7-1982

Computer Abuse -- Controld and Cases

Robert J. Thibedeau

COMPUTER ABUSE--CONTROLS AND CASES

by

Robert J. Thibedeau

Bachelor of Science, St. John's University 1967

An Independent Study

Submitted to the Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota

July
1982

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Computer Growth

In the opinion of the author, the growth of computers in all segments of society has been unparalleled by any other phenomenon in history. New technology has produced both larger and smaller systems which have increased the computer's usefulness and numbers. Not only is their growth unparalleled but so also is the dependence upon their capabilities. With every great technological revolution there is a movement from one capability to another which is perceived to be preferable. Some examples are steam to electricity, horse to automobile, steel to specialty metals and plastics. However, in previous history the move was relatively slow and the prior capability was retained at least long enough to insure the adequacy of the new capability, if not retained as a permanent backup. This has not been the case with the growth of computers. Originally a return to simpler automation or manual procedures may have been possible. Today it is not. With the introduction of transistors and integrated circuits, computer capability and utilization exploded beyond the capabilities of any backup system. In the words of Donn B. Parker, Senior Information Processing Specialist, Stanford Research

Institute,

> All but the smallest business and government agency own, lease or use computer services. Most large organizations are discovering that they can function for only a few hours or a few days at most without the correct functioning of their computers.

In 1973,

> at least 60% of all banks are automated and would be unable to function unless their demand deposit accounts were successfully processed on computers.[1]

This explosion has greatly contributed to the capacity and capability of our society but it has also given rise to new facets of an old problem--control and security. Control and security problems are not new, of course, but the centralization of such large amounts of data is. There are other new facets of the problem also.[2] Machines do not make human errors and being incapable of inductive "thought", they remove many of the human checks performed during processing. (Appendix B). Electronic Data Processing (EDP) facilities are populated by individuals with various backgrounds and skills with little knowledge of accounting control and no professional code of conduct to guide their actions.[3] EDP combines many procedures and obscures the audit trail of manual processing. The sheer volume of transactions makes control difficult. Often many jobs are run concurrently, each one involving hundreds of transactions. Data can be changed without leaving telltale erasures or other evidence. Computer usage has resulted in expansion and increased use of data transmission

capabilities. Lastly, many accounting personnel are not skilled in the use of computers and can be overwhelmed by their imagined complexity. (Appendix C).

Purpose of Independent Study

The purpose of this section is to discuss the control procedures which have been designed to prevent or detect intentional circumvention of system controls and breaches of security called computer abuse. Two control categories will be considered--those specifically designed to present or detect computer abuse and those general controls which protect against computer abuse but not as a primary function.

Donn B. Parker defines computer abuse as

> all types of acts distinctly associated with computers and data communications in which victims involuntarily suffer or could have suffered losses, injuries or damage, or in which perpetrators receive or could have received gains.[4]

This study is organized into the three phases of systems development; design, implementation, and follow-up; as a convenience to organize thoughts. Much more important than this organization, however, are the two underlying points to be emphasized. First, security against computer abuse must be a fundamental part of computers from design through audit, involving coordination of manufacturers, EDP personnel and accountants/auditors. If this is not the case, the weakest part of the system can be used to compromise stronger parts.[5] Second, no system is or can be 100 percent safe and still be

usable and cost effective. Complete protection requires that the costs to invade a system exceed the value of assets being protected. This raises the question as to the value of assets to an irrational perpetrator. The most secure systems attack the problem on two fronts, first by limiting unauthorized access and second by making any abuse as difficult as possible so that only a unique few have both opportunity and capability. "Possible" is defined by determining the value of a potential loss of assets and then expending assets and resources in sufficiently cost effective ways to protect against already known and theorized types of attack.[6] This discussion will demonstrate that security procedures are available to prevent computer abuse, that these procedures are efficient and cost effective, and that the computer itself is the greatest aid in maintaining a secure system.

---

[1]Donn B. Parker, Susan B. Nycum and S. Stephen Oura, Computer Abuse (Menlo Park, California: Stanford Research Institute, 1973), p. 17.

[2]John M. Horne, "EDP Controls to Check Fraud," Management Accounting 56 (October 1974): 43-46.

[3]Parker, Computer Abuse, p. 17.

[4]Parker, Computer Abuse, p. 5.

[5]Harold Weiss, "Computer Security An Overview," Datamation 20 (January 1974): 46.

[6]Computer Security Research Group, Douglas B. Hoyt, Chairman, Computer Security Handbook (New York: Macmillan Information, 1974), Chapter 1.

CHAPTER II

DESIGNING THE SYSTEM

Introduction

Development of any system involves two basic elements of the EDP system--hardware and software.[7] It is important for the understanding of design controls that the distinction between these two systems elements be understood; however, controls must be designed for an entire system rather than for individual elements. Coordination of separately designed elements is inefficient and sometimes very expensive.[8] It should also be noted the distinction between design and implementation controls is often fuzzy. For the discussion that follows, design involves a capability while implementation is concerned with the application and utilization.

The Computer as a Control Element

The computer itself is one element of control. Input and output from the central processing unit (CPU) can only occur through hardware connected to the CPU. This can be through many different types of equipment including the common telephone but the equipment must be connected to the CPU to have access to it. This concept is simple and basic and not unique to computers but is a major difference between an EDP system and a manual system. A manual system can be accessed with nothing more than opportunity and a pen, and

read with nothing more than eyes.

Combined Hardware and Software Design Controls

Most design controls involve a combination of equipment design and operating system software design. A supervisor or monitor program that controls all the interactions between hardware and software elements is available which will allow only necessary and approved access to the computer's input and output.[9] Two very simple but common examples of this control are one way access or limited access. Examples of one way access would be equipment which has been limited to inputing data only or outputing data only. For many abuses both accesses are required. Examples of limited access include program or equipment blocks which allow access to only certain machine storage areas, programs, files or processes. This control is discussed more fully in conjunction with systems implementation but a simple example will help illustrate the concept. Most systems control access through an identification system which is one way only. An individual with the proper identification code can access the computer however the computer is programmed and the identification system stored in such a manner that codes cannot be transmitted from the CPU to an output terminal.[10] Thus the integrity of the codes can be protected. This feature can be strengthened by the addition of an alarm system when unauthorized access is requested.

Equipment and supervisor program controls can also be designed to control access based on the identification of the user and the location of the equipment. (Appendix R). Thus data release and data input requests can be associated with an almost unlimited combination of equipment type, location and user identification.[11] For example, the same piece of equipment may be allowed to input or output data from different programs or areas of the CPU depending upon the identification provided by the user.

Access controls are strengthened by two other controls which can be designed into the supervisor or monitor program. First, a storage area or remote terminal can be established which will keep a record of accesses to the computer.[12] The amount and type of information recorded is a function of systems implementation, however the capability is a function of systems design. Second, an alarm or shut down capability called a program check interrupt can be designed into the system to be triggered by unauthorized requests for access or an improper use of instructions or data. The hardware response to the program check interrupt is called trapping. When an exception is detected, an unconditional branch is taken to a predetermined location where control is transferred to the supervisor or monitor program for appropriate action.[13]

Design controls have also been developed for individual storage locations inside the CPU. An overflow capability will allow only data falling within certain limits to be

stored. Two examples would be restrictions as to numerical size or restrictions as to character type (alphabetic or numeric).[14] Cryptographic encoding capabilities have been developed to protect sensitive data from compromise should access controls fail. When unauthorized access to a storage location has been accomplished, the data may be useless because of necessary decoding. This control has also helped protect data during transmission from one location to another--a particularly vulnerable time.[15]

---

[7]Hardware: Physical equipment and devices comprising an EDP system. Software: Programs and routines, (including assemblers and compilers, utility routine, operating systems and application programs), usually furnished by a computer manufacturer, to facilitate the operation of a computer.

[8]Jack F. Thorne, "Control of Computer Abuses," The Journal of Accountancy 138 (October 1974): 49-50.

[9]Computer Security Research Group, Computer Security Handbook, p. 76.

[10]Jacob Palme, "Software Security," Datamation (January 1974): 51.

[11]Donald L. Adams, "Alternatives to Computer Audit Software," The Journal of Accountancy 140 (November 1976): 56.

[12]Christopher Podgus, "Outwitting the Computer Swindler," Computer Decisions 5 (September 1973): 16.

[13]Computer Security Research Group, Computer Security Handbook, p. 49.

[14]Ibid.

[15]Ibid., p. 57. Palme, "Software Security," p. 55.

CHAPTER III

IMPLEMENTING THE SYSTEM

Introduction

There are three main categories of damage which are of concern in this section:

1. Illegal access to data
2. Illegal modification, addition or destruction of data.
3. Interference with the ordered working of a computer.[16]

A fourth category, physical destruction, will not be included since the same methods used to guard any valuable property also apply to a computer: locks, alarms, personnel control, etc.

In the previous section, the point was made that computer system security and control must be a coordinated part of designs, implementation, and follow-up. This requirement also holds for the implementation phase itself. A safe system cannot be created by applying selected protective measures against those risks which carry the greatest potential for damage and loss. If there is one unprotected access route or area in a computer, then these can often be used to circumvent other protective measures.[17] Protective measures and controls must be thought of as a series of controls in order to evaluate their total effect. A more practical consideration of this concept is that, besides being more

effective, it is far cheaper and efficient to design a total system than it is to try to bring cohesion and coordination to a disjointed system. (Appendix B).

In addition to this general principle, several more specific principles have been developed with regard to the security of EDP systems.

1. Assume the potential attackers will know as much about the system and its security features as its designers.
2. Provide the least amount of privilege to a process within the system and users of the system to accomplish adequately their authorized purposes.
3. Minimize the penetration and subversion of other processes when one process is penetrated or subverted.
4. Provide detection mechanisms for all anticipated penetration and subversion methods and produce a reporting facility for monitoring and auditing a secure computer system.
5. Isolate the security processes from the system with formal interfaces to the system.
6. Structure security processes to allow complete auditing for integrity.
7. Authorization should be based on permission to access as opposed to one which is based on exclusion from access.
8. Security restrictions must be acceptable to personnel and uniformly enforced.[18]

Personnel Controls

The first step in implementing a secure EDP system is choosing honest, qualified persons to staff it. This is the logical starting point because the basic security of an EDP system is no better than the integrity of the employees. Computer employees must be educationally qualified, stable, and dependable. Reference checks should be made in an attempt to relate an applicant's past to his job potential in

these areas. Even after hiring an individual to work in the EDP system, personnel checks should continue. Such checks should be alert for borrowing, gambling, drinking, unpaid bills, questionable associates, extravagance, refusal of vacations and refusal of promotions.[19] (Appendix T).

Ongoing operations. Since any screening process is at best subjective, it is necessary to augment it with two other controls. First, all employees in sensitive positions should be bonded. Second, no employee should have exclusive and permanent contact with a sensitive function. This is accomplished through mandatory vacations, requiring dual control or a second verification of sensitive functions, especially where separation is weak, and by rotating sensitive duties.[20] (Appendix J).

A secure personnel system must not stop with hiring and assignment of duties; it must be part of daily operations. Controls are normally not publicized lest the knowledge of their existence be taken as a challenge or a key to abuse. Personnel security controls are an exception. Employees should be continually made aware that they are relied upon to preserve the integrity of the computer system. This can be done in a positive way so as not to imply a lack of trust and can be an integral part of daily operations and staff training. Each employee must understand the reasons and objectives of the system and his or her role and responsibility. Management must continuously show by its actions the importance

of security by monitoring observance and citing violations. If the proper attitude and climate exist, security procedures can be instituted which will be followed.[21] (Appendix N).

Another necessary personnel control is to control access to the computer itself. EDP personnel should be identified; if necessary, badges should be issued. Entrances and exits to the computer area should be controlled. Unauthorized personnel should be denied access, unidentified personnel should be challenged, and outside individuals, such as auditors, engineers and maintenance technicians must be escorted and their actions controlled. Along with this, discharged employees must be immediately escorted from the facility and denied readmittance.[22] Wages paid in lieu of a discharge notice is a small price to pay to avoid a large theft. (Appendix S).

Separation of duties. Another area of personnel security involves the separation of duties within the computer facility. In manual systems this principle, while not always achieved, is simply stated: the duties of authorization, recording and controlling must be separated. In automated systems, these functions may be performed in the computer. As a result, the above rule restated would be: the duties of programming, operating and controlling must be separated.[23]

Programming procedures have to do with the development of source programs, the detailed written instructions to

the computer system. Programming procedures must be specific and standards of documentation must be detailed. The required documentation should be prepared as the program is developed and tested, not after the fact. Since only in very rare instances is observation of the computer actually running the program an aid to program development, programmers should submit their program for acceptance testing to others. Such testing should be the duty of internal auditors who observe the actual running of the program by the computer and evaluate the results. Once accepted, the program and documentation should be safeguarded and not returned to the programmer.[24] (Appendix E). A typical list of program documentation would include:

1. Block diagram system flow
2. Summary description of system
3. Layout of program hard copy to the program
4. Layout of program hard copy output
5. Description of program logic or logic chart
6. List of source program and diagnostics
7. Copy of reports after test
8. Operator instructions with error corrections and halt routines
9. Keypunch or typing instructions for input
10. List of operations control deck
11. Sample and layout for data cards and control cards
12. Estimates for run time based on volume.[25]

Computer operations has to do with the daily hands on contact with the computer. It is often stated operators should not be able to program but this is an unrealistic requirement since it cannot be monitored, inhibits promotion and does not do much for morale. Operators should,

however, have access to only what is needed to perform an operation. Program documentation includes many items not needed by operators which would be helpful to embezzlers, like flow charts and other information on program logic. Operators need only the instructions to run the program and to deal with operation interrupts. There should also be detailed procedures covering the submission, running, and return of all programs and data. Incorporated into these procedures should be specific standards for data preparation and program and file storage. These functions should be independent of operations personnel. A good system will have a library, independent of operations personnel, and a librarian charged with controlling data files and programs. All programs and data should be logged out only to specific personnel and returned to the library immediately after use. Complete and accurate records should be maintained which show who has used the data or programs, for what and when.[26] Data preparation should be the responsibility of the user. Efficiency may dictate a centralized key punch organization. In any case, users should be required to prepare data for input. This allows output checks through batch totals, batch counts and various other means to be performed by data users.[27] (Appendix F).

Operations personnel should be organized to allow supervision of all computer operations. Particularly sensitive operations should be controlled by more than one

person. (Appendixes M and O). Overall operations of the facility should be under the control of one person.[28] This individual, besides providing overall control of operations personnel and monitoring their activities, prepares production schedules which are used to monitor the use of the EDP facility. These production schedules include run authorization, time estimates, data file and program library release memos, data preparation instructions, output routing, and input and output checking guides. Logs of actual computer usage should also be maintained by this individual which reflects the status of the computer at all times. The foregoing has stressed the need for segregating the responsibilities in the EDP facility. This separation would accomplish little if operators are allowed in the computer room after hours and on weekends unless normal controls are in effect.[29]

The third data processing function which must be separated is control. This function has to do with overall monitoring of the system. Ideally this is performed by internal audit personnel who are independent of computer facility managers. They must have sufficient technical competence to test programs for validity and accuracy and to monitor system operations. Procedures to perform these functions should be detailed, especially those dealing with acceptance testing of new programs and program changes. Verification of new programs and changes should involve exhaustive testing since the damage caused by programs

erroneously accepted can be devastating. Other functions performed by the control staff include accounting for computer time usage, data verification, checks of operating programs, and examination of audit trails. Some of the tools and methods to be used will be discussed more fully in the follow-up phase but one item should be noted here. Internal control personnel must not only be independent of operations managers, they must also have the authority to perform any authorized check without prior notice or coordination.[30]

Identification Controls

As mentioned earlier, the computer has the capability to control access when a potential user is identified to the system. This section is concerned with the methods which can be employed to identify a user. Two types of keys or codes are most commonly used to identify persons calling the computer, memorized keys and physical keys. Memorized keys are groups of numeric, alpha numeric or alphabetic characters which are unique to an individual or a group of individuals. Individual keys are preferable to group keys since they are easier to control, more flexible and allow more detailed access records to be maintained. Simple logic is the best way to choose a safe key. The number of combinations should be large, keys should be assigned instead of chosen by the users, keys should be selected randomly, and

keys should be easy to memorize. Key check routines should be stored in an area that allows comparison checks only and under no circumstances will output a key. Key storage should be strictly controlled by internal audit personnel. The physical key is an identity card or slip which can be read at a terminal. The key can be punched into a card, on a magnetized slip or embossed on a plastic card. Such keys are more convenient to frequent users and can't be stolen merely by watching it being entered. As an alternative to entering keys, a user may input his name under supervision or have sole use of a terminal which is kept locked.[31]

Remote terminals and telephone terminals have created a new identification problem. In some cases, it is important to know not only who is calling but from where the call is coming. Where such a need exists, computer keys should be augmented by the addition of a terminal identification process. This can be accomplished in two ways. First, the terminal itself can transmit a secret key along with the user key. The terminal key should be locked in the terminal and unknown to the user. The second method involves a call back verification. The user calling the computer identifies his location and identity, the computer then calls the terminal and asks it if the user is using the identified terminal. A response is required before the user is allowed access to the computer. Since terminal usage should be documented, the risk of disclosure would be a strong deterrent. (Appendix R).

Access Controls

The obvious intent of computer users in identifying themselves to the computer is to gain access to certain data and programs of the computer. The computer, in effect, grants permission to the user based on an authorization table which has been inserted into the computer's memory. One important rule is that permission should always be permissive as opposed to exclusive. Each different access privilege and group of data should be associated with those who are authorized access, never with those who are to be denied access. There are two generally used types of authorization tables. The first and most general type is to associate every group of data with a list of the people authorized access to the data. The second type involves the organization of personnel and data into hierarchial structures. The arrangement would resemble a pyramid with generally accessible data close to the bottom. As authorization moves toward the top of the data and personnel structure, the data becomes more restricted and the persons authorized access is reduced. Each method has a distinct advantage and disadvantage. The general authorization is more flexible but less efficent. Access can be granted individually and in any desired combination, however, the search and match routine takes a long time by EDP standards. The hierarchial authorization sacrifices flexibility but gains efficiency. Access granted to any individual requires like access be given all those

above in the structure, however the search and match routine is quick by EDP standards.[32]

Once access is authorized, control can be refined further by defining access in terms of the privileges to be granted an individual user. There are six generally listed rights which can be differentiated by the computer and matched by user or data or a combination of both.

1. The right to read a group of data
2. The right to add to a group of data
3. The right to change existing data
4. The right to delete from a group of data
5. The right to execute a program[33]
6. The right to change a program.

The distinction between the term data, used above, and files recognizes the full versatility designed into the computer. Access can be to fields, records, or files. This authorization capability is a very powerful security tool and one of the major compensators for the problems created by EDP. (Appendix P).

The authorization function can be contained in either the operating system of the computer or the user program. As a practical matter, file access is usually a function of the operating system which is more secure. However, because of the numerous data combinations possible with field and record access, user programs usually control access at those levels. To avoid compromise, user programs which control access must be executable only and, if possible, stored in the computer. Access tables would reference the user and the user program by name instead of the user, and the program

would have an additional access table identifying authorized users and the data and rights associated with them.[34]

Software manufacturers are a big help in this area. They have developed programs which will perform this access control function. The programs, like IBM's 750/V52 Programming Control Facility, are a form of access table which matches users with the data rights each is allowed. The programs allow definition of the authorized relationships between users' programs and files. Attempts to violate the defined relationship will be stopped and trigger an alarm. The IBM program offers an additional dimension which has special applications to time sharing facilities, that of restricting the use of operating commands to designated users.[35]

The purpose of an identification and authorization system, of course, is to deny access to unauthorized intruders. As mentioned earlier, it must be assumed the potential intruder knows the security system as well as its designers; therefore, to make even the best planned system as safe as possible from a knowledgeable intruder, monitoring controls must be connected to the access controls.[36] First, all access should be recorded at a remote terminal. (Remote terminals will be discussed with follow-up controls.) Second, "errors" in identification or requests for access should trigger an alarm since they may, in reality, be unauthorized access by trial and error. The alarm (hardware check interrupt) can be silent--at a remote terminal--or a system shut down which necessitates inspection before restart. Some systems allow

one error, however, a second error triggers the alarm.[37] The hardware check interrupt was explained as a design feature of present computers.

Sensitive Data

The problem. Protection of secret or sensitive data, more specifically, limiting access to it, involves additional controls beyond the ones explained above. But first a digression is necessary to appreciate the unique nature of the problems involved. No one would argue with the statement that programs can be a valuable asset. Indeed a unique program can be the single most valuable asset of a firm. If uniqueness is lost, a definite financial loss has indeed resulted. Unfortunately, current law defines theft in terms of depriving another of the use of property. When a program is taken by copying, the deprivation is absent. New laws are being enacted which better describe theft of computer software but the problem still exists in many areas and as a result, the courts have tried to issue guidelines to shore up existing law where it is deficient. Most commonly, courts will recognize theft under existing law if the data is identified as to owner and restrictions, and is protected by an identification and authorization system. This test itself is a mixed blessing. Clearly the greatest priority is prevention of compromise--not criminal punishment, and identification of sensitive data is a red flag which marks it as valuable.

There is no one answer and each facility must analyze the situation based on the pros and cons of data identification. The same considerations are involved in application of cryptography controls and storage controls which attract attention to the data.[38] (Appendix K).

Controls. Regardless of the decision to identify or encode sensitive data, it must be segregated and access authorization should be strictly controlled. Access authorization has been previously discussed and the same principles apply to sensitive data. Segregation involves a distinct storage location with machine blocks to ensure access only via proper identification through the operating system. This segregation allows accumulation of a log indicating all access and attempted access to sensitive data and denies access to the routine user. Segregation of sensitive data also involves input, processing, and output phases. Input and processing should be performed only by authorized personnel. Unauthorized personnel should not even be in the computer room during processing, and hardware stops should be inserted to disconnect all remote terminals except the internal auditor's terminal. When sensitive data is output, the same procedures should be in effect. In addition, distribution must be rigidly controlled and all scrap should be treated as sensitive, including items like printer ribbons, paper and cards.

Cryptography can also be used to protect sensitive data

but since coding routines and processing of uncoded data must be protected by other means, cryptography without segregation seldom makes the data secure. Coding, however, can be valuable if there is a weak point in an otherwise safe system, such as during data transmission.[39] There are various coding systems available; however, their description is beyond the scope of this study.

Defining sensitive data. The definition of sensitive data should be a function of threat monitoring with the controls applied depending upon the protection desired. Too narrow a definition may provide opportunities for abuse which are not easily detected and, even if discovered, once the data has been compromised, the reason for its protection may be lost. Certainly this is the case where unique programs have been copied or secret data is no longer secret. Recovery in these cases is seldom complete and the laws in this area are inadequate to insure compensation or punishment. Sensitive data should also be considered in light of its potential abuse. (Appendix T).

User Programs, Operating Systems, and Memory Access. The heart of any EDP system is the operating program (or programs) which actually commands the computer's operation. It is the link or gate between the user program and computer operations and primary memory. Because of this sensitive function, it must be very securely guarded. The documentation of its logic and all copies of the program

must be closely guarded. Of the "six rights" only the right to execute should be allowed. Hardware controls should be established which prevent the dumping of the operating program or additional writing in the program.[40]

In spite of thorough testing and protection of the operating program, it remains a weak link in an EDP system. It is accessed so often and in so many different ways that it is impossible to anticipate all the possible compromises. In addition, since access is the primary obstacle to abuse, the almost universal access to the operating program by users and the universal access to memory and the CPU of the operating program means the very access, which is necessary for correct utilization of the system, makes it the most vulnerable. Given access to an important program, the intelligent programmer can neutralize any controls inside the program. This conflict between needed access and necessary security can be resolved by dividing the operating program into subprograms which are all protected from each other in the same ways the system is protected from unauthorized users. Thus, access can be gained to the operating program, access which all authorized users must be granted, without causing the internal operation of the computer to be vulnerable. After gaining access, the user must still ask permission to enter a subprogram and each subprogram can have a separate authroization table matching users and the rights granted each.[41]

Separation of data. After accessing the operating program

the user normally requests access to primary memory to perform authorized functions. The subprogram of the operating program insures only the proper memory is entered; however, if there is no separation of memory, the skilled programmer or operator will have little trouble using this authorized access to enter any area of memory. An analogy would be to lock the doors while opening all the windows. This can be prevented by use of a hardware control mechanism which divides the memory into sections. Even with subprograms and sectioned memory, a system can be vulnerable if more than one program has access to a common memory area. Usually such areas must be write protected or a user may try to enter through one subprogram and return to the other. Once in a forbidden subprogram he is free to access memory areas of the second subprogram and the process goes on. One general rule suggested by this is that sensitive memory and subprograms should not be directly or indirectly linked to other subprograms or memory areas.[42] This discussion of primary memory is also applicable to secondary or external memory. The procedures may vary since a central file handling system, a part of the operating program, is usually used to access external storage using authorization procedures previously discussed.[43]

Separation of functions. Besides the multi-module subprogram concept, operating systems can be protected via utility programs or cusps. The objective is to move as many operating system tasks as possible into these special user

programs. This gives better security but only if these utility programs are fully protected from the user which means they must be coded "execute only." The operator should not be able to even read or dump the program, let alone write in it or change it. Reading and writing would be performed by a tightly secured program--usually controlled by internal auditors. Thus a hardware memory check must be able to differentiate between the right to read, the right to execute, and the right to write in a certain memory area.

A variation on the utility program method involves compilers, interpreters and file handling programs which are placed between the user program and the operating program. The discussion of utility programs is applicable to these; however, the compiler warrants additional comment. The compiler can also be used to check for illegal program entries. First of all, a compiler should be designed for the facility that not only translates programs but also checks for illegal instructions or conditions, such as an array size which is too large. Programmers must be required to use only one programming system or language and the entire program should be compiled at one time so that all modules can be checked at one time and against each other. This same time compiling requirement is especially important when several programmers are working on modules of a single program. The compiler can be used to insure the separate modules only interface in permitted ways. It goes without saying that programmers should be prohibited

from setting switches which disconnect protective checks of the compiler.[44]

Auditability. In addition to the control requirements during interface with the operating program, a secure user program requires other considerations. Logic should insure auditability of the system. The program must be designed with this requirement in mind as opposed to designing an audit system after the fact. Closely related to this is the requirement that data checks be built into the program which are designed to detect processing or data errors. One method of achieving this is to use a series of programs to process data. Thus, while the data may be processed from start to finish at one time, the multiprogram aspect will produce subtotals and check figures which can be used to check processing. The same results can be obtained by using call routines in a main program which execute subprograms. Also user programs should not be designed to perform unrelated functions.[45] User programs should contain exception checks for data and totals. Such controls include check figures to verify processing results. Code testing should verify the data and its identification are within parameters consistent with files to be affected, consistent with other data in the transaction, and does not exceed limit tests of quantities or amounts. Checks should also be made for too much or incomplete data. Any exceptions noted should trip a branch to the supervisor program for follow-up action.[46]

Data Preparation

Data preparation is very vulnerable to abuse, especially since the data is often put into computer readable form by individuals other than those who collect the data. Since manipulation of data before processing can be virtually undetectable after processing, controls should be used which allow data verification. As previously mentioned, audit trails and module programming are useful; however, they are made more useful if the agency submitting the data also develops check figures, batch totals and document counts which can be used to verify data during and after processing. (Appendix G).

The major burden for the development of error free input data must fall upon the procedures used in the data preparation department and the checks described above. A programming philosophy should be developed, however, which emphasizes the audit function and encourages program logic which minimizes the likelihood of successful undetected criminal manipulation of the EDP system through the alteration of any single element, program or data file in the system.[47] (Appendixes D, H and I).

Data Storage

It is a well established rule that three generations of files should be retained, the grandfather, father and son concept. While the primary emphasis is on destruction of data and a back up capability, theft of data must be considered and

such a system is a protection against theft also. Data can be lost without theft or destruction however. If a system has been penetrated and the data is no longer reliable, a back up file, even though not current, can be very valuable protection against the sophisticated criminal where there is no trace of the abuse and discovery was through pure happenstance. Baseline data from which to work may be the key to unraveling the scheme.

Sensitive data should be given special considerations. Duplicate copies of sensitive data should be stored off site for the same reasons three generations of files are retained for the data. The level of security should be determined by the value of the data. Data is a valuable asset, sometimes worth literally millions of dollars, without which some businesses could fail. They should be treated as an asset and stored accordingly.

Last Lines of Defense

Even with the best of systems there is potential for abuse. The preceding discussion has described the methods for making abuse difficult and for denying access to the individuals intent on harming through the computer. The controls discussed will deter most people and catch most of the others. For those undaunted by the difficulty and sophisticated enough to gain access, there is a last line of defense called "mining" the system. Basically, mining is the setting of traps to catch the unauthorized user.[48] First of all, if security is good, the intruder probably has gained the knowledge needed to abuse

the EDP system by studying a similar system. This fact can be used to advantage if subtle program variations are installed and important data fields are changed. The operating program is normally sensitive to commands which are not precisely accurate and will show such inaccuracies easily. If things are not exactly as expected, this is doubly so. These changes can betray an intruder by causing a system to shut down when an input error occurs. Other mining techniques include dual entry of some data and periodic checks for equivalence, tagging data so it can be followed through a system or monitored on a terminal which will output every processing step involving the tagged data. Lastly, the introduction of random data errors can be used to protect sensitive data by making it inaccurate and unusable to outsiders.

---

[16]J. Walker Voris, "How the Computer Can Be Used to Commit Fraud," Practical Accountant 8 (March/April 1975): 63.

[17]Palme, "Software Security," p. 55.

[18]Parker, Computer Abuse, p. 23.

[19]Brandt R. Allen, "Computer Fraud," Financial Executive 39 (May 1971): 40.

[20]Thorne, "Control of Computer Abuses," p. 42.

[21]Ibid.

[22]Weiss, "Computer Security An Overview," p. 46.

[23]Thorne, "Control of Computer Abuses," p. 42.

[24]Ibid.

[25]Computer Security Research Group, Computer Security Handbook, p. 134.

[26]Ibid., p. 159.

[27]Computer Security Research Group, Computer Security Handbook, p. 77.

[28]Thorne, "Control of Computer Abuses," p. 44.

[29]Allen, "Computer Fraud," p. 44.

[30]Thorne, "Control of Computer Abuses," p. 44.

[31]Palme, "Software Security," p. 51.

[32]Ibid., p. 52.

[33]Thorne, "Control of Computer Abuses," p. 44.

[34]Palme, "Software Security," p. 52.

[35]Adams, "Alternatives to Computer Audit Software," p. 56.

[36]Parker, Computer Abuse, p. 23.

[37]Computer Security Research Group, Computer Security Handbook, p. 32.

[38]Gerald McKnight, Computer Crime (New York: Walker and Company, 1973): 170.

[39]Palme, "Software Security," p. 55.

[40]Ibid., p. 54.

[41]Ibid., p. 55.

[42]Computer Security Research Group, Computer Security Handbook, p. 37.

[43]Palme, "Software Security," p. 54.

[44]Ibid., p. 53.

[45]Thorne, "Control of Computer Abuses," p. 46.

[46]Computer Security Research Group, Computer Security Handbook, p. 77.

[47]G. Hunter Jones, "D. P. Error and Frand--and What You Can Do About It," Price Waterhouse & Co. Review 2 (1976): 10.

[48]Palme, "Software Security," p. 55.

[49]Horne,"EDP Controls to Check Fraud," p. 43.

CHAPTER IV

FOLLOW-UP

Auditors

Follow-up involves every effort to insure the system is operating as intended and the design meets the changing needs of the business enterprise. Any enterprise large enough to benefit from computerization of a major portion of its activities should be reviewed by internal and external auditors. The internal auditors design and/or review controls from the implementation phase through the follow-up phase. The external auditor is an independent accountant hired to examine the records and give an opinion as to the fairness of the financial statements. His review is not directed at uncovering abuse; however, his review of internal control is invaluable as an aid to discovering and correcting weaknesses, and a thorough audit increases the chances of uncovering an ongoing abuse. Probably equally as important, the mere existence of regular audits by outsiders will discourage most attempts to abuse the computer.

Before discussing the methodology of security follow-up, a brief editorial comment on the qualifications and attitude of auditors, both internal and external, is appropriate. Computer science is a very sophisticated field. A programming expert, if unmolested, can manipulate programs and data so skillfully that no trace can be found of the handiwork, and so

little will be amiss there is little chance of a tipoff leading to discovery of the thefts. For an auditor to arrogantly assume accounting skills will be sufficent to match wits with an equally skillful computer programmer is inviting disaster. Equally as dangerous however is to treat the computer as a secure black box with data in and data out but void of a significant processing stage in between. Computers are becoming an important element in the information system of even small companies. Internal auditors who are concerned with theft must constantly monitor the EDP system using persons with expertise in computer processing. External auditors who are concerned with a fair presentation of financial information must "develop sufficient evidential matter" to insure the information produced by the EDP system is accurate. For either group to accomplish their objective requires an evaluation of the EDP system and the controls in affect. It must be a thorough evaluation and it must be performed by persons who are knowledgeable in both computer processing and auditing. To do otherwise is to invite a second Equity Funding scandal where there were no internal auditors and the external auditors gave an opinion without even setting foot into the EDP center. (Appendix A). Newer computers are being designed with security systems built in and auditing tools are available which help to balance the auditor's skills with the computer programmer's skills. The auditor does not have to be a computer programming expert

because the tools are being developed by experts. The auditors, however, must understand computer operations, have basic programming skills, and experience, and most important--understand that the sophisticated computer thief is normally a genius at computer science and cannot be underestimated or ignored. With this preamble, the following discussion will expose some of the follow-up techniques available to the auditor. No attempt will be made to do more than identify and briefly describe the tools available. The purpose of this section is to identify the tools and their applications, their skillful use will require further study.

Using the Computer

One characteristic which makes a computer vulnerable is the fact it has no scruples. It is loyal to whomever has its attention and does not make judgements as to right or wrong, only decisions on correctness when compared to stored programs.[49] This same vulnerability is, however, the trait which makes the computer the most valuable single audit tool when performing audit procedures on an EDP function. It will give paragraph, chapter and verse of every abuse in process, or which has been perpetrated, if the examiners ask the correct questions. This is the heart of any follow-up system--knowing the correct question to ask the computer. Auditors must continually monitor the controls in effect to determine weaknesses; they must continually monitor the computer's operation to determine abuse opportunities and they must ask

the computer for key data which will indicate if in fact abuse has taken place or is taking place.

Internal Audit

Control of the EDP facility should vest outside of the operations personnel. Ideally there should be a group of internal auditors, which answers to top management, to perform this function.[50] Properly organized and staffed, their mere presence adds significantly to the difficulty of any computer abuse. Add to this the application and testing of controls by qualified personnel and the resultant difficulty in manipulating the computer will eliminate all but the most expert crook.

The internal control unit must have authority sufficient to do its job. This authority should include the power to shut down the computer at any time to perform a test. Examples of application of this authority would be dumping of a storage area, verification of a program, or the insertion of a test program. This authority gives internal auditors the ability to perform surprise checks and, since they must answer to top management, there is little danger such authority will be used imprudently. Other authority which should belong to the internal auditors includes access to the library, access to the operations log, power to design and implement controls, and access to input data to develop check figures.[51]

Responsibilities should also be assigned to the internal auditors. They should be responsible for developing procedures

and documentation standards for programs and program changes. In addition, they should be responsible for testing and approving all programs and modifications.[52] (Appendixes E and F). These responsibilities are a subject unto themselves, however, two points are worth mentioning here. First, program documentation standards must be very detailed and must be followed during program development, not prepared after the fact. This is especially important when programs are prepared by more than one individual. Second, testing procedures must be exhaustive and, as much as possible, test the potential effect of various data forms, especially unusual data. Test personnel should test what the program will, and equally important, will not do when combinations of data are processed.[53]

Monitoring the Computer

The continuous monitoring of operations is a very important function of control and follow-up. Again the computer is uniquely compatible with this type of oversight. In only milliseconds the computer gladly tells what it is doing. Such reporting is not a burden and does not materially affect processing time. The ideal method of monitoring the computer is a combination of hardware and software. A secure remote terminal can be established which will print out the information requested by a Systems Measurement Facility (SMF) program. The SMF program can be developed to print out any required data but as a minimum it should print out console

entries. Anyone "talking" to the computer should be monitored. Operations personnel should maintain a log of console entries and this also should be compared with the terminal print out. All unsuccessful attempts to gain access to the computer should be printed out. All unauthorized attempts to access data files should be printed out. Computer operators should be required to develop schedules and all unscheduled runs or operations should be investigated. The SMF can be requested to print out certain check figures, flag unusual transactions, or search for unusual combinations of transactions. In addition it can be programmed to analyze data and prepare reports.[54]

To fully utilize the SMF program, certain data should be prepared manually for later comparison with computer operations. As already mentioned, operations personnel should prepare run schedules. Other comparison data should be prepared by data users and preparers (not the separate keypunch unit but the initial data preparer). The comparison totals can be amounts or quantities but ideally should be easy to gather but offer meaningful comparisons. Frequently used comparison totals are numbers of personnel for payroll, batch totals for account updates, or hash totals for sensitive operations.

Two SMF programs currently in use include ABUCUS (Time Brokers, Inc.) and SMS/CAS (Boole and Babbage). Another commercial program worth mentioning is IBM's Log Tape Analysis. The central processing unit of the computer produces a log tape which contains a record of all systems activity. The log

tape is difficult to read because of the machine language it uses; however, this program performs an analysis of the log tape to provide the monitoring information needed to compare the actual computer usage with the schedules and logs maintained by operations personnel. Also, it can be programmed to analyze the log tape to highlight data which can be used to control computer usage.[55] (Appendix L).

Threat Monitoring

The key to any successful follow-up program is that it is continuously updated and that controls are in fact monitored. Continuous update is called threat monitoring. Control personnel should continuously check for weak areas of the system. In analyzing a threat, control personnel should assume a potential intruder has complete knowledge of the existing security system. Not only will such a procedure suggest new control measures for implementation, they will also suggest where to look when auditing the system.[56] Controls monitoring is performed in two phases. First, every system should be subject to periodic operational security audits which are a total audit of 100 percent of the security system. Such audits may be scheduled or unscheduled, depending on the situation, and determine whether controls are in fact being used and whether they are effective. The second phase involves spot checking of individual controls, especially those which leave no visible trail, on a rotating basis, to insure controls are not being

ignored and are effective. Ideally, these spot checks should be unannounced. (Appendix J).

The audit of security controls deserves special comment. As in most cases, controls involve conflict and trade off. The conflict involves efficiency and the trade off involves less efficiency for more security. This is important because auditors must understand that operations personnel may regard controls as inefficient and deliberately disregard them. If periodic checks are made, it will not be possible to ignore the controls, but control personnel must also understand their responsibility does not end there. They must insure that efficiency is a criteria of every control and they must insure operations personnel understand the controls and have input into their development.[57]

External Audit

An important part of any follow-up system is the external audit performed in conjunction with annual statement preparation. While the audit is concerned with the fairness of presentation by the statements, the evaluation of internal control is a valuable check on the performance of internal auditors and can uncover security weaknesses. The audit itself is a valuable tool in detecting abuse if the auditors relate weakness in internal control not only to reliance on the data produced by the system but also to the abuses which could exploit such weaknesses. Donn B. Parker has identified significant

weaknesses which auditors should be aware of. His studies of computer abuse resulted in, among other information, a listing of the characteristics of computer facilities most vulnerable to fraud or embezzlement. He lists six major characteristics:

1. The computer system is used for financial processing applications including payroll, accounts payable and receivable, and storage and maintenance of files of financial data.
2. Among the employees, there is more loyalty to each other than to the employer.
3. The organization does not separate sensitive job functions and lacks dual control of important tasks.
4. The system services and physical facilities are available to some employees during nonworking hours and without supervision.
5. Computer programs, including the operating system, are not under modification control, and ownership is not sufficiently displayed or otherwise established.
6. Disgruntled employees are not identified and removed from sensitive jobs.[58]

Audit and Control Tools

In the past, auditors have become aware of systems weakness in manual systems which were significant enough to cause a report of the weakness to management and to consider the weakness during the audit. The same sort of awareness is to be expected of EDP systems weaknesses. A system having some or all of the above six characteristics should reasonably be reported to management and such weaknesses should be considered when performing the audit.

In the past three to four years, as a result of the work of persons like Mr. Parker and the big computer abuse cases

uncovered, there has been more progress in designing securer systems than in the 15 previous years. Because of this increased emphasis on security, there has been an equally spectacular increase in the sophistication of audit tools available, particularly in the software fields. These audit tools, for discussion purposes, have been classified as audit packages and support packages. In this discussion, the term audit package will mean computer software developed to allow an auditor to get at and manipulate the contents of data processing files for the purpose of performing an audit. The designation "package" is used because they are a coordinated group of programs. Thus the package may contain different programs for different purposes and is extremely flexible. In this discussion, the term support package will mean software packages developed by systems designers and EDP users to support the design, installation and operation of an EDP system. The key element of the definitions is who developed the packages because, until recently, many auditors knew little of the availability of support packages and most EDP personnel were unaware of the audit applications of their support packages. This point is emphasized because in reality there is little difference between support packages and the programs which make up an audit package.[59]

The number of packages available of both kinds is large and growing rapidly. It serves no purpose to try to list them all since they are similar and this short discussion

would not make a reasonable comparison possible. Instead, a description of a single audit package in sufficient detail to allow evaluation of their usefulness will be provided. Similarly, a description of a representative sample of support packages to allow appreciation of their variety will be provided.

Illustrative audit package. The audit package that will be described is called AUDEX for computer Audit Extract System and was developed by Arthur Anderson and Co.[60] This choice does not imply superiority of the system, but merely the availability of sufficient descriptive material on the system.[61] AUDEX was first developed in 1969 as a replacement, or more precisely, an improvement of existing "custom" programs and packages used to audit specific firms and specific industries. It utilizes the client EDP system and is designed to maximize the capabilities of computers to save time and costs. In the words of its developers:

> Specifically, AUDEX is a library of computer routines which can be linked together to perform the desired audit procedures. It contains no "standard" computerized audit procedures; rather, by applying various combinations of the routines contained within the package, the auditor is able to tailor it to accomplish the desired procedures on each audit engagement.[62]

AUDEX does not replace the auditor nor eliminate audit procedures. It performs many of the functions which are performed in auditing a manual system and produces data which can be used by the auditor in evaluating whether the client's data supports the financial statements. Utilization of

AUDEX does not require expertise in data processing but does require knowledge of the audit steps being performed and knowledge of the client's data processing system and the data stored in it. This background is required because the auditor must "tell" the AUDEX package where data is located, the functions to be performed on the data, and describe the output desired. These instructions are input through specification sheets, composed of a series of narrative questions, each of which requires a predefined coded answer. The coded answers are then converted to punch cards and read into the computer's memory. The specific functions which the computer will perform for the auditor are: select, extract, sort, merge, match, accumulate, summarize, sample, format, calculate, sequence check, and print. The AUDEX package consists of two programs--AUDEX I, a general program which retrieves the desired information, processes it, and transcribes it into an output file, and AUDEX II which generates the desired reports or confirmations. Again, using the words of its developers, the AUDEX system can perform, among others, the following functions:

1. Extract selected data from a computer file for further processing,
2. Sort selected fields of data within a file,
3. Include or exclude data records with specified characteristics,
4. Summarize groups of like data,
5. Perform mathematical calculations on specified data fields,
6. Select data records on a random basis,
7. Read two files of data for selecting, merging, extracting, comparing, or performing mathematical calculations on specified data,

AUDEX does not require expertise in data processing but does require knowledge of the audit steps being performed and knowledge of the client's data processing system and the data stored in it. This background is required because the auditor must "tell" the AUDEX package where data is located, the functions to be performed on the data, and describe the output desired. These instructions are input through specification sheets, composed of a series of narrative questions, each of which requires a predefined coded answer. The coded answers are then converted to punch cards and read into the computer's memory. The specific functions which the computer will perform for the auditor are: select, extract, sort, merge, match, accumulate, summarize, sample, format, calculate, sequence check, and print. The AUDEX package consists of two programs--AUDEX I, a general program which retrieves the desired information, processes it, and transcribes it into an output file, and AUDEX II which generates the desired reports or confirmations. Again, using the words of its developers, the AUDEX system can perform, among others, the following functions:

1. Extract selected data from a computer file for further processing,
2. Sort selected fields of data within a file,
3. Include or exclude data records with specified characteristics,
4. Summarize groups of like data,
5. Perform mathematical calculations on specified data fields,
6. Select data records on a random basis,
7. Read two files of data for selecting, merging, extracting, comparing, or performing mathematical calculations on specified data,

8. Accumulate and print subtotals or print selected data in a specified form.

AUDEX I can:

1. Extract and reformat data from client files,
2. Check input fields for proper sequence or order,
3. Perform mathematical calculations such as addition, subtraction, multiplication, and division on input fields, (up to ten mathematical calculations in a single reading of a data file).
4. Accumulate input fields,
5. Based on specific characteristics choose data to be passed on to AUDEX II.

AUDEX II can:

1. Edit and print reports and confirmations,
2. Analyze a field and place it into one of several categories,
3. Summarize records with similar characteristics,
4. Accumulate data for control and grand totals,
5. Select or reject records with specific characteristics,
6. Sample input records on a random, systematic or block basis.[63]

Of particular importance in an evaluation of computer abuse is a particular characteristic of packages such as AUDEX. They are extremely powerful and virtually take over the EDP system being audited. This of course is also their greatest danger since it is possible to destroy a record system with such a package, and only experts should attempt to design one and must thoroughly test it before implementing it. However, this feature, because of programming and system characteristics, makes most forms of abuse involving manipulation of the EDP program stand out. Such abuses must be designed to be compatible with the existing operating

system and as such are probably not compatible with the audit package which is in sophisticated machine language and need not go through the client's operating system. Other forms of abuse, such as data manipulation, may also be detected since the speed of the computer allows sampling for exceptions or attributes on a far, far larger scale than is possible with manual systems.

Illustrative support package. Support packages developed by hardware manufacturers and EDP personnel add a significant dimension to the audit packages like AUDEX. As mentioned earlier, audit packages replaced "custom" programs and packages because of the expense of such custom programs. The wide range of support packages, estimated to be in the hundreds, partially compensates for this lost customizing but at a fraction of the cost since programmers are not required to reprogram for each engagement. One sample of the packages available has been chosen as an illustration of what is available. The sample is from IBM because of the number of computers it has manufactured and because of the information available. The following listing of such programs has been compiled by Donald J. Dashefsky, Auditability Program Administrator for IBM.

1. Data Base/Data Communication Driver System: This is a simulation program which provides powerful testing capability of a terminal network. Its use is mainly in designing a system of on line real time terminals which will access and update a data base system through data communication

facilities. Its greatest value in preventing computer abuse is in threat monitoring and evaluation of the weakness of a terminal network.

2. Test Data Generator: This program will create a file of test data and print it. Its value lies in the fact that characteristics of each field within the file can be individually described and can be related to another field if desired. This can be accomplished by extracting two related fields or creating the second field by manipulating mathematically the first field. Fields can be selected at random or in sequence. The result is that many hours of tedious work extracting test data can be reduced to minutes or even seconds of computer time.

3. CICS Network Activity Simulator: This program evaluates a system ability to handle projected loads of communications traffic. However, it can also be used to evaluate the vulnerability of a system to abuse via attempts to overload it.

4. Interactive Query and Report Processor: This program is a powerful but flexible tool for obtaining report data from a file. Information desired is described and the program locates and prints it. The program interacts with the computer, however, so that if the data generated is unsatisfactory the instructions can be revised and the job rerun. Such a program allows "selective dumping" of information and saves both processing time and reduces the amount

of superfluous information received when searching for data with the desired characteristics. In addition it contains "bells and whistles" which can be used to highlight data with specific characteristics and relationships to other data.

5. Batch Query Facility: This program is a cheaper and smaller version of the Interactive Query and Report Processor and handles only one file of fixed length records. While the instruction set is limited and lacks many of the "bells and whistles" of IQRP, it is cheaper and may be sufficient for the task being performed.

6. SMF-Graphical Analysis Program: IBM calls this program Systems Management Facility; however, it is basically the same as the Systems Measurement Facility described earlier when the use of a remote terminal in monitoring computer usage was discussed. This program can prove unwieldy since it often produces massive amounts of data; however it does accomplish two important tasks. It provides audit trails of data transactions and of jobs run. It uses the computer's own power to search and compare massive amounts of data, and produces listings, histograms and a chart which highlights data. The amount of data produced makes it difficult to use, however, when understood it can be used to trace a particular transaction or evaluate the processing performed by a specific computer run or program. It can be invaluable when the sheer volume of data obscures

audit trails.

7. System for Online Tape and Disk Libraries: This program can be used in conjunction with the access control systems described earlier. It allows the computer to select tapes and disks from a library at the request of a terminal user while retaining control over who has access to particular data, and control over the rights which the user will be granted.

8. Data Dictionary/Directory with Data Element Glossary: This program can be invaluable for internal auditors trying to monitor a large facility. It maintains a central record of all data elements in a system, their location and their format. One point should be considered, however, when using this program and protecting access to it. A potential abuser of the system will get an invaluable asset from the information it provides. The program provides five basic output reports:

a. A list of all names used in the system and their definitions.
b. A layout of each unique record identified within the system.
c. A key Word Out of Context list of each word used within the names defined in the system.
d. An alphabetical list of all names used within the system.
e. A report of all additions, changes, and deletions made to the data dictionary.[64]

Each of these reports will aid in controlling a system, however "e" may prove most invaluable if the program is tied into the operating system and the program is automatically updated whenever a unique name is introduced to the system.

Since data names must be unique, an unauthorized patch and storage location may be discovered when it is printed out in the report.

9. Audit-Source Code Compare: This program is a very powerful tool in discovering program patches and other unauthorized modifications. When internal auditors approve a program or program modification, a control copy is duplicated and secured away from the EDP facility. The Audit-Source Code Compare program performs a sophisticated match of this control copy and the program in use and produces a report that highlights any differences between the two programs. Additions, deletions and changes are flagged and identified on the output listing.

10. DOS DBDUMP Utility Program: This program can be used to dump the contents of a data base or selected parts of a data base to the printer. It is selective and can be used to scan a data base looking for an individual item with specific characteristics. This program also has a danger associated with it, however. Since it is selective, it can be used to find the location of a specific item or data which, while valuable to auditors, is also valuable to a potential system abuser looking for the location of an unguarded access point.

These 10 programs give an idea of the potential, largely untapped, audit and control tools already available.[65] Again it should be emphasized that this is a very, very small

sample of the hundreds available. Their use is relatively inexpensive because of the number of users, which allows the programming costs to be spread out, and the fact that hardware manufacturers have designed such programs to be compatible with the software and equipment they offer. Thus, the program modifications required are reduced. An additional point to be considered is that, as with AUDEX, their use is dependent upon audit skills and knowledge of the EDP system, not EDP skills. Also, while AUDEX was presented as a tool for external auditors, audit packages offer equal uses to internal auditors, especially in performing operational security audits.

Audit Concepts

In addition to the retrieval packages there are audit concepts which are significant enough to warrant mention. Initial attempts at auditing computers settled on two approaches, the test deck and the program simulation.[66] A test deck involves the creation of input data which is similar to that normally processed by the system. The deck, however, using a threat monitoring type concept, contains examples of every possible data error and is processed through the computer system being audited. The output is examined to determine if the system discovered the concealed errors and is compared with the expected output. One obvious caution when using this method is to avoid contaminating existing data of the system. Program simulation is the

corollary of the test deck approach and involves running data from the system being examined through a similar system and comparing the results. One caution of this method is that it may not fully test the programs involved since unusual transactions and errors may trigger an illegal patch or logic error that will not be discovered unless such data is run through the program simulation. A more modern concept of program simulation, integrated test facility (ITF), is being used with sensitive data in some installations. A particularly sensitive phase of processing is reproduced in another computer or an area of the computer controlled by test personnel. Data is processed by both programs and the results compared. Such a procedure can only be justified for very sensitive data; however, when skillfully designed, only parts of a process need be duplicated to test the entire process.

Two other procedures include flow charting and dumping. Flow charting programs are a useful tool in verifying programs. They can print out the flow chart of a program which greatly simplifies examination of a program while increasing the value of such a review. Dumping is the process of printing out the information in a particular area of memory. Its most valuable use is in looking for program patches and in examining data for signs of manipulation.

Summary

The foregoing review has attempted to demonstrate the

vulnerability of present EDP systems and the need for complete computer security systems to combat this vulnerability. Security systems must start with a recognition of the existing capabilities of computer hardware and software. Once their capabilities are recognized, the threats to the system must also be recognized. The implementation phase is the bridge to utilize the computer's capabilities to reduce the threats. However, given the present state of computer technology, implementation must also utilize controls exterior to the computer system. An example of these exterior controls is personnel organization and the separtion of critical duties. By this one, two, three presentation it is not intended to imply a separation of analysis and implementation. They must be coordinated and continuous and indeed continue over the life of the system.

The required integration of the system is proved by the follow-up phase. Follow-up involves continuous monitoring of existing controls and continuous threat monitoring to determine the effectiveness of existing controls and the need to implement further controls. In addition, effort must be directed towards detecting unauthorized penetration of the system. The relationship of the steps can best be illustrated by Figure 1, as shown below.

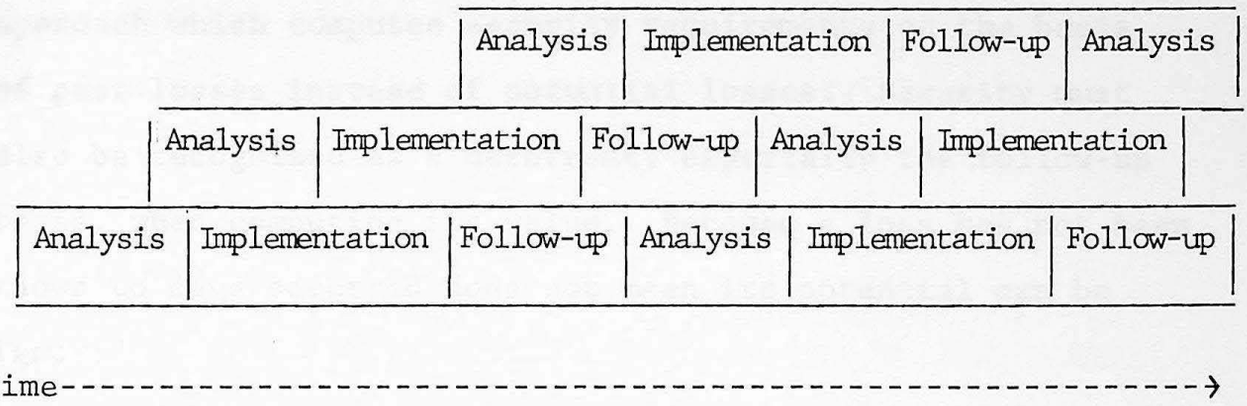

Figure 1. Relationship over time of the phases of computer controls.

As can be seen from the figure, the cycles are continuous, are constantly repeating, and are occurring simultaneously. The system cannot be static or it will be easily breached.

In an earlier part of this paper, the perceived conflict between control and efficiency was discussed. There exists another conflict between cost and control. A system of interrelated controls has been described which, while intended to prevent and detect most forms of computer abuse, may cost more than the related abuses. Cost/Benefit analysis is outside of the scope of this study; however, a familiar theme deserves to be restated. Security does not override management's responsibility to effectively utilize resources. Security is intended to conserve resources with the amount of effort expended directly related to the potential loss.

Another conflict which should be recognized, and in this case just as quickly disposed of, is the short sighted

approach which computes security requirements on the basis of past losses instead of potential losses. Security must also be recognized as a deterrent, especially the follow-up phase, when computing its value. Because a loss has not been known to have occurred does not mean its potential can be ignored.

This completes the overview of the available security controls designed to prevent computer abuse. They have never been more important than today because of reliance on the computer. Without a good prevention system, large numbers of computer users have the capacity to subvert present computers at both the system and application levels. The prospect of a technological solution to the problem is a realistic hope for the future; however, even the new systems currently being designed which have included security as a design criterion will not totally replace the controls described. When or if the ultimate state of perfect security is attained, the conversion process would last many years and security systems well-designed, coordinated and monitored will remain the primary defense against computer abuse in the interim. This vulnerability becomes even more significant and security systems even more critical in light of the fact that recent studies have shown not only an increase in computer abuse activity but also an increase in the dollars involved.

The importance of these controls can be better under-

stood when the potential results of ignoring controls are examined. To graphically illustrate the results, the appendix presents the results of research into cases involving computer abuse.

---

[50]Richard C. John, Thomas J. Nissen, "Evaluating Internal Control in EDP Audits," The Journal of Accountancy 129 (February 1970): p. 31.

[51]Computer Security Research Group, Computer Security Handbook, p. 160-161.

[52]Thorne, "Control of Computer Abuses," p. 42-43.

[53]Computer Security Research Group, Computer Security Handbook, p. 79-80.

[54]Robert L. Stone, "Who is Responsible for Computer Fraud?" The Journal of Accountancy 139 (February 1975): 37.

[55]Adams, "Alternatives to Computer Audit Software," p. 56.

[56]Parker, Computer Abuse, p. 24.

[57]Podgus, Outwitting the Computer Swindler," p. 12-13.

[58]Donn B. Parker and Susan B. Nycum, "The New Criminal," Datamation 20 (January 1974): 57.

[59]Adams, "Alternatives to Computer Audit Software," p. 54.

[60]Touche Ross calls their audit package STRATA.

[61]Milton H. Fortson and Eugene L. Delves, "AUDEX," The Chronicle 30 (December 1969): 16.

[62]Ibid., 18-19.

[63]Ibid.

[64]Adams, "Alternatives to Computer Audit Software," p. 56.

[65]Ibid., p. 54-57.

[66]Horne, "EDP Controls to Check Fraud," p. 45.

APPENDIXES

ILLUSTRATIVE CASES

Appendix A: Equity Funding

Scope of the Problem

Theft has been with us as long as there has been things worth stealing. Computer theft, or more broadly, computer abuse, has been with us as long as we have had computers. The significance of computer abuse when compared with ordinary theft is the sophistication of the persons and equipment involved. When computers were first introduced they were little understood or utilized outside of very large installations. There were few people, with even fewer opportunities, to abuse the computer. This situation has drastically changed. There are now 2,200,000 people working directly with computers with approximately 10,000,000 more in positions where they can claim to be associated with computers. This large population supports the approximately 184,000 full-sized computers currently in use today.[1] Add to this the fact that computers have taken over the processes and environs where white collar crime has occurred in the past,[2] and you have, by the estimate of Edward Bride, Vice President of Computer World Magazine, computer related losses annually amounting to billions of dollars.[3] This figure may be disputed; however, losses are certainly over a billion dollars a year if you accept the estimate of 10 cases of abuse for each case

reported. In the only acceptable scientific research into the problem, Stanford Research Institute has determined the average loss per quantifiable case as of July 1973 was $1.4 million, and that figure does not include the $2.0 billion Equity Funding case.[4]

Summary

The reason Equity Funding is excluded from most data is because, technically, it was not a computer crime. The crime itself was the sale of phony insurance policies to reinsurers and inflation of earnings to support a high stock price. However, its scope would not have been possible without a computer to keep track of the phony policies. As a computer related crime, however, it may be as significant to accountants as the McKesson & Robbins case of 1939. Equity Funding graphically illustrated the need for auditors to be concerned with computer processing when attesting to the fairness of financial statements.

Equity Funding first began to market its "product" in 1960. The basic plan called for purchase of mutual fund shares from the parent company. These shares were then security for "program loans" to purchase insurance from Equity Funding Life Insurance Company, a principal subsidiary of Equity Funding Corporation. At the end of ten years, the mutual fund shares were to be sold to pay the program loans and hopefully also allow a profit to participants. As early as 1964, program loans which were carried on the

books as funded loans or accounts receivable were being duplicated, falsified and inflated. By 1972, the overstated assets amounted to $74 million from phony loans and another $36 million in overstated or ficticious assets.

With the creation of phony loans came the creation of phony insurance policies and the second half of the scheme. As early as 1966, Equity began creating the phony policies to bolster its income in an era when insurance profits were falling. The phony policies were supported by commissions revenue and operating expenses which resulted in earnings being recognized on the policies. The last element of the fraud was the sale of the phony insurance policies to reinsurers beginning in 1970. Through inflated promises of a persistency rate of 85% (no more than 15% of policy holders would cancel during year 2), Equity was able to sell the policies for 180-190% of annual premiums. The revenue realized from these sales represented 20% of their after tax profits. These sales, however, led to the ultimate collapse of the pyramid. Since Equity was now dealing with outsiders, phony assets and internal juggling could no longer cover for the fictitious policies. Second year and beyond premiums had to be paid by cash to the reinsurers. This resulted in more phony policies, more sales to reinsurers, and ultimately $2 billion dollars in phony insurance policies. The scheme finally collapsed on March 7, 1973 when Ronald Secrest told Raymond Dirks, a

securities analyst, about the phony policies.

Over its 13 year history, Equity reported $75 million in income and by 1972 the distortion in its financial statements could be summarized as follows:

Assets	
Branch account	$36 million
$24.6 million	fictitious bonds
8.0 million	fictitious commercial paper
3.4 million	fictitious deferred taxes and policy reserves
Funded Loans and Accounts Receivable	$74 million fictitious loans
Owners Equity	
Capital	$35 million in unrecognized losses
Retained Earnings	$75 million
$75 million	reported earnings

Audit Follow-up

None of this was a direct result of any direct computer abuse; however, it could not have been attempted without access to a computer to keep track of the fraud. Upper level management operated an IBM 360/40 computer on weekends and used it to insert 60,000 phony insurance policies into the files. The phony policies were assigned to "Department 99" which was programmed for special handling. For example, no bills were sent to "Department 99" policy holders while normal address changes, deaths, cancellations,

etc. were generated to meet statistical probabilities. With the computer keeping track of the phony policies, hard copy files were only needed on the small number of them requested by the auditors. The computer played another role in the fraud. The company's auditors were not capable of auditing the Equity computer records. One member of the investigating firm which uncovered the fraud stated the auditors never set foot in the EDP center and "were forced to take what Equity said on faith. They were not equipped to do anything else."[5] Since direct confirmation of insurance policies with policy holders is not a normal audit step, the auditors were reduced to asking for proof from Equity itself that the policies on the computer printout sheets were real.

While the fraud may not have been a computer abuse, its long duration can only be blamed on the total lack of any follow-up procedures and controls. There was no internal auditor and the external auditors lacked the skills to audit EDP records. Had there been an independent internal auditor, or had the external auditor examined the computer files, the fraud could not have been accomplished. Such a massive scheme could not have been concealed. Indeed, one of the duties of internal auditors would have been to monitor the use of the computer. Sixty thousand policies entered on weekends would have certainly gotten them started. The external auditors would have been alerted equally as easily by fictitious policy premium payments and billings.[6]

Appendix B: Airline Volume

Summary

In the early days of its computerization, the airline industry was being cheated out of thousands of dollars each year because of inadequate controls. The airlines knew of the fraud but could not detect nor stop it because of the sheer volume of the transactions camouflaging it.

The typical airline ticket is in three parts:

Part one is the "auditor's coupon." It is sent to the airline Revenue Accounting Department with the money collected (less any commission). After the revenue is recorded, the coupon is entered in the Master Audit File.

Part two is collected at the boarding gate and is also forwarded to Revenue Accounting to be matched with part one in the Master Audit File. A part two with no part one indicates a theft by the ticket seller or a theft of the tickets themselves.

Part three is the copy retained by the customer.

This system did not work, however, for two reasons. First, the volume caused errors and omissions in input to the computerized Master Audit File, and second, the volume offered opportunities to merely insert part one in batches being sent to the computer for input. Had these problems been anticipated and provided for, many dollars of losses

would have been saved and many dollars spent patching the system after the fact would have been saved.[7]

Prevention

This type of abuse can be prevented through proper design of control features when the system is being designed and installed. Once the problem was discovered, there was little doubt that corrective action would be taken. The problem should have been anticipated during the systems design phase and adequate provision should have been made for the eventual volume. Controls should have been designed to counter anticipated threats. In addition, data preparation controls should have been designed to prohibit tampering with data prepared for input.

Detection

In this case detection was easy. The revenue being received did not match the tickets being processed. However, because of the volume and inadequate input controls, the shortage could not be related to specific documents. Adequate input and batch controls would have made such a matching possible.

Appendix C: Eyeshade Accountant

Summary

A sales organization installed a new computer to take over procedures previously handled by hand. The accountant charged with implementing the system did not understand it and had no desire to learn. Because of this he was grateful to and did not question the motives of "Bob" the sales manager who offered to help him prepare data to be used in computing commissions. Commissions were computed twice each year and paid over the next six months to insure a level income. "Bob" was to receive an "overriding" commission on total sales. The accountant approved a procedure whereby, based on past experience, "Bob" would anticipate repeat orders of certain products by certain customers. His estimates were very accurate and dutifully entered into the computer to be used to compute commissions. When the repeat order was made, however, this sale was also recorded for commission purposes and a double commission on the repeat order was paid to "Bob." The fear and lack of understanding of computer operations on the part of the accountant resulted in overpayments of several thousand dollars.[8]

Prevention

This type of abuse can be prevented through training

and implementation controls. The persons charged with using a system must be properly trained. If they refuse to be trained, they must be replaced. During the implementation phase the old system should be continued and compared to the EDP results to insure the computer system is performing as desired.

Detection

Follow-up is a constant requirement. Transactions should be tested on a sample basis to check for errors and irregularities.

Appendix D: Carl Joseph Maggio, Inc.

Summary

For six years, 1962-1968, the chief accountant Eldon Royce of this fruit and vegetable shipping firm carried out a "salami fraud," the manipulation of many accounts to accumulate a balance in an account controlled by the thief. This case is unusual in that the accountant also ran a computer service bureau which he used to simulate the accounting system of his employer. After determining, via simulation, which expense and inventory accounts he could take money from without being detected, he charged these accounts with phony purchase orders and receipts. The difference between what was actually owed and the inflated payable was credited to dummy vendor companies with accounts in a local bank. He embezzled over $1 million before getting caught. He got careless or greedy or both and stopped using the computer to determine which accounts to manipulate and how much could be stolen.

Prevention

This type of abuse can be prevented through controls on input. Over a six year period, the accountant had to make hundreds of entries into the files. There must be specific controlled procedures for the preparation of data in terms

of back-up, authorization, and source.

Detection

With adequate follow-up controls this fraud would have been uncovered rather quickly. First of all, on-going personnel checks of persons in sensitive positions should be in effect. $180,000 per year is a great deal of money to conceal. Probably the most effective follow-up control, however, would have been a properly conducted audit. A study of attribute sampling indicates a high probability one of the false entries would be chosen for investigation. When a transaction is chosen for investigation, it must be thoroughly scrutinized from start to finish. In this case purchases and expenses were entered without invoices or receiving documents. Such entries cannot stand scrutiny. The transactions themselves should have betrayed Royce also. Their unusual nature should have alerted internal control personnel.[9]

Appendix E: Zzwicki

Summary

In this case a bank programmer inserted a patch into a program and caused funds to be accumulated in the last deposit account which he had opened under the name of Zzwicki. There are two versions of the theft. The first involves a patch to the service charge program so that it added ten cents to every service charge less than ten dollars and one dollar to each service charge over ten dollars. The additions were then credited to his account. The second version involves a patch to the interest program which caused all computations to be rounded down with the round down credited to his account. Whichever version you believe, he was caught when the bank, under a new marketing program, tried to honor the first and last names on their customer list.

Prevention

This fraud is easily prevented if there is proper control of programs. First of all, programs must be documented to include a flow chart. Programs must be tested by internal auditors to determine if they are performing as designed and to determine the effect of unusual transactions. One indispensable testing procedure is using a flow charting program to produce a flow chart of the program being tested.

With proper testing, the patch will be uncovered and, since tested programs are not returned to the programmer and programmers are not allowed to operate the computer, insertion of the patch later is extremely difficult.

Detection

If the patch is undetected during testing or is inserted later, there are still follow-up procedures which can detect fraud. All programs should be periodically, on a surprise basis, checked against the approved program. Such testing will discover the patch when the programs are compared. Check figures should be developed and manually compared to the computer generated figures. In the case of the service charge, internal and external auditors would select a sample and compute the service charge manually. The interest scheme would be more subtle and less susceptible to manual checks, however, a conscientious internal auditor would notice the pattern in this case also. A last control measure which may uncover the fraud is verification of accounts. Again a sample is taken, this time of accounts, and verified as to proper bank records and existence of the depositor via city directories.[10]

Zwana

A variation of the above scheme occurred in an English mail order company. The programmer collected sales commission round downs in the last account, opened by him under the name Zwana.[11]

Appendix F: National City Bank of Minneapolis

Summary

This was another banking embezzlement. A young programmer was responsible for programming and occasionally operating an IBM 1401 computer. He placed a patch in the computer to have it ignore his account when checking for overdrawn accounts in exception reporting. His original purpose was a three day loan of $300, however four months later, when he was discovered because of a computer breakdown, the patch was still in and he was overdrawn $1,352.

Prevention

The most obvious prevention technique is to remove temptation and separate the duties of programming the computer and operating the computer. Programmers should not be allowed to operate the computer. If this rule is followed and all program modifications as well as initial programs are thoroughly tested, the patch cannot be inserted into the program.

Detection

This fraud can be detected if programs are periodically tested on a surprise basis. Comparison of the program with the original documentation will uncover the patch.[12]

Appendix G: Diner's Club

Summary

This case involved collusion in the submission of overstated invoices and their payment. In 1970, Gerald Branco, a Diner's Club vice-president, was assigned the task of setting up a computerized record system. He contracted with two small data service companies to provide temporary computer operators and key punch operators. The two companies, Action Temporaries, Inc. and Action Computers, Inc., controlled by Joseph Fryzer and David Alexander, submitted $1.8 million in invoices for their services. These invoices, approved by Branco, were overstated by $800,000. To avoid detection Fryzer and Alexander created time sheets and employee records to substantiate the overbillings. Checks were issued by the data service companies payable to the nonexistent employees and, with the aid of a fourth conspirator, a teller at the East Chester Savings Bank in Manhattan, the funds were accumulated safely out of sight of Diner's Club auditors.

The computer was involved in two ways. First, as stated above, it provided the opportunity. Second, it camouflaged the phony invoices and gave them an air of authenticity. Once the invoices were approved by Branco they were inserted

into the computer and the computer dutifully wrote checks for the approved invoices. On close inspection the invoices may have raised questions, especially since billings were overstated by 80%. Mixed in with the millions dispersed annually, however, the final payments did not cause concern. To quote McKnight in "Computer Crime," "Human tolerance is not so accurate (as a computer's), maybe, but it is far more suspicious."[13]

The detection of this scheme can only be described as a fluke, combined with carelessness on the part of Branco. To avoid attracting attention, the $800,000 was spread over several accounts. The teller at the East Chester Savings Bank allowed accounts to be opened easily; however, he could not prevent the routine checking of depositor's references. In opening one checking account for $30,000, Branco used a Diner's Club card as a credit reference and listed a lower Manhattan address for the depositor. When the bank routinely checked with the Diner's Club, the address caught the eye of Jack O'Toole in their Security Office. Unfortunately for Branco, O'Toole recognized the address and knew the persons living there did not have $30,000 to deposit in a checking account. Investigation soon uncovered the scheme.

Prevention

As with any case involving collusion, prevention is difficult. Personnel controls are very important. Branco

was known to be a big spender and on one occasion even bragged to O'Toole that he had given his wife a $10,000 Christmas present. One of the ongoing personnel checks should be of spending beyond one's income. Another control designed to prevent such a fraud involves separation of duties and dual control. Branco should not have been allowed to be the sole approver of invoices for payment and insertion into the computer. Normally the person responsible for a contract approves payment, however, a second approval before such data goes into the computer should be standard.

Detection

Detection involves the same personnel controls as above. The scheme can also be uncovered by a non-EDP control technique, the auditing of such contracts by internal auditors. Such a large overbilling would not stand scrutiny if the data chosen for testing is thoroughly examined.[14]

Appendix H: John Players and Sons

Summary

In 1969, John Players Tobacco Company offered cash for coupons attached to their tobacco products. Mike Micallef, a senior accounts clerk, inserted punch cards into the computer which credited him for redeeming non-existent coupons. His total theft was only about $50 and he asserts his sole purpose was to graphically demonstrate security weaknesses in the John Players EDP system. The truth of his assertion may never be known since he was quickly uncovered when the owner of the apartment he used for his cover address became suspicious.

Prevention

The coupons were exchangeable for vouchers and, in the words of Micallef, the vouchers "were as good as cash." Prevention merely involves the same controls long used for cash receipts. An individual receives the coupons, verifies their count, and passes the necessary data to the keypunch department. A batch total is also computed by this individual and the coupons are cancelled at the same time. The coupons and batch total are routed to another individual who verifies the total and compares it to the data input into the computer. Since all vouchers must be supported by coupons, control of

the coupons will prevent this fraud. All computer input should be subject to a double check.

Detection

Detection assumes a successful fraud which assumes absence of the controls described above. Absent these controls. detection may indeed rely on the suspicious landlord. Once the data from the non-existent coupons is entered into the computer, it can no longer be identified as fraudulent.[15]

Appendix I: Fortes Holdings Ltd.

Summary

This is another case which illustrates the need for a double check of computer input. In the late sixties, an accounts clerk named Masood Hason Ansari passed on bills from a fictitious supply firm which he was authorized to certify for payment.

Prevention

This type of fraud is preventable by requiring such authorizations to be approved by a second individual. This individual should require that each transaction be substantiated by a purchase order and receiving document.

Detection

Assuming Ansari was in a position to create the necessary documents to fool the individual checking his input data, the fraud can be detected by the computer. Purchase orders and receipt documents should be input to the computer by the individuals charged with creating them. The approved invoice can be matched with these documents when the invoice is input. This is nothing more than an automated voucher system but effectively detects solo abuses.[16]

Appendix J: National Bank of North America

Summary

This case involved a two bank float or "kiting." Bank deposits are coded as to check or cash. Cash deposits are available for immediate credit while check deposits are often frozen until after the normal float period. In this case a bank vice president and four others coded check deposits as cash deposits and over a four year period stole $900,000. The scheme was discovered when a bank messenger failed to deliver $440,000 worth of checks to the clearing house and their accounts were overdrawn.

Prevention

Prevention of this fraud may be dependent upon detection as a deterrent.

Detection

Kiting requires constant control. A check drawn on bank #1 must be covered by a check from bank #2 which must be covered by a check from bank #1 . . . It also requires collusion between individuals at two or more banks. This volume of activity and collusion is an aid in detection. Rotation of duties and mandatory vacations can disrupt the chain and uncover the fraud. Internal auditors can also discover the fraud by periodic audits of bank terminals.

Such an audit would disclose a cash shortage which, when compared to deposit slips and checks, would reveal the fraud. The computer itself may also be a tool to detect this fraud. The large volume of covering transactions become "patternized" over a period of time. Computer programs can be written which test for kiting patterns.[17]

Appendix K: Youth Corps Payroll

Summary

Employees of a New York welfare department data center stole $2.75 million in only nine months by creating a fictitious workforce with equally fictitious social security numbers. The conspirators would intercept the checks, endorse them, and cash them. They were discovered when a policeman found a batch of over 100 checks in an overdue rental car found illegally parked.

Prevention

To prevent any fraud of this type requires that the payroll file be treated as a vulnerable asset. This file needs update only prior to a payroll run. The update can be easily controlled via the monitoring program and dual control during the update. An attempt to add to this file at any time, except scheduled updates, should cause an alarm to be triggered.

Detection

Assuming the false names are in fact inserted in the payroll run and checks are printed, this fraud should not avoid detection beyond the first payroll run. One of the functions of the personnel office should be the preparation of batch totals to be compared against the payroll run.

Since all new employees must be required to report to personnel personally, the employee list can be controlled. As an added check, the batch total from personnel should contain a social security listing. Another detection technique involves a surprise distribution of checks. On a surprise basis, internal control personnel should observe the distribution of payroll checks to insure the checks are valid.[18]

Appendix L: Payroll Check Duplication

Summary

No discussion of computer abuse is complete without adding the case of the EDP operator in West Germany who pushed the repeat button 200 times when his payroll check was printed. He was caught when he tried to cash 37 of the checks at the same bank.

Prevention

As with the padded payroll, such simple schemes as this should be prevented by treating the payroll run as a sensitive operation. All commands to the computer must be controlled via human or computer controls.

Detection

The repeat command during a payroll run is incompatible and should certainly cause an alarm in the monitor program.[19]

Appendix M: Penn Central Railroad

Summary

One of the most interesting cases of inventory theft involved 277 freight cars, valued at $1 million, which were stolen from the Penn Central tracks. The computer part was very simple. The cars were routed to the LaSalle & Bureau County Railroad in Illinois. Once off of Penn Central tracks, inventory records in the computer were altered to show the cars were scrapped. Although no one was ever caught at Penn Central, it is estimated at least four persons would be needed to work such a scheme. Three are needed to misroute the cars and at least one to create the computer input. The interesting part of the scheme involved the cars themselves and what happened to them. Under a debt settlement in 1970, the Equitable Life Assurance Society received 466 cars of Penn Central. Equitable had a contract with Penn Central calling for the leasing of the cars by Penn Central. When this lease contract expired, Diversified Properties Inc., under a contract with Equity, was to refurbish the 466 cars and lease them to the LS & BC Railway. This is where the theft began. At least 743 and maybe as many as 771 cars were actually delivered to the LS & BC Railway for refurbishment by Diversified. The conspirators were greatly aided by very poor transfer procedures, sloppy inventory records, and the chaos of the bankruptcy reorganization, which accounts for the inaccuracy of

the numbers. The cars were given LS & BC markings and loaded with freight. Many subsequently traveled Penn Central tracks. While Joseph Bonanno, owner of Diversified, returned the 277 cars and paid Penn Central $150,000, one of the amazing aspects of the case is that no one was ever charged in the theft.

Prevention

This may be the best time to again reemphasize the deterrent effect of controls designed to detect abuse. The fact that Penn Central employees knew they could not be caught even if the scheme were uncovered must encourage abuse of this type. One very necessary ingredient of prevention is a reasonable chance of detection. The main controls which would prevent such an abuse are a proper division of responsibilities, computer cross checks, and the monitoring of sensitive functions by internal auditors. The scrapping of inventory should involve authorization, disposal and the updating of computer records. All three functions should be separated and individually entered into the computer which can then match the transactions. If one transaction is missing, the updating of computer inventory records should be blocked. Each transaction should be verified before it is entered into the computer. Internal auditors should monitor access to inventory records which should be allowed only at scheduled times.

Detection

Large scale inventory theft will be uncovered by internal auditors when they sample and examine inventory transactions. One of the basic rules of detecting abuse is that samples must be thoroughly investigated. In this case, should one of the fraudulent deletions be chosen for investigation, a thorough check would discover the car was not scrapped. Once this is known, the computer and computer logs will reveal who entered the fraudulent data.[20]

Appendix N: Public Telephones Company

Summary

In 1970, Jerry Schneider, owner of an electronics supply company, stole $1 million worth of telecommunications equipment. Schneider bribed keypunch operators, stole codebooks, and even posed as a magazine writer to learn the Pacific computer system. With this knowledge and the stolen codes, he used an ordinary telephone and remote terminal to place orders for equipment while commanding the computer not to produce a bill. The computer prepared invoices for the goods which directed the warehouse to pack the goods and place them on the loading dock. The warehouse operations were not on the computer. Inexperienced and poorly trained individuals allowed him to pick up the equipment at night in a truck painted to look like a Pacific transport. The normal daytime staff would have required bills of lading which Schneider did not have. He was discovered when one of Schneider's employees demanded money and turned him in when Schneider refused.

Prevention

The Pacific personnel involved did not have an appreciation of their responsibilities and how much their employer relied upon them. Proper staff training reduces the oppor-

tunity for an individual to obtain the detailed information required. The individuals Schneider bribed did not understand the consequences of their actions and the night shift at the warehouse had not been trained to issue equipment. The weakness of the system was increased by the fact the warehouse was not integrated into the system. Threat monitoring should reveal such weaknesses and produce controls to balance them. Proper monitoring of the computer will also prevent such thefts. A shipping order without a charge to a customer or department should trigger an alarm and print out at the monitoring terminal and suspend the transaction until proper authorization is received.

Detection

The personnel controls mentioned above also offer the best method of detection. Computers are not suspicious while people are. Properly trained personnel will report contacts and occurrences which violate their training. Indeed, part of their training should be to encourage such reporting. Internal auditors should detect such a fraud in two ways. First, the monitoring program should, as was mentioned above, be programmed to flag unusual transactions such as shipping instructions without a billing. As a part of the continual threat, monitoring such transactions should be examined. Second, internal auditors should also sample inventory transactions and verify proper authorization.[21]

Appendix O: Canadian Department Store

Summary

A systems analyst used his knowledge of the sales-order processing system to order expensive appliances which he coded as "special pricing orders." When these orders were processed in the EDP department he intercepted them and priced the items at $6. He then paid for the items when they were delivered. Systems analysts hired to review the adequacy of EDP controls discovered the fraud.

Prevention

This type of data manipulation can be prevented if all transactions affecting computer input are subject to a control which verifies their validity. In this case, special pricing orders should be segregated and logged before entering them into the computer. The separate listing should be compared with the computer listing of such transactions. Proper data control and separation of responsibilities will also insure the persons in a position to initiate sales orders cannot later alter them.

Detection

Internal auditors are again responsible for detecting such abuses involving data manipulation. Threat monitoring will identify such transactions as vulnerable to abuse which

should cause internal auditors to review them. If samples are properly investigated and the monitoring program is designed to search for patterns which indicate abuse, such frauds will be uncovered.[22]

Appendix P: Hugh Jeffrey Ward

Summary

In this case the item stolen was a program which converted aircraft specifications into design characteristics. Ward worked for University Computing Company and planned to use the program to compete with its owner, Information Systems Design Corporation. As a computer programmer, Ward had contact with EDP personnel from Shell Development Company, one of ISD's customers. He used these contacts to obtain Shell's access code to ISD's Univac 1108 computer. Using this code, he dialed up the ISD computer and simply asked for the program. Unfortunately for Ward, Shell was billed for the transaction which prompted an investigation.

Prevention

Such programs if truly valuable or proprietary should be protected by machine controls which prohibit their being dumped to customers. Since ISD was not selling the program but merely the output from it, such a control would not interfere with its authorized use. Proper appreciation of access codes and their secrecy is also a requirement for prevention.

Detection

The control in effect at ISD is the most effective at detecting unauthorized access. In this case a remote printer

monitored access. The print out was used to bill customers. To this, however, should be added the control function of evaluating the print out for transactions which are unauthorized, such as program dumping.[23]

Appendix Q: E. F. Hutton

Summary

Three employees in a Texas office stole securities valued at $500,000 from customers' accounts. To cover the thefts they altered account data to delete securities from inventory. When customers complained about the errors on their account transaction statements, the customers were told the computer had made a mistake.

Prevention

When a securities transaction is carried out, a written authorization is obtained from the customer. This authorization should be required back up for any computer entry. Given the volume of transactions and the ability of brokers to forge authorizations, detection is the most effective prevention however.

Detection

Customers themselves offer the most effective control on theft from their accounts. Since this may not control abuse where the customer is likely to contact the thief, as in this case, specific instructions should be furnished to customers with each statement which instructs the customer to contact internal auditors if statements are incorrect.[24]

Appendix R: Bank of America

Summary

A disgruntled executive with knowledge of computer codes knew that multinational operations and the worldwide flow of funds often caused Bank of America cash to be on deposit in foreign banks. For example, a debt to a Bank of America client could be satisfied by depositing funds in a foreign bank. The foreign bank would notify Bank of America of the deposit and the client's account would be credited. Often the funds would then be held on deposit for Bank of America since this facilitated foreign transfers by Bank of America clients.

A complex code involving the date, time, amounts and parties involved was used via a telex system. The executive had access to the code and successfully caused large sums to be paid to unauthorized persons because there was no terminal identification system or verification used which would have identified the message as coming from an unauthorized source.[25]

Prevention

Sensitive transactions should be under dual control. In this case the code was designed to insure only authorized Bank of America personnel could send a message. The code

also provided anonymity, however, since the sender and the location were not verified by a call back or other means. Such an identification system would prevent unauthorized transfers.

Detection

Without the controls described above, detection becomes very difficult. Since the overseas transferor did not look beyond the code, once the transaction is entered and accepted, it is not unique.

Appendix S: French Patch

Summary

A young programmer from a French firm was fired in January 1968 for stealing computer time. Instead of immediate dismissal, he was given the normal notice and continued to work during the notice period. Each January 1, an automatic updating of all records was performed and it was this programmer's job to prepare the program for the years ahead. The 1968 and 1969 updates were performed without a problem; however, the programmer had left a patch in the update program which said in effect, "Mr. Computer, if you are updating 1970 records, please destroy them." It was and it did.[26] It's hard to imagine a patch like this going undetected for almost three years, but why test fate.

Prevention

The obvious preventative measure in this case would have been to remove the programmer from the job immediately, especially since he was being fired for dishonesty. The problem goes deeper however. Program controls should have prevented such an unauthorized patch to be inserted. Approval of any program change should be required. In addition, entry of the change should not be made by the programmer.

Detection

Follow-up controls must include a periodic check of programs. The check should include a flow charting and a comparison with authorized versions on file. Both of these procedures are designed to look for both errors and unauthorized changes.

Appendix T: Union Dime Savings Bank

Summary

Roswell Stefan was a chief teller with access to a remote terminal in a branch office. Using this terminal he identified inactive accounts from which, over a period of three years, he stole $1.5 million. He simply pocketed cash deposits made by regular customers and covered it by creating false withdrawals from the inactive accounts. He was clever enough to steal only a portion of the interest on these accounts to avoid detection by manual auditing controls. He was caught only because police raided a gambling operation and discovered he was betting up to $30,000 per day on professional sports. When asked how he was able to feed such a habit on his $11,000 salary, he confessed to his scheme.[27]

Prevention

Had Stefan been honest, this theft would not have occurred. A thorough screening of employees is necessary to help detect future problems. The threat of detection may, however, be the surest form of prevention.

Detection

In this case, Stefan's personal habits should have been a key to detection. Personnel checks of employees in sensitive

positions should continue after hiring to uncover unusual borrowing, gambling, unpaid bills, questionable associates, extravagance, refusal of promotions, or refusal of vacations. Follow-up controls designed to uncover unusual transactions will also be effective. The volume of transactions required to perpetrate Stefan's scheme should have been detected by a sampling process.

FOOTNOTES TO APPENDIX

[1]Donn B. Parker, Susan B. Nycume and S. Stephen Oura, Computer Abuse (Menlo Park, California: Stanford Research Institute, 1973), p. 79.

[2]Donn B. Parker and Susan B. Nycume, "The New Criminal," Datamation 20 (January 1974): 56.

[3]Robert S. Strother, "Crime by Computer," Readers Digest (August 1976): 144.

[4]Parker, Computer Abuse, p. 78.

[5]Christopher Podgus, "Outwitting the Computer Swindler," Computer Decisions, 5 (September 1973): 13.

[6]Ibid., p. 13-15. Wyndham Robertson, "Those Daring Con Men of Equity Funding," Fortune 88 (August 1973): p. 81-85, 120.

[7]Brandt R. Allen, "Computer Fraud," Financial Executive 39 (May 1971): 40.

[8]Gerald McKnight, Computer Crime (New York: Walker and Company, 1973): 170.

[9]Allen, "Computer Fraud," p. 39-40. Parker, "The New Criminal," p. 56. Brandt R. Allen, "Embezzler's Guide to the Computer," Harvard Business Review 53 (July-August 1975): 82.

[10]Allen, "Embezzler's Guide to the Computer," p. 87. Thomas Alexander, "Waiting for the Great Computer Ripoff," Fortune 90 (July 1974): 144.

[11]Parker, Computer Abuse, p. 102.

[12]Allen, "Computer Fraud," p. 40. Donn B. Parker, "What to Do to Keep Light Fingers Off a Bank's Computer," Banking (May 1973): 34.

[13]McKnight, Computer Crime, p. 155.

[14]Ibid., p. 155-157. Wall Street Journal, 1971, 3/18-12; 3, 6/3-5; 2.

[15]McKnight, Computer Crime, p. 186.

[16]McKnight, Computer Crime, p. 186.

[17]Allen, "Computer Fraud," p. 40. Parker, Computer Abuse, p. 96.

[18]Allen, "The Embezzlers Guide to the Computer," p. 85. Parker, Computer Abuse, p. 92.

[19]Ibid., p. 102.

[20]McKnight, Computer Crime, p. 152. Wall Street Journal, 1971, 6/15-10; 2. 6/17-4; 2. 3/19-1; 4. 10.6-6;5. 11/15-2;4.

[21]McKnight, Computer Crime, p. 36. Allen, "The Embezzler's Guide to the Computer," p. 83. Parker, Computer Abuse, p. 98.

[22]Ibid., p. 101. Allen, "The Embezzler's Guide to the Computer," p. 84.

[23]McKnight, Computer Crime, p. 130-147.

[24]Allen, "Computer Fraud," p. 39. Parker, Computer Abuse, p. 92.

[25]McKnight, Computer Crime, p. 22.

[26]Ibid, p. 103.

[27]Allen, "The Embezzler's Guide to the Computer," p. 79. Alexander, "Waiting for the Great Computer Ripoff," p. 142.

SELECTED BIBLIOGRAPHY

Books

Computer Security Research Group, Douglas B. Hoyt, Chairman. Computer Security Handbook. New York: Macmillan Information, 1973.

McKnight, Gerald. Computer Crime. New York: Walker and Company, 1973.

Parker, Donn B., Susan B. Nycume, S. Stephen Oura. Computer Abuse. Menlo Park, California: Stanford Research Institute, 1973.

Thorsen, June-Elizabeth. Computer Security: Equipment, Personnel & Data. Los Angeles: Security World Publishing Co., Inc., 1974.

Magazine Articles

Adams, Donald L. "Alternatives to Computer Audit Software." The Journal of Accountancy 140 (November 1970): 54-57.

Alexander, Thomas. "Waiting for the Great Computer Ripoff." Fortune 90 (July 1974): 143-150.

Allen, Brandt R. "Computer Fraud." Financial Executive 39 (May 1971): 38-44.

Allen, Brandt R. "The Embezzler's Guide to the Computer." Harvard Business Review 53 (July-August 1975): 79-89.

Fortson, Milton H. and Eugene L. Delves. "AUDEX." The Chronicle 30 (December 1969): 11-20.

Gilson, Milo. "Computer Fraud--Who Gets the Axe?" Data Management 13 (April 1975): 22-23.

Horne, John M. "EDP Controls to Check Fraud." Management Accounting 56 (October 1974): 43-46.

John, Richard C., Thomas J. Nissen. "Evaluating Internal Control in EDP Audits." The Journal of Accountancy 129 (February 1970): 31-38.

Jones, G. Hunter. "DP Error and Fraud--and What You Can Do About It." Price Waterhouse & Co. Review 2 (1976): 3-11.

Kramer, Loren B. "Auditing Computerized Records." The Practical Accountant 9 (January-February 1976): 68-71.

Menkus, Belden. "Computerized Information Systems are Vulnerable to Fraud and Embezzlement." CPA Journal 43 (July 1973): 617-619.

Moore, Richard A. and Thomas J. Kroger. "Computer Generated Documentation." The Journal of Accountancy 139 (June 1975): 82-90.

Palme, Jacob. "Software Security." Datamation 20 (January 1974): 51-55.

Parker, Donn B. "What to Do to Keep Light Fingers Off a Bank's Computer." Banking 65 (May 1973): 34, 35, 50.

Parker, Donn B. "The New Criminal." Datamation 20 (January 1974): 56-58.

Podgus, Christopher. "Outwitting the Computer Swindler." Computer Decisions 5 (September 1973): 12-16.

Robertson, Wyndham. "Those Daring Con Men of Equity Funding." Fortune 88 (August 1973): 81-85, 120-132.

Stone, Robert L. "Who is Responsible for Computer Fraud?" The Journal of Accountancy 139 (February 1975): 35-39.

Strother, Robert S. "Crime by Computer." Readers Digest (August 1976): 143-148.

Thorne, Jack F. "Control of Computer Abuses." The Journal of Accountancy 138 (October 1974): 40-50.

Voris, J. Walker. "How the Computer Can be Used to Commit Fraud." Practical Accountant 8 (March-April 1975): 63-64.

Weiss, Harold. "Computer Security an Overview." Datamation 20 (January 1974): 42-47.