REQUIREMENTS SURVEY

Torben G. Grønning  
Peter F. Hollings  
Stuart E. Madnick  
Susan E. Rudd  
Grant N. Smith  
Bruce E. Stangle

## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 INTRODUCTION</td>
<td>7</td>
</tr>
<tr>
<td>1.1 SUMMARY OF GENERAL FINDINGS</td>
<td>7</td>
</tr>
<tr>
<td>1.2 COMMUNITIES STUDIED</td>
<td>7</td>
</tr>
<tr>
<td>2.0 GENERAL HYPOTHESES</td>
<td>8</td>
</tr>
<tr>
<td>2.1 SECURITY AWARENESS</td>
<td>8</td>
</tr>
<tr>
<td>2.2 SECURITY AWARENESS RELATED TO JOB FUNCTION</td>
<td>9</td>
</tr>
<tr>
<td>2.3 SECURITY AS A FUNCTION OF INDUSTRY</td>
<td>10</td>
</tr>
<tr>
<td>2.3.1 Financial Community</td>
<td>10</td>
</tr>
<tr>
<td>2.3.2 Medical Community</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 Education Community</td>
<td>13</td>
</tr>
<tr>
<td>2.3.4 Service Bureau Community</td>
<td>13</td>
</tr>
<tr>
<td>2.4 IMPLICATIONS OF FINDINGS</td>
<td>13</td>
</tr>
<tr>
<td>3.0 SPECIFIC FINDINGS FROM INDUSTRY SURVEYS</td>
<td>14</td>
</tr>
<tr>
<td>3.1 FINANCIAL COMMUNITY</td>
<td>14</td>
</tr>
<tr>
<td>3.2 MEDICAL COMMUNITY</td>
<td>16</td>
</tr>
<tr>
<td>3.3 SERVICE BUREAU COMMUNITY</td>
<td>17</td>
</tr>
<tr>
<td>3.4 EDUCATIONAL COMMUNITY</td>
<td>18</td>
</tr>
<tr>
<td>4.0 TRENDS IN COMPUTER SECURITY</td>
<td>18</td>
</tr>
</tbody>
</table>

Massachusetts Institute of Technology  Volume 4
1.0 INTRODUCTION

The past decade has seen the rapid development and proliferation of computers in organizations of all sizes and functions. Often, as this growth proceeded, security was not considered in the system design stages. However, recently it has been recognized that computing systems are easily compromised. This is especially true since most security systems have evolved on an ad hoc basis with patches made in elements of the system to thwart any perceived weaknesses. One objective of this study, therefore, was to evaluate the impact and implications of this practice of ad hoc security provision.

Preliminary evaluation of a cross section of current systems revealed a number of issues that warranted further study. These included, for example, an identification of those who perceive a need for security and the differential effects of alternative technological configurations and organizational roles on this perceived need. Data was then collected on these and other variables by pencil-and-paper questionnaires and by personal interviews.

1.1 SUMMARY OF GENERAL FINDINGS

After analyzing this data, we conclude that:

* Presently, only a small proportion of computer users use security features. These users who do require security are also those who are aware of potential security problems.

* In organizational settings, those individuals who are close in job function to the computer system tend to be more aware of potential security problems.

* Significant differences exist in the levels of awareness in the various user communities. These industries use different approaches to security because of the nature of processing and the perceived value of the information being processed.

This section presents a reasonable structure for analyzing the security issue: to define problems, to validate hypotheses and, finally, to discuss the implications of this research for the future of the computer industry.

1.2 COMMUNITIES STUDIED

To test the empirical validity of our hypotheses, we chose four distinct computer communities—the medical, financial, educational, and service bureau industries. We believe that these four areas are a reasonable sample of the real world. Extensive field interviews were undertaken to determine the state of the art in computer utilization and, more particularly, in the development and usage of security systems. The organizations that were visited displayed a wide range of technical and administrative capability and sophistication. (The study teams received the fullest cooperation at all sites; we are grateful for this cooperation.)

After the field interviews were completed, questionnaires were developed to test the various hypotheses. Because of differences in
the nature of processing, a different questionnaire was required for each industry. The same questionnaire was used for organizations within a given industry. The questionnaires were prepared so that they could be completed by both technical and nontechnical managers. In each organization, as many subjects as possible were included.

2.0 GENERAL HYPOTHESES

Our research led to the development of both general and specific hypotheses. The former are testable assumptions that cut across all of the industries that were sampled, whereas the latter are those that seem to hold true only in one particular setting or industry. Specific hypotheses will be examined more carefully later in this report.

2.1 SECURITY AWARENESS

Our research supports the following general hypothesis:

H1: The security demanded by a user depends on the user's awareness of security threats.

We have observed that the degree of a person's concern for security depends on that individual's awareness of security threats and vulnerabilities. People perceive the instigators of security violations as mirrors of themselves. If they know thousands of ways to subvert their system's security, they assume that their "enemies" are equally knowledgeable. Conversely, if they are not aware of any defects in their security system, it is assumed that the system cannot be penetrated. For example, our survey of the service bureau industry indicated that numerous security techniques are available. Furthermore, the potential for security threats is high, because of the number of different organizational users and the proprietary nature of their data. Nevertheless, of all the available techniques, only a handful are actually used and these are used by the most sophisticated users. The majority of users assume that the computer system is secure and that they are adequately protected. Only among the users with a high awareness of the probability of security breaches does one find a high usage of security techniques.

Consider the implications of this hypothesis. If awareness merely depended on advertising and explicit education, then the computer industry could control the user's requirements for security by a restrictive advertising and educational policy. However, other sources of awareness exist. Thus, the computer industry can increase awareness by advertising and education, but they cannot unilaterally stifle awareness. Other factors, such as legal regulation and public concern, may become significant. Recently, major changes in producer behavior have occurred through these routes of influence, for example, consumerism.

After receiving all the evidence, we believe that awareness is increasing rapidly. At least three factors are involved:

- **Breadth of Exposure**: Continual press coverage of the entire area of computers, privacy, and confidentiality -- emphasis is often placed on harnessing this technology to serve the needs of mankind;
• **Increases in Personal Contact:** The pervasive nature of computers within an organization -- more and more people are in direct contact with the computer system;

• **Economics:** As the trend in costs moves from hardware to software and personnel, the normal arguments that security costs too much will diminish.

### 2.2 SECURITY AWARENESS RELATED TO JOB FUNCTION

The first hypothesis can be extended further by stating:

\[ H2: \text{In an organization, an individual's proximity (in terms of job function) to the computer system influences his awareness of security as a problem.} \]

In some of our questionnaires we asked the following three questions:

1. **How would you describe your exposure to and/or use of the company's computer system?**

   - Extensive 5
   - 4
   - 3
   - 2
   - 1 Negligible

2. **Generally, how would you describe your personal concern for such things as security leaks, bugging, and the invasion of privacy?**

   - Extremely Concerned 5
   - 4
   - 3
   - 2
   - 1 Not Concerned

3. **How would you rate the company's present system with the security of computerized information?**

   - Excellent 5
   - 4
   - 3
   - 2
   - 1 Poor

The answers to these questions indicated some very interesting general conclusions. We found that those who had direct contact with the computer system were significantly more concerned about security both in general and in their organization. User department managers and of the systems staff responded in the same way. In other words, this group (that is, high sophistication of use) answered the questions near the extremes of "extensive," "extremely concerned," and "poor." They could see the pitfalls in the system. The other group of managers who had little or no exposure to the computer system fell at the other end of the spectrum in their answers. This group (naïve use) answered the questions near the extremes of "negligible," "not concerned," and "excellent." These users had a very low awareness of the problem of security.

Based on a review of the psychological literature of attribution theory, our belief that systematic differences exist in user concern with security is, in fact, well-supported. Psychologists believe that individuals evaluate the motivations and behavior of others by attributing to them their knowledge, values, and feelings. Consequently, an individual who is exposed to the computer system knows how the system can be compromised; this knowledge can influence his attribution process.
This finding and the supporting theory indicate that further effort on developing a general theory of user behavior, user awareness, and user expectation is warranted. For example, if an organizational chart can be used to evaluate levels of security awareness within a firm, its use may be a prerequisite to answering such questions as:

- Is it in the interest of the company to extend awareness to other people and groups within the organization?
- Also, is it in the interest of the computer manufacturer to raise the security awareness levels within a firm?

2.3 SECURITY AS A FUNCTION OF INDUSTRY

Our final general hypothesis has evolved from a comparison of our findings in the financial, medical, service bureau, and educational institutions.

H3: Differences in levels of security awareness and in approaches to the security problem depend on the nature of processing in an industry and the perceived value of the information being processed.

This hypothesis clearly follows from the two preceding hypotheses because it expands on the determinants of awareness, but also accounts for some of the differences that we have noted when comparing user communities. Throughout this discussion, refer to Exhibits 1 and 2 for the definition of data security, some common threats to data security, and means for protecting a computer system from such threats.

To illustrate this hypothesis, we will discuss some of our observations on the nature of processing and the value of information in the four user communities. Some of the dimensions that characterize the nature of processing are shown in Exhibit 3.

2.3.1 Financial Community

In the financial area, the awareness level was relatively high in comparison to the other industries. Processing was done extensively throughout the organization. The perceived value of information was high because of the substantial monetary gains one could reap from penetrating the system. These factors had led to the adoption of security along all dimensions -- technical, procedural, and personnel. The primary security concern in this industry was to prevent financial loss, especially through embezzlement.

2.3.2 Medical Community

However, in the medical community, the data being processed, the structure of processing, and the resultant concerns for security are quite different. Generally, health institutions have not had to face the entire spectrum of problems raised by computerization. This is because the introduction of computers in medicine is relatively recent compared with an industry like finance where computers were introduced in the mid-1950's. Because the medical industry has not had to deal with security, it is not surprising to note that levels of security awareness are relatively low. The perceived value of the information being processed in medicine should also be considered.
Key Definitions and Concepts in Data Security

- **Data Security**: The state of data (or information) in which it is safe from unauthorized or accidental modification, destruction, or disclosure.

- **Data**: Any identifiable aggregation of bits stored on a machine-readable medium that can be manipulated or translated into some meaningful form.

The nature of data is derived from the use of it. Basically, two attributes apply, **essentiality** and **confidentiality**.

**Confidentiality**: If some data concerns a private party or person (client or patient), and if only certain persons are permitted access to it, it possesses a degree of confidentiality.

**Essentiality**: If some data has a high degree of importance for a user such that, if lost through unintentional modification or theft it can only be recovered at a high expense, then it possesses a degree of essentiality.

**Examples**: A proprietary software package possesses high confidentiality because parties outside the vendor-buyer relationship are not allowed examination or use of the package. Its essentiality may be very high for the owner, whereas for the buyer it is rather low (an extra copy can always be obtained). ZIP codes have both low confidentiality and low essentiality, because of their public nature and availability.
Exhibit 2

**Common Security Threats and Countermeasures**

Common threats against data security are computer installation sabotage, accidental system breakdown, fraud, embezzlement, interception errors, disclosure of data, theft, sabotage, or unauthorized copying of data. Data security can be created and maintained by some or all of the following elements:

**Technical Protection (automated):**
- computer system integrity (such as operating system, backup power, and fire protection)
- remote access control (such as terminal authorization, and user identification)
- data encoding (encryption).

**Procedural Protection (manual):**
- physical access control (such as guards, badges, and locks)
- data handling rules (such as offsite storage and written requisition of storage volumes)
- program modification rules
- input/output separation
- input/output controls
- audit.

**Personnel Protection:**
- preemployment screening
- supervision
- division of responsibility.

12 Volume 4 Massachusetts Institute of Technology
Although information is vitally important and extremely sensitive, its economic value is perceived as low to those who handle it. These factors have determined a more personnel-oriented approach to security. High reliance is placed on the integrity of personnel because so many are involved in the delivery of medical care. The intent in medical data processing is to prevent catastrophes, that is, health impairment or patient embarrassment, as well as to maintain the viability of the institution.

2.3.3 Education Community

In educational communities, security awareness is moderate to high because of the diverse nature of the processing environment. Every type of threat is possible (but perhaps not likely) in the university, because internal users are probably the most ingenious of all. However, information is not perceived as having as direct an economic value as in the financial industry. The result has been a more technical approach to security especially at the operating system level so that users can be protected from each other. In educational institutions, a primary objective of a security system is to prevent mischief.

2.3.4 Service Bureau Community

Service bureaus are somewhat similar to educational institutions because of the high concentration of technical expertise; however, they differ from educational institutions because the atmosphere is much more competitive. The nature of information processing is such that high premiums are placed on speed, efficiency, and performance. Also, more proprietary data generally resides on their systems. Thus, the resulting approach has been to focus on the operating system for internal security, as well as to make security options available to those customers who demand, and pay for, them. Prevention of financial loss through breakdown of system, manipulation of accounting mechanisms, or theft of propriety data have thus been the objectives of their security systems.

2.4 IMPLICATIONS OF FINDINGS

We now turn to the implications of our hypothesis for the manufacturers, user industries, individual firms, and the consumers or clients. Different awareness levels and approaches to security suggest that security interfaces may have to be tailor-made for the particular industry in question, so that different needs can be met. Although the interfaces may vary, the underlying security facilities needed are the same for all industries studied. In the future, it is quite possible that agreement could be reached on the minimum amount of security required for various industries or classes of data. The result of this could be the promulgation of industry standards with respect to good practices of secure operations.

At the organization or industry level, we have observed that those working closest to or most frequently with a particular data item will be most concerned with the security of that data. This proprietary interest in one's data could cause problems if top management were forced by budget constraints to protect one group of data over another.
Finally, the third hypothesis has implications for altering consumer behavior; if different approaches to security are taken in different industries, then the consumer may perceive that personal data may be more secure in one industry than in another. Consumers may fight to impose the security standards of one industry on the operations of another industry.

3.0 SPECIFIC FINDINGS FROM INDUSTRY SURVEYS

Consider some of the more specific findings for the four industries that we studied. A brief profile of the user community will provide background against which to view the security-related findings of our research.

3.1 FINANCIAL COMMUNITY

We have previously mentioned that the financial community is one of the more advanced areas since it has been using data processing techniques for the past two decades. Our sample was composed of three banking institutions, a mutual funds company, and a mutual life insurance company. Being a mature set of users, many of the security issues were originally dealt with in these sorts of institutions. We will now present some of the more significant points that were uncovered by our questionnaire research are:

- Sixty percent of the managers surveyed believe that a central decision maker is required in data security matters, but there is wide disagreement as to where in the organization this decision maker should be (both in terms of functional area and level of management).

- Computerization does not substantially increase the vulnerability of data.

- Data security in the organizations surveyed has received moderate to high attention in the systems development stage.

- Managers are generally quite concerned with the confidentiality of data handled by their departments. Their attitude toward protection of data against disclosure has increased moderately over the past three years because of the following reasons:
  1. Social forces
  2. Company policies
  3. Personal ethics
  4. Experience with harmful disclosure

However, managers do not expect their attitude toward data disclosure to change over the coming few years.

- The security precautions currently used cause little operating inefficiencies. The technical precautions (see Exhibit 2) are more bothersome than the procedural-personnel precautions in organizations with sophisticated processing. Thus, although technical precautions may be perceived as more important than nontechnical precautions, managers have been willing to spend extra security dollars on the less bothersome nontechnical aspects like input/output controls, audits, and personnel policies before spending dollars on sophisticated software techniques. In the future, additional physical protection will not be effective.
Exhibit 3

**THE NATURE OF PROCESSING**

The two important ways to characterize the nature of processing are: the technical nature of processing and the organizational nature. For data security considerations, both are important.

**Technical Aspects:**
- Ratio between use of central processor and use of input/output devices.
- Multiprogramming
- Multiprocessing
- Data communication
- Time-sharing
- Online/Batch.

**Organizational Aspects:**
- Dependency on data processing within organization
- Pervasiveness of processing, for example, number of employees in direct contact with processed data
- Location of users of data, that is, inside or outside of organization
- Purpose of data, for example, for immediate operations or for long-range planning
- Degree of decentralization (dissemination) of processing
- Time horizon for the use of processed data
- Organizational experience with processing.
Obviously, the financial community perceives a secure operating environment as a combination of technical, procedural, and personnel precautions. Financial institutions more than most other industries have already dealt with many of the substantial security problems caused by computerization. Many of the problems that occurred were effectively countered by a strong security system of non-technical components. After all, data processing in the financial institutions has been predominantly the automation of manual processes.

3.2 MEDICAL COMMUNITY

The medical community is faced with costs that are rising rapidly at the same time that many people do not have access to the care that they need. Many of these problems are due to temporal characteristics in organization and financing; therefore, these problems can certainly be ameliorated in the future. However, the entire system of medical care is ready for broad change.

Change could be in the areas of national health insurance or numerous prepaid group practices. As change occurs, the information needs of the industry will also be altered. Currently, information processing is both administrative and medical. Larger inroads have been made in the first type of processing and this will continue to be a fruitful area of endeavor. However, the next decade will see great advances in the realm of medical information processing. As these systems are developed and as the user community becomes more sophisticated, greater demands will be made for more secure operating environments.

In our study of this industry, we have purposely focused on the user; we have assessed his perceptions of the security issue in establishing what problems exist and what the necessary conditions are felt to be for counteracting any of these perceived security problems.

These are some of the observations that we have made in the field with managers and physicians:

- Computers are perceived as a threat to confidential relationships, for example, physician-patient and manager-employee.
- As computer usage expands in the industry, there will be an increasing need for security.
- In a Medical/Management Information System numerous types of data are processed and stored, including employee payroll and patient diagnosis. These data items are fundamentally different in nature and content, some being public knowledge and others being highly confidential. We can hypothesize that:
  1. These different data types can be identified.
  2. The different data types have varying needs for security due to their various degrees of sensitivity.
  3. The access rights of an employee to different kinds of data depends on his need-to-know and on the sensitivity of the data.

Massachusetts Institute of Technology
Most physicians and managers see the role of the computer dramatically expanding in their institutions over the next few years. As this growth takes place, it will be incumbent on computer manufacturers to provide assurances to medical users that information can be secure from interference. Without these assurances, the medical industry may become even more concerned that security threats constitute potential breaches of medical ethics.

3.3 SERVICE BUREAU COMMUNITY

We included the service bureau industry in our survey because we believe that most computer systems in the future will be operated as service bureaus of some kind. In universities, this has been true for several years. Also, in private industry, the trend is toward running the Electronic Data Processing division as an in-house service bureau since economies of scale in both personnel and hardware seem to justify this trend. In our survey, we interviewed several establishments whose services ranged from simply selling raw machine time to selling the use of proprietary data bases.

Findings common to all service bureaus surveyed were:

- Numerous relatively sophisticated security mechanisms are currently implemented in the service bureau industry.

- These mechanisms are usually site-implemented and site-maintained -- in only a few cases were they provided by the manufacturer.

- Almost without exception, these mechanisms are difficult to use, given the naivete of the average user.

- Users of the service bureau services are far more security conscious in word than in deed.

- Despite their general availability, these mechanisms are not used by the users.

- In general, somewhere in the region of 90% of the resources of a service bureau are consumed by about 10% of its clients. This 10% is a sophisticated set of users, for the most part, and tend to make extensive use of the available mechanisms -- often implementing their own.

One may wonder why these sophisticated techniques exist primarily in the service bureaus, and also why the use of the mechanisms is so limited. First, service bureaus are usually founded and operated by technical people who generally are concerned about security and are aware of the problems caused by security breaches. Second, in the service bureau environment, users are isolated from one another, and the system is independent from the users to permit time-sharing. These requirements encourage the use of security systems with strong access features. Third, and perhaps most important, security is an option on most service bureau systems, and, therefore, real costs are incurred by the user in both effort and dollars. Most users cannot comprehend the techniques and are not entirely aware of the potential problems inherent in default procedures; therefore, the optional techniques are not used. As one
service bureau official described the situation: "With regards to
security, some of our customers are sensitive, some are not -- but
they are all naive." If these rather unsatisfactory conditions are
to be rectified, the average user must be better informed on how to
use security techniques; the techniques must be easier to use; they
also should not be offered as an option but rather made an integral
part of the system.

3.4 EDUCATIONAL COMMUNITY

The security requirements of the educational (university)
community are a combination of the requirements of the financial and
the service bureau communities. A large part of educational
information processing is centered around research and teaching
(student problem data and solutions). This type of processing is
often based on online time-sharing and, thus, resembles the security
needs of the service bureau community. The requirements for integrity
and prevention of destruction or disclosure are similar in both
communities. Because large amounts of administrative data, such as
payroll, financial accounting, and so forth, are processed in the
educational communities, its security needs are also similar to the
financial community. Security needs for educational information will
be discussed in detail later.

4.0 TRENDS IN COMPUTER SECURITY

In summary, users demand security only when they are aware that
security can be a problem for them. From this base, we have
developed theories that relate to the security awareness levels in a
firm and how these levels depend on the proximity of the individual
to the computer system. This same awareness phenomenon determines the
approach to security that an industry in general or a particular firm
will take.

In addition to presenting this general theory, we have tried to
convey a sense of the concern for security in four distinct
industries -- finance, health care, education, and computer services.
We will consider some of the more likely trends in the computer
industry. And finally, we present The Data Security Game in the
Appendix as a way of putting all of the disparate elements of this
problem into a meaningful context.

4.1 IMPACT OF SECURITY ON THE ORGANIZATION

When extensive computer security is first introduced into an
organization, the personnel may react in a negative manner. Usually,
these reactions occur because of: (1) a feeling of loss of power
and/or (2) difficulty in getting their work accomplished.

The feeling of loss of power occurs because of the nature of a
secure system. People no longer can have unrestricted, unlimited
access to the entire system. Because most good system designs allow
the installation to determine each person's accessing ability, this
problem is primarily a management issue. Generally, when each
employee realizes that his information is better protected because of
the security, he is more willing to accept his own limitations.

A serious problem exists if the secure system poses real
difficulties to a user in accomplishing his work. Besides
introducing discontent, such a situation can lead to bad practices. For example, in a noncomputer environment, if a door requires 30 minutes to lock and 30 minutes to unlock and this door must be used frequently, it is likely that the employee will rapidly learn to leave the door open -- thus defeating the purpose of the lock.

As another example, consider the case of Doctor A who desperately needs information about a patient who has been admitted in an emergency. The patient is a regular patient of Doctor B, who, for some reason, is unavailable to give the attending physician access to the patient's file. Does this leave Dr. A helpless to attend the patient? In such cases, it should be possible to use a formal procedure whereby Dr. A can request access to the patient's file. The system will record this fact, and Dr. A's action will be subject to review.

In most contemporary systems, there is usually either no "escape" mechanism or the effect is accomplished by "turning off" the entire security mechanism (such as by the computer operator or systems programmer). In neither case is the procedure responsive, well-defined, nor is the action recorded for later review.

A possible implementation may be based upon the establishment of three levels of user access to information. The normal "access is allowed" and "access is prohibited" can be augmented by "access may be allowed." Thus, in an environment with high ethical standards and/or other constraints that tend to encourage ethical behavior, certain users may be given "access may be allowed" permission to other users' information. In this case the access would be automatically recorded and the final decision as to the appropriateness of the access is deferred to human review at a later time.

4.2 SECURITY AS AN ISSUE

We foresee that security will become an increasingly important issue in the computer industry within the next decade. This belief is based upon a number of diverse factors. As we have pointed out, one of the strongest trends in the future will be the changes within the user communities. As awareness expands, security will become a more and more important issue in the purchase of both hardware and software. Other key factors are at play as well. For instance, regulation in the computer field has been discussed. Also, the clientele of computerized organizations will make their voices heard more in years to come. The strength of the response from these areas will undoubtedly depend on the extent to which security problems continue to develop and the concomitant ability of the computer industry to adapt to its rapidly changing environment. We have identified four areas that have the power to effect increases in the level of awareness: (1) computer manufacturers, (2) computer users, (3) the general public, and (4) the government.

4.3 FORCES AT WORK

The computer industry can play an important role in changing the general level of security awareness. Numerous options are available to this group -- especially, the familiar routes of advertising, marketing, and education. Major product lines can be offered that stress their ability to solve many of the common security problems. The three major technical problems to be overcome are integrity,
Exhibit 4

Integrity --- The predictable and meaningful behavior of the system (i.e., Does the system follow orders correctly? Can it be confused?).

Authorization or Access Control --- The facilities, procedures, and restrictions for establishing access controls to be enforced by the system.

Certification --- The extent to which the integrity of the system can be proven or measured.

Example by analogy: Let us consider a guard for a warehouse.

(1) Integrity --- We must make certain that the guard understands all aspects of his job -- leaving no uncertain aspects. For example, if his replacement is late for work, he should not just leave. Every situation must be identified and appropriate actions specified.

(2) Authorization --- The warehouse would be rather useless if there was no way to get things in and out. Thus, there must be rules established that the guard can follow to determine whether a request for access should be permitted. Since the guard cannot be instructed, when hired, on the specific handling of each of these future requests, he is, instead, told how to determine the correct action (e.g., if Joe says it is "OK," then allow the access -- thus, Joe can authorize accesses).

(3) Certification --- Given that we have carefully instructed the guard on his duties, can we be certain that he will follow these rules faithfully? We would like to certify our guard's competence by some test.
authorization, and certification as depicted in Exhibit 4. Competitive advantages can be gained by marketing secure hardware and software. To emphasize the importance of security, a broad educational program could be undertaken that indicated the more frequent threats to computer systems. Currently, the manufacturer has a great deal of leverage in raising the level of security awareness.

Another constituency whose voice will be heard is the buyers or users of computing equipment. As attempted and actual violations increasingly occur, the demands of the users will increase. The experience of the CODASYL Data Base Task Group (DBTG) serves as an example of how security consciousness can be impacted in this area. For some time now, DDTG has been involved in the design of a data base management system that has security built into the system. This activity has been undertaken because of the general lack of security in existing data base management systems. Other user-oriented impact areas are the various journals specific to the user community. For instance, accounting journals often devote space to the issue of security from a controller's or treasurer's perspective; computer journals, and even medical journals, contain articles on computer security. It is difficult to estimate the magnitude of the effect that such exposure might cause. A large potential exists in the user constituency for increasing security awareness.

A third group that could cause increases in awareness is the general public, that is, the clientele of the various user communities. One cannot disregard the influence of this group because of the advances that the consumer movement has made in the past decade in the United States. Policy, as made by legislators, relies on the input of pressure groups such as unions, lobbyists, activists, and professional organizations. The success of these efforts depends ultimately on the volatility of the issue at hand. For example, automobile safety, pollution, and ecology became very controversial issues in the 1960's. The result was that the automobile industry was forced to comply with some stringent regulations. If public concern over computerized recordkeeping and the right to privacy were to grow substantially, then the computer industry could be another target of consumer groups. Because computerization is a potentially inflammatory public issue, the computer industry and/or user groups cannot even together stifle public security awareness.

As we have implied previously, one final constituency is the government that can exercise its control over the industry through legislation. The possibilities for regulation are mounting. In fact, Sweden has recently enacted the world's first law intended to prevent computerized invasion of privacy. The United States is also considering legislative action in this area. The Department of Health, Education, and Welfare's (HEW's) study of the problem recommends a "Code of Fair Information Practice," which would be enforced against all automated personal data systems, governmental and private. Unless manufacturers take the initiative, the immediate impact of such legislation would probably fall most heavily on the user communities as they were forced to comply with regulations. However, after a time there would be second-order effects that the industry as a whole would have to adjust to. It is very difficult at present to assess the total, long-run impact that such legislation might have.

These, then, are four groups that play significant roles in the process of changing security awareness. Note that this process of change is truly dynamic with leverage points and attitudes in constant flux.
4.4 LIKELY ACTIONS

To accomplish the requirements of education and standardization, organizations may evolve to protect both users of computer products, as well as the rights of the general public. This observation is made because it is not clear that current conditions offer protection from either poorly manufactured products or the misuse of present computer technology. This organization should ideally represent the manufacturer, the user, the government, and the general public.

This organization would have the responsibility for determining and promulgating classes of minimum requirements appropriate to various categories of user requirements. It would also certify particular systems as meeting the standards of a security class when operated in a certain manner. By adopting such a plan, users would have more knowledge of the capabilities and limitations of the products they buy, and the general public would see the extent to which their rights are protected.

It is not clear how such an organization should be formed. One possibility would be to separate the privacy policy-making from the security validation. For example, an independent organization could take responsibility for classifying and testing the security of systems (in a manner similar to the Underwriters Laboratory). The security requirements to be used for a specific system would be determined by the appropriate government regulations, industry standards, and the peculiarities of the company. In medicine and finance, for example, there already exist groups responsible for establishing standardized procedures and codes of ethics. We plan to make our findings available to such groups to aid them in making recommendations regarding computer security.
5.0 APPENDIX: THE DATA SECURITY GAME

In summary, we have dealt with a complex web of interrelated problems that interact to form a system. A basic means for putting this system into a meaningful perspective is by the use of scenarios that include the major actors in the "drama." Scenarios are valuable because they force one to look ahead and to plan. The Data Security Game puts much of the discussion of the previous pages into an understandable form.

The Game is usually played by two players:

1. **Manufacturer (M)**, who designs and produces computer technology.

2. **The User (U)**, the organization that uses computer technology.

At a more sophisticated level, a third or fourth player may request (or demand) participation.

3. **The Public (P)**, which by pressure groups and campaigns, influences decisions about the application of computer technology.

4. **The Government (G)**, which through agencies, commissions, and bureaus controls the application and perhaps design of computer technology.

An almost unlimited number of games can be played by some or all of these parties. A few examples are:

- M creates a secure operating system for U according to M's assessment of U's needs (without ever actually ascertaining them).

- M creates a secure operating system for U according to the specifications of G.

- U orders a secure operating system from M, but U fails to realize that he is asking for a system that is technically infeasible.

- P exerts pressure on U to comply with minimum acceptable standards of security. U cooperates because of concern for public image.

- U changes its recordkeeping systems and introduces expensive security measures in response to legislative action from G.

- M has to scrap a costly new product line because it does not conform with G's regulations on the design and manufacture of computers.

Obviously, the Game generates a rich set of paradigms for consideration. The reader is encouraged to create more variations. The exercise helps to focus attention on the important determinants of policy security.
DATA SECURITY IN UNIVERSITY ENVIRONMENTS

Donald O. Hewitt and
Peter F. Hollings

CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTERS AND EDUCATION</td>
<td>37</td>
</tr>
<tr>
<td>ADMINISTRATION</td>
<td>37</td>
</tr>
<tr>
<td>RESEARCH</td>
<td>39</td>
</tr>
<tr>
<td>CLASSROOM SUPPORT</td>
<td>41</td>
</tr>
<tr>
<td>COMPUTER SYSTEM SECURITY</td>
<td>43</td>
</tr>
<tr>
<td>SECURITY--THREATS AND PROMISES</td>
<td>43</td>
</tr>
<tr>
<td>THREATS--THEIR ORIGIN AND TARGETS</td>
<td>43</td>
</tr>
<tr>
<td>SECURITY IN EDUCATIONAL COMPUTING</td>
<td>49</td>
</tr>
<tr>
<td>THE CLASS MONITOR SYSTEM</td>
<td>54</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>54</td>
</tr>
<tr>
<td>GOALS OF THE MONITOR</td>
<td>56</td>
</tr>
<tr>
<td>THE MIT CLASS MONITOR SYSTEM</td>
<td>58</td>
</tr>
<tr>
<td>SECURITY AND THE MONITOR</td>
<td>65</td>
</tr>
<tr>
<td>RSS AND CLASSROOM SUPPORT SECURITY</td>
<td>66</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>66</td>
</tr>
<tr>
<td>THE POTENTIAL OF RSS</td>
<td>66</td>
</tr>
<tr>
<td>CONCLUSIONS</td>
<td>67</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>69</td>
</tr>
</tbody>
</table>

Massachusetts Institute of Technology  Volume 4  35
As one might expect, the role of the computer in education has been a rapidly expanding one. In describing the current uses of computers in this field, it seems convenient to impose a simple framework. We can describe data processing in the educational environment as falling into three major categories:

I. Administration
II. Research
III. Classroom Support

ADMINISTRATION

Within the educational environment, administrative data processing can be subdivided into three categories for discussion. It is important to realize that these categories represent only one of several breakdowns of this area. The hand study (1), for example, uses a slightly different one. The analysis in this paper is done from an information security point of view, and thus, has a functionally different set of criteria for subdivision.

The first category is Corporate Processing, so called because it is virtually identical to the central administrative data processing of a large corporation. The financial and accounting applications of the institution are found here, as well as the personnel processing common to all employers, such as employee payroll and benefits. In addition, programs for resource allocation are often included, among these being programs for scheduling classrooms and physical plant operations (analogous to production scheduling in the business world). In many ways this is the most "standard" of the various campus applications.

The second category is Student Processing. Here we find the set of data processing functions which are unique to students. In something of a chronological order, we can name Admissions, Registrar, Bursar, and Alumni Offices as groups that engage in Student Processing. Although it may seem somewhat arbitrary to separate Admissions from, say, the processing of applicants for jobs, it is in fact done almost without exception in universities. Even payrolls are kept separate, with students and staff in separate runs. Further, from a security standpoint, the school often has more confidential information on its students than on its employees. Therefore, we keep Corporate and Student Processing separate.

The final category is Departmental Processing. This subset of administrative data processing is a direct outgrowth of the decentralized departmental organization of most universities, and it includes many of the same functions as Corporate and Student Processing. Graduate admissions, for example, are commonly handled both at the departmental and central level. Resource allocation is also handled as a cooperative effort between the master schedulers in a central office, and the departments who staff the various courses. Payroll, on the other hand, is an example of a function that is generally not duplicated. However, the department for which a faculty member teaches is likely to have to worry about which account or research fund should be tapped for salary and expenses, so that there is still some interaction. A great deal of aggregate planning is done at the department level, in the design of curricula and
research programs. In summary, Departmental Processing is a decentralized subset of administration which has a more microcosmic view, and which requires interfacing with the central programs and data bases in order to be effective.

Having examined the components of administrative data processing, let us review some general characteristics of this area: timing, accountability for funds, security. This kind of processing operates mostly on fixed short-term deadlines -- registration day, budget submission day, not to mention payday. This is often in sharp contrast to the demands of Research or Classroom Support Processing, which, as we shall see, are quite flexible. Such a need for accurate timing dictates a high degree of system continuity and reliability. The manager of such an operation is likely to be quite conservative, a fact which underscores the similarity of this type of processing to its industrial counterpart.

Another facet of this area that resembles commercial enterprise is the accountability of managers for funds used. Because of the relatively standard nature of the tasks being performed, techniques have been established for their valuation and management. Further, because the users of administrative systems are employees rather than students, the resources of these facilities can be more closely (and more formally) managed. Noteworthy here is the absence of the "interesting area of research" syndrome, which often allows feasibility studies to be bypassed in research when considering the acquisition of new hardware or software. Thus, innovation is more difficult in the administrative area.

A final aspect, and one which is important to this analysis, is the need for security. It is clearly necessary to protect confidential data on both students and staff from unauthorized disclosure. Further, as in most corporate applications, it is essential to protect programs and data bases from accidental or deliberate modification or deletion, as well as to protect historical data files on tape or other media. A wide spectrum of physical and programming measures is needed to ensure adequate protection.

The combination of differing programming needs (e.g., FORTRAN versus COBOL), the need for conservative, reliable, short-term service, and the need for system security has led almost every university to physically separate administrative data processing facilities from all others. This has automatically solved the majority of their security problems so far. There is, however, a terrific price for this security. First, the physical separation has necessitated duplication of hardware and programming staffs. In an era of budget throttling and cost cutting, this duplication tends to be very unpopular. Second, and perhaps more subtle, is the fact that deliberate isolation of the administrators from the facilities utilized by Research and Classroom Support systems has meant that the administrators have been isolated from the technological progress being made in these areas. Although one would not expect this type of processing to keep up with the forefront of research (nor would this be desired from the standpoint of continuity and reliability, as mentioned earlier) the gap between the two technologies has been allowed to widen to such an extent that even if all security problems were solved tomorrow, it would be several years before most administrative facilities would be able to consider a merger of any sort with existing research facilities. The MIT Office of Administrative Information Systems is a good example of this problem (2).
One interesting exception to this general method of solving the security problem is found in Departmental Processing. Because this processing is at the department level where a large supply of cheap, high-quality labor exists (thesis students), more and more work is being done on these applications. Oddly enough, because the students who work on such projects do not have access to the administrative facilities, the work has been forced onto the newer research facilities, with the expected benefit of the newer technology. Thus, we see information systems for management and aggregate planning being developed, for example, in the Sloan School of Management at MIT (3) which far outstrip the capacity of the central administrative facility in the sophistication of information handling techniques. However, these applications are in desperate need of solutions to the security problems which the central facility solved by physical separation. Thus, we find the departments pioneering applications for the central facility by pushing into research facilities with innovative and complex planning systems which will force the security problems to be faced, probably well in advance of the needs of the central facility.

It should be noted, in conclusion, that there are two possible ways for the administrators to proceed. First, they may find it advantageous to combine facilities with the administrative organizations of other universities. An example of this type of venture is WICHE (Western Interstate Commission on Higher Education), which combines many universities in the design of standard administrative programs (4). This would provide a stable system, and also promote innovation through common development of systems. However, the problem of security still exists because many schools would be sharing common hardware if this cooperation were carried to its logical conclusion. The other alternative is a merger into a central campus computing facility either as a complete move, or as a remote satellite facility. This brings more risks in terms of system instability, but has the advantage of allowing a much smoother interface among the various kinds of central and departmental programs. In any case, it seems clear that over the next five years security problems will come to the point where they can no longer be avoided.

RESEARCH

The research community can be divided into two groups, those who research on computers, and those who research with computers. The former category is composed mostly of computer scientists, electrical engineers, and a few computer-oriented information systems specialists. The latter includes more of the campus every day, and even now, on the MIT campus, there is not a single department (including the Music Department) where the impact of the computer has not been felt heavily.

There are, of course, other distinctions that could be made here, for example, thesis versus non-thesis research, and sponsored versus non-sponsored. While we may make these distinctions here as examples of different accounting methods, our primary emphasis will be upon the differences between research on and with computers. This emphasis allows us to focus on the security aspects of research processing.

By nature, Research Processing is highly decentralized. This is primarily due to thesis projects, which are generally pursued individually under the supervision of an advisor or committee. In
terms of the number of persons participating, this constitutes more than half of the research effort of most universities. In addition to extreme decentralization, Research Processing can be characterized by a lack of definite deadlines. This is quite intuitive -- one can hardly set deadlines on pure research and development. The kinds of time constraints generally encountered are in the form of arbitrary limits such as project expiration dates and thesis deadlines, which are generally quite flexible. A related facet of managing research efforts is the allocation of funds. When examining administrative processing, we observed the formal structure for financial accountability. A different problem is encountered with the thesis user. Generally, the thesis user is not spending personal funds on computing. The department will often have a set procedure for computer resource allocation. The problem occurs when the user consumes the full amount of the allocation and is not yet finished. It seems absurd to think that the student must quit. Somehow, more funding must be obtained. Control of not only the allocation of funds, but the use of funds thus becomes a problem.

A related problem is the expertise of the individual user. The researcher who is a computer specialist is likely to have computer expertise which permits efficient use of the computer resources, while the person who merely wants the computer as a tool for other research may be totally unfamiliar with the proper methodology, thus wasting vast amounts of money. Further, it is possible that the non-computer researcher may be unwilling to invest time in learning efficient techniques. From this perspective, we must realize that these users have their own priorities, and it is unrealistic to assume that the computer should be allowed to detract appreciably from the primary goals of research in the user's selected field.

The last aspect of Research Processing to be discussed here (research on computers) can be viewed as a mixed blessing. Certainly, the campus has been the scene of the most significant advances in computer technology -- the computer itself, core memory, software systems such as the MIT Compatible Time Sharing System (CTSS) and others. In addition, the research on computers within a given campus community often results in "tuning" improvements within existing equipment, such as improved device management, scheduling algorithms, etc. Sometimes, entire new systems, such as the MULTICS system at MIT, become available to the community as a direct outgrowth of campus research. Moreover, this technological fallout has two bad effects. First, educational systems tend even more than others to become "tailored" with a series of modifications that render them incompatible with all other systems. Second, when new resources are introduced, there ensues a great deal of contention about system stability and documentation for general use. One example of this problem is the development of the MULTICS System at MIT from 1968 to 1971. During this time, the system underwent constant "tuning" changes, as well as several major revisions. Thus, many non-computer users who might want to use the machine for service were in direct contention with the systems development people, who wanted constant improvement.

For purposes of this discussion, the Research Processing area can be regarded as an extremely decentralized group of users, whose expertise ranges from superlative to negligible. It is characterized by a lack of structure both from time and budget perspectives. Finally, we find that research on computers sometimes requires experimentation at a machine level which can seriously conflict with the standard service needs of the rest of the research community.
In recent years, the computer has begun an important and many-faceted role in the classroom environment. It comes to the classroom both as subject and tool. We shall divide Classroom Support Processing along these lines, into support for computer courses and support for non-computer courses. The reason for this breakdown is the fundamental differences in the teaching objectives of these two areas, which are reflected in the types of activity produced.

There exists within the curricula of most large universities a set of courses designed to teach computer programming and other aspects of systems design and utilization. Whether the computer is being taught in its own right or in the context of some application such as mechanical engineering, the orientation of such courses is toward the exploration of the capabilities and limitations of the computer system. This focus upon the computer results in a great deal of student use, and, as we shall see, in a serious security problem from imaginative and mischievous student programmers.

In contrast to the computer courses, a rapidly growing segment of the computer resources on campus is being consumed by students who have no interest whatever in computers or programming. These students are using pre-packaged programs that perform simulations, linear programming, and other computations that aid the students in their work. A good example of this type of program is the area of financial management, where a student might have access to small utility programs for present value analysis and discounting, as well as large packages for linear programming and modeling. The important distinction here is that the non-computer student is neither trained in nor (in most cases) interested in the computer. Thus, the kind of in-depth exploration of the computer system characteristic of computer courses is not found here. However, from the security point of view, we find a different problem -- the control of large numbers of inexperienced and sometimes indifferent users. As we shall see, there is an implied problem of usage control by the instructor, who must see that the class budget is efficiently and equitably distributed.

In summary, we have divided the educational computing environment into three areas: Administration, Research, and Classroom Support. Figure 1 shows the overall breakdown, along with some rough figures on each area as a percent of total activity. It should also be noted that the type of processing is relevant, ranging from batch processing to interactive time-sharing. All of the categories in our schematic can be regarded as existing in a continuum which has batch monoprogramming at one end, followed by multiprogramming, limited inquiry systems, and finally interactive systems at the opposite end. The reason for this view is the difference in security requirements, for example, the problem of collection and distribution of decks in a batch environment versus terminal access control in time-sharing systems.
Note: An estimate of the three areas as a percentage of total university data processing expenditure is included. The percentage in 1967 (See Reference 1), and the author's estimate of the percentage in 1973, respectively, reflect a relative increase in Classroom Support Processing. 2% is allowed for other uses.

*Author's Estimate

Figure 1. A Schematic of the Educational Environment

The next section will discuss a framework for describing security requirements of our three areas in terms of this framework.
COMPUTER SYSTEM SECURITY

SECURITY--THREATS AND PROMISES

Security, whether in computer systems or in some other context, is a topic that eludes positive definition. It is usually described negatively. For example, security is not having your house broken into, or not having your medical records printed in the newspaper. It is not required here that we arrive at a definition which will be accepted with great finality by all readers. However, it is essential that some working definition be developed that will facilitate the comparison of the security requirements of the different areas described previously, as well as permit a general discussion of computer system security.

An "ideal" computer system can be defined as one that provides uninterruptible and fully controlled service, and in which all data and programs are available only to authorized users, and only in a particular mode. While the notion of an "ideal" computer system is admittedly a vague one, it does serve us well enough to establish the nature and function of computer security. Whatever we define as our desired system, computer system security is the ability to prevent it from being changed. One distinction that should be made here is the difference between programming errors and system errors. It is consistent with the notion of an "ideal" system to expect it to generate errors in the output of jobs whose input is incorrect. To execute exactly the program submitted is all we may ask (for the present) of any system. The system deviates from its desired performance when, for example, the accidental or intentional errors of some task are permitted to interfere with the execution of supposedly independent tasks. Thus, we see security as the act of maintaining a set of system-wide relationships among programs, data, processes, users, and other system entities that define a computer system. A secure computer system, then, is one that promises that its security mechanism is capable of dealing with every component of threat without failure, insuring consistent performance of the system.

THREATS--THEIR ORIGIN AND TARGETS

Having defined security in terms of coping with threats to the system, we now turn to the problem of classifying those threats. The first qualifier would logically seem to be the source, or origin of the threat. The source of a threat can be viewed along several different dimensions. In a company, for example, threats can be classified as originating within the firm, or outside. For our purposes, however, we seek a dimension independent of the purpose of the machine, and one that is relevant to the organization of the computer system itself. Therefore, we will regard threats as originating in one of three modes:

1) External Mode
2) Supervisor Mode
3) User Mode

The first mode reflects all threats that do not involve the execution of an instruction under control of the operating system. This includes most of the kinds of security problems referred to in...
the literature as "physical problems," such as fire, vandalism, theft of tapes, etc. It also includes the type of programs known as "background utilities" where the accidental mounting of a wrong disk pack can result in loss of good data through accidental initialization. In this case the threat is not under the control of the computer system, but is in direct control of an operator, who mounts a pack and pushes a button. Power surges are another common type of problem that originates outside the system framework of sequential execution of instructions.

The two remaining modes represent a distinction made in the design of many modern computer systems, such as the IBM System/360 and System/370 ("Supervisor State--Problem State"), and the GE 6000 ("Master Mode--Slave Mode"). Because the experience of the author is mainly on IBM equipment, most of the examples herein will be drawn from that environment. (See 5,6.) The most general description of the difference between Supervisor Mode and User Mode is that the entire instruction set of the machine is executable from Supervisor Mode, while User Mode permits only a restricted subset (in the 360/370 Series, all I/O instructions and several control instructions require Supervisor State) (5).

It should be noted here that not all programs in the operating system run in Supervisor Mode. This mode is generally reserved for important control routines, such as the I/O controller. If we think of Supervisor Mode as a mode of operation restricted to special system routines and User Mode as the mode of all programs that execute under the control of these routines, then for any non-external threat, the threat may be thought of as originating within Supervisor Mode or outside Supervisor Mode (User Mode). Assuming that the ability to switch from User to Supervisor Mode is closely controlled in most machines (which it is), we say that there is a difference in the nature of the two threats, because one is a threat of internal disruption while the other is a threat of system penetration.

One example of a Supervisor Mode threat is the IBM Attached Support Processor (ASP). In the current version of ASP being used at MIT in conjunction with the IBM Resource Security System (RSS), ASP runs in Supervisor State, with all protection disabled. ASP has been modified by the local programming support staff to a large extent. Any bugs in ASP have an excellent chance of clobbering the operating system because protection is disabled. Thus, we see that there can be threats from within the Supervisor Mode. Threats may also originate, of course, from the programs that run in User Mode. Common threats of this type include the attempt to switch to Supervisor Mode without authorization, and the issuing of invalid requests to system service routines. (One of the classic examples of this in the System/360 is requesting the system clock to deposit the current time into the middle of the operating system programs, which it does on some versions without checking.)
In summary, we have categorized threats according to their origins: those external to the computer itself, those internal to the computer and its privileged mode of operation, and those internal to the computer in normal operating mode. Having established where the threats are coming from, let us turn our attention to their targets.

For our purposes, it is convenient to divide threats into two major categories:

1) Threats against Operating System and Subsystem Integrity.
2) Threats against data set access control.

We shall not attempt the impossible task of enumerating all of the possible threats against a computer system. We will instead mention some of the salient types of threats, and note their places in our framework. Hopefully, this will suggest to the reader other particular cases that we have omitted. The breakdown of our two major categories is outlined in Figure 2. As can be seen,
Operating System and Subsystem Integrity includes a wide range of system functions, including accounting and control functions as well as system stability. One important feature of this category is its recursive nature. All of the kinds of features inherent in system-wide operation, such as accounting, validation, and continuity also occur in subsystems, to an extent that depends upon the sophistication and design objectives of the subsystem. An example of system and subsystem integrity problems is the design of an interactive program for use under CP/CMS (Control Program/Cambridge Monitor System, a set of software used on the IBM System/360 Model 67). The environment is shown in Figure 3. As can be seen, there are three levels of system, each with a separate set of control and accounting problems: first, the CP System, which provides a virtual machine environment that must be protected and isolated; second, the CMS System, which provides the command language and file system and must manage requests, and finally, the interactive program, which might have its own accounting and complete user environment to support and protect.

OPERATING SYSTEM AND SUBSYSTEM INTEGRITY

1) Accounting Mechanisms
2) User Validation
3) Priority and Process Scheduling
4) Integrity of Actual Code
5) Memory Access Control
6) Continuity of Operation

DATA SET ACCESS CONTROL

1) Read
2) Write
3) Execute
4) Append
5) Delete
6) Restrict Access to specific programs
7) Control of access to offline files (tapes, cards, etc.)

Figure 2. Target Categories
The other category of threat is against Data Set Access Control, where we define data sets in the most general sense as aggregations of data (or instructions), which may be either on line, or off line in the form of cards, tapes, or printout. Here we find the conventional read, write, and other access categories for online data sets, as well as the full range of physical access controls necessary for offline files. All control of programs and data within the computer system (with the exception of those programs and data specifically designated as protected parts of the operating system) is thus placed in this category.

A schematic diagram showing the relationship between the origins and targets of several sample threats appears in Figures 4 and 5. Having outlined our security framework, we will now turn to a brief comparison of the security requirements of the educational environment.
Figure 4. Schematic of Example Threats
EXAMPLES OF SECURITY THREATS

1. A power surge.
2. A tape is stolen from the library.
3. A user shuts off a dedicated terminal in a retrieval system.
4. A bug in the operating system causes a core lockup.
5. When no more space is available in the OS/360 job queue, new jobs are entered on top of jobs already in the queue, thus destroying spooled input before they can be processed.
6. The teleprocessing access method interprets a line error as an "attention" interruption, and discontinues the program in progress.
7. User gives the time of day command a bad address, causing it to overwrite part of the OS.
8. In OS/360, users have almost unlimited access to delete data sets, even those belonging to others.
9. A user enters a character in a numeric input line and causes the program to end.

Figure 5. Example Threats

SECURITY IN EDUCATIONAL COMPUTING

Given our simple framework for classifying threats, the most convenient way of surveying the security requirements of the educational computing environment seems to be the use of the simple charts shown in Figures 6, 7, and 8. These charts outline the most important aspects of the three areas defined in the first section. They will be supplemented by brief individual discussions here, followed by some general observations on the environment as a whole.
<table>
<thead>
<tr>
<th>SYSTEM AND SUBSYSTEM INTEGRITY</th>
<th>DATA SET ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXTERNAL</strong></td>
<td>Relatively standard set of operator and environmental problems.</td>
</tr>
<tr>
<td><strong>SUPERVISOR</strong></td>
<td>Minimal problems here due to use of very standard software designed to minimize operating system problems.</td>
</tr>
<tr>
<td><strong>USER</strong></td>
<td>Restricted user community and standard software minimize these problems.</td>
</tr>
</tbody>
</table>

*Figure 6. Administrative Security Profile*
Standard set of operator and environmental problems, plus research problems if experimental equipment is interfaced with the system. Also problems of remote terminal control.

For classified research, input/output handling problems will occur. Also, standard risk of operator error.

Experimental operating systems (and modifications to existing operating system software) may cause the system to be less reliable, increasing the problems in supervisor mode.

Operating system modifications increase the risk of data set access problems while in supervisor mode.

Most users are not interested in challenging the system. Experimental operating systems may allow accidental penetration. Computer science users may need to experiment at this level, and may cause problems if not isolated.

In an environment like OS, considerable problems with many inexperienced users having access to one another's files. Also, there is a problem of protecting users from their own errors, which is not possible under OS/360.

**Figure 7. Research Security Profile**
Figure 8. Classroom Support Security Profile

The Administrative area, as expected, has minimized many of the threat categories by its use of standard hardware and software and its restriction of users (almost all are members of the programming staff). However, two areas are important here. First, External Data Set Access threats are extremely important, due to the large number of confidential and historical data sets, and the large amount of physical handling of these files. Also, threats from User Mode to Data Set Access come from the possibility of programming errors within the staff, as well as large scale possibilities if facilities were shared with other areas.

In the Research area, we find increased threats for two major reasons: first, the use of non-standard hardware and software vastly increases the potential threat from External and Supervisor Mode respectively, and second, the great increase in the number of users and the corresponding decrease in control makes User Mode security much more important. While it is true of most researchers that their preoccupation with research makes tampering with the system unlikely, it is also true that for those who are researching on computers at the most basic level, some simulation or virtual machine support is necessary if they are to use common facilities. This implies a great deal of subsystem support. Overall, we find our security problems expanding.

The last area, Classroom Support, carries the research problems one step further, involving even greater numbers of students, some of whom are "hackers," and all of the problems inherited from attempting
to gain from research technology in the form of hardware or software modifications. We find here the full range of educational security problems. External, Supervisor, and User Modes seem very well filled out with standard and experimental equipment, and cooperative and malicious users.

There are clearly discrepancies in importance between compromising the solutions to a problem set and the medical records of students. If, however, one sets aside the intrinsic value of the information and observes the security profile, it can be asserted that all of the security problems of the educational environment show up within the Classroom Support environment. It follows, then, that a computer system that satisfactorily solved those problems for the classroom environment would be satisfactory for general use.

The remainder of this paper deals with a specific aspect of the Classroom Support environment, namely, the creation of a monitor subsystem to allow controlled, time-sharing in a secure student environment. External and Supervisor Mode threats will be left for other research, and we will concentrate here on the implementation of a subsystem that must deal directly with User Mode threats in a time-sharing environment.
THE CLASS_MONITOR_SYSTEM

INTRODUCTION

This section discusses one attempt to satisfy the need for controlled time-sharing for Classroom Support processing. The Class Monitor System for OS/360/TSO (Time-Sharing Option) was implemented during the summer of 1972 at MIT. In order to explain its design features and implementation strategy, it is first necessary to briefly describe the TSO environment.

At the time, the MIT Computation Center ran OS Release 21 with TSO on a System/370 Model 165 with 1.5 megabytes of core storage. Several IBM 3330 Disk Storage Drives are available, as well as two IBM 2301 Drum Storage Units for swapping system data sets. A hybrid of ASP (Attached Support Processor) and LASP (Local Attached Support Processor) is used in support of the Main 165. Although several MIT modifications have been made to the system, the basic structure and user interface of TSO remain standard (7,8). It should be noted here that MIT has a standard data set naming convention for user data sets, which reflects the account number and programmer to be billed for the online storage. A standard MIT/TSO data set might have the following name:

U.M1234.9999.HEWITT.FORT

where U = User File, M1234 = Account Number, 9999 = Programmer Number, Hewitt = Arbitrary Name, and Fort = FORTRAN Source. Each programmer on the system has a unique programmer number, and each project is assigned a unique account (or problem) number. Thus, only valid problem-programmer combinations are accepted within the system. Under TSO, a unique problem programmer number is associated with each USERID-PASSWORD combination, and the prefix "U.PROB.PROG" is automatically added to all data set references (see Reference 7 for details of TSO data set conventions).

Under standard TSO, as might be expected, "all users are treated equally." That is, each entry in the User Attribute Data Set (UADS) is assumed to be a "full" user, with all of the privileges available to users of the system. The only exceptions to the notion of the standard user are users whose UADS entry contains either an administrative or operator flag. Thus all standard users have the same command repertoire and data set access rights. Under OS/360, these data set access rights are quite extensive, because they include all data sets in the system. In fact, until recently there have been only two ways to remove access to data sets in the system -- first, they may be password protected (a clumsy and not often used facility), and second, the physical device containing the data set may be placed in read-only mode (this is done at MIT with the system residence volume). Note that the second method cannot protect against read access, while even the first cannot hide the existence and location (volume serial number) of the data set in question. Thus, we see that the standard TSO user has a quite potent facility at his command, which includes the ability to modify or delete more than 90 percent of the data sets on the system, either accidentally or deliberately. Rounding out our discussion of standard users, consider the administration of USERID's within the system. The central user accounting office must handle both batch and TSO accounting. At any given time, there are several thousand active
combinations of problem-programmer numbers, as well as hundreds of valid TSO USERID's. As classroom use of computers increases, the potential load on this office is obvious -- several thousand students, each involved in several courses, combine to produce many thousands of USERID-PASSWORD combinations. If we assume that the instructor in any given course would probably exercise fairly tight control over the course budget, the workload of the central office in closely overseeing all users becomes immense.

The Classroom Support environment lends itself naturally to decentralized control of computer resources. Because the student users are already organized into classes, it seems logical to allow the person in charge of a particular class to allocate computer resources in any desired manner. Further, some controls should be provided for the class administrator to ensure that the class budget is spent equitably. For example, in the current MIT accounting system (which, although primitive, is not unlike many university centers), if a class of ten students were each assigned standard TSO USERID's, all ten users would be billed against the class budget for the course. Now because they are "full" users, there is absolutely nothing to prevent a student from logging onto the system and using it to create, compile, and run FORTRAN decks instead of executing the desired tutorial program. Moreover, because all users in the class are being billed against the master budget (a separate account for each user is unthinkable) a single enthusiastic student can "play" until the entire budget is consumed, preventing the other nine students from ever logging on at all. This rather disturbing scenario at least has the advantage of assuming non-hostile intentions on the part of the student. If a student who wished to cause trouble was released on the system, almost every user data set on the system could be deleted without fear of discovery.

It is clear that there is a need for a subsystem that can achieve the dual purpose of relieving the central accounting office and providing control of student users. Figure 9 indicates an expanded set of design goals, which will be discussed individually. Following that discussion, the implementation of the MIT Class Monitor will be described.
DESIGN GOALS FOR CLASS MONITOR

1) Get immediate and uninterruptable control at logon.
2) Validate users and log invalid users completely off the system.
3) Give the valid user a pre-designated "subset" of full TSO.
4) Protect itself and its files.
5) Handle user file maintenance.
6) Get uninterruptable control at logoff or console shutoff.
7) Handle on-line accounting.
8) Provide a mechanism for maintenance of the master accounting file.
9) Have low operating overhead.
10) Be easy to use for inexperienced students.
11) Be easy to maintain for center personnel.

Figure 9. Design Goals for Class Monitor

GOALS OF THE MONITOR

No matter how we implement our system, it is still necessary to use the concept of the TSO USERID. Now, however, we assign several "open" USERID's to our class and install the Class Monitor in each one. The first goal of our system, then, is that the Class Monitor must intercept the standard logon in some manner and gain control of the logon session, thus encapsulating the user in a new environment immediately and without failure.

Because we have given out "open" USERID's, the process of validation and account balance checking must be undertaken by the Class Monitor logon processor, so that only valid users are admitted. If students fail to give the proper identification, they must be logged off the computer system. Note that the USERID had to be logged completely onto the TSO system before the execution of the CMS processor could begin. Therefore, it is necessary for CMS to fire a direct call to the system logoff routine to prevent the user from taking any action prior to being logged off the system.

Having admitted the student to CMS, we now face the problem of control. The Monitor must be able to allocate the user only those commands that are necessary to do the assigned work, and in a manner that will make it difficult to perform any but the intended tasks. As will be seen later, extreme care must be taken to avoid giving the user commands with which he may bootstrap into a more powerful environment. For example, if we give the user a restricted command library, and also the ability to copy command processors, the user will simply copy more powerful commands into the library and then execute them. This is a difficult area in the design of subsystems that attempt to contain student users.
If the Monitor is to have online accounting, as well as a command library of some sort, it is clear that these data sets must be protected from user tampering. Further, it is desirable that the existence of such files be hidden from the user, to eliminate the temptation of such tampering.

Perhaps the most difficult area for the monitor system is the handling of user file maintenance. First, we encounter the problem of giving the user the ability to create and delete data sets. This entails releasing commands that constitute a direct threat to other data sets on the system. Second, many interactive teaching programs use online data sets to save intermediate results between login sessions. Thus we should like to provide some means of creating and deleting data sets from within a higher level language such as FORTRAN or PL/I. Finally, we encounter the problem of billing the individual student for data set space because the entire file maintenance system of IPC TSO is built around the notion that all of the files under one USERID should be treated as a single group. Ideally, the Monitor should allow for program control of data set allocation, automatic distinction between different students' files under the same USERID, and billing procedures for these files.

It is essential that the Monitor get uninterruptible control at logoff time, to ensure a clean CMS termination, and proper billing at the student level. Further, it is essential that control be received in the event of an abnormal termination such as console shutoff or telephone disconnect, because these events could easily bypass the subsystem accounting mechanism.

The concept of online accounting is important for class use. This is primarily true due to the scarcity of computer resources at this level. The MIT TSO system does not, at this time, have online accounting, primarily for reasons of security. That system produces punched card records of each logon session, and performs a daily update Monday through Friday. With the limited resources available to classroom users, a student could easily log on several times during a weekend, and consume much more than the parcel of time allotted, knowing that the billing would not catch up until Monday. This is quite unsatisfactory for the classroom environment. In fact, there is some reason to believe that the accounting should be done periodically during the logon session, so that the session could be terminated when the balance reached zero instead of waiting for LOGOFF processing to update the balance. At the very least, the Monitor must compute a reasonably accurate cost figure for the session, update an online accounting file, and display both the session cost and balance to the user, so that the students may budget their time properly.

A necessary adjunct of online accounting is the ability to maintain the on line accounting file by adding or deleting users, allocating funds, changing passwords, printing reports, etc. All of these standard maintenance functions must be provided in conversational form so that the class administrator can maintain the file without assistance from the central accounting office.

The final three design goals are common to all types of subsystems. Obviously, the Monitor must not consume large amounts of computer time in providing the student environment, or the benefits of this environment will be outweighed by its consumption of resources. Because the Monitor is to be used by students of all disciplines, it must provide a set of interactions that are extremely straightforward, so that there is little chance of confusing
inexperienced students. Finally the Monitor must be easily maintained, because the programming support staffs of the university generally have a fairly high turnover rate.

We have now outlined a set of design goals for a Classroom Support subsystem. In the second part of this section, we will discuss the implementation of the MIT Class Monitor System, which attempts to achieve these goals.

TMF MIT CLASS MONITOR SYSTEM

The Class Monitor System consists of three major components:

- Logon Processing
- Session Control
- Logoff Processing

We will examine each of these components and the implementation strategy used, concluding with a summary of the strengths and weaknesses of the system as currently available.

When a TSO user logs onto the system, the Terminal Monitor Program (TMP) is invoked. This is the standard IBM control program, which supervises the console session. It is the TMP that accepts and executes all user commands. The TMP also has a very important feature which we use to get control. At the beginning of the console session, the TMP looks at the PARM field of the EXEC statement which invokes it. (See Figure 9 for details on the operation of the invocation process and catalogued procedure used.) If any non-blank characters are present in this field, they are placed into the command buffer and taken as the first command line to be executed. Further, no information is accepted from the terminal until this first command is processed to completion. This includes processing of "attention" interruptions, which are suspended until the first command terminates normally, and until the TMP issues the first READY message to the console.

The availability of the "first command" option means that the Monitor can specify a single load module that will be executed before any user intervention is permitted. This load module must, of course, contain the CMS user validation scheme as well as some ability to assure that invalid users are logged off the system. As a matter of convenience, the Master File Update Routine is included as a subroutine in the LOGON module, thus allowing us to implement this version of the Monitor using only two load modules, one at logon, and one at logoff. The modules are both written in PL/I. Figures 10, 11, and 12 give the basic flowcharts of the system, and each one will now be explained.
Figure 10 depicts the operation of the LOGON processor. A standard TSO "nopass" option is used for all TSO CMS USERID's which allows the invocation of the TMP without a TSO password, that is, TSO logon is accomplished by merely typing "LOGON USERID." At this time, when finished with initialization, the TMP passes control to the CMS LOGON routine. The first function performed is the basic user validation. CMS uses group numbers for user identification. This was done so that the class administrator would not have to cope with both USERID's and passwords. Under this scheme, the administrator may assign group numbers based on a signup list or class enrollment sheet and either assign or collect unique passwords. This scheme also tends to assist students by giving them a number instead of a character string as their identifier. If they forget their password, they can always identify themselves to the administrator by group number...
number and have their memory refreshed. The user is given two tries to enter a valid group number/password combination. If unsuccessful, the user is logged off the system completely. This is made possible by the mechanism of the system logoff routine. It is called directly from within the LOGON processor and simply turns on a bit in one of the system control blocks for the process. Whenever control is returned to the TMP, this bit is automatically checked and, if on, causes the logon session to be cancelled. Because control is returned directly from the LOGON processor to the TMP, this screening method cannot be subverted. The next task is to determine the account balance. If there is any money left in the account, the user continues processing. If not, an appropriate message is printed, and the user is logged off. At this time, we are certain that the user is a valid one, either a student user or an administrator. We must now solve an important problem caused by the modular nature of TSO.

Although we now know the valid user's group number, as soon as the LOGON processor is finished executing, all of the current information disappears as a new program is fetched into the TSO region for execution. How does the LOGOFF processor know which group number to charge for the session? It is clearly undesirable to have the group number and password reentered at the end of the session. The problem is solved using the User Profile Table, one of the control tables that remain in use throughout the session, and in which, for example, the characters for character and line deletion are recorded. Several bytes of installation-usable space are reserved in this table, and one halfword is used to record the group number. Each time the TMP invokes a command, the address of this table is passed in the argument list. This allows our LOGON and LOGOFF commands to communicate.

Having recorded the identity of the user, we check an administrative flag in the accounting record. If the user has administrative privileges, we give him an opportunity to access the Master File for maintenance. Otherwise, we simply return to the TMP, our work completed. If the user accesses the Master File, we return to the TMP when finished.

The Master File Access Routine is a PL/I procedure callable from the LOGON processor. It is simply a conversational file maintenance routine, which provides the obviously necessary functions of administration, such as adding and removing group numbers, changing passwords, allocating money, printing reports, etc. Most of the programming and design involved are relatively mundane, and available to the interested reader in the CMS Programmer's Guide (10). One feature that is relevant here, however, is the hierarchical scheme currently employed for administration. Users are classified either as non-administrative (no access to Master File), administrators (access to the records of all student users in the file), and super-users (access to all records in the file). Without judging the merits of the scheme, it is interesting to note that almost all subsystems of this nature possess the "deity syndrome," that is, there is always one particular user (in this system called a "super-user"), who has absolute accessing rights to all information within the system. This usually starts as a protective measure during the debugging stages of the development, when it is clearly advantageous to have an override mechanism for emergencies. But somehow, as time goes on, this facility is never removed, so that the creator of the system always has "the power." It is likewise true of CMS that the current CMS administrator has absolute access to all Master Files for all classes. Because the class administrator is often an inexperienced student assistant, it has been found that the override facility has been very useful so far.
LOGON processing, then, includes user validation and screening, account checking, posting of group numbers, and Master File access for administrators. Control is then returned to the TMP.

When the TMP regains control from the CMS LOGON processor, the system logoff bit is immediately checked. If this bit is set, the session is terminated. If not, the user is presumed to be valid, and a READY message is issued to the terminal. (For a description of the CMS LOGON and LOGOFF user interface, see Figure 11.) At this point, we are at the TSO READY State. This is often referred to in other time-sharing systems as the Supervisor Level, or Command Level. When a user is in READY state, the TMP is ready to process commands from the terminal.

Having performed the necessary functions at LOGON, we are now faced with the problem of controlling the user console session, as depicted in Figure 11. As noted earlier, we would like to give the user the absolute minimum number of commands necessary to perform the assigned task. In doing so, we would like to use standard TSO commands as much as possible, so that when a user has a question about a command, all of the existent TSO documentation (such as the Command Language Reference Manual) will still be relevant. The procedure, then, is to start with the standard set of TSO commands and simply remove those that are not needed. This turns out to be an extremely easy task, because of the straightforward way in which TSO processes commands. When a command line is sent to the command buffer for processing, the TMP has a standard search path for locating the proper program to fetch. The part that concerns us here is the location of the commands themselves. They all reside in a partitioned data set called SYS1.CMDLIB. Each command is a separate member of the data set, with its command name and abbreviation corresponding to a member name and alias. In order to restrict the commands available to the user, we simply copy the desired subset of the TSO commands into a new partitioned data set, and substitute that data set into the search path in place of the standard command library. The DDNAME of this new data set is LIMITLIB, and the TMP is modified to make the substitution. Thus, we tailor the command set of a given class by varying the members of the LIMITLIB used for that class. When commands are requested which are not included in LIMITLIB, the standard TSO error message "COMMAND XXXX NOT FOUND" is received.
For some applications, the LIMITLIB alone is a satisfactory solution to control problems. However, in many cases it is necessary to allocate and delete data sets in order to run instructional programs. If we release the ALLOCATE and DELETE commands to users for this purpose, then they will be able to allocate any data sets they wish (or accidentally allocate large data sets by incorrectly specifying the allocation parameters) as well as delete any data set on the system. Therefore, we use the TSO EXEC feature, which allows
us to store commands in a data set for execution. Then we rename the commands we need, placing the renamed members in and executing them with our EXEC file to maintain the confidentiality of the command name. Finally, we remove the option of the EXEC command which allows the command data set to be listed, so that the user cannot learn the new command names (otherwise, the user could simply invoke them from LIMITLIB by their new names). This procedure allows us to include commands in LIMITLIB which we wish to restrict to a subset of CMS users. The LIMITLIB data set and the EXEC command, then, provide user control during the console session.

At the end of the console session, CMS must get control to perform accounting functions and print charges. This is accomplished by inserting a CMS module into LIMITLIB as LOGOFF. The flowchart for LOGOFF is shown in Figure 12. When the user types the command LOGOFF, the CMS module is fetched instead of the standard system program. After CMS performs its accounting functions, it calls the system LOGOFF routine, assuring that when control is returned to the TMP, a normal session termination will occur.
Referring to Figure 9, let us evaluate the current version of the Monitor in light of our design goals. Following that, we will discuss some of the security aspects of the Monitor.

We have seen that CMS LOGON gets the desired control at logon time, and properly validates all users. The combination of LIMITLIB and EXEC provides the ability to tailor the TSO command language. Protection is accomplished mainly by hiding essential commands and data sets from the user. At the present time, no file maintenance is performed by CMS, due to the lack of an interface between higher level languages and DAIR (Dynamic Allocation Interface Routine). Several schemes are being studied to implement a file subsystem. The LOGOFF routine gets control when the user issues a logoff request. In Version 2 of the Monitor, abnormal termination such as console shutoff will also be handled. Online accounting and Master File maintenance are provided. The system has proven to be extremely easy to use and maintain, due to its modular design and close resemblance...
to normal TSO. Finally, the overhead has been measured at approximately one dollar per student session, and this figure will be reduced in the next version.

SECURITY AND THE MONITOR

Although the Class Monitor has proved a useful tool in non-computer courses at MIT, it is of limited value in the control of computer-oriented courses due to its vulnerability. It lacks protection in two major areas.

First, because the data sets used to implement the monitor are simply normal user data sets, they may be edited, listed, or deleted by other "full" TSO users. A specific example of this occurred last fall, when a staff member at the Computation Center, in an effort to assist a Class Monitor user, listed his EXEC files using a command not available within LIMITLIB. Thus the code names (and access) to all commands were given to this user. Further, students who are given full TSO access for other projects are free to alter or destroy all of the essential data sets for the Monitor. Therefore, the Monitor is open to sabotage from non-Monitor users.

A second area of vulnerability arises from the fact that all of the programs within the Monitor are written in PL/I, and run in User mode. If we give a PL/I programming class access to a version of the Monitor that includes the ability to edit, compile, run, and delete PL/I programs, we face two problems. First, anything that the Monitor does, they can do (for example, gain access to the User Profile Table, and change the group number to be billed for the session). Second, the logical structure of TSO provides no method of keeping different users' data sets apart in a single USERID, and releasing file maintenance commands gives unlimited access to all CMS data sets. Therefore, we have relied upon the good nature of our students when using the Monitor for more sophisticated applications.

The final section discusses the IBM Resource Security System, and what additional security it will bring to the Class Monitor in the OS/TSO environment.
INTRODUCTION

In May of 1973, the IBM Resource Security System (RSS) was installed at the MIT Computation Center as part of a study on operating system security. RSS is designed as an addition to OS Release 21 which, when fully implemented (and debugged), provides the additional software necessary to "secure" the operating system. Because extremely detailed documentation is available for RSS (12, 13, 14), only a very brief description will be given here.

RSS is a system primarily concerned with data security, and one that clearly reflects its military ancestry (it was originally designed for use in the World Wide Military Command and Control System). System resources (programs, data sets, and terminals) are accessed by users on the basis of security levels, access categories, and need-to-know. Security levels reflect the sensitivity of data, in a manner directly analogous to the military "confidential, secret, and top secret" classification. Data sets are assigned one of eight security levels, and each user has an attribute which sets a maximum permissible level of data access. Access categories provide a means of implementing the concept of a group need-to-know by associating groups of users with groups of system resources. For example, all of the administrative users in a given department might be authorized to the set of confidential files for that department. Finally, in the extreme case, individual users can be authorized to specific data sets on a specific need-to-know basis.

Perhaps the most important reflection of the military design strategy in RSS is the concept of a Security Officer. The control of all security procedures within the system rests with the person (or persons) designated as Security Officer. This control includes the definition of access categories, maintenance of the security profiles of all users, and the control of the authorization procedure for all controlled data sets.

Through the authorization procedure described in 12, users are given rights to data sets on the basis of their code words, which specify access categories, levels, and need-to-know. The RSS System then monitors the use of all controlled data sets and attempts to prevent any access to the system that might subvert the control mechanism.

After taking a brief look at the potential effectiveness of RSS in alleviating the security problems of the Class Monitor, this section will conclude with some comments on the design of operating systems for use in the Classroom Support Environment.

THE POTENTIAL OF RSS

When OS/360 was originally introduced, the designers were very proud of the ease with which data in the system was accessed. It was a very "open" system, and the most flexible available in terms of file system organization. As the need for data protection became clear and TSO was added to the environment, RSS was developed to gain control of the system. For the Class Monitor, this meant the ability to protect its control data sets from outside disturbances, and
further, to authorize the contents of those data sets to specific programs. For example, the accounting file could be authorized only to LOGON, LOGOFF, and the Master File Access Routing, thus preventing access by student programs. However, it should be noted here that even though this is a major improvement in OS/TSO, the result is no advance in the state-of-the-art. Indeed, there is some reason to believe that the design of a time-sharing system in which one user can access another user's files (or even know that they exist) was a terrible mistake at best, and that data set protection in TSO in fact brings the design of the system up to a level just below that of systems such as VM/370, because the user in TSO can still find out that controlled data sets exist from the system catalog.

In the Classroom Support Environment, as noted previously, a very decentralized user community exists. Unfortunately, in RSS only the Security Officer can protect data sets and assign privileges. This military notion of security centralization is in direct conflict with the needs of our environment. This and other problems of RSS in a "service bureau" environment are discussed by Daley (15).

RSS provides no assistance in the other major area of difficulty, that of preventing the user from accessing the User Profile Table and other sensitive control tables during execution. Although RSS in most cases can catch a user before the OS environment is affected, it offers no assistance in maintaining the subsystem environment needed in our application.

In summary, the addition of RSS to OS/360 provides some useful control of sensitive data sets, but fails to provide the mechanisms necessary for subsystem control, such as automatic user exits from various sections of the TMP, and authorization mechanisms that operate without the Security Officer. Much modification of TSO is needed before the Class Monitor can be secured, and RSS must be extended to include more specific authorizations, such as the execute-only access and program-program-file permission discussed by Daley. RSS, while solving many data set access problems, falls short of the requirements of the classroom environment.

CONCLUSIONS

Experience in the Classroom Environment at MIT has shown that a decentralized approach such as the Class Monitor System provides the necessary simplicity and computing power for students, as well as the control required for administrators. Severe problems occur, however, in attempting to provide adequate subsystem integrity. It seems clear that any operating system that intends to service this environment must include mechanisms for subsystem implementation which include access (either by user exits or open entry points) to most major modules of the system. File systems, accounting systems, command processors, and many other areas must be available to provide adequate subsystem security.

In conclusion, we should note that there are some viable alternatives to the OS environment for Classroom Support. Madnick and Donovan (16) make a strong case for the use of virtual machine systems in areas where security is a problem. Certainly this idea is appealing in attempting to isolate computer research, for example, from other campus activities. Even with a class virtual machine, however, some mechanism will be required to protect members of that class from one another, making some subsystem necessary. The MULTICS design (17) makes the implementation of subsystems somewhat easier.
The needs of the Classroom Support environment, then, are mainly in the area of subsystem security. Hopefully, the work of the RSS Study Group at the Sloan School of Management will provide a framework for analyzing these needs. Such a framework will assist designers in meeting the security needs of the classroom support environment more completely.
REFERENCES


2. Interview with Joseph Patten, Director, Office of Administrative Information Systems, MIT, on April 26, 1973.


5. IBM System/360 Principles of Operation, IBM Corporation, Form GA22-6821.


DATA SECURITY IN THE SERVICE BUREAU INDUSTRY

Grant N. Smith

CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 INTRODUCTION</td>
<td>73</td>
</tr>
<tr>
<td>1.1 Questionnaires</td>
<td>73</td>
</tr>
<tr>
<td>1.2 Overview of the Industry</td>
<td>73</td>
</tr>
<tr>
<td>2.0 ANALYSIS OF MECHANISMS</td>
<td>75</td>
</tr>
<tr>
<td>2.1 Security at Login Time - Authentication</td>
<td>75</td>
</tr>
<tr>
<td>2.2 User-controlled Run-time Security</td>
<td>77</td>
</tr>
<tr>
<td>2.3 System-controlled Run-time Security</td>
<td>77</td>
</tr>
<tr>
<td>2.4 Enforcing Access Control to Data Sets</td>
<td>78</td>
</tr>
<tr>
<td>2.4.1 Access-type Specification</td>
<td>78</td>
</tr>
<tr>
<td>2.4.2 User Categorization</td>
<td>78</td>
</tr>
<tr>
<td>2.5 Monitoring and Back-up</td>
<td>79</td>
</tr>
<tr>
<td>3.0 EXPLANATION OF FINDINGS</td>
<td>79</td>
</tr>
<tr>
<td>4.0 ACTION FOR THE FUTURE</td>
<td>80</td>
</tr>
<tr>
<td>5.0 CONCLUSION</td>
<td>81</td>
</tr>
<tr>
<td>6.0 REFERENCES</td>
<td>82</td>
</tr>
<tr>
<td>7.0 APPENDIX 1</td>
<td>83</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

The Service Bureau Industry has been included in the user requirements survey because there is good reason to believe that most computer installations of the future will be run as service bureaus of some kind. In universities, this has been the case for some time. There is a definite trend in industry toward running the EDP division as an in-house service bureau. Returns to scale, in terms of both personnel and hardware, justify this trend.

We have primarily focused our attention on time-shared and remote job entry service bureaus. These seem to be most advanced in terms of security facilities, in comparison with conventional batch operation.

1.1 Questionnaires

A questionnaire was developed to ascertain the attitudes of the service bureau industry. A sample copy of this questionnaire is included in Appendix 1.

Due to time constraints, the sample surveyed was quite limited and, as such, did not yield statistically significant data. Thus, in the observations that follow, the questionnaire data was augmented by personal interviews and the experience of members of the Sloan Information System Security Project.

In addition to a general study of the service bureau industry, five service bureaus were studied extensively:

- Interactive Data Corp. - IDC
- National Computer Software Systems - NCSS (3)
- First Data
- Computility
- Interactive Sciences Corp. - ISC

These service bureaus are considered representative of the range of services typically offered.

1.2 Overview of the Industry

The service bureau industry is one which has recently experienced, and continues to experience rapid growth. It is also a highly competitive industry. There has been much talk of late about regulation in a non-economic sense - namely by introduction of legislation that will exert control over all personal data, be it in computerized data bases, or not. The service bureau may be required to provide a sufficiently secure system in which to keep such data. The ability to do so may well appear to have secondary, economic regulatory effects (1).

An outstanding facet of the service bureau industry is the wide variance exhibited on a number of fronts. Some of these fronts are:

- Users
  The users of the service bureau range from the most naive, to the most sophisticated of computer users.
Size of Operation
There are several very large, country-wide service bureaus - two examples of which are IDC (Interactive Data Corporation) and NCSS (National CSS) - and a myriad of small-to-medium companies - such as First Data and Computility.

Type of Service
Services offered by service bureaus range from selling raw machine time to selling the use of proprietary programs and data bases.

An example of the latter category - sale of the use of proprietary data bases - is IDC. IDC, along with various third parties, supplies a variety of packages which the user may use on the IDC-supplied data base. The use of this proprietary data base is one of IDC's primary services.

There is belief expressed by some that proprietary programs and data bases will become the major service of the industry.

Customer Base
As a result of these varied services, the service bureaus cater to a widely varying customer base.

The classes of customer correspond to the classes of service offered by the industry: some buying only raw machine time; others buying the use of data bases; others buying the use of various packages.

All of these different classes have different security needs, and are at different levels of sophistication. But more of this later.

Despite the large variance in such factors as users, services, and size of operation, certain findings were common to all the service bureaus surveyed:

1. There is not much variance in the responsibility to the customer which the service bureau agrees to assume. All those service bureaus surveyed undertook to take "all reasonable precautions" against loss or disclosure of user data. It does not take a wary eye to notice the great legal latitude allowed by the word "reasonable."

This essentially places the responsibility with the user; he must decide whether to make use of the supplied security facilities.

2. The supplied security mechanisms are often site implemented and site maintained. These mechanisms augment those supplied by the manufacturer, and are often more sophisticated.

3. Almost without exception, these security mechanisms are difficult to use, given the naivete of the average user. As one service bureau official described the situation: "With respect to security, some of our customers are sensitive, some are not -- but all are naive!"

4. Users of the service bureau are far more security conscious in word than in deed. Despite the general availability of security mechanisms, the great majority of users make little use of them.
Estimates ranging from 30-90 percent of the resources of a service bureau are consumed by about 10 percent of its clients. This 10 percent is a sophisticated set of users, for the most part, and tends to make extensive use of the available security mechanisms - often augmenting those supplied by the service bureau, with their own.

2.0 ANALYSIS OF MECHANISMS

We stated above that the service bureau industry in general - and this includes all those surveyed - make available a wide range of relatively sophisticated security mechanisms.

In this section we will present a breakdown of the security mechanisms offered by those service bureaus surveyed. A detailed summary of similar data for four of the service bureaus included in the survey appears in Table 1.

The discussion of the security mechanisms available on the systems surveyed will be divided into five parts:

- Security at login time
- Security at run-time controlled by user routines -- e.g., traps.
- Security at run-time controlled by system status-indicators
- Control of access to data sets
- Monitoring and Back-up facilities.

2.1 Security at Login Time - Authentication

Standard on all systems surveyed is the common "user-ID/Password" combination for identification of the user.

The facility for associating several user-IDs with a Project-ID is also common.

In addition to these well-publicized login procedures, there are several other mechanisms that are quite common, but not as well publicized. These include:

- time limits for logging in - if the user takes too long to log in, the system hangs up. This is to make it unpleasant for the user to try many different IDs or passwords.
- on some systems (e.g., NCSS) only one user of an ID may be active at any time.
- passwords are generally easily changed if a user should suspect that his password has leaked out.
- many systems permit specification of terminals from which a password will be accepted. This allows use of physical security - lock the terminal away - to augment the password mechanism.
- the user has the option of "extending" the login procedure (e.g., the "PROFILE EXEC" of NCSS). This takes the form of automatically invoking a set of user-written commands when someone attempts to log in under a certain ID, that will attempt to further authenticate the user attempting the login. The commands may request additional passwords, or other information,
<table>
<thead>
<tr>
<th>TECHNOLOGICAL ENVIRONMENT</th>
<th>NCSS</th>
<th>ISC</th>
<th>FIRST DATA</th>
<th>COMPULUTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>7 complete IBM System/360/370's</td>
<td>2 complete PDP-10 systems</td>
<td>Dual processes</td>
<td>Single processor</td>
</tr>
<tr>
<td></td>
<td>180 2314 Spindles</td>
<td></td>
<td>PDP-10 with KA-10</td>
<td>PDP-10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU's</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Single processor</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PDP-10 with KI-10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU (virtual memory)</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>VP/CSS Operating System (superset of CP-67/CMS)</td>
<td>DEC PDP-10 Monitor with modifications</td>
<td>DEC PDP-10</td>
<td>DEC PDP-10</td>
</tr>
<tr>
<td>Processing Mode</td>
<td>Batch and Online</td>
<td>Batch and Online</td>
<td>Batch and Online</td>
<td>Batch and Online</td>
</tr>
<tr>
<td>Security Features</td>
<td>ID use limits; time limits; PROTECT EXEC; PROFILE EXEC; Attention key disabled</td>
<td>File use authorization; time limits; Mode of use authorization; PSW bit set to disable functions; user file directory lock</td>
<td>KI-10 hardware protection; time limits; PSW bit set to disable functions; encrypting; Password-terminal associations; Mode of use authorization; File use authorization; PSW bit set to disable functions</td>
<td>Password (easily changed); Access control lists on files; Mode of use authorization; File use authorization;</td>
</tr>
<tr>
<td>Integrity Features</td>
<td>Monitor file changes; Daily backup</td>
<td>Accounting System integrity; Daily backup; Continual backup of changed files</td>
<td>Daily backup; Offsite weekly backup</td>
<td>Daily backup</td>
</tr>
</tbody>
</table>
before allowing the user into the system. This facility may also be used to prevent the user from using certain system commands or facilities.

These mechanisms for restricting access to the system as a whole, are extremely powerful, but not widely used by the average service bureau client. Many of these techniques (e.g., "PROFILE EXEC") require a moderate degree of sophistication to make effective use of them.

2.2 User-controlled Run-time Security

These facilities are those available to the user to more closely specify the environment in which his routine will run. Examples are:

- With the NCSS "PROJECT EXEC", any attempt to attach a file on a proprietary disk automatically invokes a set of commands written by the file owner. These commands, stored as a file on the protected disk, may simply ask questions to further verify the user's identity, or may invoke actual programs which may be written in any of the supported languages (e.g., COBOL, FORTRAN, PL/1, Assembler). Furthermore, while the user has this file attached, special accounting is done in the supervisor to charge separately for the use of this protected file - should the file-owner so wish.

- The ISC system recognizes the file name extension of ".PRO", and sets a status indicator which disables:

  - the "examine" feature
  - the "deposit" (update) feature
  - the "save" feature (essentially the same as "copy")

- The user often has the option of setting status indicators that prevent the use of certain features, such as DEC's Dynamic Debugging Tool (DDT).

2.3 System-controlled Run-time Security

These features include:

1) Disable the "attention" key while executing a "PROFILE"--or "PROTECT EXEC" (on NCSS).

2) Traps to accounting routines if, after interrupting execution, the user attempts anything other than "RESTART".

3) On the ISC system, there is a system routine that unlocks a user directory for the life of the calling program only, if the program name ends in ".PRO". In this way, data files may be associated with a program. Only that program can access the data file while it is running.
2.4 Enforcing Access Control to Data Sets

Control of access can be achieved by the conventional restriction of access to a data set, or by rendering the data (or program) unintelligible.

This can be done by encryption. Many of the systems surveyed offer an encrypting package for a very nominal fee.

Restricting access to data sets (in the conventional sense) consists basically of two steps:

- the ability to differentiate between different types of access
- the ability to specify to whom each type of access applies.

2.4.1 Access-type Specification

The NCSS system (IBM 360/67) recognizes two types of access:

- read only
- write access

DEC's PDP-10 (KI-10 processor) systems recognize four types of access:

- read
- write
- execute
- append

Both the 360/67 and PDP-10 allow complete restriction of access.

The ISC system recognizes different types of access to a user directory:

- read (i.e., use the directory)
- write (i.e., modify the directory)
- create
- destroy

2.4.2 User Categorization

The NCSS system categorizes users by the "user-ID/password" combination. There is no notion of a group of users being treated as a "project".

The DEC PDP-10 systems categorize users into three groups:

- the owner of the file
- other members of that project (programmer number)
- all other users.

Specification of security (as in 2.4.1) can be applied to each of these three classes separately.

Much of the enforcing of these access restrictions is carried out by the hardware.
2.5 Monitoring and Back-up

All service bureaus surveyed have, in operation, daily backup of all files from disk to tape.

Some generate weekly backup tapes which are stored.

Some systems (NCSS, ISC) monitor file changes continuously, and keep this information in the system to allow recovery of a quicker nature than reloading all files from tape.

3.0 EXPLANATION OF FINDINGS

Given that we have the mechanisms described in Section 2.0, which are certainly commensurate with the state of the art, a question poses itself:

"Why is it that these sophisticated techniques exist in the service bureau industry, and not, in general, outside of it?"

By way of providing an answer, we offer the following explanations:

* The people that conceive of, and run service bureaus, are in the main, technical people. Couched in terms of the belief that security consciousness is a function of the level of awareness, these are people who are operationally close to the computer, and so would be expected to exhibit a high degree of concern. Being technical people, they have the wherewithal to bring into existence such mechanisms as will, in their experience, provide adequate security.

* The very nature of the service bureau environment requires that the system be protected from the user, and that users be isolated from each other. This is strictly enforced, except in cases where specified access control allows sharing of programs or data files. Such a requirement is mandatory if time-sharing is to be possible.

There is much evidence of personal, and proprietary data being kept in the system. Therefore, some form of access control is necessary.

The service bureau is a natural outlet for the software vendor, or the person selling the use of a data base. The seller may be the service bureau itself, or a third party. In either case, there is definite concern on the part of the seller for the safety of his product.

This goes a long way toward explaining the existence of security mechanisms in the industry.

* As opposed to all the other industries surveyed, the service bureau industry is the only one in which the use of computers is the "raison d'être" of the industry. This means that there were no "traditional" approaches to dictate policy, as there were, and still are, in industries such as the Financial and Medical.
The service bureau industry began with a "clean slate", so to speak.

We have implied throughout that the available mechanisms are not often used by the average service bureau user. This, too, begs a question: "Why is this so?" Again, in answer, we propose the following:

- The available mechanisms were designed and implemented by technical people, for technical people. The average user of a service bureau is not a technical person.
- Security is an option for which the user must choose to pay - in terms of both effort and dollars. It is an option because it is generally an "add-on" to the system, rather than an integral part of the design.

The sophisticated 10 percent of service bureau clients is prepared to expend the effort and the dollars to make use of the available mechanisms for two major reasons:

1. They have the ability to understand and make use of these security mechanisms.
2. They are aware of their vulnerability if no precautions are taken to protect their programs and data.

4.0 ACTION FOR THE FUTURE

We have described a situation in which a variety of sophisticated mechanisms exist, but are not widely used. By way of understanding this situation, we proposed the explanations of Section 3.0.

Now we reach the stage where we are faced with a decision; a decision which will set the scene for the immediate future. What courses of action are open to the service bureau industry?

One course is simply: Don't do anything.

No one has done very much to date, beyond making security mechanisms available, and things seem to have worked out fairly well. There have been a few cases of disgruntled users, stolen data sets, and lost data, but in the main, the situation has been far from serious.

Inaction, however, is not what we advocate. There is a definite goal that can be set, consisting of two major parts:

1) Ease of use of the available mechanisms be brought to a level where the average user can understand, and make effective use of these facilities. This includes:

   - make security an integral part of the system, which will lead to the situation in which security is not an option. In cases where this is already so - e.g., in DEC PDP-10 systems in which the hardware access control features are built into the system - the default should be no access to anyone other than the
owner of the file. Explicit specification of security for cases where sharing is desired is then necessary.

- Once the optional nature of security is removed, there will be great pressure to bring about "simple to use" mechanisms.

This is based on the philosophy that the demise of security as an option will require all users to utilize the facilities. If the user - the naive user - is forced to use the facilities, these facilities will have to be suitably simplified.

2) Education of computer users in general, and service bureau users in particular, should be directed along such lines as will result in a general increase in the level of awareness. There is strong reason to believe that any program with such intent would be well received, given the attention currently being paid this topic by a major report to the Department of Health, Education, and Welfare (2). Legislation in this area is expected to be forthcoming in the very near future.

5.0 CONCLUSION

In summary, we have determined that many sophisticated security mechanisms do exist, and are in use by a small percentage of service bureau users. Their widespread use, however, is clearly not a reality. Several explanations were offered that attempt to explain why use is so low amongst the great majority of service bureau clients. Chief among these is the general naivete, and lack of awareness of the average service bureau client.

Finally, we propose that positive steps be taken to educate the average computer user so as to make him aware of his vulnerability.
6.0 REFERENCES


7.0 APPENDIX 1

SERVICE BUREAU QUESTIONNAIRE

1. Three major computer services are listed below. Select the service(s) which your firm offers, and if there is more than one, number them in order of importance, 1 = most important, etc.
   ___ time-sharing
   ___ batch processing
   ___ proprietary programs and/or data bases

2. How many unique user entities does your primary system recognize? (e.g., in time-sharing, USERID-password combinations, or in batch processing, account-programmer combinations)

3. Do you provide services to firms or individuals who are in direct competition with one another?
   ___ Yes
   ___ No

4. Rate, on a scale of 1-7, the security-consciousness of your users regarding their programs and data.
   not concerned 1 2 3 4 5 6 7 very concerned about security

5. Rate, on a scale of 1-7, the security-consciousness of your staff regarding user programs and data.
   not concerned 1 2 3 4 5 6 7 very concerned about security

6. Rate, on a scale of 1-7, the security-consciousness of your staff regarding your own accounting data and proprietary software.
   not concerned 1 2 3 4 5 6 7 very concerned about security

7. It is often true that a few active users consume a large percentage of a service bureau's resources. Among your users, taking the most active 10 percent by number, what percentage of your resources do they consume?

Massachusetts Institute of Technology Volume 4 83
9. It has been suggested that the most "sophisticated" users of computers generally indicate future trends for other users. To what extent do your most sophisticated users make use of security techniques (either those provided by the system, or techniques of their own design)?

no use 1 2 3 4 5 6 7 heavy use

9. What brand of mainframe is used for your primary system?

___ DEC ___ CDC ___ XDS ___ UNIVAC
___ IBM ___ Burroughs ___ Honeywell
___ Other (specify): ____________________

10. On this system, you use (check one):

___ Standard Operating System software supplied by vendor.
   (Name) ________________________________
___ Modified Version of Standard Operating System.
   (Name) ________________________________
___ Your own operating system software.
   (Name) ________________________________

11. Do you use passwords for user identification?

___ Yes
___ No

12. The file system can best be described as:

___ a master catalog structure, open to access by all users.
___ a tree structure, with open access for search by all users.
___ a tree structure with locks at each level.
___ a decentralized directory structure, with each user's directory neither known nor accessible to others.

13. Does the operating system permit sharing of files

___ so that multiple users can concurrently access a single file in read or write modes?
___ so that multiple users may access, but only if they are all in read-only mode?
___ so that multiple users may access a file, but only one user may have it in write mode?
___ such that only one user at a time may access a given file?
___ such that a separate copy of a file is the only means of sharing?

14. How many simultaneous processes does your primary system support?

___
15. What access do your systems programmers have to user data?

- Unrestricted access
- Restricted access (e.g., by permission of service bureau)
- Controlled access (e.g., no access except by explicit permission of the owner).

16. How are the privileges of your systems programmers administered?

- Automatically, by association with the user identification.
- By special password protection.
- By physical interventions such as special terminals or lines.
- By the standard file system.
- Other

17. If your operating system is modified, or of your own design, note the relative importance of each of the following goals in your design process. (1 = least important, 10 = most important)

- improve throughput or service level
- maximize number of concurrent processes
- interface to special equipment
- protect operating system from user processes
- increase reliability
- provide special accounting or billing
- protect system files
- protect user files
- simplify command language
- simplify file access or sharing

18. Which of the following protection mechanisms are currently available on your system?

- Access control lists
- Multi-level password protection
- Encryption
- Memory protect hardware
- "Trap" programs activated by certain accesses
- Date protection of files
- Read-only access to files
- Execute-only access to files
- Restricted terminals for special access
- Periodic automatic backup
- Virtual machine capability
- Special files containing commands which execute automatically at every logon. (Profile feature)

19. Rate the ability of your system to insulate itself from malicious user tasks.

unprotected 1 2 3 4 5 6 7 completely safe
20. Rate the ability of your system to insulate concurrent processes from one another.

unprotected 1 2 3 4 5 6 7 completely safe
For each of the following situations, choose:

1. Responsibility rests with users
2. Responsibility rests with service bureau
3. Responsibility shared between users and service bureau
4. Responsibility rests with manufacturer of system in use

21. ___ User accidentally deletes his own file, and has not provided a backup copy.

22. ___ System crash destroys an online file belonging to user. User has not provided backup.

23. ___ A user steals a copy of another user's proprietary program.

24. ___ A staff member steals a copy of user's private database.

25. ___ A user taps another user's proprietary program by gaining execute access.

26. ___ An unauthorized person steals a registered user's password, and plays on the system at that user's expense.

27. ___ An unauthorized person steals a registered user's password, and causes the system catalog to be destroyed.

28. ___ A user program causes a system crash.

29. ___ A user modifies a competitor's program in core while it is executing.

30. Suppose that your system was modified to implement an access control matrix, which contained authorization information on various user-file pairs. The matrix would not be practical if permission had to be specified for every user-file combination on the system. For efficiency, there must be a default condition, against which matrix entries are weighed. Mark one of the following schemes:

___ A file is, by default, open to all users, unless specifically designated as restricted in the access matrix.

___ A file is, by default, regarded as the property of its creator, and no other user may access it in any mode unless permission is explicitly granted in the access matrix.

30a. Would you consider such an access matrix a valuable addition to your system (assume no performance degradation)?

negative value 1 2 3 4 5 6 7 positive value
31. The access control matrix must, of course, be continually updated to reflect the extension and revocation of privileges, as well as the addition and deletion of users and files. At what level should the access matrix be maintained? Check one.

___ The access matrix should be accessible only to the system security officer, and all changes should be made through this person.

___ Users should be given the right to set access on any file which they create. The system administrator should only install and remove users.

___ Users should be given the right to set access on any file which they create. However, the system administrator, in addition to adding users, also should have access to all files as a precautionary measure.

31a. In your opinion, what percentage of your users would make use of the security features just described, if the default was no security?

____

32. Does your system process any data which falls into any of the following categories (check all that apply)?

___ sensitive personal data, such as payroll, medical, etc.
___ data which has a government security classification
___ proprietary corporate information, such as marketing.

33. Are any special precautions taken on your system when data of this type is processed (leave blank or check all that apply)?

___ No special precautions
___ All non-essential users barred from system
___ Special operators used
___ Special printers used for sensitive output
___ Other ____________________________

34. Are you aware of any state or federal statutes which affect your operations regarding the handling of personal data?

___ Yes (cite law if possible)

___ No ____________________________

35. In what state(s) does your firm operate?

______________________________
In the next section, several statements are presented. Mark one of the five numbers to indicate your agreement or disagreement with each statement. The numbers are:

1 = strongly agree  
2 = agree  
3 = neutral  
4 = disagree  
5 = strongly disagree

Subjects on whom records containing personally identifiable information are maintained should have the following rights:

36. 1 2 3 4 5 To be informed of the existence of such records when they are started.

37. 1 2 3 4 5 To review on demand the contents of records concerning them.

38. 1 2 3 4 5 To correct, rebut, update, and expunge incorrect or obsolete information concerning them.

39. 1 2 3 4 5 To be furnished periodically with an accounting of the uses made of information concerning them.

40. 1 2 3 4 5 To stop the exchange of information concerning them among information suppliers.

The following actions regarding data banks containing personally identifiable data are necessary:

41. 1 2 3 4 5 Registration as to purpose and contents.

42. 1 2 3 4 5 Standards of hardware and software security.

43. 1 2 3 4 5 Standards concerning the acquisition and dissemination of information.

44. 1 2 3 4 5 Periodic site inspections.

The following persons and organizations trafficking in personally identifiable information should be licensed and certified:

45. 1 2 3 4 5 Data bank proprietors.

46. 1 2 3 4 5 Information brokers (suppliers).

47. 1 2 3 4 5 Data processing centers.

48. 1 2 3 4 5 Computer programmers.

49. 1 2 3 4 5 Data gatherers.
50. Rate, on a scale of 1-7, the ability of currently available operating systems and hardware to provide adequate security over the next three years.

    inadequate 1 2 3 4 5 6 7 adequate

   COMMENTS

   Your comments on our questionnaire are solicited, especially as regards your opinion of its merit as: A) an information-gathering tool for survey work, and B) a consciousness-raising device for you as a participant.
DATA SECURITY IN THE MEDICAL COMMUNITY

Bruce E. Stangle

CONTENTS

1.1 INTRODUCTION .................................................. 93
1.2 STATE OF THE ART IN MEDICAL INFORMATION SYSTEMS .......... 94
  1.2.1 MEDICAL CENTER 1 (MC1) .................................. 97
  1.2.2 MEDICAL CENTER 2 (MC2) .................................. 97
  1.2.3 MEDICAL CENTER 3 (MC3) .................................. 97
  1.2.4 MEDICAL CENTER 4 (MC4) .................................. 98
1.3 PAST EFFORTS IN MEDICAL DATA SECURITY ....................... 99
2.1 ANALYTIC FRAMEWORK ............................................. 100
2.2 METHODOLOGY ................................................... 101
  2.2.1 HYPOTHESES ................................................ 101
  2.2.2 PRINCIPLES OF MEASUREMENT ................................ 102
3.1 THE SECURITY ATTITUDE SURVEY ................................ 103
4.1 CONCLUDING REMARKS ............................................ 104
5.1 REFERENCES .................................................... 105
6.1 APPENDIX ....................................................... 106
1.1 INTRODUCTION

The use of computers in health systems is becoming more extensive. Currently, computers are used in such varied application areas as payroll and patient monitoring systems. Because health expenditures form such a large part of the U.S. Gross National Product (GNP) and since medical costs have recently risen so rapidly, it will be important for this sector to develop even more effective means for applying computer technology in the future. U.S. health expenditures were 7.4 percent of the GNP in 1971. Furthermore, health-care costs have been rising at a rate that is twice the average increase in costs in the economy. This is a more rapid rise than any other component of the consumer price index. It seems increasingly important in these times of growth and innovation to evaluate the scope and breadth of automation so that fundamental institutions and relationships are maintained despite technical change. As applications of computers increase, one must recognize that computers can be subverted for means which were never intended.

The information processing capabilities of the computer allow health organizations to consider large-scale applications which could potentially revolutionize patient care, but which also present serious problems in the area of individual privacy. It is not outside the realm of possibility that in the future all patient records from every hospital, clinic, or other health organization could be shared. The advantages of such a data base are clear when a doctor must treat a new patient or especially in an emergency when the doctor is not familiar with the patient's history and the patient is not able to give this information. Equally as clear, though, is the potential threat to personal privacy present in such a database. Without proper protection, the database could be read by anyone with access to the computer.

The aim of this paper is to analyze and discuss some of the issues relating to data and information security in the medical environment. This analysis will include an assessment of the state of the art in computer usage, a brief review of the literature, a framework for analysis, a statement of the problem, some tentative hypotheses, and a discussion of the questionnaire that will be used to test these hypotheses in a medical setting. It is important to note that the findings and observations reported herein are part of an ongoing research effort.

Before proceeding further, it is best to define some of the terms that will be used. Data security is a term that denotes safety or freedom from any threat. Data, itself, can have several states or qualities. Data can be private, confidential, operational, or public knowledge. Common threats to data are disclosure, modification, or destruction whether they be accidental or malicious. Often, such countermeasures as passwords, encryption, and physical security are employed against these threats. In Exhibits 1 and 2 these terms are defined.

At this point it also is useful to provide the reader with some examples of threats to medical information systems. The following three scenarios are suggested as possibilities:

1. Many medical facilities have large automated files of patient demographic information. Suppose a computer programmer who was heavily in debt decided to sell all
patient names and addresses to a direct mail advertising firm.

2. The use of automated patient scheduling systems is expanding, especially in outpatient or ambulatory clinics. Doctor's schedules are stored in the computer and can be accessed or updated via a CRT or television like device. Consider the impact of appointment information leaks. Suppose a business firm sends their employees to a clinic and one of the patients notices on the CRT screen that his boss is scheduled to see a psychiatrist.

3. Many medical facilities are automating patient medical records. Assume a patient were allergic to penicillin but for some reason (data input error, accidental modification, or deliberate tampering) the patient's automated record reports no such allergy. The patient goes into shock and dies after a penicillin innoculation administered by a doctor who thought the medical record was correct.

These scenarios are meant to offer a context in which to view the medical data security problem. They represent threats to information which are possible. It should be noted that many of these same threats would pertain to non-computer systems as well. The difference is that with computerized systems one may be able to bring data security under tighter control and thus successfully deter these threats.

1.2 STATE OF THE ART IN MEDICAL INFORMATION SYSTEMS

Before analyzing medical data security, it is necessary to assess the extent to which computers are actually being used by those in the medical community. (For a general overview of current computer applications in medicine, see Ryan and Monroe (9).) In order to make a thorough assessment of the state of the art in computer usage, a series of field interviews were conducted. Sites were chosen so that a reasonably diverse set of organizational criteria could be observed. This diversity will become apparent as one reads through the brief site visit summaries that follow.
Key Definitions and Concepts in Data Security

Private: Data are private if they relate to a specific individual and should not be known by anyone but that individual. For example, one could claim that the amount of one's contribution to a charitable fund was a private matter of concern to no one else.

Confidential: Data are confidential if they can be shared openly within the context of a professional relationship, i.e., doctor-patient, manager-employee, etc. However, the data should not be available to anyone outside this professional relationship.

Operational: Data are operational if they are needed for the normal functioning of a business. The information should be available to all personnel who require the information in order to perform adequately their job(s).

Public Knowledge: Data are public knowledge if they can be openly known by all persons either within or outside a business.
Exhibit 2

Common Security Threats and Countermeasures

Common threats against data security are computer installation sabotage, accidental system breakdown, fraud, embezzlement, interception errors, disclosure of data, theft, sabotage or unauthorized copying of data. Data security can be created and maintained by some or all of the following elements:

**Technical Protection (automated):**
- Computer system integrity (operating system, backup power, fire protection).
- Remote access control (terminal authorization, user identification).
- Data encoding (encryption).

**Procedural Protection (manual):**
- Physical access control (guards, badges, locks, etc.).
- Data handling rules (offsite storage, written requisition of storage volumes).
- Program modification rules.
- Input/output separation.
- Input/output controls.
- Audit.

**Personnel Protection:**
- Preemployment screening.
- Supervision.
- Division of responsibility.
1.2.1 MEDICAL CENTER 1 (MC1)

MC1 is a large, fee-for-service, specialty clinic that has used computers for many years to improve the delivery of ambulatory medical care. Numerous applications have been undertaken on both the medical side and the management side of operations. Examples of each are:

<table>
<thead>
<tr>
<th>Medical</th>
<th>Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Medical History</td>
<td>- Appointment Making</td>
</tr>
<tr>
<td>- Diagnosis History</td>
<td>- Billing</td>
</tr>
<tr>
<td>- Laboratory</td>
<td>- Accounts Receivable</td>
</tr>
<tr>
<td>- Research</td>
<td>- Payroll and Personnel</td>
</tr>
<tr>
<td>- Minnesota Multiphasic</td>
<td>- Patient Data Base</td>
</tr>
<tr>
<td>- Personality Inventory</td>
<td>- Budget</td>
</tr>
<tr>
<td>- Other Test Results</td>
<td></td>
</tr>
</tbody>
</table>

One of the most important systems for any outpatient facility is its appointment-making function. Because the amount of physician time available for scheduling with patients is a critical resource, the organization must effectively manage this area or face a loss of revenue. MC1 has developed an online appointment system that permits coordinators to query doctor availability files while talking on the phone with patients. Key features of the system are the dynamic update capability and the automatic generation of numerous hard copy reports such as the doctor's daily schedule. This system alone has allowed MC1 to provide service to many more patients without having to increase manpower substantially.

1.2.2 MEDICAL CENTER 2 (MC2)

MC2 is a university health plan serving over 10,000 students, staff, and faculty. Currently, MC2 is converting to a new information system with the following planned features:

1. A patient master file for storing details on all patient activities. Data included would be medical history, diagnosis, treatment, and follow-up.

2. A pseudo-diagnostic routine would match patient symptoms with a medical reference file and would suggest to the physician the possible diagnosis and tests.

3. A general analysis system would scan subsets of the history file for trends in the health of the patient population and would evaluate the effectiveness of certain drug treatments.

4. Anonymous information would be made available to university researchers who might want to use the clinical data base.

1.2.3 MEDICAL CENTER 3 (MC3)

MC3 is a newly established, prepaid, group practice or more popularly a health maintenance organization (HMO). HMOs are an emerging mode of delivering health services that stress the aspects of preventive health maintenance and comprehensiveness. An important application at MC3 is the online medical record system. This is an
heroic undertaking which has not been successfully implemented at more than a handful of the nation's medical institutions, primarily because of the unwieldy nature of most medical records. The MC3 approach has been to segment the record into two parts -- encounter reports and status reports. The former detail the results of any physician visit and the latter summarize the up-to-date health of the patient.

1.2.4 MEDICAL CENTER 4 (MC4)

MC4 is a large, urban, voluntary hospital. Within MC4 are two separate information processing centers. One unit focuses entirely on medical applications, while another develops administrative systems. For the past decade both groups have been working on innovative methods of applying computer technology to delivering in-patient medical care. An important product of this effort is a high-level, interpretative, computer language with special hospital-related features. Some of the representative systems tackled by the MC4 medical unit are:

- Clinical Laboratory System
- Sequential Problem Solving
- Computer-Based Examinations for Medical Students
- Radiology Report Generation System
- Computer-Based Medical Record for Intensive Care Unit
- Computer-Assisted Acute Respiratory Care
- Diabetic Ketoacidosis Program

Many of these systems although initially developed at MC4 are now being implemented at other medical centers throughout the country.

These, then, are the four organizations that were visited by the study team. The sites were selected both for their diversity in objectives and for their pursuit of effective computer applications to medicine. These visits demonstrated that a dichotomy exists in most medical information systems between applications primarily in support of management or administration and applications primarily in support of delivery of medical care. The knowledge gained from these visits facilitated the construction of a framework suitable for analyzing the security issue in medical organizations. Before we consider this framework, however, a brief review of the literature of medical data security is needed.
1.3 PAST EFFORTS IN MEDICAL DATA SECURITY

Not a great deal of work has been done specifically in the realm of data security in the medical community. The majority of the efforts to date have a strong legalistic bent and expound upon the importance of maintaining basic patient rights to privacy.

Curran et al. (2) advocate the adoption of a code of ethics and clearly defined rules and regulations to govern the protection of information in all health data systems. A similar recommendation is made by the U.S. Department of Health, Education, and Welfare (HEW) in its request for a Code of Fair Information Practice (7). The key safeguard requirements of this proposed HEW code are:

- There must be no personal data record-keeping systems whose very existence is secret.
- There must be a way for an individual to find out what information about him is in a record and how it is used.
- There must be a way for an individual to prevent information about him that was obtained for one purpose from being used or made available for other purposes without his consent.
- There must be a way for an individual to correct or amend a record of identifiable information about him.
- Any organization creating, maintaining, using, or disseminating records of identifiable personal data must assure the reliability of the data for their intended use and must take precautions to prevent misuse of the data.

Debate on the merits of these recommendations is under way in Congress, and reliable opinion is that a law may be passed soon in this area.

Freed (4) takes an in depth look at the legal aspects of using computers in medicine. He expresses that because hospital record systems offer less valuable returns than such computer crimes as check or credit forgery, less rigorous security systems will probably be acceptable in automated health systems.

Two instances of private institutions analyzing the problem of information security have been reported by Hulac (5) and Davis (3). The former describes an urban, comprehensive, health system in Denver, Colorado. A list of standards for ensuring the proper handling of patient data is presented. The latter reports on the Kaiser-Permanente Health System, a large, nationwide, prepaid, group practice, that has been very active in the application of computer technology to medicine.

Except for the above, the area of medical information security has received little attention (1, 6, 10, 11, 13, and 14). Our present intention is to provide a more systematic analysis of the requirements for data security in the medical industry. To achieve this end it is necessary to compile a framework suitable for analyzing health information systems.
2.1 Analytic Framework

From a study of numerous user sites in various industries, an MIT Sloan group (12) developed a framework consisting of these major components:

1. Organizational Environment
2. Data Environment
3. Technological Environment

The major policy variable to be analyzed is the amount of requisite security for any given information system. Security features have some cost associated with them. The amount of time and money expended on security by an organization is strictly a function of the three independent variables: organization, data, and technology. By carefully examining the dependent variable, security requirements, in this manner one is then able to make comparisons across organizations and across groups within the same organization. We will return to this later.

At least four means are available for expressing the organizational variable. These measures are organizational activity, organizational size, organizational goals, and organizational structure. In the health sector a great diversity exists across each of these measures. For example, although all medical organizations are concerned with the improvement or maintenance of health, this goal can be met by any number of activities. Some organizations serve only a small group of patients; others try to meet the needs of entire communities. Many organizations are active in providing inpatient (hospital-based) care and others concentrate on outpatient or ambulatory care.

With respect to the second independent variable, data, there are two dimensions which describe its nature. These are data types and data attributes. Data types refer to data which are either personal (i.e., medical histories) or not personal (i.e., software). The distinction here is sharp: either data are personal or they are not. Data attributes, on the other hand, are more relative as they refer to the private, confidential, operational, or public nature of the data (see Exhibit 1). In health systems there are generally large files of personal data relating to all aspects of a patient's medical and financial history. With the high demand for these data by different groups with varying needs-to-know, it becomes an important matter if one can differentiate on a confidentiality scale between personal data relating to diagnoses and personal data relating to payments. In other words, to construct the data access rights for any information system, one must, first, determine who the individuals or groups are that require access, second, the attributes of the data must be rigorously specified, and third, the links between data attributes and individuals can then be mapped out.

The third independent variable, technology, is included because it is clear that the need for security can be to a large extent dictated by the type of installed configuration. For example, a time sharing system using remote terminals connected to a computer via telephone lines has a number of security threats to deal with that do not exist in a batch system which is physically isolated from the outside world. Similarly, the operating systems of the major computer manufacturers generally are more or less vulnerable to different threats.
This then is a framework which facilitates data security analysis. The three key independent variables of organization, data and technology are the major determinants of a user's requirements for data security. We now turn to an application of this framework.

2.2 METHODOLOGY

Any scientific investigation moves from initial observations about some condition to more refined, general statements that attempt to explain the nature of a system. This process of defining the problem, testing the hypotheses, and stating the theory has been employed in this analysis as well. Before we could adequately define the problem, however, it was necessary to talk at length with those in the field who were working with medical information systems. Numerous interviews were conducted with physicians and managers in health organizations actively involved in automating the process of delivering medical care. From these discussions emerged a good feeling for the problems concomitant with using computers in medicine. The best way to state the problem is that current users of computer technology are uncertain as to what level of security is needed for their system. This uncertainty exists because computer growth in the industry has been rapid with an attendant lack of planning for security. Uncertainty also exists due to the general lack of experience with data security risks.

2.2.1 HYPOTHESES

In order to examine the dimensions of this problem we developed a number of hypotheses. Some of the hypotheses are general in that they apply to data security in any context whether it be the financial, educational, or medical setting. Other hypotheses relate more specifically to the problem of medical data security. The more general hypotheses are:

G.H1: Security demanded by a user is a function of the user's awareness of security as a problem.

G.H2: In an organization an individual's proximity (in terms of job function) to the computer system will determine his awareness of security as a problem.

G.H3: Differences in levels of security awareness and in approaches to the security problem are a function of the nature of processing in an industry and the perceived value of the information being processed.

Besides making inferences from these, an important part of our work has been to investigate the special problems of the medical industry. From our work in the field, it became clear that the following areas warranted further study:

- The threat of computers to confidential relationships.
- The perceived adequacy of present medical data security systems.
- The sensitive nature of medical data.
- The access rights of different individuals to medical data.
These four areas of concern capture the essence of the uncertainty problem.

The above areas can be stated more formally as testable hypotheses. These are the hypotheses specific to the medical industry:

**M.H1:** Computers by their very nature are perceived as a threat to confidential relationships, e.g., physician-patient, manager-employee.

**M.H2:** As computer usage expands in the industry in the future, there will be an ever increasing need for security.

**M.H3:** In a Medical/Management Information System numerous types of data are processed and stored from employee payroll to patient diagnosis. These data items are fundamentally different in nature and content, some being public knowledge and others being highly confidential. It is hypothesized that:

1) these different data types can be identified,
2) the different data types have varying needs for security due to their various degrees of sensitivity, and
3) the access rights of an employee to different types of data is a function of his need to know and the relative sensitivity of the data.

**M.H4:** Most physicians and managers see the role of the computer dramatically expanding in their institutions over the next few years. As this growth takes place it will be incumbent upon computer manufacturers to provide assurances to medical users that information can be secure from interference. Without these assurances, the medical industry may become even more concerned that security threats constitute potential breaches of medical ethics.

2.2.2 PRINCIPLES OF MEASUREMENT

In order to test these hypotheses, many of which deal with attitudes, it was necessary to construct a measurement tool. This tool had to adhere to a prescribed set of principles. Let us examine these principles briefly before going on to consider the tool itself. (For a more detailed discussion of this entire subject, see Oppenheim (8) which is summarized below.)

1. Undimensionality or homogeneity -- One thing at a time must be measured not three. If one is measuring length then you cannot measure temperature or viscosity. A problem with attitudes is that questions may stimulate numerous unmeasurable (or unanticipated) responses.

2. Linearity and equal intervals or equal-appearing intervals -- A scale should follow the straight line model and a scoring system should be devised with interchangeable units. Attitude scales assume linearity (although this may be inadequate), but the creation of scoring units is difficult, and they are not generally interchangeable.
3. Reliability -- Measurements taken today and next week should be identical provided the object has not changed. Consistency can be achieved by greater length and diversity in attitude scales, but complete consistency is difficult to achieve since people are bound to react differently to a scale when they are confronted with it a second time.

4. Validity -- Are we measuring what we thought we were measuring? One may obtain unidimensionality by keeping only those items which intercorrelate highly, yet the scale may not measure what we want it to measure. For example, instead of measuring authoritarianism it may just be a measure of acquiescence.

5. Reproducibility -- This principle relates to a continuum of attitudes and cumulative scaling. This may not be required when dealing with constant and interchangeable units, such as pounds or inches, but, say, if one is dealing with symptoms of a disease it is helpful if the symptoms could be scaled in terms of seriousness. In that way the presence of symptom D would dictate that a patient also must have symptoms A, B, and C.

In attitude research with questionnaires these five principles must be observed. Now, we proceed to consider the actual measurement tool.

3.1 THE SECURITY ATTITUDE SURVEY

We have stated the problem with respect to data security in medical information systems, and we have presented a list of hypotheses to be tested. The actual test of these hypotheses is achieved by administering a set of questions to those who work directly in the medical community. We have put together a five-part Security Attitude Survey which assesses the needs of a particular organization for data security (see Appendix for the complete document). The questionnaire is divided into sections as follows:

1. Computer utilization -- this section reveals the extent of the subject's familiarity with computers and any biases pro or con they may have toward computers in medicine.

2. Security and Privacy in the 1970's -- these questions deal with general topics about perceived threats of computers to confidential relationships and the security of computerized data vis a vis other forms of data.

3. Computer Security at the Medical Center -- this section asks the subject to specify the nature of a given number of data items. The respondent must differentiate between data which are private, confidential, operational, and public knowledge. Next, the relative sensitivity of each data item is rated on a scale from extremely important to extremely unimportant. Finally, an access control matrix is presented which requires that a type of access be specified for each person to each data item. For example, all employees may be permitted to have access only to summaries of patient diagnoses. This is the most crucial section of the questionnaire.

4. Respondent Profile -- Job functions and years of employment are queried.
5. Comments.

This tool has been extensively pre-tested to eliminate ambiguity in wording and to further refine the measures, i.e., as far as possible we have adhered to the principles of measurement. It is now being administered to a sample of physicians and managers at a health institution in the Boston area. Results from this test and from tests at another site will be forthcoming, thus allowing for the rejection or acceptance of the underlying hypotheses.

4.1 CONCLUDING REMARKS

This paper has sought to describe an on-going research project dealing with data security in the medical environment. Results of several site visits have been forwarded as a means for establishing the state of the art in medical information systems. The literature on the subject, sparse as it is, has been reviewed. A framework for analyzing the matter of data security has been described. It was stated that the major problem with medical data security is the uncertainty surrounding the decision as to the required level and degree of security. A number of general and specific hypotheses have been advanced which seek to explain the data security problem. These hypotheses are currently being tested by means of questionnaire research. Once the responses are in and have been analyzed, a thorough report of the findings of this study can be made.
5.1 REFERENCES


6.1 APPENDIX

Security Attitude Survey
The Medical Center (MC)

As you know the Medical Center has been doing things with computers for a number of years. Currently the computer is used to help in such different areas as printing the payroll, assisting in the appointment scheduling process, and recording answers to patient medical histories. It is probably quite likely that you have had several experiences in the past with the Medical Center computer system.

This questionnaire is designed to survey your attitude on some of the important issues raised by the uses of computers in medical organizations. Please complete the questions as carefully as possible. Less than thirty minutes of your time will be required to finish the survey.

Part_1: Computer Utilization

1. My work is such that I often come in contact with computerized information. (Circle one.)

   Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

2. The use of computers in medicine offers little promise for providing a higher level of care to patients and for achieving greater efficiency in managing health institutions.

   Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

3. In the next few years at the Medical Center, the computer will play a larger and larger role in assisting in the delivery and management of health care.

   Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

Part_2: Security and Privacy in the 1970’s

1. In general, I am quite concerned about such things as security leaks, bugging, and the invasion of privacy.

   Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

2. The computerization of information increases the likelihood that such data will be used for unintended purposes. In other words, automated data is more susceptible than other data to either
accidental or intentional disclosure, modification, or destruction.

Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

3. The use of computers in health organizations poses a threat to the confidential relationship between a doctor and patient.

Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

4. The use of computers in health organizations poses a threat to the confidential relationship between a manager and employee.

Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

5. If you want to make sure that information is kept from the knowledge of others, then it is better to keep information in written form and lock it in your desk rather than storing the information in a computer.

Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

**Part 2: Computer Security at the Medical Center**

1. How would you rate the present Medical Center system with respect to the security of computerized information? (Note: If you feel unequipped to answer this question, leave it blank.)

Excellent Poor

5 4 3 2 1

2. In the Medical Center computer system there are basically two types of information, medical and administrative. The two columns at the bottom of this page present examples of these data types. One method of ranking the sensitivity of a particular item of data is shown in lettered code below. How would you scale the sensitivity of each type of data? In answering this question, use the following code by putting the correct letter next to each data type.

<table>
<thead>
<tr>
<th>Code</th>
<th>Letter</th>
<th>Code Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>These data are private in that they relate to a specific individual and should not be known by anyone but that individual. For example, one could claim that the amount of one's contribution to a charitable fund was a private matter of concern to no one else.</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>These data are confidential in that they should be shared openly within the context of a professional relationship; i.e. doctor-patient, employee-manager, etc. However, the data should not be available to anyone outside of this confidential relationship.</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>These data are operational and are necessary for the normal functioning of the Medical Center. The information should be</td>
<td></td>
</tr>
</tbody>
</table>
available to all personnel who require the information in order to perform adequately their job(s).

These data are general knowledge for all persons either within or outside the Medical Center.

None of the above.

### Medical Data

<table>
<thead>
<tr>
<th>Code</th>
<th>Data Type</th>
<th>Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>______</td>
<td>Responses to Automated Medical History</td>
<td>______</td>
</tr>
<tr>
<td>______</td>
<td>Diagnoses</td>
<td>______</td>
</tr>
<tr>
<td>______</td>
<td>Lab test results</td>
<td>______</td>
</tr>
<tr>
<td>______</td>
<td>Personality inventory results (MMPI)</td>
<td>______</td>
</tr>
<tr>
<td>______</td>
<td>Services rendered to a patient</td>
<td>______</td>
</tr>
<tr>
<td>______</td>
<td>Name and address of patient's local MD</td>
<td>______</td>
</tr>
<tr>
<td>______</td>
<td>Patient surgical procedures</td>
<td>______</td>
</tr>
</tbody>
</table>

### Administrative Data

<table>
<thead>
<tr>
<th>Code</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>______</td>
<td>Payroll data--name, check amount, deductions, etc.</td>
</tr>
<tr>
<td>______</td>
<td>Patient names, addresses, phone numbers, etc.</td>
</tr>
<tr>
<td>______</td>
<td>Appointment data--MC MD, availability, bookings</td>
</tr>
<tr>
<td>______</td>
<td>Patient billing and payment history</td>
</tr>
<tr>
<td>______</td>
<td>Blue Shield codes for Medical Center services</td>
</tr>
<tr>
<td>______</td>
<td>Prices of all medical ctr services &amp; procedures</td>
</tr>
<tr>
<td>______</td>
<td>Total revenue generated by a staff member</td>
</tr>
</tbody>
</table>

3. Computers like people, are both fallible and vulnerable. For instance, it is possible to intercept or disrupt the transmission of computerized data. As protection certain security mechanisms have been developed which can deter the threat of accidental or intentional disclosure, modification, or destruction of information.

You have just scaled the relative sensitivity of several data items. The next question is, how important is it that these same data items be secure from interference of the above mentioned threats? To answer this, enter a number from 1 to 5 which signifies the relative importance of security for the particular data type.

<table>
<thead>
<tr>
<th>Extremely Important</th>
<th>Extremely Unimportant</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

### Medical Data

<table>
<thead>
<tr>
<th>Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>______</td>
</tr>
</tbody>
</table>

### Administrative Data

<table>
<thead>
<tr>
<th>Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>______</td>
</tr>
</tbody>
</table>
4. In many computer systems security mechanisms exist whereby certain types of data can be assigned different kinds of access according to various functions, "needs to know", or levels of authority. For example, technological capabilities are such that an automated record can be fully disclosed to one individual, while only a portion of the same record is displayed to another person. One can easily conceive of a situation in which this capability would be useful; on a newspaper it is likely that a sports editor would need to have access to different information than a fashion editor. Keeping in mind the needs and constraints of the Medical Center, what types of information would you allow each type of person named in the matrix on page 6 (of this Questionnaire) to access?

In order to complete this final (but most critical) question, please fill in the matrix on the next page. The rows in the matrix correspond to the data types of the previous two questions, and the columns refer to those who either should or should not have access to each particular type of data. Enter the appropriate number in the box under each "type of person" according to the type of access which you believe should apply to that type of data.

To indicate your opinion as to the access rights of each type of person please use the following codes.

<table>
<thead>
<tr>
<th>Code Number</th>
<th>Code Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>This type of person should have <strong>routine access</strong> to any of this data.</td>
</tr>
<tr>
<td>S</td>
<td>This type of person should have <strong>routine access only</strong> to summarized data of this class; e.g. data which could <strong>not be connected to a particular person</strong>.</td>
</tr>
</tbody>
</table>
This type of person should have special access to some of this class of data provided appropriate authorization has been granted.

This type of person should have absolutely no access to this type of data at any time.

If the list of codes seems inadequate for a certain situation, then simply leave that box blank or make any comments you wish.

To illustrate how this coding might work a portion of a completed matrix is shown below. In this particular case it was felt that all Medical Center employees (in general) should not have access to patient diagnoses or lab test results. Thus there are "4" (no access) codes next to these data types. Also it was felt that Medical Center prices could be widely known by all employees, so there is a "1" (routine access) code for that item.

Now go on to the next page to complete the matrix.

<table>
<thead>
<tr>
<th>All M.C.</th>
<th>Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient diagnoses</td>
<td>4</td>
</tr>
<tr>
<td>Lab test results</td>
<td>4</td>
</tr>
<tr>
<td>Prices of M.C. services</td>
<td>1</td>
</tr>
<tr>
<td>DATA TYPE</td>
<td>A</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>Responses to Automated Medical History</td>
<td></td>
</tr>
<tr>
<td>Patient diagnoses</td>
<td></td>
</tr>
<tr>
<td>Lab test results</td>
<td></td>
</tr>
<tr>
<td>Personality inventory results (MMPI)</td>
<td></td>
</tr>
<tr>
<td>Services rendered to a patient</td>
<td></td>
</tr>
<tr>
<td>Name and address of patient's local MD</td>
<td></td>
</tr>
<tr>
<td>Patient surgical procedures</td>
<td></td>
</tr>
<tr>
<td>Payroll data--name, check amount, etc.</td>
<td></td>
</tr>
<tr>
<td>Patient names, addresses, phone numbers, etc.</td>
<td></td>
</tr>
<tr>
<td>Appointment data--MD, availability, bookings</td>
<td></td>
</tr>
<tr>
<td>Patient billing and payment history</td>
<td></td>
</tr>
<tr>
<td>Blue Shield codes for Medical Center services</td>
<td></td>
</tr>
<tr>
<td>Prices of all medical center services</td>
<td></td>
</tr>
<tr>
<td>Total revenue generated by each staff member</td>
<td></td>
</tr>
</tbody>
</table>

**Code**

A. Director of Medical Center Research
B. Medical Center Staff Department Chairmen
C. All Medical Center physicians
D. Only certain Medical Center physicians*
E. All Medical Center nurses
F. Only certain Medical Center nurses*
G. Director of Medical Center Data Processing Appt. Office
H. Director of Medical Center Laboratory
I. All Medical Center managers
J. Only certain Medical Center managers*
K. All Medical Center employees
L. Only certain Medical Center employees*
M. General public
N. Only certain outside parties (BC/BS, etc)*
"Only certain..." refers to those who in the course of their job would be intimately concerned with a particular situation whether it be treating a particular patient, managing a particular employee, etc.

**Part 4: Respondent Profile**

1. Position with Medical Center (check one).
   - _____ Staff
   - _____ Management

2. Years of employment with the Medical Center
   - _____ less than 1 year
   - _____ 1-2 years
   - _____ 2-5 years
   - _____ 5-10 years
   - _____ over 10 years

**Part 5: Comments**

Thank you for taking time to complete this questionnaire. If you are interested in the results, I would be glad to supply you with a copy.

Any comments you might have on the administration or content of this survey can be made below.
DATA SECURITY AND THE FINANCIAL COMMUNITY

Torben G. Grønningen

CONTENTS

ACKNOWLEDGMENTS .............................................................. 115
INTRODUCTION .............................................................................. 115
HISTORICAL REVIEW ................................................................. 115
PHASE ONE ................................................................................. 116
PHASE TWO ................................................................................. 117
THE SURVEY ................................................................................ 118
MANAGERS' ATTITUDES ABOUT THE EFFECTS OF COMPUTERIZATION ......... 121
SECURITY CONSIDERATIONS AT PLANNING STAGE .............................. 121
MATCH BETWEEN MANAGERS' ATTITUDES AND ACTUAL PROTECTIVE MEASURES .................................................. 122
CLIENTS' SECURITY AWARENESS AND SPECIFIC SECURITY DEMANDS ...... 123
INEFFICIENCIES RESULTING FROM DATA SECURITY PRECAUTIONS ......... 123
THE RELATIVE IMPORTANCE OF DIFFERENT TYPES OF SECURITY MEASURES . 124
WHERE WOULD ADDITIONAL SPENDING OF SECURITY MONEY BE MOST IMPORTANT .................................................. 124
CENTRALIZED OR DECENTRALIZED CONTROL OF DATA SECURITY .......... 125
CONCLUSION ............................................................................. 125
REFERENCES .............................................................................. 127
APPENDIX .................................................................................. 128
ACKNOWLEDGMENTS

This paper is the result of a questionnaire survey of five financial institutions in the Boston area. Much support and interest have come from all five institutions. All participants in the survey are thanked for their help. Special thanks are due the survey contact manager of each site, without whose assistance in urging completion and then collecting the completed questionnaires, the survey would have taken a considerably longer time. Thanks are also extended to all members of the Sloan Information Systems Security Project. A special thank you to Mr. Philip Johnson of the Sloan School of Management who helped with the design of the questionnaire, an effort for which the author is very thankful, and to Mr. Peter Hollings of Cambridge Management Associates, Inc., who was most helpful by thoroughly reviewing the manuscript. The author alone, however, is responsible for the data analysis, presentation, and conclusions.

INTRODUCTION

Of all users of automated administrative data processing, the financial community has the longest tradition and experience. Most banks, insurance companies, investment and brokerage houses, and other financial institutions have used computers extensively for the past twenty years. Long experience in a field naturally creates expectations of knowledge, sophistication, and advanced methods. It was therefore with considerable excitement and interest that we took a close look at data security as it is found in the financial community.

The target of our analysis was not so much what we could see at each computer installation. We were less interested in the way things are and appear to be, and more interested in the way things are going to be, judged from a managerial viewpoint. In our survey, we were concerned with the managerial attitudes toward the increasing and changing nature of data automation. What do managers in organizations strongly based on automated data processing think of data security today, and what changes do they anticipate in the future? What do managers think of the relative importance of different kinds of protective measures? How concerned are the organizations’ clients with data security? What organizational changes do managers foresee as a result of increased data security considerations? These were some of the questions we asked the financial community. The answers that we obtained are discussed in this paper. Before this discussion, however, a short historical review of data processing in the financial world is in place.

HISTORICAL REVIEW

The evolution of automatic data processing in banks, insurance companies, and investment companies has been long, diverse, and unpredictable. Two distinct phases in this evolution are discernible, however.
PHASE ONE

The initial efforts in automation in the financial community took place in the late 1950's. Typical of this first phase transformation of information processing in the industry was the conversion of basic clerical functions. The primary goal was to reduce costs and to develop a capacity to absorb the ever-increasing work load. During the period from 1940 to 1955, business institutions experienced a "transactional surge" (exemplified by the increase in numbers of checks written, telephones in use, social security payments, tax returns filed, airline passengers, and so forth). The computer was unleashed to handle this transactional surge. The Phase One projects represented a degree of challenge, but they did not require, nor did they attract the talents and involvement of top management. The impact on top management's thinking, planning, and decision making, and overall organizational structure was minimal. There was a strong tendency to view the new groups of systems designers and programmers as a necessary evil, relegating their work to second-level status and, in many cases, associating it with the historic clerical jobs it replaced. The impact of the new systems was limited primarily within each organization, and there were few, if any, competitive advantages to being a leader.

The role of data security in Phase One is close in impact and profundity to the tasks and projects implemented on the computer system in the same phase of information processing. Concurrent with the first changes toward automation was the attention given to the most obvious needs for protection of data. The computer installation and the data storage media were the vulnerable objects of the first era because they were the new and obvious elements of the changed processing concept (the people who did the processing and handling of data were often recruited from other departments within the organization). The computer site became protected against such dangers as fire, flooding, heat, and power failure. A new small industry sprang up, providing fireproof file cabinets for storage of punched cards and magnetic tapes, and the organizations fabricated extensive backup procedures and data duplication combined with offsite storage in remote locations.

As more and more of the transactional surge was transferred to the computer, new types of educated specialists became common in the financial community: the computer operator, the computer programmer, and the Management Information Systems (MIS) analyst. These specialists often knew little about the organization and its data, and were considered separated in job, responsibility, and remuneration, from the rest of the people in the organization. Because of the increasing awareness of the potential dangers of having large amounts of data kept at one concentrated point, internal auditors pointed out that those who created the computer programs should be separated from those whose job it was to run the programs. Thus, it became commonplace to insist that only the computer operators had the right to be in the computer room during production processing. Of course, because of tight time-scheduling, this was sometimes hard to enforce. After all, programmers did need testing opportunities for their programs. An attempt to overcome this was the setup of a "closed shop" testing procedure. In some instances, a remote terminal for testing purposes was acquired.

The initial concerns with data security were mostly induced by the fear of either loss of data through physical damage or theft, or
accidental destruction through errors and mistakes. Later came the recognition of the needs of protecting certain particularly sensitive kinds of data against disclosure (for example, browsing and copying) and modification (for example, fraud and embezzlement). A safeguarded data archive was created, and data acquisition procedures were brought into use. In principle, data volumes were strongly guarded, though often not in reality.

Next came two important developments in data security. The first was caused by the spreading use of online systems with a large number of terminals outside the usual, controlled domain of the computer room. Ways of verifying that an authorized person was using a terminal became common to prevent errors and mistakes, and also to discourage attempts by unauthorized personnel to make intentional, illegal changes in data. Password identification and authentication were now necessary parts of processing. This often became a sophisticated way of distinguishing between different rights to data manipulation of different individuals. Some could create new data and change existing data, and others were only allowed to see certain portions of a file record.

The second important development in the security arena was the intensive refinement of the physical safeguards for computer facilities. This consisted of techniques such as security guards, television monitors, special locks requiring identification cards, and pushbutton combinations. This was partly a response to the publicity given such things as sabotage of computers, and partly because of the increased independence and organizational separation of the data processing department.

PHASE TWO

Many companies in the financial community today are in Phase One. This is true both in terms of data processing development and with regard to data security protection systems and procedures. A number of financial institutions, however, are on their way into Phase Two of data processing. This phase has begun with automated cost reporting, budgeting, profit planning, and, in some cases, incentive systems. Portfolio management centered around coded objectives, security analysis, bond trade analysis, and investment analysis are being transferred to computers. A whole new aspect of data processing has evolved with the centralized administration and validation of credit cards. Checks begin to show early signs of ultimate elimination because of the growth of direct transfers and debits. Automatic securities transfer has also been introduced. The cashless, checkless society is slowly becoming a reality.

The impact of these changes in the financial community will have profound effects throughout society. Where Phase One was mostly the automation of well-established and well-defined procedures, Phase Two will need strong managerial support and skills. New business policies will emerge, and structural changes in organizations will result.

An example of the latter is the creation of the so-called "automated services division." This division is usually created as a single profit center within the organization. In addition to serving in-house customers, an increasing part of its revenue will come from sales of data processing services to outside organizations, often customers of the main line of the business (for example, banking and insurance). The automated services division can, in addition to
providing such common services as financial accounting and payroll administration, also grant customer access to valuable data about the economy and selected industry sectors, initially compiled and updated by the financial institution for its own internal purposes.

What developments and changes in data security are likely to occur, as Phase Two extends its effects? In other words, to what extent are the security precautions of Phase One, such as current personnel and procedural protection and employed hardware and software safeguards, insufficient? It is not within the scope of this paper to delve into the complexities of hardware and software protection in such terms as voice-print patterns, read-only memory, residual data, antibugging and anti-interception devices.

Here it is more to our interest to see if it is possible to point to the changes, if any, in future data security in the financial community. As mentioned in the introduction of this paper, we have done this by asking questions of managers in the financial community. We did this by means of a questionnaire survey of a number of managers in several selected financial institutions.

THE SURVEY

Forty-two managers in five different financial institutions (three banks, one mutual-funds company, and one life insurance company) in the Boston area participated in the study. Seventeen of these managers were responsible for information-processing-related functions, such as systems design and development, programming, program maintenance, computer operations and internal data processing education, or they were in processing-related functions as liaison to the users of processing services in the organization. We shall call these individuals technical managers. The other 25 managers had responsibilities in data user areas such as auditing, treasury, controllership, personal trust, insurance, deposit accounting and actuary functions. We shall refer to these individuals as nontechnical managers.

The method of surveying was through a standard paper-and-pencil questionnaire, distributed to each individual, to be completed independently with no interference from superiors or subordinates. Most of the answers were given on a 1 to 5 scale. The five values of the scale can be characterized in a number of ways, depending on the nature of the question. Examples are: 1 = Low, 2 = Quite Low, 3 = Medium, 4 = Quite High, 5 = High; or: 1 = Not At All, 2 = A Little, 3 = Somewhat, 4 = A Lot, 5 = Fully.

The questionnaire contained, as a point of reference, an explanation of certain often-used terms. Three different kinds of data security were defined: technical, procedural, and personnel. Exhibit 1 shows some examples of these three categories. Furthermore, in some of the questions, a distinction was made between disclosure (for example, illegal browsing) and destruction or modification of data (for example, fraud). Exhibit 2 gives examples of the meaning of these terms. The questionnaire with the aggregated findings is shown in the Appendix. (The discussion in the following sections is based on the responses given to the questions in this questionnaire.)
Exhibit 1

Common Security Threats and Countermeasures

Common threats against data security are computer installation sabotage, accidental system breakdown, fraud, embezzlement, interception errors, disclosure of data, theft, sabotage, or unauthorized copying of data. Data security can be created and maintained by some or all of the following elements:

Technical Protection (automated):

- Computer system integrity (for example, operating system, backup power and fire protection)
- Remote access control (for example, terminal authorization, user identification)
- Data encoding (for example, encryption).

Procedural Protection (manual):

- Physical access control (for example, guards, badges and locks)
- Data handling rules (for example, offsite storage written requisition of storage volumes)
- Program modification rules
- Input/output separation
- Input/output controls
- Audit.

Personnel Protection:

- Preemployment screening
- Supervision
- Division of responsibility.
Exhibit 2

**Data Disclosure**: The revelation to an unauthorized party of data possessing a certain confidentiality. Examples of such data are client names and addresses, credit data and payroll data.

**Data Modification or Destruction**: Accidental or intentional acts performed on data, and resulting in undesirable change or loss of data. Examples of such acts are errors, system malfunction, sabotage, fraud and embezzlement.
MANAGERS' ATTITUDES ABOUT THE EFFECTS OF COMPUTERIZATION

It was anticipated that most managers would feel quite comfortable about computer-based data processing. It was also expected that managers would be more concerned with the dangers of data destruction and modification than with the dangers of data disclosure. The survey confirmed this. (See questions I.1 and II.1 in the Appendix.) Two-thirds of the managers surveyed were of the belief that computerization does not increase the likelihood of data disclosure. The technical managers were particularly of this opinion. When asked about the chance of data destruction or modification created by the usage of computers, the feelings were slightly different. The perceived likelihood for security violation here (for example, fraud or embezzlement) was higher than for disclosure. Twenty-seven percent believed that the chance of destruction or modification was considerable. The majority of those who believed so was nontechnical (only 18 percent of the technical managers thought so, while more than one-third of the nontechnical managers were of this belief).

An explanation of the generally higher perceived risk of violations such as fraud and embezzlement as compared to the risk of disclosure can be that managers more frequently referred to bad experiences from cases of computer-aided fraud and embezzlement in their organizations than to bad experiences from harmful disclosure of computerized data. Also, this finding may reflect that fraud is considered more harmful to organizations than disclosure.

SECURITY CONSIDERATIONS AT PLANNING STAGE

To the question of what the managers' attitudes were toward data disclosure are in general, 78 percent said that they were medium to highly concerned with this issue. (See questions I.2 - I.5 and II.2 - II.4 in the Appendix.) As one manager said, "You may get away with telling others what diseases a person has, but you can never reveal how much money he has in his bank account!" (This remark should not be taken as indicative of the author's view of the relative importance of protecting medical and financial data.) The concern for data disclosure has been on the upswing over the past three years, according to two-thirds of the managers, but one-third has not noted any recent change in their concern for disclosure. Major reasons for the change have been an overall trend in society, and the increasing degree of computerization, often combined with the desire for faster data processing. Personal ethics and company-wide policies were other reasons given for the concern for data disclosure.

Little change in attitude toward disclosure is expected in the near future, that is, the next three years. Two-thirds of the managers believe that the concern for data disclosure will remain unchanged.

The change in the managers' concern for fraud and embezzlement has typically been increasing much more than the concern for disclosure. Almost seven out of ten managers characterized their change in concern for fraud and embezzlement as either "more" or "much more" when asked about the changes over the past three years. The major reason given for this was the increase in degree of computerization of data. Some managers attributed the increase to factors such as "company-wide trend," the demand for faster
processing operations, and personal experience with cases of computer-aided fraud or embezzlement.

The anticipated risk of security breach caused by the computerization of data was reflected in the considerations given in the planning stage of system development. Here, again, opinions were that disclosure is less of a potential danger than fraud and embezzlement. Protection against fraud and embezzlement was characterized as "considerable" to "extensive" by 70 percent, but the similar figure for the planning of protection against data disclosure was only 49 percent. A majority of managers felt that the security considerations in their organizations against disclosure was "medium" to "quite high."

MATCH BETWEEN MANAGERS' ATTITUDES AND ACTUAL PROTECTIVE MEASURES

The managers were asked if they felt that their concern for data security was adequately matched by three different types of security precautions: technical, procedural, and personnel (See Exhibit 1 for examples of these three.) Again, a distinction was made between disclosure and destruction or modification. (See Exhibit 2 for examples.)

The protection of data from disclosure by technical measures was predominantly (85 percent of the responses) in the range of "somewhat adequate" to "fully adequate." (See questions I.7 and II.5 in the Appendix.) It was mostly nontechnical managers (68 percent) who were willing to state that their current information systems were "quite" to "fully" adequate. Only 35 percent of the technical managers felt this degree of adequacy of the technical precautions. Assuming that technical and nontechnical managers have understood the question the same way, this finding points in an affirmative direction to the psychological findings manifested in the attribution theory. Psychologists believe that one way individuals evaluate the motivations and behavior of others is by attributing to them the same knowledge, values, and feelings that they themselves have. Consequently, the technical manager whose proximity to the computer provides him with the knowledge of the many ways the system can be compromised is likely to have this knowledge influence his attribution process.

Consequently, our survey seems to confirm the hypothesis that a manager's concern for data security is a function of the manager's awareness of data security as a problem. Obviously, technical managers are more aware of technical problems and shortcomings in the area of data security than nontechnical managers. Consequently, the technical managers are more likely to feel any existing inadequacies of technical precautions.

A similar result was found in the match between concern for destruction or modification, and technically oriented precautions. Ninety percent of all respondents believed that technical precautions covered their concerns "somewhat" to "fully" adequately. Of those who felt that the match was "quite" to "fully" adequate, the majority were nontechnical managers.

Speaking of data disclosure, the managers felt that procedural protection was more efficient than personnel protection. One-half of the managers believed that data procedures, controls, and audits did a "quite" to "fully" adequate job of protection. Only one-third
would admit to the same opinion of personnel precautions, such as preemployment screening, supervision, and division of responsibility.

We then asked the managers of the effectiveness of nontechnical precautions in the prevention of data destruction or modification (here meaning largely computer-aided fraud or embezzlement). The prevailing opinion was that personnel and procedural countermeasures are equally effective. However, personnel precautions were only believed "fully" adequate by 15 percent, and procedural precautions were considered "fully" adequate by 23 percent.

CLIENTS' SECURITY AWARENESS AND SPECIFIC SECURITY DEMANDS

Clients are perceived by management to be very concerned with data security, but specific requirements have rarely been set forth. This was the finding of our survey with regard to clients' data security awareness and demands. (See questions I.8 and I.9 in the Appendix.) Two-thirds of the managers believed that their clients were "very" or "quite" concerned with the prevention of disclosure of private data (for example, depository accounts, portfolio data, operating data and payroll or insurance data). Only 20 percent thought their clients to be little concerned with security. Of these 20 percent, the majority were technical managers. A possible explanation of this surprising finding may be that technical managers are not in direct contact with clients and, therefore, do not have an accurate feeling for their needs.

Despite our observations of high client concern, few specific security measures have been demanded by the clients. More than half of the managers said that they had never or rarely come across such client demands. Of the 15 percent of the managers who said that they had experienced demands "quite often," all of them were in nontechnical business functions. This result may again point to the fact that nontechnical managers are in more direct contact with customers than technical managers. Furthermore, there is evidence that demand for security from clients more often is posed by procedural and personnel precautions than about strictly computer-technical precautions. Also, the rather low reported frequency of computer-based security violations in the financial organizations may well have given the clients a feeling of security.

INEFFICIENCIES RESULTING FROM DATA SECURITY PRECAUTIONS

When we asked the question of whether security precautions had given rise to any detectable degree of operating inefficiency, we had the suspicion that this would be true for technical aspects such as online terminal user authorization, and for some procedural precautions, such as data requisitions, access control and input/output control.

However, the survey showed that neither technical, procedural, nor personnel precautions were considered noticeable impediments to efficient daily operations. (See question III.1 in the Appendix.) Substantial hindrances due to personnel precautions were largely absent, and only one-fifth of the managers felt that the different security precautions were causing any efficiency problems.

These results, paired with the knowledge of the existing protective measures in the five surveyed organizations, seem to indicate a good integration of basic security features. The daily
operating efficiency obviously has been a major consideration in the
determination of the adopted level of security, combined with the
perceived level of security threats.

THE RELATIVE IMPORTANCE OF DIFFERENT TYPES OF SECURITY MEASURES

An attempt was made to evaluate the perceived importance of the
three types of security measures described in Exhibit 1: technical,
procedural, and personnel. By way of ranking, technical security was
in first place, preferred by 41 percent of the managers. (See
question III.2 in the Appendix.) For comparison, the other two types
of measures were ranked first by 32 percent and 27 percent of the
managers, respectively.

Procedural precautions were ranked second in preference by most
of the managers. This type of security measure was ranked second by
52 percent of the managers. A similar number of managers ranked
personnel precautions third. The distribution of preferences for
these three types of security precautions was not significantly
influenced by the managers' type of responsibility (technical or
nontechnical). Many nontechnical managers felt that technical
security was most important, and were willing, in principle, to trade
off nontechnical with technical security.

On this background, the answers to the next question were rather
surprising.

WHERE WOULD ADDITIONAL SPENDING OF
SECURITY MONEY BE MOST IMPORTANT

We asked the managers that if additional money from the computer
budget should be spent on data security, where should the money go?
A list of five alternatives was given: physical computer site
protection, software sophistication—operating system, software
sophistication—user programs, personnel practices, or data
input/output controls and auditing.

Because of the preferences revealed in the preceding question,
we had expected that the two types of software sophistication would
be prime candidates for additional spending. However, there existed
a clear preference for data controls and auditing. (See question
III.3 in the Appendix.) This type of protection was recommended
first by 45 percent of the managers. Software sophistication for
user programs ranked only second and fourth for operating system
improvements. The most obvious explanation of this result may be
that procedural protection is currently given too little weight
relative to technical security protection, which has attracted more
attention because of its perceived higher importance.

Personnel protection ranked third, and physical site protection
fifth and last on the "willingness to spend extra money" scale. This
indicates a preference among managers (both technical and
nontechnical) for extra outlays for improving personnel practices
before extra spending on operating system improvements.

Again, this must be evaluated in light of the relatively low
frequency of computer-aided data crimes experienced over the past
years, compared with the crimes that have been committed by personnel
or others in a noncomputer-related fashion.
By ranking physical site protection last on the "willingness to spend extra money" scale, the managers seem to support the hypothesis that the current level of physical site protection is adequate.

CENTRALIZED OR DECENTRALIZED CONTROL OF DATA SECURITY

A point of considerable interest is the role of data security in a structural context within the organization. How are security problems solved today, and is it reasonable to expect any organizational ramifications of data security considerations?

Managers within the same organization frequently had different opinions as to whether security problems were solved jointly by the different user departments, or independently in a decentralized, noncoordinated fashion. On an aggregate basis, 59 percent believed that data security problems were solved jointly, and 41 percent said they were solved independently. (See question III.4 in the Appendix.) One manager stated that the approach depended entirely on the type of problem. Among the reasons given for independent solutions to security problems were dissimilar data requirements, dissimilar processing or procedures, and dissimilar security needs.

A majority (60 percent) felt that there was a need for a central decision maker or decision-making group responsible for all data security within the organization. Most of these managers believed that the responsibility in such cases should rest with top management, either directly related to the auditing department, or as a special data security office. One manager felt that more important than the actual functional level in the organization was the need for someone thoroughly familiar with the nature of the organization's data and its users, and with a strong knowledge of software, hardware, and data procedures.

CONCLUSION

Data security has been the focal point in a questionnaire survey of managers in the financial community. It was found that managers generally feel quite confident in the adequacy of the current level of protection of data from destruction, modification, or disclosure (either intentional or accidental). The attitude toward data security is not expected to change substantially over the next few years. Although technical security precautions such as system integrity and access control are believed most important of all protective measures, managers are more interested in improving nontechnical protective features such as data handling rules, input/output controls, and auditing. A slight majority of the managers surveyed preferred a central decision-making unit within the organization to adopt and implement data security policies. Most of these managers felt this to be a top management responsibility.

Security features were integrated into daily operations to such a degree that little or no inefficiency was felt. The prevention of disclosure of data was believed to be less imperative than prevention of destruction or modification (for example, fraud and embezzlement).

The technical managers surveyed were generally more concerned with data security than their nontechnical colleagues. A possible explanation is the technical managers' deeper understanding of the existing data systems' vulnerabilities. Technical managers, however, were less concerned with the confidentiality of client data, possibly because of less client contact.
Despite the limitations of sample size (42 survey respondents), our findings appear generally reasonable and consistent, and, on this basis, may be taken as representative of the financial community at large. We conclude that financial institutions are concerned with data security, and that they have taken action to adopt what is considered a reasonable level of data security given current needs.

It is important for the financial community to recognize the deep changes that data processing currently is undergoing. In the introduction we touched upon the transformation from a Phase One to a Phase Two in automated processing. We briefly mentioned the impact on data security that such a processing transformation may have. We can now speculate about the effects of Phase Two on the security needs and demands of the financial community. What can we say about the kind of change in data security that will take place in financial institutions in the years to come?

Based on our survey results, the obvious expectation is one of little or no change at all. The state of the art in data security is characterized as largely satisfactory by our sample of managers. This raises the serious question of whether managers in the financial community have adequately anticipated the changing demands for security brought upon it by Phase Two. This question, in turn, leads to the consideration of forces outside the financial community with potential for policymaking.

All through our discussion we have distinguished between two basic dimensions. These two dimensions are conceptually related to privacy and integrity. We have approached the potential violation of these from the information user's side, here the financial community. But our discussion of data security would be incomplete without mentioning those other constituencies which have a possible impact on data security policymaking.

Necessary conditions for privacy and integrity are safe information processing practices and procedures, paired with usage of computer hardware and software which is uncompromisable to a high degree. This brings the computer manufacturing industry into the picture as a potential policymaking factor. No recordkeeping system can attain a high degree of security without being used in a computer system which recognizes these needs.

Computer systems in Phase One have mostly been designed to accommodate large amounts of data, and processing of these data at a high speed with a satisfactory reliability. Parallel with the efforts of the computer manufacturers to enhance these features, an additional component in the design is emerging: the integration of security features in the basic architecture of computers, such as hierarchical access control, read-only memory, voice pattern recognition authorization, erasure of residual data. Add-on security considerations which are not deeply integrated into the basic architecture of the computer system will, slightly overstated, be like the automobile which is sent on the market, equipped with safety belts and shatter-proof windshield, but without brakes.

The changes necessary to create a high degree of privacy and integrity are so fundamental and far-reaching that it seems unrealistic to expect data security policymaking merely to be limited to the information systems users and the computer manufacturers. A third constituency is very likely to be active: the government.
Sweden has recently taken legislative action to adopt fair information practices. The primary purpose of this law, the first of its kind, has been to eliminate the keeping of secret records, and to allow the individual knowledge of existence, contents, and use of all data file records pertaining to that individual. A Data Inspection Board is the formal agency for the implementation of this law. All data files must be registered with this board. Requests for the use of a general individual identifier (such as the social security number) will only be accepted from information users which can document a genuine need for this. The result of such a law in the United States would probably be the abolishment of the use of the social security number for commercial purposes, such as bank accounts, insurance policies, credit cards, etc.

Much seems to indicate that the United States is headed in the same direction, although at a slower pace. Senator Ervin's congressional hearings a few years ago brought increased attention to the rights to information privacy. The Department of Health, Education, and Welfare has recently published its report on automated personal data systems, recommending the enactment of a federal "Code of Fair Information Practice." (1)

Such a legislative step could place heavy administrative burdens on the organizations which process personal information. Furthermore, it might, directly or indirectly, require computer manufacturers to adopt basic security features in computer systems design.

Thus, in a climate where issues involving the rights of the individual attain increasing attention, policymaking in the area of data security may well occur as a result of legal imposition. And, if legal forces fail to act, the public may well let its voice be heard through spokesmen and activists, forcing organizations, manufacturers, and governments to take action in what may be called Phase Two of data security.

REFERENCES

**APPENDIX**

**Tabulations from Survey Questionnaire**

Sample Size = 42

T = Technical Managers (17)
N = Nontechnical Managers (25)

**Important Note:**

All numbers in the following are percentages, with the total number of responses (42, unless otherwise stated) equal to 100 percent.

**Part II: Data Disclosure**

The phrase "data disclosure" is used here to mean the revelation to an unauthorized party of data possessing a certain confidentiality. Examples of such data are client name and address lists, credit data and payroll data.

1. Is it your opinion that the use of the computer in your department increases the likelihood of disclosure of confidential data? Please circle or check one of the numbers on the scale.

<table>
<thead>
<tr>
<th>No increase</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Great increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>36</td>
<td>31</td>
<td>17</td>
<td>14</td>
<td>02</td>
<td>02</td>
</tr>
<tr>
<td>T:</td>
<td>19</td>
<td>17</td>
<td>05</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>N:</td>
<td>17</td>
<td>14</td>
<td>12</td>
<td>14</td>
<td>02</td>
<td>02</td>
</tr>
</tbody>
</table>

2. How much attention has protection against data disclosure received in the planning and implementation of computer application systems currently in use in your department?

<table>
<thead>
<tr>
<th>Negligible</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Extensive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>00</td>
<td>12</td>
<td>39</td>
<td>34</td>
<td>15</td>
<td>02</td>
</tr>
<tr>
<td>T:</td>
<td>00</td>
<td>02</td>
<td>17</td>
<td>15</td>
<td>05</td>
<td>00</td>
</tr>
<tr>
<td>N:</td>
<td>00</td>
<td>10</td>
<td>22</td>
<td>19</td>
<td>10</td>
<td>02</td>
</tr>
</tbody>
</table>

3. What is your attitude toward disclosure of data being handled by your department?

<table>
<thead>
<tr>
<th>Little concern</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Great concern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>02</td>
<td>02</td>
<td>17</td>
<td>38</td>
<td>40</td>
<td>02</td>
</tr>
<tr>
<td>T:</td>
<td>02</td>
<td>00</td>
<td>07</td>
<td>17</td>
<td>14</td>
<td>00</td>
</tr>
<tr>
<td>N:</td>
<td>00</td>
<td>02</td>
<td>10</td>
<td>21</td>
<td>26</td>
<td>00</td>
</tr>
</tbody>
</table>
4. Has there been any change in your attitude toward data disclosure over the past three years, and if so how?

<table>
<thead>
<tr>
<th></th>
<th>Much less concern</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Much more concern</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total:</strong></td>
<td>00</td>
<td>00</td>
<td>36</td>
<td>40</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td><strong>T:</strong></td>
<td>00</td>
<td>00</td>
<td>14</td>
<td>16</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>W:</strong></td>
<td>00</td>
<td>00</td>
<td>22</td>
<td>24</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>

5. To what reasons do you attribute this change? Please check one or more of the reasons listed. If no change, go on to 6.

(Total number of reasons given: 67.)

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>11</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>10</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>21</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td></td>
</tr>
<tr>
<td><strong>5</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Society trend</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Company-wide trend</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Trend within department</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Change in type of data being responsible for</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Change in degree of computerization of data</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Concern for faster data processing operations</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Witnessed consequences from actual cases of data disclosure</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Personal ethical reasons</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Other:</strong></td>
<td></td>
</tr>
</tbody>
</table>

6. Assuming you have responsibility over the same kind of data as today, do you think your attitude toward data disclosure will change over the coming three years. If so, how?

<table>
<thead>
<tr>
<th></th>
<th>Much less concern</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Much more concern</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total:</strong></td>
<td>00</td>
<td>00</td>
<td>64</td>
<td>21</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td><strong>T:</strong></td>
<td>00</td>
<td>00</td>
<td>31</td>
<td>02</td>
<td>07</td>
<td></td>
</tr>
<tr>
<td><strong>W:</strong></td>
<td>00</td>
<td>00</td>
<td>33</td>
<td>19</td>
<td>07</td>
<td></td>
</tr>
</tbody>
</table>

7. Do you feel that your attitude toward data disclosure is adequately matched by:

**Technical precautions**

<table>
<thead>
<tr>
<th></th>
<th>Inadequate</th>
<th>Adequate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total:</strong></td>
<td>00</td>
<td>14</td>
</tr>
<tr>
<td><strong>T:</strong></td>
<td>00</td>
<td>07</td>
</tr>
<tr>
<td><strong>W:</strong></td>
<td>00</td>
<td>07</td>
</tr>
</tbody>
</table>

**Procedural precautions**

<table>
<thead>
<tr>
<th></th>
<th>Inadequate</th>
<th>Adequate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total:</strong></td>
<td>05</td>
<td>14</td>
</tr>
<tr>
<td><strong>T:</strong></td>
<td>02</td>
<td>12</td>
</tr>
<tr>
<td><strong>W:</strong></td>
<td>03</td>
<td>02</td>
</tr>
</tbody>
</table>
Personnel precautions

<table>
<thead>
<tr>
<th>Inadequate</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Adequate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>05</td>
<td>19</td>
<td>45</td>
<td>17</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>02</td>
<td>12</td>
<td>17</td>
<td>02</td>
<td>07</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>03</td>
<td>07</td>
<td>28</td>
<td>15</td>
<td>07</td>
<td></td>
</tr>
</tbody>
</table>

9. What is your estimation of the overall level of concern among your clients with the issue of data disclosure?

<table>
<thead>
<tr>
<th>Little concern</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Great concern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>33</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>08</td>
<td>08</td>
<td>05</td>
<td>10</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>02</td>
<td>02</td>
<td>10</td>
<td>23</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

9. Have specific security requirements ever been proposed or demanded by your clients?

<table>
<thead>
<tr>
<th>Never</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Very frequently</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>05</td>
<td>49</td>
<td>31</td>
<td>15</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>02</td>
<td>23</td>
<td>18</td>
<td>00</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>03</td>
<td>26</td>
<td>13</td>
<td>15</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

PART II: Modification or Destruction of Data

Here we refer to accidental as well as malicious modification or destruction of data. Examples are errors, system malfunction, sabotage, fraud and embezzlement.

1. Is it your opinion that the use of the computer in your department increases the likelihood of data destruction or modification?

<table>
<thead>
<tr>
<th>Great increase</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>No increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>17</td>
<td>10</td>
<td>23</td>
<td>25</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>02</td>
<td>05</td>
<td>08</td>
<td>15</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>15</td>
<td>05</td>
<td>15</td>
<td>10</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

2. How much attention has protection against data modification or destruction received in the planning and implementation of computer application systems currently in use in your department?

<table>
<thead>
<tr>
<th>Extensive</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Negligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>38</td>
<td>33</td>
<td>25</td>
<td>02</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>15</td>
<td>16</td>
<td>08</td>
<td>00</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>23</td>
<td>17</td>
<td>17</td>
<td>02</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

3. Has there been any change in your concern for computer-aided fraud or embezzlement over the past three years, and if so, how?

130 Volume 4 Massachusetts Institute of Technology
Much more concern 1 2 3 4 5 Much less concern

Total: 32 37 29 02 00
T: 07 20 12 02 00
N: 25 17 17 00 00

4. To what reasons do you attribute this change? Please check one or more of the reasons listed. If no change, go on to 5.

(Total number of reasons given: 54.)

<table>
<thead>
<tr>
<th>N</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>08 09 17</td>
<td>Company-wide trend</td>
</tr>
<tr>
<td>03 02 05</td>
<td>Trend within department</td>
</tr>
<tr>
<td>10 05 15</td>
<td>Change in type of data being responsible for</td>
</tr>
<tr>
<td>09 17 26</td>
<td>Change in degree of computerization of data</td>
</tr>
<tr>
<td>02 13 15</td>
<td>Concern for faster data processing operations</td>
</tr>
<tr>
<td>06 07 13</td>
<td>Witnessed consequences from actual case of computer-aided fraud or embezzlement</td>
</tr>
</tbody>
</table>

07 02 09 Other:

5. Do you feel your attitude toward data destruction or modification is adequately matched by:

**Technical precautions**

<table>
<thead>
<tr>
<th>Adequate</th>
<th>1 2 3 4 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>35 17 38 10 00</td>
</tr>
<tr>
<td>T:</td>
<td>15 02 23 02 00</td>
</tr>
<tr>
<td>N:</td>
<td>20 15 15 08 00</td>
</tr>
</tbody>
</table>

**Procedural precautions**

<table>
<thead>
<tr>
<th>Adequate</th>
<th>1 2 3 4 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>23 35 25 15 02</td>
</tr>
<tr>
<td>T:</td>
<td>13 10 10 10 00</td>
</tr>
<tr>
<td>N:</td>
<td>10 25 15 05 02</td>
</tr>
</tbody>
</table>

**Personnel precautions**

<table>
<thead>
<tr>
<th>Adequate</th>
<th>1 2 3 4 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>15 35 28 20 02</td>
</tr>
<tr>
<td>T:</td>
<td>10 12 11 10 00</td>
</tr>
<tr>
<td>N:</td>
<td>05 23 17 10 02</td>
</tr>
</tbody>
</table>

**PART III: Data Security in General**

The following questions relate to the overall concept of data security, embodying issues of disclosure, modification, and destruction.
1. Are existing data security mechanisms in your department perceived as an impediment to efficient daily operations?

**Technical precautions**

<table>
<thead>
<tr>
<th>No inefficiency</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Great hindrance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>46</td>
<td>32</td>
<td>20</td>
<td>02</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>19</td>
<td>12</td>
<td>08</td>
<td>02</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>27</td>
<td>20</td>
<td>12</td>
<td>00</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

**Procedural precautions**

<table>
<thead>
<tr>
<th>No inefficiency</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Great hindrance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>29</td>
<td>37</td>
<td>15</td>
<td>20</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>14</td>
<td>13</td>
<td>08</td>
<td>08</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>15</td>
<td>24</td>
<td>07</td>
<td>12</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

**Personnel precautions**

<table>
<thead>
<tr>
<th>No inefficiency</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Great hindrance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
<td>46</td>
<td>27</td>
<td>22</td>
<td>02</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>T:</td>
<td>19</td>
<td>10</td>
<td>10</td>
<td>00</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>N:</td>
<td>27</td>
<td>17</td>
<td>12</td>
<td>02</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

2. How do you in general perceive the relative importance of the following types of data security precautions (please give percentage estimates if possible, with sum equal to 100 percent)?

**Ranking (percentages)**

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical precautions</td>
<td>41</td>
<td>22</td>
<td>28</td>
</tr>
<tr>
<td>Procedural precautions</td>
<td>32</td>
<td>52</td>
<td>21</td>
</tr>
<tr>
<td>Personnel precautions</td>
<td>27</td>
<td>26</td>
<td>51</td>
</tr>
</tbody>
</table>

3. If your organization were to commit an additional 5 percent of its annual computer budget to data security measures, rank your personal choice among the following alternatives (1 = most preferred, 5 = least preferred).

**Ranking (percentages)**

<table>
<thead>
<tr>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>17</td>
<td>10</td>
<td>19</td>
<td>49</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>20</td>
<td>38</td>
<td>14</td>
</tr>
<tr>
<td>28</td>
<td>14</td>
<td>40</td>
<td>09</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>23</td>
<td>10</td>
<td>31</td>
<td>19</td>
</tr>
<tr>
<td>45</td>
<td>23</td>
<td>20</td>
<td>03</td>
<td>08</td>
</tr>
</tbody>
</table>
4. How do different departments in your organization solve their data security problems?

<table>
<thead>
<tr>
<th></th>
<th>Jointly</th>
<th>Independently</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total:</strong></td>
<td>59</td>
<td>41</td>
</tr>
<tr>
<td><strong>T:</strong></td>
<td>18</td>
<td>22</td>
</tr>
<tr>
<td><strong>N:</strong></td>
<td>41</td>
<td>19</td>
</tr>
</tbody>
</table>

If independently was checked, give reason.

(Total number of reasons given: 37.)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong></td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td><strong>N:</strong></td>
<td>08</td>
<td>22</td>
</tr>
<tr>
<td><strong>Top management</strong></td>
<td>Expedience</td>
<td></td>
</tr>
<tr>
<td><strong>Dissimilar data requirements</strong></td>
<td>16</td>
<td></td>
</tr>
<tr>
<td><strong>Dissimilar processing or procedures</strong></td>
<td>22</td>
<td></td>
</tr>
<tr>
<td><strong>Required by outside authority</strong> (for example, audits)</td>
<td>03</td>
<td></td>
</tr>
<tr>
<td><strong>Dissimilar security needs</strong></td>
<td>05</td>
<td></td>
</tr>
<tr>
<td><strong>Other:</strong></td>
<td>03</td>
<td>03</td>
</tr>
</tbody>
</table>

5. Do you think there is a need for a single decision making individual or group responsible for all data security in an organization?

<table>
<thead>
<tr>
<th></th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total:</strong></td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td><strong>T:</strong></td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td><strong>N:</strong></td>
<td>36</td>
<td>24</td>
</tr>
</tbody>
</table>

If so, which level of management should be charged with such a responsibility?

(Total number of replies: 25.)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong></td>
<td>16</td>
<td>44</td>
</tr>
<tr>
<td><strong>N:</strong></td>
<td>08</td>
<td>24</td>
</tr>
<tr>
<td><strong>Top management</strong></td>
<td>Expedience</td>
<td></td>
</tr>
<tr>
<td><strong>Middle management</strong></td>
<td>16</td>
<td></td>
</tr>
<tr>
<td><strong>Line management</strong></td>
<td>08</td>
<td></td>
</tr>
</tbody>
</table>

If so, in which department should this decision-making individual or group be?

(Total number of replies: 24.)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong></td>
<td>05</td>
<td>08</td>
</tr>
<tr>
<td><strong>N:</strong></td>
<td>08</td>
<td>13</td>
</tr>
<tr>
<td><strong>Controller's office</strong></td>
<td>Expedience</td>
<td></td>
</tr>
<tr>
<td><strong>Internal auditor's office</strong></td>
<td>17</td>
<td></td>
</tr>
<tr>
<td><strong>Computer systems development</strong></td>
<td>29</td>
<td></td>
</tr>
<tr>
<td><strong>Computer operations</strong></td>
<td>04</td>
<td></td>
</tr>
<tr>
<td><strong>Special data security office</strong></td>
<td>04</td>
<td></td>
</tr>
<tr>
<td><strong>Other:</strong></td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

Massachusetts Institute of Technology Volume 4
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0   Introduction</td>
<td>165</td>
</tr>
<tr>
<td>2.0   Integrity</td>
<td></td>
</tr>
<tr>
<td>2.1   Data Integrity</td>
<td>167</td>
</tr>
<tr>
<td>2.2   System Integrity</td>
<td>167</td>
</tr>
<tr>
<td>2.2.1 Receiving As Expected</td>
<td>168</td>
</tr>
<tr>
<td>2.2.2 Consistency Of Operation</td>
<td>168</td>
</tr>
<tr>
<td>2.2.2.1 Multiple-State Machines</td>
<td>168</td>
</tr>
<tr>
<td>2.2.3 Requirements for Integrity</td>
<td>170</td>
</tr>
<tr>
<td>3.0   Access Control</td>
<td></td>
</tr>
<tr>
<td>3.1   System Access Control</td>
<td>171</td>
</tr>
<tr>
<td>3.2   Address Space Access Control</td>
<td>171</td>
</tr>
<tr>
<td>3.2.1 Data Set Access Control</td>
<td>171</td>
</tr>
<tr>
<td>3.2.1.1 Keyword Access Control</td>
<td>172</td>
</tr>
<tr>
<td>3.2.1.2 Procedure Control</td>
<td>173</td>
</tr>
<tr>
<td>3.2.2 Subfile Access Control</td>
<td>173</td>
</tr>
<tr>
<td>4.0   Conclusion</td>
<td>174</td>
</tr>
<tr>
<td>5.0   References</td>
<td>178</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

For the purposes of this report, we will define the state of practice to be what the more advanced user installations are actually doing, as opposed to those that might qualify as "academic" or "laboratory" schemes. We intend to present some of the more evident global trends, rather than an exhaustive technical expose. Space considerations demand that we further confine our scope to a representative sample of the emerging trends in "conventional" system design. Omission of many of the schemes in operation today in no way passes judgement on these schemes.

As the level of user awareness increases, so the demand for secure systems is expected to increase. This is both logical, and also more formally verified by responses to questionnaires on this issue. We intend to show that current approaches to system design go a long way toward facilitating secure systems.

What issues are of concern when we use the term "secure system"? It appears reasonable to specify two requirements for a secure computer system: 1) system integrity, and 2) access control.

For our purposes, system integrity concerns the predictable behavior of the system under any and all conditions. Access control is concerned with prohibiting and/or controlling access to sub-parts of the system, user address spaces, and data.

This raises the issue of certification. It is all very well to state the requirements of a secure system, and to claim that a system - the "ideal" system - satisfies these requirements. But we are then faced with the issue of certifying that a system is secure. Certification of a system of the size and complexity of, for example, OS/360 presents a formidable - indeed impossible - task. We contend that by following some of the trends presented below, certification becomes viable although still extremely difficult.

Exhibit 1 serves to introduce the key concepts underlying integrity, access control, and certification.

We contend that system integrity is a logical prerequisite for uncompromisable access control, and both integrity and access control are prerequisites for certification. Everything, therefore, hinges on integrity.

An example may help to indicate the importance of system integrity. Mechanisms, such as the IBM System/360 storage protection keys, are frequently used to prevent access by user programs, the areas of storage used by the operating system. An "integrity hole" exists if it would be possible for a user program to gain access to supposedly protected areas. How can such a hole exist? Let's consider a possibility.

Most systems provide a facility whereby the user program may request a copy of the current date by means of a supervisor request such as REQUEST DATE (LOC), where LOC is the location in which the date is to be placed. Presumably, LOC is within the user's authorized area of storage. What if it is not? Then the operating system -- if it does not carefully check -- may put the date into a
Exhibit 1.

Integrity, Access Control and Certification

Integrity --- The predictable and meaningful behavior of the system (i.e., Does the system follow orders correctly? Can it be confused?).

Authorization (Access Control) --- The facilities, procedures, and restrictions for establishing access controls to be enforced by the system.

Certification --- The extent to which the integrity of the system can be proven or measured.

Example by analogy: Let us consider a guard for a warehouse.

1) Integrity --- We must make certain that the guard understands all aspects of his job -- leaving no uncertainties. For example, if his replacement is late for work, he should not leave his post. Every situation must be identified and appropriate actions specified.

2) Authorization --- The warehouse would be rather useless if there was no way to get things in and out. Thus, there must be rules established that the guard can follow to determine whether a request for access should be permitted. Since the guard cannot be instructed, when hired, on the specific handling of each of these future requests, he is, instead, told how to determine the correct action.

3) Certification --- Given that we have carefully instructed the guard on his duties, can we be certain that he will follow these rules faithfully? We would like to certify our guard's competence by some test.
location not authorized to the user and thereby destroy supposedly
protected information. The storage key is designed to restrict the
user's access. If the integrity of the system is such that it
permits the above incorrect action, then the storage key is rendered
ineffective. This example has its counterpart in a manual system.
Many guards have been tricked into letting penetrators into locked
areas under the guise that they had left something behind or have
come to repair the telephone, etc.

It appears that many systems in operation today supply access
control without a high degree of system integrity. This is analogous
to building a 20-foot high wall, and then leaning a ladder up against
it. Because we contend that system integrity is a logical
prerequisite for uncompromisable access control, this is clearly not
conducive to providing a secure system.

We shall structure the remainder of this report as follows:

A) Integrity
   I) Data integrity
   II) System integrity

B) Access control
   I) System access control, i.e., control of access to the
      system as a whole, by the login procedure
   II) User address space access control
       a) Data set level
       b) Sub-file level

2.0 INTEGRITY

2.1 DATA INTEGRITY

By "data integrity" we mean the ability to insure that only
valid data is entered into the computer and that inconsistencies are
prevented. For example, we recognize 030-34-7261 as a valid social
security number, whereas 134-736-123 is not. Also, if a record
indicates a person's age as 28, a birth date of July 19, 1932, for
that same person is clearly inconsistent.

At the present state of the art, extensive data integrity is not
common and is usually situation specific. For these reasons we do
not pursue this topic in this paper. It is assumed that future data
base management systems will direct more attention to this area.

2.2 SYSTEM INTEGRITY

Two key issues are involved in system integrity:

1) The system will do what it is expected to do,

2) The various parts of the system behave as specified,
   irrespective of the environment in which they operate, i.e.,
   they are consistent.
System integrity subsumes such issues as detection and recovery abilities, as well as predictable and meaningful behavior in all situations. It is of major importance that a system have a high degree of integrity if, as in our prior analogy, the ladder is to be removed from the wall - or at least shortened to 10 feet.

2.2.1 Behaving As Expected

The first of these requirements - behaving as expected - can be approached through modularization of the system's subtasks. This is common practice today, and has been for some time.

In order to allay any possible confusion, two separate issues are presented: modularization, and structuring.

Modularization is the breaking up of tasks into multiple, smaller subtasks, or "modules". This implies nothing about the interrelation of the modules.

Structuring, on the other hand, implies a clear, well defined, limited set of interfaces between modules, i.e., there is a structure imposed on the interaction between modules and subsystems (5).

Here we are concerned with both modularization, and structuring. If modules are of sufficiently small size, and module interactions of a precise nature, it is far more probable that the behavior of the system will not give cause for dismay. This is a major step toward certification.

2.2.2 Consistency Of Operation

The second requirement for system integrity - consistency of operation - is of somewhat more concern. It is far more complicated to be able to certify that a module will behave as expected in varying environments, than it is to do so for a constant environment. This can be approached by clean, well-structured system design. "Hole plugging" has at least two serious drawbacks:

1) Correcting a bug, or "plugging a hole", may well give rise to other bugs, i.e., there is a propagation effect.

2) The cost may become exhorbitant as a result of the propagation effect mentioned above.

The trend toward well structured systems is thus an extremely sensible one, and one that is gathering an increasing following. Despite the fact that the motivation behind this trend may be to decrease and control costs, rather than the provision of secure computer systems, the ramifications are of great importance in the provision of secure computer systems.

2.2.2.1 Multiple-State Machines

Part of the evidence of this trend can be found in the change from the more common "2-state" computer (problem- and supervisor states) to the multiple-state machine. In the 2-state machine, which is the common third generation computer, certain instructions are designated as being privileged, and so may be executed only in
privileged - or supervisor - state. Notice that the designation of certain instructions as "privileged", designates certain functions as being "privileged". For example, in the IBM System/370, the "SIO" (START I/O) instruction is a privileged instruction. This, then, makes I/O a privileged function.

Similarly, more than two states allows more than two levels of function privilege. In computers with multiple states, the most privileged functions will exist in the most privileged state. For example, the paging mechanism might be in a more privileged state than the file system. The file system will, in turn, be in a more privileged state than a user process.

Examples of multiple-state computers are:

- DEC PDP-10's 3 state machine with kernel, supervisor, and problem state (9). Kernel state will contain such sub-systems as the paging mechanism, the more basic interrupt handlers, etc. Supervisor state will contain such sub-systems as the I/O- and device management routines. The user program will run in problem state.

- The Honeywell MULTICS System has a multi-state computer embodied in its ring structure (7). These rings can be viewed as a series of concentric annuli, numbered from zero in the center, sequentially to its outer ring. The most privileged processes run in ring zero, less privileged system routines in ring one, and so on out to the user processes. Transfer of control into a more privileged ring than the one in which execution is currently, has to pass through a logical "gate", where the credentials of the caller are checked by a "gate keeper". Information passed from a more privileged to a less privileged ring is checked against the original caller's security clearance, before being permitted to pass.

- IBM's OS/VS2 is utilizing the storage key to provide a 7-state operating system with 8 being the problem state (10). The storage key is no longer needed for separation of user address spaces, as this is now accomplished by making use of page tables. Again, the most privileged system functions - the paging mechanism for example - runs under key 0, and successively less privileged system functions run under successively higher keys. User processes run under key 8.

- Structured programming is facilitated on the Burroughs B6700 by the use of "display-" or "lexical levels" (2,8). The B6700 is a stack-based machine. The display level is the index in the processor's current stack. Entry 0 -- lexical level 0 -- in this stack is the System Stack Descriptor. The System Stack contains descriptors for all system routines. These system routines all execute at display level 1. All user routines operate at display level 2, and higher. Each called user routine (subroutine) operates at a display level that is one level higher than the calling routine. Notice that this also makes available a history of calls, so that data passed to the original caller can be validated as data that he has clearance to see. This history is available in the user stack. This multi-state feature is inherent in Burroughs machine architecture. It is not an added feature.

The multi-state feature is being implemented in the hardware of these systems, and is evidence of the trend toward structured programming. The importance of this approach is that it accomplishes separation of the address spaces of:

- the system components,
subsystems placed on top of the operating system, from the
operating system, and,
• system and subsystem components from their users.

This clearly delineates the bounds of operation of each process;
the most privileged process being in the "kernel state" -- or "ring
0", or "storage protect key 0" -- and moving outward from there to
less and less privileged processes. Notice that these processes of
differing privileges are now separated by the hardware -- much as the
user address spaces were separated by the storage keys in the IBM
System/360. Similarly, in multi-state computers, the processes of the
same state are separated from each other by the use of the
segment/page tables. In this way, all interfaces can be clearly
specified, controlled and monitored. The repercussions of the
ability to do so are far reaching, and have their ends in detection
and recovery procedures.

The integrity of the system is greatly improved with the
multi-state approach, and the results are applicable to access
control. (This is inextricably bound up in the concept of the virtual
machine -- (6).) This is perhaps the major trend to be observed in
this section of the report. As stated previously, this trend may be
in evidence for other reasons -- most notably cost control -- but it
is a trend that has great applicability in the design and
implementation of secure computer systems.

At a somewhat higher level of system function, is the notion of
the hierarchical file system, with successively higher levels of
privilege needed to access higher nodes of the file structure. The
hierarchical file system separates logical file references from
physical devices. In general, it is safe to say that all of these
systems allow logical file references only, which is in keeping with
the concept of the virtual machine, and separation of user address
spaces. It appears that these trends are going to continue, and that
this continuation is going to be a major step toward making the
system of high integrity a reality.

2.2.3 Requirements for Integrity

Let us now propose some of the major requirements for building
an integral system:

1) All sub-parts of the system are isolated from their
   environment.
2) Detection of errors of any type occurs.
3) On detection of an error, recovery procedures are available to
correct that error. If no procedure exists for a particular
error, the error will be logged, and the system will shut down.
4) There is continuous monitoring of the system, for the following
   reasons:

   a) detection of errors is facilitated, and
   b) a record of activities can be kept which will enable the
      system to return to the same point as that at which the
      error occurred - i.e., recovery is facilitated.

We postulate further that, if isolation of a module from its
environment is, in fact, made a reality, then the only way in which
it communicates with that environment is via the use of parameters. Furthermore, we can assume that the module will check that any parameters passed to it are valid - as part of the "will perform as expected" requirement. The only point, then, that has to be monitored, is the output of all modules. It is only possible for the module to influence its environment by its output, and so detection of errors is simply a function of monitoring the output of all modules. This ability relies also on the fact that, in a well structured system, all interfaces can be identified.

Recovery is a function of detection. If we can identify all interfaces, and are able to detect all errors at the interfaces, then we can produce a recovery mechanism to take care of the situation. In all cases, the key to success is the isolation of the modules from the environment in which they operate. This isolation is facilitated by the existence of the multiple-state machine.

3.0 ACCESS CONTROL

3.1 SYSTEM ACCESS CONTROL

By "system access control" we mean control of access to the system as a whole, via the login procedure. The standard approach here has been to supply each user with a unique password which he must use to verify his identity. User identification is quite a critical issue. Even in a secure system, a user can negate the effect of the most sophisticated access control mechanism if he is able to masquerade as another user. Positive identification of a user is the first step in the process of access control, and, thus, is crucial. Speculation on this subject would tend to indicate that the way of the future is something along the lines of a hand- or voice-print identification -- i.e., physical identification mechanisms. At the present, the user-ID/password combination seems to be considered sufficient for most applications.

3.2 ADDRESS SPACE ACCESS CONTROL

3.2.1 Data Set Access Control

As stated in Section 1, system integrity is a prerequisite for uncompromisable access control. It does not make sense to have a sophisticated access control mechanism which permits any user of moderate sophistication to bypass it. (It is, in fact, with this very issue that we are dealing in the Service Bureau Industry.)

We should point out that there are several other approaches to data set access control in what might be called the "laboratory" stage, in that they have not yet been implemented in a working environment. We thus exclude them from our discussion of the state-of-the-art. We can categorize the approaches seen to date into two very broad classes:

a) Password/access-control-word oriented
b) Procedure oriented.
3.2.1.1 Keyword Access Control

3.2.1.1.1 Password Control

Under "password/access control word" schemes, fall several mechanisms. These range from the simple "you can have it if you know the password," with a "0-1" type of control, to the complex "SET ACL/CACL" of the MULTICS system, where each node of the file hierarchy (be it a directory or a program- or data-file) has a list of people who may see that node, along with their various privileges.

It is noteworthy that MULTICS has changed to a technique where different people and/or projects are given different privileges. Each person may be given privileges as an individual (7).

Filling in the spectrum between "all or nothing", and "per-person" specification of access privileges, is Digital Equipment Corporation (DEC) (9), where the users are divided into three groups, each of which may have different privileges. These groups are:

- the owner of the data,
- other users with the same programmer number, and,
- all other users.

There is also the Dartmouth Time-Sharing System (1,3) -- DTSS -- which divides users into two groups: password and non-password holders. The Dartmouth scheme provides two access fields for each file: the first specifies the allowable access without a password, the second specifies the allowed access to those that know the password. There are also facilities for procedure oriented controls, via the use of the "trap" facility (see below). Controls are along the lines of:

- execute only,
- read only,
- append only,
- write only,
- owner privileges,
- delete only,
- trap facility,

with one group for password holders, and an identical group for non-password holders.

It should be pointed out that this system has been designed along clean, logical lines, and all aspects of the system are sensible. The result is a system with a record of reliability, integrity, and security, which may well be unparalleled for a system of comparable magnitude. An important feature of DTSS is that there is no logical way to bypass the access control mechanism and the integrity of the system is exceptionally high.

3.2.1.1.2 Cryptography

Under "password/access control word" schemes, there is also the approach of "rendering data useless". The major technique here is cryptography. Again, this hinges on knowledge of the cryptographic key (a form of password) and so falls into this section, rather than being in a separate class. There is an important distinction to be made, however, between the above password schemes and a cryptographic
key. The distinction is that the password schemes mentioned above require that the password be stored internal to the system. The cryptographic key is not internal to the system. This means that there is no way for the system to determine whether the key is correct. The file specified will be decrypted irrespective of the key entered. However, unless the correct key is used, the data will be as meaningless as in the encrypted form.

3.2.1.2 Procedure Control

Under "procedure oriented techniques," are such schemes as traps and the "PROTECT EXEC" of National Computer Software Systems (NCSS). Both of these facilities provide for the specification of routines that are to be invoked at the time that a user attempts access. These invoked routines can do further checking as to the authority of the process making the request. Such routines are generally user-written, and can be as simple, or as complex as the individual user desires. A problem with this technique is that it tends to require a moderately high degree of sophistication on the part of the user. Such techniques are, nevertheless, available and currently in use.

3.2.2 Subfile Access Control

By subfile access control, we mean control of access to different records of a file, and in some cases, different fields of a record. This is of particular importance in Information Systems that use shared data banks.

As in the case of data integrity, we feel that control of access on a subfile level falls outside the range of responsibilities of the manufacturer. However, it is the responsibility of the manufacturer to provide facilities that make this type of extension of the access control mechanism easy to implement. These facilities - often called "hooks" - are exemplified by such mechanisms as the DEC convention of two types of calls to the operating system:

- Calls in which the code number passed as the parameter is positive; these calls are to standard manufacturer-supplied facilities.
- Calls to the operating system in which the parameter is negative. Users may use negative parameters to invoke either their own routines or installation-written routines.

The fact that all negative parameters are available for provision of site-specific facilities, provides the user with "hooks" on which to hang his own utilities.

One such hook, namely program access control, essentially solves the problem of subfile access control. By "program access control" we mean the ability to treat a program as a user, and deny or permit it access to a data set. We could thus write our own routine that would perform the function of controlling access on a subfile level. This routine could intercept all requests to the data set, and only permit those that were legal. We could further control access to the data set as a whole in such a way that it would be restricted from access by any other user or program. (This is the "hook": the program access control.) In this way, our routine supplies the only interface with the data set, and that routine could be as simple -- asking for additional passwords -- or as complex -- selecting only the permissible fields of permissible records -- as we care to make it.
These "hooks" should not affect the integrity of the manufacturer-supplied system. The fact that we have a facility (as in the DEC system) to call the operating system to invoke our own packages should not mean that we have free access to all, or any, parts of the system.

4.0 CONCLUSION

The need for certifiably secure computer systems is destined to become a pressing one indeed.

Two requirements for a secure system are:

1) A system of high integrity
2) Uncompromisable access control mechanism(s).

We contend that (1) is a prerequisite for (2).

A major trend can be observed in the computer manufacturing sector: well designed, well structured systems appear to be emerging as the norm. There is a strong probability that the reason for this trend is financial rather than for the provision of secure systems, but it certainly has far reaching benefits in this direction.

Judging from these trends, there appears to be a strong indication that major systems in the future will be designed and built along these lines.

Exhibit 2 summarizes the various security techniques in general use, and indicates examples of systems that use these techniques.
### Exhibit 2. The State of the Art and Computer Security

<table>
<thead>
<tr>
<th>Feature</th>
<th>DEC PDP-10</th>
<th>IBM OS/VS2</th>
<th>Burroughs 6670A</th>
<th>IBM VM/370</th>
<th>Multics</th>
<th>DTSS²</th>
<th>IBM OS/360 (TSO¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modularized</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Multi-state Machine</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Virtual Memory</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Virtual Machine</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>A</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Virtual I/O (Logical refs only)</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Structured System</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Extensive Recovery Procs.</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Back-up Facilities</td>
<td>A</td>
<td>?</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>ID/Password Combinations</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Multiple Passwords</td>
<td>A</td>
<td>N</td>
<td>N</td>
<td>A</td>
<td>A</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Password/Terminal Combination</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>N</td>
</tr>
<tr>
<td>Extended Handshakes</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Password</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>N</td>
<td>N</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>User Categorization</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Access Types Differentiation</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Cryptography</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Access Control Lists/Data Set</td>
<td>N</td>
<td>N</td>
<td>A</td>
<td>N</td>
<td>A</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Process Access Control</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Traps</td>
<td>A</td>
<td>N</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>N</td>
</tr>
</tbody>
</table>

**Key to Exhibit.**

- **A** = Implemented
- **B** = Implementable, using "hooks"
- **N** = Not available
- **?** = Information not available

**Notes:**

1. Included for purposes of comparison.
2. Dartmouth Time Sharing system.
Explanations of Terminology of Exhibit 2.

**System Integrity**

**Modularized:** See 2.2.1

**Multi-state:** See 2.2.2.1

**Virtual Memory:** Each user is given a virtual address space, part of which is in main storage while his job is running. Not all of the address space need be in main storage to run a job. This is accomplished via segmentation, paging, or a combination of the two.

**Virtual Machine:** The ability to run several operating systems on the same machine concurrently, e.g., OS/360, DOS, and OS/VS2 may all run concurrently under VM/370.

**Virtual I/O:** All I/O references logical devices only. The system may link any physical device to that logical device, and vice-versa. The user never has access to a physical device directly.

**Structured System:** See 2.2.1

**Extensive Recovery Procedure(s):** See 2.2.3

**Backup Facilities:** See 2.2.3

**System Access Control**

**ID/Password Combinations:** When signing onto the system, the user enters an ID, and the system then requests a password.

**Multiple Passwords:** The system may request additional passwords, over and above that of the ID/Password Combination to authenticate the user.

**Password/Terminal Combination:** Certain passwords will only be accepted from specific terminals.

**Extended Handshake:** The login procedure is extended beyond ID/Password Combinations. Several additional tests (other than addition passwords) will be administered to the user for purposes of authentication.
Explanation of Terminology of Exhibit 2. (continued)

**Data Set Access Control**

**Password:** Before allowing access to a data set, the system requests a password. (See 3.2.1.1.1)

**User Categorization:** Dividing the user community into groups, each with specific access rights. (See 3.2.1.1.1)

**Access Types Differentiation:** Permitting specification of the type of access to be allowed to a data set, along the lines of Read, Write, Execute, etc. (See 3.2.1.1.1)

**Cryptography:** See 3.2.1.1.2

**Access Control List/Data Set:** Each data set has associated with it a list of people (or groups) that may have access to it, as well as their type of access. (See 3.2.1.1.1)

**Process Access Control:** See 3.2.2 and 3.2.1.1.2

**Traps:** See 3.2.1.2
5.0 REFERENCES


# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>THE SECURITY DECISION</td>
<td>199</td>
</tr>
<tr>
<td>THE DECISION CRITICAL PATH MODEL</td>
<td>200</td>
</tr>
<tr>
<td>ISOMORPHISM OF THE DECISION CPM WITH THE SECURITY DECISION MODEL</td>
<td>200</td>
</tr>
<tr>
<td>THE SECURITY PROBLEM TRANSLATED INTO CPM TERMS</td>
<td>200</td>
</tr>
<tr>
<td>NETWORK STRUCTURE OF THE MODEL</td>
<td>201</td>
</tr>
<tr>
<td>PROBABILISTIC ASPECTS OF THE MODEL</td>
<td>202</td>
</tr>
<tr>
<td>VALUE OF THE NETWORK REPRESENTATION</td>
<td>202</td>
</tr>
<tr>
<td>FORMULATING THE PROBLEM TO FIND THE MOST COST-EFFECTIVE SOLUTION</td>
<td>205</td>
</tr>
<tr>
<td>CRITERIA FOR IDENTIFYING DOMINATED SOLUTIONS</td>
<td>206</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>207</td>
</tr>
<tr>
<td>FORMULATING THE DECISION CPM MODEL</td>
<td>207</td>
</tr>
<tr>
<td>I. DEFINITIONS</td>
<td>207</td>
</tr>
<tr>
<td>II. HEURISTIC TECHNIQUES</td>
<td>208</td>
</tr>
<tr>
<td>SAMPLE FORMULATION FOR VS2 RELEASE 2</td>
<td>211</td>
</tr>
<tr>
<td>CONCLUSION</td>
<td>216</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>217</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>218</td>
</tr>
</tbody>
</table>

Massachusetts Institute of Technology  Volume 4  197
THE SECURITY DECISION

Our studies have approached the security problem from both ends -- (i) the user environment and (ii) the state of the art. In this paper we attempt to show how the gap between the two may be closed.

The user (D.P. Manager or Systems Engineer) is confronted with a number of difficulties concerning the appropriate set of security features for his system. He must enumerate the items to be protected and the directions from which they are threatened (sources, paths and impact points). These must be translated into technical specifications that are matched with the available techniques (constrained by the installed system if the user is not considering a new purchase).

Difficulties arise because there are often multiple ways in which to meet the user's needs. Choosing between these alternatives is not simple as multiple dimensions exist for comparison, e.g., dollar cost, CPU load, response time and availability. The significance of these dimensions depends on environmental conditions (e.g., present CPU capacity utilization). These dimensions are not reducible to a single dimension such as dollars, without loss of information critical to the decision maker. The manager or systems engineer is not able to optimize his choice because the trade-off across dimensions varies with external conditions. However, there may be solutions that dominate others on all relevant dimensions.

The use of a network representation (nodes representing available protection techniques and paths representing avenues through the system) offers us two advantages. It enables us to represent the logical prerequisite relations between different techniques and reveals 'cunning' solutions that clearly dominate others.

We are thus able to offer the user a set of dominant solutions from which to choose. These are 'cunning,' economical, and correctly located within the system. The output must be translated into choices from which the user can determine the appropriate techniques.

THE DECISION CRITICAL PATH MODEL

The Decision Critical Path Method is a way of formally considering the interaction between the scheduling and planning phases of a project. (See Crowston and Thompson 1967 (2).)

1. Several competing methods are available for performing some of the jobs. Each method has a different cost, different time duration, and different logical dependencies. These possibilities are represented as a network so that the alternatives may be compared.

2. In the actual decision-making phase, consideration is given to the effect of the alternate methods on the total cost of completing the project. The alternatives that minimize the cost are then calculated.
To understand why this model is useful in describing the Computer Security Decision, it may be helpful first to point out the similarities and then to translate the model into our particular decision.

ISOMORPHISM OF THE DECISION CPM WITH THE SECURITY DECISION MODEL

Four major similarities are:

1. We have alternative techniques for achieving the same purpose, and are seeking feasible configurations.

2. Although we have multiple dimensions for measuring cost, we can demonstrate solutions that are dominant and that offer the decision maker trade-offs.

3. Security techniques have clear precedence relations which must be formalized.

4. We can make changes in a system where a number of features (nodes) are already determined.

THE SECURITY PROBLEM TRANSLATED INTO CPM TERMS

1. For choosing appropriate security control techniques within a computer system, several competing methods (nodes) are available for the same task. Each method has a different cost and a different impact on the system's resources. Multiple dimensions can be attached to the cost coefficient; e.g., availability can be represented as zero for techniques currently available, and as very large for those that are not available.

Logical dependencies are critical in configuring a given set of security mechanisms. For instance, limiting data or program combinations to particular users may be dependent both on preceding identification and on some kind of reference monitor. The network representation has some major advantages in handling precedence relations of this kind.

2. In the actual decision-making phase, the method will enable us to determine which set of techniques is best for a given cost dimension. The trade-off between the dimensions is a difficult one. Obviously, where one dimension is overriding, it is simple to pick the optimal set. Mixed strategies present more difficult problems unless the trade-off criteria can be formalized. For example, both response time and CPU utilization may be important, but at different levels of activity or user satisfaction, the trade-off will be different. Thus, the trade-off is not uniquely determined; it is situation-dependent. At any one point, however, it might be formalized. The following example may help to clarify these points: suppose we have four alternate feasible paths (configurations ranked and measured as costs on three dimensions). (See Exhibit A.)
If the decision maker is more concerned with 'Availability and Support' than with any other dimension, he will choose D for his IBM system. However for CPU Utilization and Impact on Response Time, D is clearly inferior. No single configuration dominates on all dimensions.

NETWORK STRUCTURE OF THE MODEL

The first step in formulating the Decision Model of the Security Technique is to develop a network representation of the possible configurations. Each node represents a particular technique for control or checking. The particular node will either be chosen or dropped compared to other nodes that fulfill the same function. Particular configurations will be chosen over others for their efficiency in fulfilling a set of functions. Later in the paper we develop a particular example based on VS2 Release 2.

Exhibit G shows how the network might look. Control points in a column represent techniques. The column represents a function or group of techniques. Each branch of the tree represents a transition between control points, during which activities not involving security checks are done. (See Appendix on "The Logical Nature of Security." In a correctly designed system, any attempt to cross an interface between activities will be monitored or inhibited by a control mechanism.) The arrowheads indicate the sequence of controls. Other sequences of functions are possible, but Exhibit G indicates a particular order based on the hierarchical nature of the functions to be performed.

The model enables us to find the most constraining set of techniques that can be used to satisfy a given set of requirements.
PROBABILISTIC ASPECTS OF THE MODEL

It is stated elsewhere (Appendix) that all security controls, by their very nature, have less than 100 percent probability of detecting errors. Even if a system is logically perfect and secure from integrity problems, reliability is less than 1. The normal procedure is to use the probability of the mean value to measure the 'expected value.' For problems that fluctuate only with reliability, this is a reasonable estimate. However, where a loophole exists that makes systematic exploitation possible, as in an imperfectly designed computer system, we must abandon our 'expected value' approach.

VALUE OF THE NETWORK REPRESENTATION

The network representation is used to configure the security system because it can specify precedence relations between the different control mechanisms. The cost of the mechanisms may also vary with the particular configuration chosen -- a kind of 'synergy.' This can be represented by the different values attached to the paths connecting the nodes. The direct costs are the price of obtaining and installing individual mechanisms. The incremental changes, caused by grouping in one way rather than another, are reflected indirectly in the cost of each path. For instance, grouping program, data, or user control with both a reference monitor and a data-base management system, may result in a different cost being attached to the configuration than if a traditional file structure was used. Changes in one mechanism may, thus, affect the costs of all others.

Where efficient combinations result in overall savings, the path costs can reflect this. We can recognize that the efficiency of the 'cunning' solution is due to cost reductions that are dependent on the particular path by which a node was reached. The system is, therefore, not 'memoryless' as the traditional CPM model is: the costs to be attached to any configuration are a function of the paths used. The choice of control mechanism is related to the particular controls exercised elsewhere in the system.

One procedure for calculating the cost-coefficient matrix consists of starting at the end of the network (i.e., at the point of final impact or object. See Graham and Denning (1972) (3)) and calculating the relative costs of using different configurations of controls to reach this object. For further discussion of Graham and Denning's work, see Stepczyk (1974) (7). The most constraining series of checks will not necessarily be the most expensive to implement. Thus we can demonstrate 'cunning' solutions, whereby we get better control for less cost.

An example may be helpful here. (See Exhibit B.) A user (1) can obtain access to the system in multiple ways, e.g., by terminal (2), a central batch input (3), or a remote system (4). Each of these access points has one or more ways of checking, for example, verifying the identification of users. These are labeled A, B, C, D, and E. Five ways are available to gain access to the system. We can, therefore, construct a matrix. (See Exhibit C.) The numbers in the matrix are cost figures.
Exhibit B

Exhibit C

Source Identification Procedures

<table>
<thead>
<tr>
<th></th>
<th>2A</th>
<th>2B</th>
<th>3C</th>
<th>4D</th>
<th>4E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

The user is asked qualitatively to evaluate the rankings of different techniques (with the help of questions). Suppose, for simplicity, we have no other checks in the system during processing (5) in Exhibit B until the user's program asks for data from the database (6). We have three techniques for authorizing access: F, G, and H. Exhibit D shows that G and H will be cheaper to implement if we already checked for both ID and Program (or other user information that enabled us to deduce the program e.g., from conversation with the user).
Here B represents program/ID combination and D remote authentication.

The cost matrix is shown in Exhibit D. We must, therefore, calculate the cost for arriving at each node as a function of the path chosen.

Exhibit D

<table>
<thead>
<tr>
<th>Path to Node 5</th>
<th>Data Control Procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6F</td>
</tr>
<tr>
<td>1-2-A-5</td>
<td>2</td>
</tr>
<tr>
<td>1-2-B-5</td>
<td>1</td>
</tr>
<tr>
<td>1-3-C-5</td>
<td>2</td>
</tr>
<tr>
<td>1-4-D-5</td>
<td>1</td>
</tr>
<tr>
<td>1-4-E-5</td>
<td>1</td>
</tr>
</tbody>
</table>

The most constraining check is H, but if combined with D, it is cheaper than if combined with E. We, therefore, have a dominated solution: E. The total cost matrix is shown in Exhibit E. This table indicates that the cheapest way to implement H is 1-4-D-5-6-H, whereas the cheapest procedures are 1-3-C-5-6-F and 1-4-E-5-6-F (if we only wish to implement F). 1-4-D-5-6-G is preferable if we wish to implement G.

Exhibit E

<table>
<thead>
<tr>
<th>Path to Node 5</th>
<th>Cumulative Control Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6F</td>
</tr>
<tr>
<td>1-2-A-5</td>
<td>4</td>
</tr>
<tr>
<td>1-2-B-5</td>
<td>6</td>
</tr>
<tr>
<td>1-3-C-5</td>
<td>3</td>
</tr>
<tr>
<td>1-4-D-5</td>
<td>4</td>
</tr>
<tr>
<td>1-4-E-5</td>
<td>3</td>
</tr>
</tbody>
</table>
FORMULATING THE PROBLEM TO FIND
THE MOST COST-EFFECTIVE SOLUTION

Dynamic programming is the traditional technique for finding the most cost-effective path out of a set of alternatives. Normally, the following are true:

1. The cost-coefficient matrix is static, and independent of previous choices in the network.
2. The network can be decomposed into subproblems, the costs of which can then be minimized.

We have costs that are a function of the path chosen. Thus, we cannot decompose the network into separate subproblems. If we were to have precise knowledge as to how costs change as a function of previous configuration choices, we would be able to handle the interdependencies. Currently, it is not clear that we can formulate these. For example, how does the cost of field-level access control change as a function of a data-base management system? Although any one specific case can be calculated, we cannot state the general case a priori.

The procedure described in the last example involves complete enumeration. This becomes increasingly difficult to handle as the system becomes more complex and detailed. Therefore, we should establish criteria and logical ranking schemes that enable us to omit 'clearly dominated' solutions from our network (at least for given cost dimensions). The truncated network will then consist of alternatives that are not obviously superior or inferior to one another. Rather, they will differ only in cost and this may be evaluated differently at different times or by different users. Note: A switch of the dimensions for measuring cost will change the evaluation.

The decision maker has to consider:

1. Techniques that perform the same control for less resource utilization than others.
2. Controls that are more powerful for the same (or less) resource utilization.

Massachusetts Institute of Technology  Volume 4  205
The user must determine the trade-offs between different levels of resource utilization.

**CRITERIA FOR IDENTIFYING DOMINATED SOLUTIONS**

We can use a framework like that of Graham and Denning (3) to construct tables that classify a technique on the basis of the characteristics of the Subject (Origin, Process, and Domain) and the Object (Impact Point to be protected).

Having translated our requirements into subjects and objects we can establish the following criteria. (See Appendix.)

1. Control techniques that provide separation of users dominate those that do not. Referring to the matrix developed by Grant Smith (1974) (6), it can be seen that one machine design will dominate another. Often we are not facing a choice between machines, but rather a choice between versions of the operating system. If we consider an IBM System/370 installation, it is clear that the VS2 Release 2 control of separate user address-space is superior to that of OS/370-MVT. Note: A solution that dominates on one cost dimension may be dominated on another. For example, VS2 currently may be dominated by OS on the dimension of system response time. The trade-off between security and response time must be made by the user.

2. All access-control mechanisms that are logically impossible to bypass dominate those that contain 'escape clauses,' e.g., loopholes designed for systems programmers. If the access-control mechanism can be bypassed, the degree of difficulty in doing so will cause some techniques to dominate others (but all are far inferior to techniques that cannot be bypassed). For example, techniques requiring operator collusion dominate those where a lone programmer can avoid the system control. These techniques, in turn, dominate situations where the default is 'No Access Control.'

3. Techniques that perform more checks for the same resource utilization are superior to those that perform less. If greater control is exercised while using fewer resources, we have a 'cunning' solution.

4. Any solution that logically prevents increase in one's priority level is superior to one where it is sometimes possible. Although difficulty is a factor, because systematic exploitation of a loophole is feasible, the relative difficulty may be academic; it may be a challenge! The experience of those involved in the Dartmouth Time Sharing System supports this.

5. Control mechanisms located close to the object to be protected are preferable to those located farther away, unless the latter provide additional features, for instance, context-dependent access checking - such as: Which type of account is being accessed by the payroll rounding routine? In this case, more control may be exercised if the control is close to the source rather than the object. Generally though, remote controls will be less efficient and will have a greater probability of being bypassed. Comparisons for a particular system must be made here.
6. Unmonitored interfaces are incompatible with a secure system. A change in source, domain, process, or access should initiate certain checks for illegality, i.e., situations where harm to other than the originator can result. Any techniques whose default is "no security" undermine the validity of the network control model. Note: An analogy may be made here with structured programming. Paths that do not enter or leave functional modules by a standard interface can cause havoc.

7. Control mechanisms require their own types of protection. Mechanisms must be able to be modified occasionally, but must be used frequently, without possibility of modification. Mechanisms with the capability of owner-access dominate those without.

CONCLUSION

We have discussed several control dimensions on which one feature dominates another. Obviously the cost dimensions are equally important. Determining the appropriate cost measures for a particular user must be done through an interactive process. Questionnaires presented elsewhere in this Volume illustrate the type of questions to be asked.

FORMULATING THE DECISION CPM MODEL

I. DEFINITIONS

Let $C = C(1), C(2), C(3), \ldots$ be a set of control functions required to ensure adequate security. To ensure the desired level of security, some of the control nodes within each control function must be implemented. For each control function define the set of nodes to perform that particular security function,

$$C(i) = C(i, 1), \ldots, C(i, k(i))$$

With each set of control nodes, associate $k(i)$ variables, $d(i, 1), \ldots, d(i, k(i))$ so that $d(i, j)$ is 1 if the control node $C(i, j)$ is to be implemented and is zero otherwise. If exactly one of the controls in $C(i)$ must be implemented, then $C(i)$ requires the mutually exclusive interdependence condition. In our problem this means that we will only select one technique to perform a particular security function. Multiple functions can be performed, however.

If groups of components combine to form a particular control mechanism $C(i, k(i))$, they can be subsumed in a given node. Within a given node, choices may have to be made between groups of components. This is another network of component mechanisms within the node $C(i, k(i))$. These could be represented as members of the set of mechanisms for a node, equivalent to the set of techniques for a given function. We have not defined such subsets in detail, (e.g., different techniques for Memory Protection), as each lower level involves an entire network representation itself. Unlike the problem that we have defined, alternatives are not mutually exclusive; they may be complementary. As a result, formal techniques that are dependent on heuristics (because of the dynamic cost difficulty) may become unduly cumbersome. The most valuable part of the network model for such choices is often the enumeration of alternatives. The Decision CPM model will enable alternative designs for the same control mechanism to be grouped to form the solution.
Where complete searches of the network are required, this is often most efficiently performed by the decision maker. Constructing rules for eliminating dominated solutions is a helpful prerequisite to systematic search. Complete automation of the search is often undesirable. Heuristic methods, however, enable us to handle the dynamic cost problem, i.e., cost can change as a function of a particular configuration. We can thus hand back to the user of the model two decisions: (1) the trade-off across cost dimensions and (2) the evaluation of a minimum acceptable control level.

There may still be no feasible way to satisfy a user's needs for his present computer system. Calculation of prices for introducing better configurations, or relaxing other constraints, is possible with the heuristic method, but it is necessary to evaluate specific alternatives. Having found an "optimal" solution using a heuristic approach, the effort involved will be worthwhile if it suggests areas for developing new techniques. We should beware, however, of major reconfigurations caused by small changes in cost. The cost of moving between solutions may be very great. D.P. managers are well aware of the system disruption that may result from apparently small changes.

II. HEURISTIC TECHNIQUES

We have already discussed the information needed to create the network. The procedure described below is useful because it forces the decision maker to structure his thinking.

1. Construct the network configuration including the precedence relations and dummy nodes. This is a technological ordering.

2. Eliminate solutions not currently feasible or not under consideration.

3. Rank all possible paths through the network according to their control capability.

4. Identify the costs of each path on the user-determined dimensions.

5. Reduce the number of feasible paths in the network. If all nodes may be connected to all others, we have, in the example in Exhibit G, (eliminating dummy variables) 2,304 paths. (Order matters, and only one can be taken from each group.) Fortunately, if we eliminate nodes that represent unavailable configurations, we can eliminate a good deal of work. In addition, we can 'a priori' eliminate nodes or techniques that we consider unsatisfactory (or dominated). We can constrain the particular alternatives by the current computer system and its current state, allowing some limited system changes. We can also find node configurations that fall within a given budget constraint before setting up the network. Where a major system change is expected, it is wise to examine the alternative systems contemplated because of the security features they make available. It would be foolish to optimize the security configuration first and then find a system that made it possible. Security is only a part of the decision.
6. Having reduced the feasible paths to some practicable set, probably under 50 (see Exhibit H) we can begin to trade-off the degree of control for each path with the penalty for including it. The formulation of the trade-off matrix, as discussed earlier, is more valuable than attempting to translate all penalties and benefits to dollars. The matrix will look like that shown in Exhibit F. The rows represent paths, and the columns represent different cost and benefit dimensions.
Exhibit H

Each column represents a function.
Each node represents a control mechanism or technique.
Indicates a dummy node, i.e., No Control.

SAMPLE NODE INTERPRETATIONS

A - Terminal
B - Remote Batch
C - Password Identification
D - Password/User-identified Identification
E - Terminal/Remote identification
F - Monitor of Program requests
G - Job Queue Check/Privacy Control
H - Program Control limited by user
I - Program Control by time of day and program
J - Program Monitor (Trap Bit)
K - Read-only processor
L - Multi-state system
M - Access control Data/Program/User Combination
N - Access control Dynamic Formulary
O - Access control Reference Monitor
P - Access control Channel Program/Kernel I/O
Q - Use of VTAM Data Control Block
R - Read-only data access
S - Modification and read access
T - Encryption
U - Print Suppression
V - Particular device restriction
**SAMPLE FORMULATION FOR VS2 RELEASE 2**

Groupings of Possible Techniques Based on "Requirements for Secure Operating Systems", F. Stepczyk, (7)

(References are to Chapter 4.)

<table>
<thead>
<tr>
<th>Number</th>
<th>Name of Path or Function</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Isolation</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>User programs from user programs</td>
<td>IS1 - IS6</td>
</tr>
<tr>
<td>1.2</td>
<td>User programs from operating system</td>
<td>IS6</td>
</tr>
<tr>
<td>1.3</td>
<td>System element isolation from environmental factors</td>
<td>IS7 - IS16</td>
</tr>
<tr>
<td>1.4.1</td>
<td>Operating system from user programs</td>
<td>IS17 - IS24</td>
</tr>
<tr>
<td>1.4.2</td>
<td>Basic protection functions from user programs</td>
<td>IS25 - IS27</td>
</tr>
<tr>
<td>1.5.1</td>
<td>Basic protection controls from operating system</td>
<td>IS28 - IS33</td>
</tr>
<tr>
<td>1.5.2</td>
<td>Operating system from operating system</td>
<td>IS34 - IS39</td>
</tr>
<tr>
<td>1.6</td>
<td>Information from users</td>
<td>IS40 - IS45</td>
</tr>
<tr>
<td>1.7</td>
<td>Information from operating system</td>
<td>IS46 - IS48</td>
</tr>
<tr>
<td>1.8</td>
<td>Resources from user programs and operating system</td>
<td>IS49 - IS51</td>
</tr>
<tr>
<td>1.9</td>
<td>Information from information</td>
<td>IS52 - IS53</td>
</tr>
<tr>
<td>1.10</td>
<td>Resources from resources</td>
<td>IS54 - IS56</td>
</tr>
<tr>
<td>2.0</td>
<td>Controlled Access Techniques</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Techniques on scope of controlled access</td>
<td>CA1 - CA10</td>
</tr>
<tr>
<td>2.2</td>
<td>Techniques on modification of access attributes</td>
<td>CA11 - CA15</td>
</tr>
<tr>
<td>2.3</td>
<td>Authorization techniques</td>
<td>CA16 - CA23</td>
</tr>
</tbody>
</table>
3.0 Techniques for Identification

3.1 Identification of users ID1 - ID7

3.2.1 Terminal identification by computer ID8 - ID13

3.2.2 Terminal identification by operating system ID14 - ID15

3.2.3 Input/output stations and their media by computer and operating system ID16 - ID19

3.2.4 Storage devices/containers by computer and operating system ID20 - ID22

3.2.5 Computer identification by terminal ID23

3.2.6 Computer by another computer ID24

3.2.7 Computer by operating system ID25

3.3 Program/operating system/data identification and authentication ID26 - ID35

4.0 Integrity

4.1 Completeness/correctness techniques IN1 - IN3

4.2 Maintainability techniques IN4 - IN13

4.3.1 Error prevention techniques IN14 - IN17

4.3.2 Error detection techniques IN18 - IN26

4.3.3 Error correction/notification techniques IN27 - IN40

4.4 Data integrity techniques IN41 - IN50
5.0 Surveillance

5.1.1 Monitor action techniques      SV1 - SV2
5.1.2 Decision action techniques      SV3 - SV5
5.1.3 Compensatory action techniques  SV7 - SV11
5.1.4 Reporting action techniques    SV12 - SV13
5.2 Security audit techniques        SV14 - SV23

Note: 1.0, 2.0, 3.0, 4.0, and 5.0 indicate major functions, the decimalized numbers indicate control techniques (e.g., 1.1, 1.2, etc.) and alternate subsets of techniques (3.2.1, 3.2.2, etc.).

Using Stepczyk's listing of functions and techniques (see Reference 7), VS2 Release 2 was subjected to analysis. Implementation at M.I.T. is scheduled within the next year, and all answers are based on the currently anticipated form. Note that this is M.I.T.'s planned implementation which is being analyzed, not VS2 Release 2 per se.

Two hours were spent with an expert systems programmer to complete the list of available techniques. The list is considerably smaller than the set of possible techniques. A number of doubtful techniques were eliminated although they may become feasible. Note: For interpretation of numbers into techniques, see Reference 7.
<table>
<thead>
<tr>
<th>NUMBER</th>
<th>POSSIBLE TECHNIQUES</th>
<th>AVAILABLE TECHNIQUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function or Column in Network</td>
<td>Possible Nodes</td>
<td>Available Techniques for VS2 Release 2 as Planned for M.I.T. -- Actual Nodes</td>
</tr>
<tr>
<td>1.1</td>
<td>IS1 - IS5</td>
<td>IS4,5</td>
</tr>
<tr>
<td>1.2</td>
<td>IS6</td>
<td>IS6 (with restrictions)</td>
</tr>
<tr>
<td>1.3</td>
<td>IS7 - IS16</td>
<td>IS7,11</td>
</tr>
<tr>
<td>1.4.1</td>
<td>IS17 - IS24</td>
<td>IS17,18,19,20,21,23,24</td>
</tr>
<tr>
<td>1.4.2</td>
<td>IS25 - IS27</td>
<td>IS25,26,27</td>
</tr>
<tr>
<td>1.5.1</td>
<td>IS28 - IS33</td>
<td>IS30,31,32,33</td>
</tr>
<tr>
<td>1.5.2</td>
<td>IS34 - IS39</td>
<td>IS34</td>
</tr>
<tr>
<td>1.6</td>
<td>IS40 - IS45</td>
<td>None</td>
</tr>
<tr>
<td>1.7</td>
<td>IS46 - IS48</td>
<td>None</td>
</tr>
<tr>
<td>1.8</td>
<td>IS49 - IS51</td>
<td>IS49,50</td>
</tr>
<tr>
<td>1.9</td>
<td>IS52 - IS53</td>
<td>IS53</td>
</tr>
<tr>
<td>1.10</td>
<td>IS54 - IS56</td>
<td>None</td>
</tr>
<tr>
<td>TOTAL</td>
<td>56</td>
<td>23</td>
</tr>
<tr>
<td>2.1</td>
<td>CA1 - CA10</td>
<td>None</td>
</tr>
<tr>
<td>2.2</td>
<td>CA11 - CA15</td>
<td>None</td>
</tr>
<tr>
<td>2.3</td>
<td>CA16 - CA23</td>
<td>None</td>
</tr>
<tr>
<td>TOTAL</td>
<td>23</td>
<td>None planned until later release</td>
</tr>
<tr>
<td>3.1</td>
<td>ID1 - ID7</td>
<td>ID3,7</td>
</tr>
<tr>
<td>3.2.1</td>
<td>ID8 - ID13</td>
<td>ID9,12</td>
</tr>
<tr>
<td>3.2.2</td>
<td>ID14</td>
<td>None</td>
</tr>
<tr>
<td>3.2.3</td>
<td>ID16 - ID19</td>
<td>ID17</td>
</tr>
<tr>
<td>3.2.4</td>
<td>ID20 - ID22</td>
<td>ID21,22</td>
</tr>
<tr>
<td>3.2.5</td>
<td>ID23</td>
<td>None</td>
</tr>
<tr>
<td>3.2.6</td>
<td>ID24</td>
<td>None</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>3.2.7</td>
<td>ID25</td>
<td>ID25</td>
</tr>
<tr>
<td>3.3</td>
<td>ID26 - ID35</td>
<td>ID26,27,33</td>
</tr>
<tr>
<td>Total</td>
<td>35</td>
<td>11</td>
</tr>
<tr>
<td>4.1</td>
<td>IN1 - IN3</td>
<td>IN3</td>
</tr>
<tr>
<td>4.2</td>
<td>IN4 - IN13</td>
<td>IN6,7,8,9,10,12</td>
</tr>
<tr>
<td>4.3.1</td>
<td>IN14 - IN17</td>
<td>IN15,17</td>
</tr>
<tr>
<td>4.3.2</td>
<td>IN18 - IN26</td>
<td>IN19,20,23,24,25</td>
</tr>
<tr>
<td>4.3.3</td>
<td>IN27 - IN40</td>
<td>IN27,30,32,34,37,39</td>
</tr>
<tr>
<td>4.4</td>
<td>IN41 - IN50</td>
<td>IN50</td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td>21</td>
</tr>
<tr>
<td>5.1.1</td>
<td>SV1 - SV2</td>
<td>SV1,2</td>
</tr>
<tr>
<td>5.1.2</td>
<td>SV3 - SV6</td>
<td>None</td>
</tr>
<tr>
<td>5.1.3</td>
<td>SV7 - SV11</td>
<td>None</td>
</tr>
<tr>
<td>5.1.4</td>
<td>SV12 - SV13</td>
<td>SV12,13</td>
</tr>
<tr>
<td>5.2</td>
<td>SV14 - SV23</td>
<td>None</td>
</tr>
<tr>
<td>Total</td>
<td>23</td>
<td>4</td>
</tr>
<tr>
<td>Overall Total</td>
<td>187</td>
<td>59</td>
</tr>
</tbody>
</table>

This formulation indicates how steps 1 and 2 of the heuristic technique will look in practice. Steps 3, 4, and 5 require further discussion with the decision makers concerned. The formulation in this paper allowed only one technique to fulfill each function. This condition is not crucial to the usefulness of the model and can be relaxed. The complexity of the search may be increased, however.

The prerequisite conditions follow from Stepczyk's hierarchical concept (7). Figure 3 of Stepczyk's paper indicates the prerequisite structures: Integrity is most basic (Level 0). Prevention is next (Level 1); below this are Controlled Access, Isolation, and Identification (all Level 2). These three have specific interrelations for each technique (Level 3). Below Detection (Level 1) the next level is Surveillance (Level 2).

Although still large (59), the actual set of techniques available is substantially less than the maximum possible, using
Stepczyk's classification (187). Attributing costs and precisely examining synergistic effects still require a good deal of effort, but the basic structure for evaluation of a particular system has been established. Comparing VS2 Release 2 with another system, e.g., installed version of IBM Virtual Machine Facility/370 -- VM/370 -- would now be very easy.

CONCLUSION

Although the heuristic procedure can become laborious and cumbersome, it helps the decision maker to formulate the trade-offs to be made between functions, and helps him to seek cunning solutions (i.e., better control for less money). Arriving at an optimal solution after making arbitrary trade-offs and cost allocations is far less desirable than confronting the user with this decision procedure.

We do not have to contend with utility functions or transitivity problems. We avoid the 'impossibility' problem (see K. J. Arrow (1963) [1]) of group decision making. We merely give to the DP Manager and his superiors a basis for formulating the trade-offs that will lead to their decision. If the situation changes, e.g., CPU capacity becomes a constraint, the trade-offs may be made differently, and a new solution selected without difficulty from the same set of information.
REFERENCES


APPENDIX: THE LOGICAL NATURE OF SECURITY

**Proving Security**

To establish that a system is secure it is necessary to show two things:

1. That the system performs as advertised. (This is the same as verifying that it meets specifications.)
2. That the system does nothing that it is not supposed to do.

To prove 1, we can use transaction simulation, sampling, test programs, even continuous monitoring, and all the standard techniques of auditing. To prove 2 has been likened to proving the Universe, or rather the Universe minus N, where N is the set of things that are supposed to be done. In fact, this turns out not to be the most fruitful way to look at the problem.

A more helpful way to approach it might be as follows. Given a starting point within N, it is necessary to transfer from N to another set of transactions. The point of transfer requires an interface. If such an interface exists, the very minimum protection would be to have a monitor. Conversely, if N is perfectly separated, there could theoretically be no interface. In practice, even where N is running in a dedicated machine, there is at least one interface with the operating system. The damage that can be done in this situation is either to N, (and, therefore, does not constitute a breach of security) or to the operating system, via the interface. To operate any program, in practice, it is necessary to abandon complete isolation.

**CONSEQUENCES**

Perfect integrity (which implies that there is no way Form or Content can be modified in an unauthorized fashion) thus becomes an ideal. The monitoring or access control mechanism at the interface(s) may be logically perfect (i.e., in software design), but the possibility of hardware failure always exists. (This is a reliability problem.)

From this we can state the following:

1. Complete separation of subsystems from the central system(s) is impossible.
2. Interfaces can and must be monitored to provide the desired access control.
3. Even a logically perfect system with maximum integrity and logically secure access control is subject to failure (reliability is never 100 percent).
We can thus formulate the following probability of having security:

\[
P = f(P_{\text{security}} \times P_{\text{integrity}} \times P_{\text{access control}} \times P_{\text{reliability}})
\]

This can never be 1.0.

LOGICAL CERTIFICATION

From the preceding arguments it follows that for maximum integrity and logically secure access control (e.g., Dartmouth Time Sharing System), certain \textit{LOGICAL} conditions must be met.

To define these it is useful to adopt the Black Box philosophy from Cybernetics. A given input is to produce certain predefined outputs. (The process by which the output is achieved is not investigated directly.) The law of requisite variety requires that an input disturbance not be passed on through the system. To ensure this, it becomes obvious that the logical point at which to monitor disturbances is either at the input or the output interface of each subsystem. The size of the subsystem can vary from a module of a structured program to a copy of an operating system for a virtual machine user. The interface control is required to ensure that the errors passed on meet certain criteria for acceptance. (Theoretically, different criteria could be established for different interfaces, but the criteria to protect the central system must be the union of all requirements, as all users may be affected by modifications here.)

Two things are logically necessary to have a certifiable system:

1. \textit{All interfaces must be practically identifiable.} For example, an attempt to replace GETMAIN in OS/360 failed because it was not possible to identify all the logical consequences of the change. The obvious extensions of this concept are Modular Programming and Structured Programming. For Modular Programming, the interface is defined. For Structured Programming, it is \textit{practical} to examine the complete set of interconnections for any one module. (2) \textit{The impact points of all outputs from a given subsystem must be practically as well as logically identifiable.} It should be possible analytically to identify the impact of a system change before it is made. Heuristic approaches should only be necessary as a final check. Structured programming is an effective way to ensure the practicality of specifying impact points. No system change whose impact is unknown should be implemented in a Production Environment.

\textbf{Summary List of Conditions that Help Prevent the Violation of the Logical Nature of Security}

1. Integrity implies separation. Users should be separated to the maximum extent possible, e.g., separate address space and virtual machine.

2. The access control mechanism must be logically impossible to bypass; i.e., the processing should be dependent on the access control authorization, e.g., Dartmouth Time Sharing System address pointers and bounds register checking on System/370.

3. Every access must be subject to the checking procedure, i.e., not just initial file opening. This is critical for item 2.
4. There must be no logical way to acquire a higher level of priority than one started with. (See (a) the method by which the Dartmouth Time Sharing System was breached, (b) recursive feature of the MULTICS system at M.I.T. to restrict priority to the lowest -- most constrained -- priority of a series of calling routines.) It is important to verify that the most restricted access control applies to all subsequent processing.

5. Where shared facilities (Schroeder (1972) (5)) exist, the monitor should be located as close as possible to the item/location to be protected. This is not only in accordance with controlling the maximum number of potential disturbances with the fewest control mechanisms, it is likely to minimize the probability of bypassing the control point. (Do not locate your police roadblocks along the major highways if you want to protect the President.) Controls located to prevent problems whose context indicates their severity will need to be used where the probability of successful trapping at the last point is too small, e.g., transfer of rounding errors to programmers paycheck. Usually, the closest control will be adequate if correctly designed, for example, trap bit in I.S.C. system (6) to prevent unauthorized operations following an interrupt.

6. Where non-shared facilities exist, it must be logically impossible to step outside authorized bounds, i.e., to exit from a module or process, a standard interface must be crossed. Unauthorized creation of an interface occurs if unmonitored exit is possible. Trap-door entry exploits an unmonitored interface. Supervisor state has to be entered by an uncontrolled procedure or instruction.

7. Effective monitoring requires that the monitor have a higher level of privilege that can be acquired by any non-owner of the item or location being monitored (i.e., hierarchical structure). At the program or user level, this means that only the owner can modify the control procedure. (See (a) Graham and Denning (3); and (b) Dartmouth Time Sharing System procedure for systems programmers to implement system changes.) Fundamental problems exist with a system where for some purpose the supervisor is merely "first among equals," as in OS. Thus, the operating system must be logically at a higher level than everything else. The two ways to achieve this are: by system protection and by user constraint. The Resource Management System of VS2 Release 2 puts the system in overall control; similarly, a virtual machine separates the user. Both accomplish or partially accomplish the levels of separation. (See Madnick and Donovan, (1974) (4). This book describes hierarchical criteria for operating system design.)

8. Unidirectional access is necessary; i.e., a program can use data or programs with a higher level of priority than itself, but cannot acquire that level (e.g., System Catalog is modifiable by a user program, but the user cannot acquire the authorization of the catalog - Dartmouth Time Sharing System.)

9. Although masquerading as another user, e.g., by erroneous physical or password identification, cannot always be prevented, internal system masquerading can be made logically impossible. Entering supervisor state is a special case of masquerading. Items 2, 4 and 6 are critical to preventing internal masquerading.
CONCLUSION

It is hoped that the preceding conditions will provide a basis for determining whether a system is logically secure. Even logically insecure systems can have PROTECTION, DETECTION, and RECOVERY mechanisms, but they are, in principle, UNCERTIFIABLE. The possibility of undetected modification always exists so that even 'brute force' checking of every instruction or simulation may be useless. To make the logical security requirements economically feasible, several system features can be exploited. Examination of these and of the preceding logical requirements establish that CERTIFIABLE SECURITY can only be an integral part of the system design, never an optional add-on feature.

IMPLICATIONS FOR VS2 RELEASE 2

The following checklist indicates the areas in which VS2 Release 2, as described so far, meets the logical requirements listed in the preceding section. The nine categories correspond to those listed previously.

<table>
<thead>
<tr>
<th>CHARACTERISTIC</th>
<th>MET</th>
<th>VS2 RELEASE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum separation</td>
<td>Yes</td>
<td>Separate address space</td>
</tr>
<tr>
<td>No Bypass of Access-control mechanism</td>
<td>Probably?</td>
<td>Not yet defined</td>
</tr>
<tr>
<td>Every access checked</td>
<td>?</td>
<td>Not clear if this could be easily incorporated</td>
</tr>
<tr>
<td>No logical way to acquire higher priority</td>
<td>Partial?</td>
<td>System Catalog</td>
</tr>
<tr>
<td>Monitor close to item to be protected</td>
<td>?</td>
<td>Possible</td>
</tr>
<tr>
<td>Not stepping outside bounds</td>
<td>Yes</td>
<td>Separate address space</td>
</tr>
<tr>
<td>Higher level monitor</td>
<td>Could be</td>
<td>System mgmt capability</td>
</tr>
<tr>
<td>Unidirectional access</td>
<td>Yes</td>
<td>System catalog</td>
</tr>
<tr>
<td>Masquerading</td>
<td>?</td>
<td>Could be prevented</td>
</tr>
</tbody>
</table>

Massachusetts Institute of Technology Volume 4 221