James P. Anderson Co.
Box 42 Fort Washington, Pa. 19034
215 646-4706

COMPUTER SECURITY THREAT

MONITORING AND SURVEILLANCE

CONTRACT 79F296400

February 26, 1980

Revised:

April 15, 1980

Consultants in Computer Technology

TABLE OF CONTENTS

Computer Security Threat Monitoring and Surveillance

February 26, 1980 - Revised: April 15, 1980

1.1 Introduction

This is the final report of a study, the purpose of which was to improve the computer security auditing and surveillance capability of the customer's systems.

1.2 Background

Audit trails are taken by the customer on a relatively long term (weekly or monthly) basis. This data is accumulated in conjunction with normal systems accounting programs. The audit data is derived from SMF records collected daily from all machines in the main and Special Center. The data is temporarily consolidated into a single file ("dump" data set) from which the various summary accounting and audit trail reports are produced. After the various reports are generated, the entire daily collection of data is transferred to tape. Several years of raw accounting data from all systems are kept in this medium.

Audit trail data is distributed to a variety of individuals for review; a DAC for GIMS applications, activity security officers for some applications located under their purview, but the majority to the customers data processing personnel! For the most part the users and sponsors of a data base or an application are not the recipients of security audit trail data.

Security audit trails can play an important role in the security program for a computer system. As they are presently structured, they are useful primarily in detecting unauthorized access to files. The currently collected customer audit trails are designed to detect unauthorized access to a dataset by user identifiers. However, it is evident that such audit trails are not complete. Users (particularly ODP "personnel" with direct programming access to datasets) may operate at a level of control that bypasses the application level auditing and access controls. In other systems, particularly data management systems, the normal mode of access is expected to be interactive. Programmers with the ability to use access method primitives can frequently access database files directly without leaving any trace in the application access control and audit logs. Under the circumstances, such audit trail concepts can do little more than attempt to detect frontal attacks on some system resource.

Security audit trails can play an important role in a security program for a computer system. As audit trails are presently structured on most machines, they are only useful primarily in detecting unauthorized access to files. For those computers which have no access control mechanisms built into the primary operating systems, the audit trail bears the burden of detecting unauthorized access to system resources. As access control mechanisms are installed in the operating systems, the need for security audit trail data will be even greater; it will not only be able to record attempted unauthorized access, but will be virtually the only method by which user actions which are authorized but excessive can be detected.

1.3 Summary

In computer installations in general, security audit trails, if taken, are rarely complete and almost never geared to the needs of the security officers whose responsibility it is to protect ADP assets. The balance of this report outlines the considerations and general design of a system which provides an initial set of tools to computer system security officers for use in their jobs. The discussion does not suggest the elimination of any existing security audit data collection and distribution. Rather it suggests augmenting any such schemes with information for the security personnel directly involved.

2. Threats

2.1 Scope

In order to design a security monitoring surveillance system, it is necessary to understand the types of threats and attacks that can be mounted against a computer system, and how these threats may manifest themselves in audit data. It is also important to understand the threats and their sources from the viewpoint of identifying other data. It is also important to understand the threats and their sources from the viewpoint of identifying other data sources by which the threat may be recognized.

To assist the reader, the following definitions are used in this paper:

Threat:

The potential possibility of a deliberate unauthorized attempt to:

a) access information

b) manipulate information

c) render a system unreliable or unusable

Risk:

Accidental and unpredictable exposure of information, or violation of operations integrity due to malfunction of hardware or incomplete or incorrect software design.

Vulnerability:

A known or suspected fl$\overset{a}{o}$w in the hardware or software design or operation of a system that exposes the system to penetration of its information to accidental disclosure.

[illegible]

Attack:

A specific formulation or execution of a plan to carry out a threat.

Penetration:

A successful attack; the ability to obtain unauthorized (undetected) access to files and programs or the control state of a computer system.

In considering the threat problem, the principal breakdown of threats is on the basis of whether or not an attacker is normally authorized to use the computer system, and whether or not a user of the computer system is authorized to use a particular resource in the system. The cases of interest are shown in Figure 1.

Another view of the representation of threats is shown in Figure 2. This representation shows the protected resources surrounded by rings of control and rings of "users". In some ways this representation is more useful for purposes of identifying where and what kind of audit data might be of use in detecting the exercise of one of the threats shown.

2.2 Gaining Access to the System - External Penetration

In the context of this report, the term "external penetration" is not confined to the usual case of an outsider attempting to gain access to a computer resource in an organization of which he is not a part. The term is meant to convey, in addition to the previous case, the notion of an employee of the organization who has physical access to the building housing the computer system but who is not an authorized computer user. These cases are of general and specific interest in that they represent in some ways the extremes of the problem of gaining access to a computer.

The true outsider has the most difficult task in some ways, if the only means (terminals, RJE stations, etc.) of accessing a computer are physically co-located with the computer in the same buildings. Where access to computer resources is granted through wire communications, the external penetrator has a substantially easier task in attempting to gain physical access. For those systems and networks

	Penetrator Not Authorized to Use Data/Program Resource	Penetrator Authorized to Use Data/Program Resource
Penetrator Not Authorized Use of Computer	Case A: External Penetration	╳
Penetrator Authorized Use of Computer	Case B: Internal Penetration	Case C: Misfeasance

FIGURE 1

General Cases of Threats

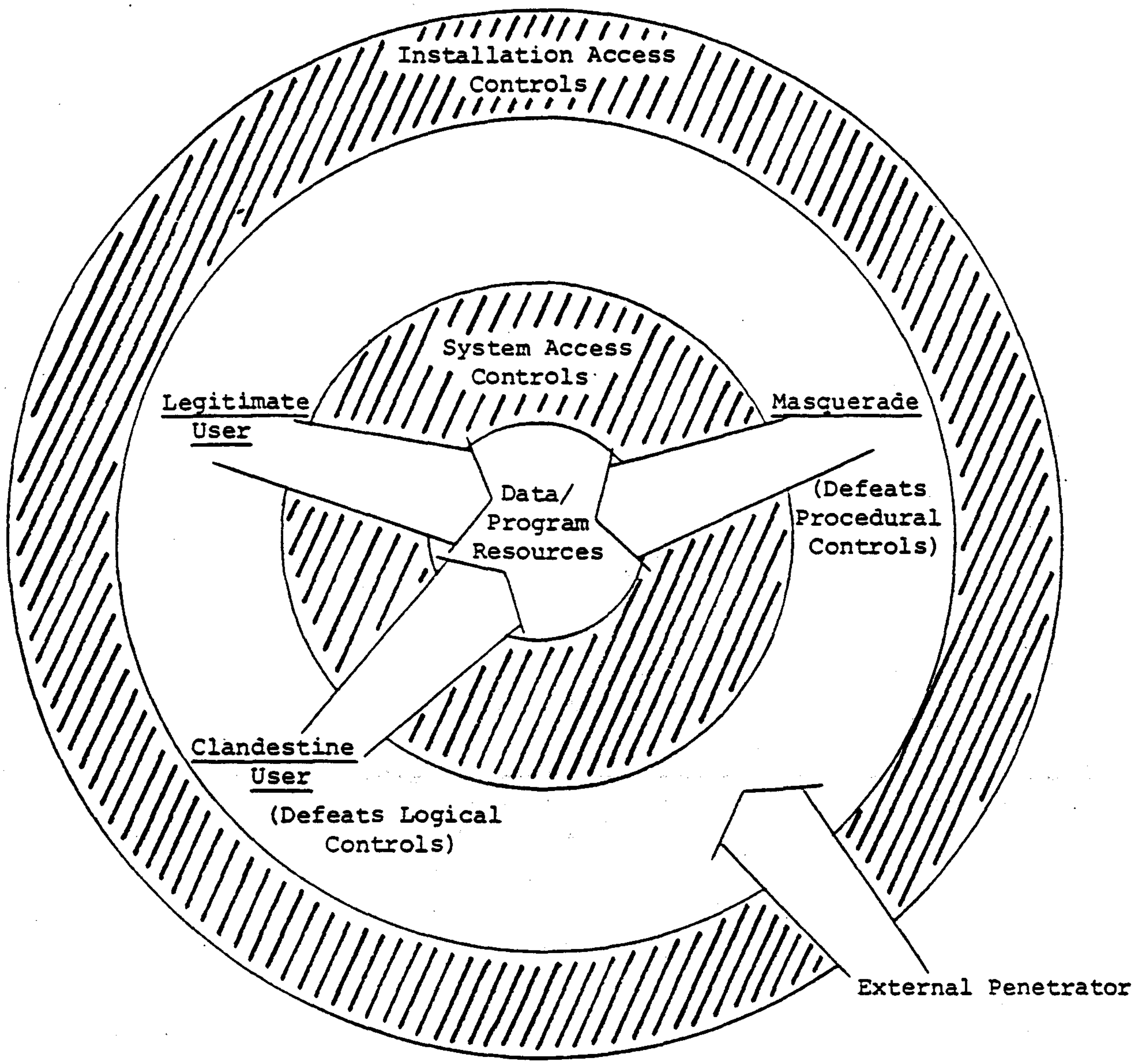


FIGURE 2

Threat Representations

has merely to wire tap a communication line to effectively gain use of the targeted system.

The individual with physical access to the building housing the computer systems or its terminals does not have to resort to such exotic methods. However, it may be more difficult for such an individual to gain access to use the system without attracting attention. Whether or not this is true in any specific instance is in part a function of how mature the insolation is and in particular, whether or not there are many terminals for use of the computer resources.

In the case of the user with physical access to the building housing the computer systems, there is a possibility of additional information that may be useful to correlate for security purposes. As an example, in those buildings that employ security logging or building access systems that record the time and point of entry and exit of all individuals, it would be possible for detected security incidents to be correlated with individuals who could conceivably be involved in the incidents.

In case of unprotected communication lines, there is opportunity for individuals to attempt to gain use of computer systems by trail and error attempts at logging on. Records of the log on attempts if collected, would provide security officers with a substantial warning of unauthorized activity, and identification of at least the location from which unauthorized access is being attempted.

In most systems such data is not collected. This is because the systems are generally large with a large number of users, and recording the presumed attempted logons would consume too many system resources to warrant their acquisition.

In addition there is a potential problem created by recording in the audit data unsuccessful logons if those logons contain the password or other user authenticator. The danger is that the audit trail will contain partial or complete user authenticators or passwords from legitimate errors made by authorized users as well as the unsuccessful external penetration attempts. This is not to say such data should not be collected, it is only to point out that in the collection it is possible that a greater danger is created.

Auditing of attempted logons can include identification of the terminal, the port through which the terminal is connected to the system, and the claimed identity of the user and the like. If the assets required it, it would be possible to trigger an immediate exception report to the security officer or other operations personnel if the number of unsuccessful longons from a given port number exceeded some threshold over time. The cost of this idea is the additional complication of maintaining logon records or even extracts from logon records on a per-port basis when the number of ports or the number of potential users of the system is extremely large. Note that the external penetrator threat translates into an internal threat as soon as the installation access controls have been penetrated.

2.3 Internal Penetration

In many installations, the internal penetration is more frequent than external penetrations. This is true for a variety of reasons, not the least of which is the internal penetrator has overcome a major barrier to unauthorized access; that is, the ability to gain use of a machine. Again for the purpose of identifying possible means of detection through audit trails, three classes of users can be identified. These are:

a. The masquerader

b. The legitimate user

c. The clandestine user

The user classes are shown in an order of increasing difficulty in detecting their activity through audit trail data. The ability, to detect activity of each category of user from audit data varies, in some cases considerably; hence the breakdown.

2.3.1 The Masquerader

As indicated in the diagram, the masquerader is an internal user by definition. He can be any category of individual; either an external penetrator who has succeeded in penetrating the installation access controls, or an employee without full access to a computer system, or possibly an employee with full access to a computer system who wishes to exploit another legitimate users identification and password that he may have obtained.

This case is interesting because there is no particular feature to distinguish the masquerader from the legitimate user. Indeed, with possession of the proper user identifier and password, he is a legitimate user as far as the computer system is concerned. Masquerade is interesting in that it is by definition an "extra" use of a system by the unauthorized user. As such it should be possible to detect instances of such use by analysis of audit trail records to determine:

a. Use outside of normal time

b. Abnormal frequency of use

c. Abnormal volume of data reference

d. Abnormal patterns of reference to programs or data

As will be discussed in the subsequent section, the operative word is "abnormal" which implies that there is some notion of what "normal" is for a given user.

In attempting to detect masquerade, a surveillance system focuses on the legitimate user as the resource being "protected". In other types of surveillance the resource being protected may be other elements of the system such as devices, specific files and databases or programs and the like.

Quite obviously the masquerader can have as his intent any of the various stated purposes of penetration. Again, since his use of a system will be extra, that is in addition to normal use by a user of the same user number, this extra use can or should be detectable.

2.3.2 Legitimate User

The legitimate user as a threat to information resources is a case of misfeasance in that it involves the misuse of authorized access both to the system and to its data. Since the user is authorized to use the system, the audit trail records would not be expected to exhibit any abnormal patterns of reference, logon times and so forth. It is for this reason that the degree of difficulty in detecting "abnormal" use by a legitmate user of a system is more difficult than the preceding case. There maybe no "extra" use of resources that can be of help in detecting the activity.

It must be recognized that small amounts of misuse of authorized access would not be detected under any circumstance. As an instance, if the authorized user misuses his authority slighty, to print Snoopy calendars or to extract two extra records of data that he is otherwise authorized to use, a statistically satisfactory method of detecting such minor abnormalities is probably not feasible.

If the legitimate user makes use of his authorized access to refer to or gain access to information that is normally <u>not</u> authorized in the conduct of his job, the audit trail should be able to reflect this. Similarly, if the authorized user misuses his access to gain large amounts of information by transferring many records or use an "excessive" amount of computer time, this too should be detectable. Initially, it may not be possible to detect a difference between a case of misfeasance and a masquerade. In general, it would be expected that the masquerade would show up as an anomaly in the time of use of a system whereas misfeasance would show up by one or more of the parameters total time used, or data transferred exceeding previously established norms.

2.3.3 Clandestine User

The clandestine user is quite possibly the most difficult to detect by normal audit trail methods. The assumption regarding clandestine users is that the user has or can seize supervisory control of the machine and as such can either operate below the level at which audit trail data is taken or can use privileges or system primitives to evade audit trail data being recorded for him. As far as most audit trail information is concerned, the clandestine user is "the little man who isn't there". There is nothing that can be done to detect this type of user unless he activates his clandestine operations in a masquerade or as misfeasance of a legitmate user that may then create individual records that show up under those categories of use.

The clandestine user who effects a technical penetration to obtain control of the most privileged state the computer system, is not capable of being audited. Where the threat of such penetrations is considered high it would be possible to augment the internal auditing mechanisms of the individual computer with external measurements of busy or idle states of the CPU, the memory, secondary storage and so forth, and from this additional data possibly (a very weak possibly) detect "pure" phantom use.

2.3.4 Clandestine User Countermeasures

The penetration issue is one which can be played measure - countermeasure through what appears to be endless variations. What is really at the heart of the difficulty of "defense" is the fact that the penetrator has a myriad of places to effect operating system changes that permit penetration. At a high level of sophisitcation, the penetrator could temporarily alter the operating system to suppress audit recording of what he's doing. Depending on a number of factors, this is virtually impossible to detect purely by analysis of the internal audit records. It involves in looking for what isn't present. However, if the operating system changes for the penetration are only temporary, the changes could be detected, if the operating system code is continuously compared in some fashion with a reference version.

The security audit data is dependent to a large extent on the integrity of the origins of the audit trail records. The audit trails are a centralized recording of information originally designed to support billing and other accounting functions. To support security surveillance, the ideal situation would be to provide independent audit trails for each major component of the machine, preferably by a micro or other computer element associated with the device or devices supporting the use of the system.

Independent audit trails for each major component or function of a machine is dervived from the experience of auditing in networks. It is clear that the suppression of audit records in a network where a number of points must be traversed through the network in order to affect the desired penetration, is virtually impossible unless one subverted every component of the network from the point of entry to the target and possibly back again. In sophisticated networks involving a transport layer, one or more systems as access systems' and then server hosts, total control of all use recording of all such affected elements would not be possible. Under any circumstance, the distribution of recording among a number of points in a system greatly compounds the difficulty for the penetrator. In fairness, it must be pointed out that it also compounds the work for the compilers and users of audit trail data.

3. Characterization of Computer Use

3.1 Introduction

The basic premise of this study is that it is possible to characterize the use of a computer system by observing the various parameters available through audit trails, and to establish from these observations, "normal" ranges for the various values making up the characterizations.

3.2 The Unit of Computer Work - The Job or Session

Considering the problem of characterizing use of a computer the first issue that must be faced is what unit or units should be used to represent how a computer is used. It appears that the most natural unit of computer use is the notion of job in batch running or session in interactive working. Both of these terms denote a continuous unit or a single unit of use of a computer with a well defined beginning and a well defined end. The parameters that distinguish one unit from another are the user identifiers on whose behalf they are operated and the list of the program and (where available) data files entering into the program.

It should be noted that if the resource being monitored is the file or device that the notion of job or session as the principal parameter of characterization may not make much sense. In these instances, a list of references by user identifier or program (if such information is available) is the principal parameters of characterization of such use.

3.3 Time Parameters

There are basically 2 time parameters of interest that characterize how a system is used for a particular job. The first of these is the time of day (and in a larger sense the day of the week) that a particular job or session is operated. For many jobs this time of use is fixed within a fairly narrow range.

The second time parameter is the the duration of length of time the job takes. While the fact that most modern systems are multi programmed and the elapsed real time for a job will vary accordingly, it is still a measure that one would ordinarily expect to have relatively little variability.

The time of day of the job initiation is one of the few use parameters with multiple values. Depending on the kind of user being characterized, the time of initiation of a particular task or job will vary, perhaps substantially. This is especially true in the case of interactive working where the choice of when to do a particular kind of task is totally up to the user under observation.

While system usage patterns can exhibit wide fluctuations from one user to another, it is expected that individual users establish patterns to their use of a system. It is these patterns that will be disturbed by masquerades.

Further, it should be evident that the ability to discriminate a particular indicator is a function of how widely the individuals own pattern of use fluctuates from day-to-day, and week-to-week.

This is well illustrated by the example given below where the ability to detect use of a resource outside of 'normal' time cannot be achieved if 'normal' time can be any hour of the day, any day of the week.

Detection of, outside of normal times of use is relatively straightforward. Individual jobs (sessions, job steps, etc.) are sorted on time of initiation and compared with previously recorded data for the specific user.

The basic question to be faced is the granularity of the analysis needed to detect 'out of time' use of a resource. For users exhibiting little variability in their use of a system, a gross measure, such as number of jobs (sessions, etc.), per quarter of the day (0000 - 0559, 0600 - 1159, ... etc.) will be sufficient to discover second or third shift use of a system under the name of the subject under observation.

For another class of user, with considerable variability in time of use, it may be necessary to record usage by the hour. Obviously, if the 'normal' use is every hour of the day, the 'outside of normal time' condition is not detectable. One would have to examine such users further to determine whether the normal use extends seven days a week, on holidays, through vacations, etc. Conceivably, 'normal' usage could extend through all of these periods. Then, the 'out of normal time' condition would not be a useful discriminant for that user.

Figure 2 shows the number of logons per hour for two different days (approximately 20 days apart) for a number of different users. Users I, II, and IV exhibit consistent patterns of logon, while users III and V exhibit more variability (in these two samples).

		Hour of Logon (GMT)																								
User		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
I	A														1											
	B														1											
II	A																				1	2	2			
	B																	1	1		3	2	3	1		
III	A		2																4	4						
	B			2										8								1	1	1		
IV	A		1			5			1			2	3		1			1	4		1			1		
	B		1			5			1			3	3	3	1		1	1	4		1			1		
V	A	3							2				4		3						1	2				2
	B	5	3										1		3					1		2	3			

Figure 3. Number of Logons/Hr.

If (for purposes of illustration) we assume that the 'A' data is the average (or cumulative) experience with the user in question, the variability in time of use could be scored by summing the squares of absolute values of the difference, i.e.,

$$\text{score} = \sum_{i=1}^{24} \left| (A_i - B_i) \right|^2$$

While not a particularly elegant measure, it does show for the several users represented, those whose logon pattern exhibit greatest variability, which might be the result of masquerade. Depending on other measures, those users might then become subjects of additional investigations.

The time of use abnormality scores for the five samples are:

User	Score
I	0
II	8
III	107
IV	11
V	41

Depending on where the cutoff point is set for reporting, one would expect to see 'III' and 'V' reported as being out of range.

In addition to the elapsed real time for a particular problem, we can measure the actual computer time used on a particular problem. This measure should not vary substantially, but a heavy system load which causes programs to be swapped in and out frequently can increase the elapsed running time for the problem. The increase should not be significant unless there is some other reason.

3.4 Dataset and Program Usage

The parameters that can be measured in this area varies significantly from one system to another. In some cases it is possible to identify the number of records read and written to a particular dataset or file while in another case on another system, the only data reference information that would be available would be a total number of pages transferred between a file system to a processor, with no indication being given whether those pages were read or written. These differences are a result of the fact that the audit data is taken for accounting purposes rather than security purposes, and as a consequence the kind of information that's collected is driven by accounting interests rather than what one would prefer for security purposes.

With regard to program usage the principal concern as far as security audit goes is whether or not a program was referred to for execution purposes or whether it is being read and written as data. This is significant for a security viewpoint because of the fact of reading and writing of programs as data is almost certainly a clue of penetration activity as opposed to normal system use. It must be understood that the reading and writing programs as data does not mean the results of compilation. Thus the principle data parameter for programs or data files is the number of records read or written.

3.5 Monitoring Files and Devices

The preceding discussion focused on the monitoring of a particular user identifier through the range of actions that the user identifier is allowed to do include submitting jobs, use of system and so forth. It is indeed the monitoring of system users that is the focus of the preceding kinds of surveillance and monitoring techniques. When one shifts the attention to monitoring a particular file or correspondingly a device, the kind of information collected, how it is collected and how it is used differs.

3.6 Group Statistics

While one could attempt to detect abnormal values of parameters against all of the job records for a single user, it is believed that better measures and better security can be obtained by grouping the job records into sets having the property that each job or session refers to the same set of files; that is, an identical set of files.

The presumption is that the session or job referring to the same file sets can be considered to belong to the same population and will exhibit similar statistical properties from run to run. An arbitrary deviation of the norm for the user is a criterion for reporting a particular use and generating an "abnormal volume of data" or an "abnormal (measure of one of the parameters discussed above) exception". With no other data available, if the observed statistic for a parameter is more than plus or minus 2.58 standard deviations from the mean, it is out in the five percent range and probably is worthy of examination.

The abnormal patterns of reference are determined simply by discovery of file references that have not been previously encountered. If the files referenced in a particular job are not identical to a set previously seen, this should be reported as a new event. In the section on the organization of a surveillance system, some of these comments are illustrated with the results of a model system.

4. Structure of a Surveillance System

4.1 Introduction

This section outlines the functional components of a security monitoring and surveillance system. It identifies the key programs that will be required and considers a number of alternatives in implementing such a design. Figure 4 is a diagram of the central function of a surveillance system. It shows elements for the automatic generation of security exception reports.

4.1.1 Monitoring of Users

The diagram, Figure 4, shows the major steps involved in producing the monitoring and surveillance system data files. The first step is the selection of audit records affecting the element or elements being audited. This step is included in the overall design on the premise that the ability to keep history records for a large number of users will be storage limited. The second reason for including this is the premise that most use of a system is benign and proper and that for large populations, the bulk of the population is not of interest to the security personnel at any one time. In practice, a security office may have 50-100 "cases" in which they are interested. Some of these cases may be merely random selections from the total user population to be audited for a period of time, not with the intent of finding any wrong-doing, but with the intent of determining any possible wrong-doing.

4.1.2 Sorting Audit Records

The audit records selected in the previous step are then sorted on, a user identifier, and then within that, job identifier, date, time

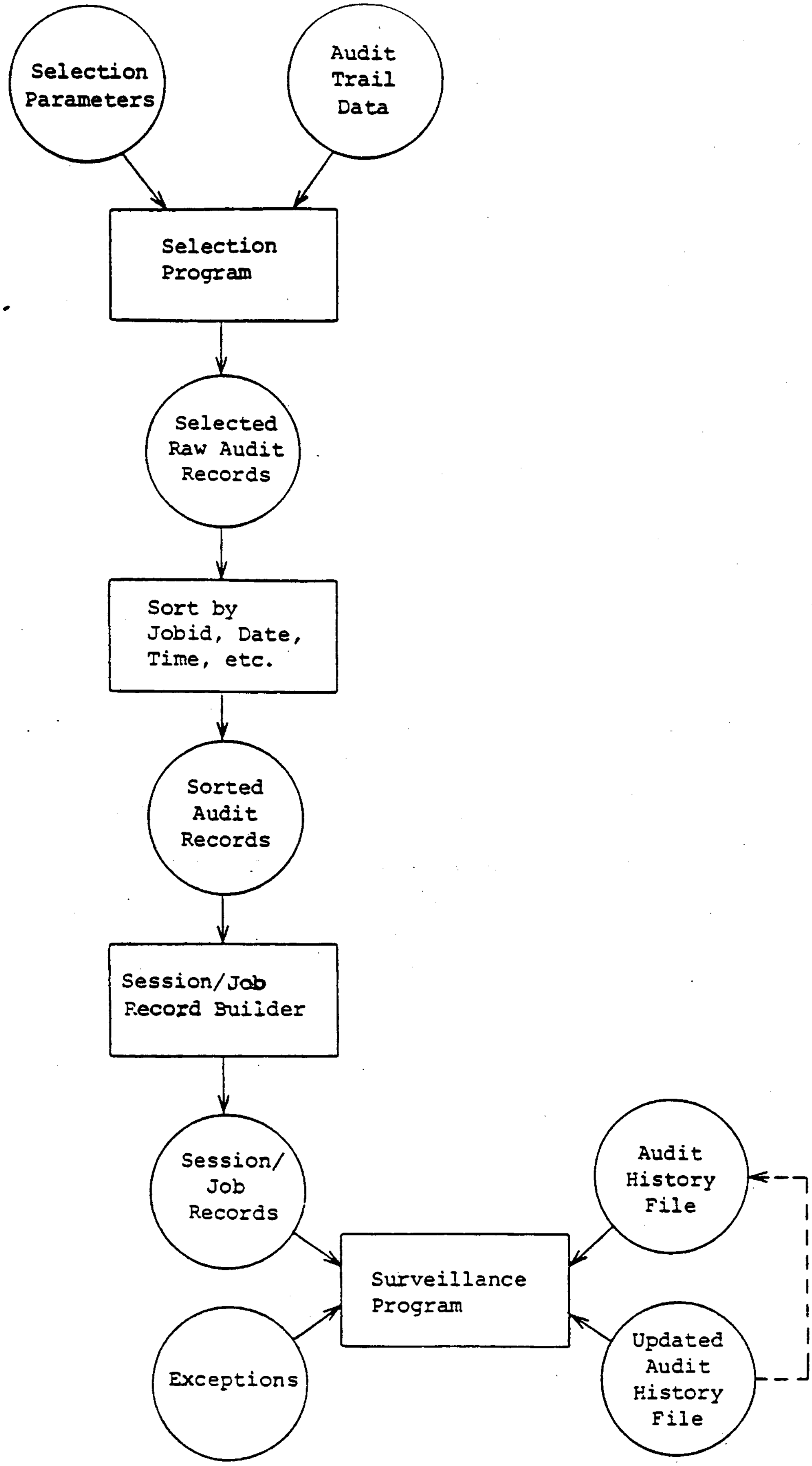


FIGURE 4

SURVEILLANCE SYSTEM

and so forth. The purpose of the sort is to collect together all records constituting a job. In most audit systems the job is represented by a number of audit records; job initiation, job termination, job execution, etc. The information of interest may be distributed over all the different kinds of records. The output of the sort is used as input to a program that builds session records.

4.1.3 Session Record Builder

Whether or not a session record builder is required, is a function of the type of audit data that is collected and possibly the type of system being employed. The model constructed as part of the project to determine the feasibility and the difficulty of doing surveillance of this type was based on a time sharing which provided a variety of records that required processing of all the records for a particular session in order to determine how much input and output had occurred. Other systems accumulate this information and make it available as part of a record identifying the termination of a job or program or as part of a program summary. The need for this step is a function of the underlying audit recording system for which it is built.

4.1.4 Surveillance Program

In some respects this is the heart of the system in that to performs a variety of functions. In the prototype or model system, the surveillance system performs the following functions:

It accumulates all instances of the same kind of job where job is defined in this case as having same program and file reference set involved (see 3.6). As it considers each job (or session) it compares the parameters measured on a session; that is the connect time, the number of input - output characters, the numbers of file references, etc., against a set of absolute limits. The absolute limits were arbitrarily chosen by taking statistics over a large member of users and setting the limits such that it would cause an exception report if an individual session was unusual in and by itself.

In addition to the absolute limits, an individual session record is subject also to the distribution test. Distribution tests are those elements that are single values treated as samples, compared against distribution represented by the mean and the standard deviations of those means. If any of the parameters measured are greater than 2.58 standard deviations from the mean in either direction, the session record is reported as an exception. After these two operations are performed the session record is accumulated with all others like it and statistics for the set are available. Nothing is done with these statistics in prototype program. However, a similar measure could be employed to say how does the mean of all of the individual runs for this day compare with the accumulated mean, etc. Finally the history master record is updated with the session summary data and the process repeated for the next set of session records.

In order to minimize file passes, the surveillance program recognizes when a master record has not been updated in fifteen days. This is an arbitrary time period established for the model program that is used to keep the history file at a reasonable size. In the event it finds such a record that has not been updated in fifteen days, it is removed from the history records, and reported as a record dropped for lack of activity.

Obviously with the records being dropped and added the other consideration is that a previous history record does not exist for a particular user. In this case, new master records are created and inserted in the correct place. No statistical reporting or distribution tests are performed in this case, but the absolute limits tests are recorded. In order to provide the security officer with some notion of what is going on, an exception report item is created for the session summary records that indicates that a new history master record is being created, and the new master record is available for display as part of the exception reporting.

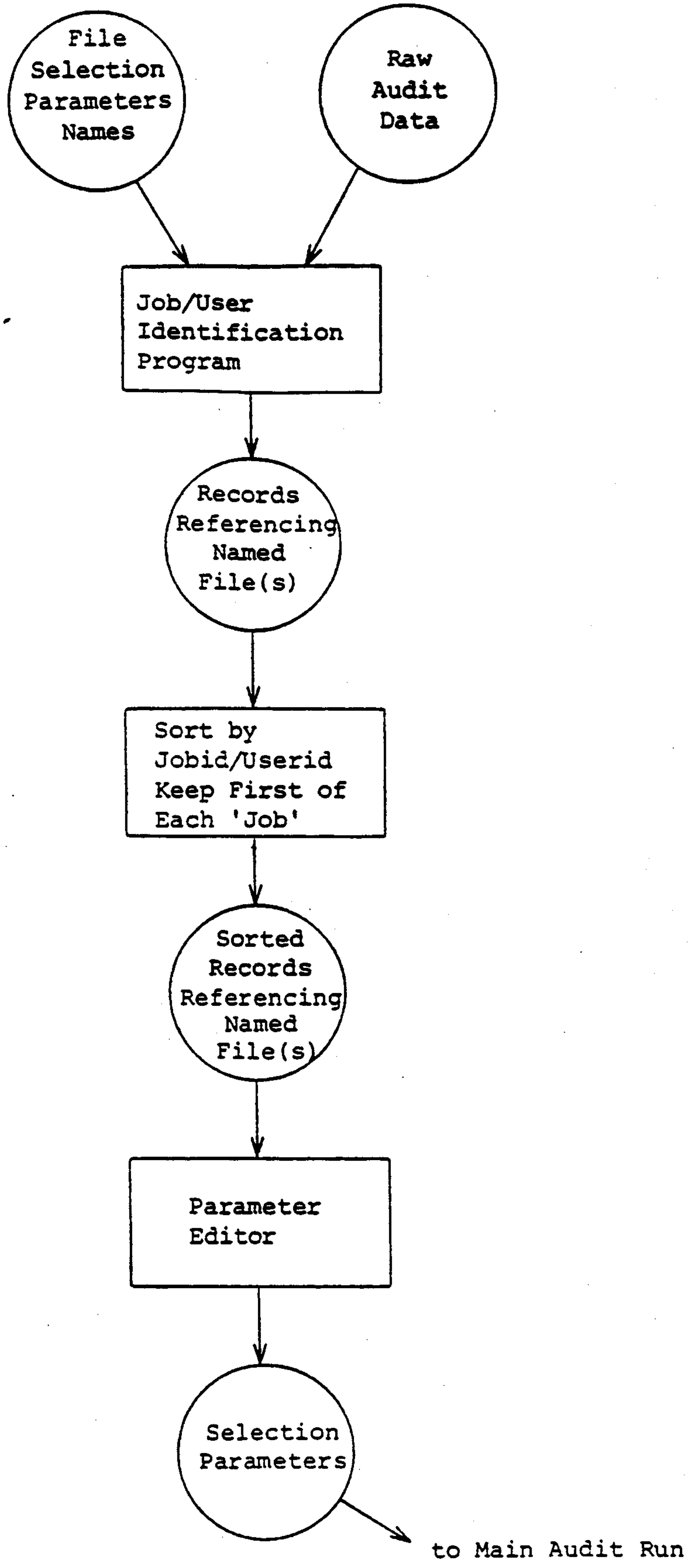

File Selection Parameters Names
Raw Audit Data
Job/User Identification Program
Records Referencing Named File(s)
Sort by Jobid/Userid Keep First of Each 'Job'
Sorted Records Referencing Named File(s)
Parameter Editor
Selection Parameters
to Main Audit Run

FIGURE 5

Processing to Audit File Use

The entire sequence outlined above of selecting records of interest sorting them creating session summaries, updating master and the like and adding to the exception report is run once a day at the time the accounting files are turned over. The exception records are accumulated until such time as the reports are actually prepared. A sample of the reports from the model system are shown in figures 6,7,8 and 9.

4.2 Monitoring Files

Producing the records necessary to monitor use of files or other objects in a system is similar to that outlined above for monitoring users activities in a system. The principal difference is that fact that the element being sorted is the 'file', and the records being kept are on a per user basis. In some ways the files are a little more complicated than the users activities files in that multiple accesses to the same file in three or four different runs are to be treated in some sense differently, particularly in terms of the amount of data read from or written to the target file.

The file or device monitoring may require more than one pass of the audit file in order to collect the necessary information. As an example, if one wanted to record against a particular file, the users identifier and the session statistics associated with that reference to that file, it may be necessary to first pass the audit data file looking for those user identifiers or other session identifiers that are associated with its reference, make a list of those and then on a second pass of the audit data file collect the session records necessary to produce session summary statistics to be recorded against the file name. An example process flow is shown, Figure 5. Quite obviously these procedures vary

FIGURE 6

```
                         TRANSACTION RECORDS
                           EXCEPTION REPORT
                                               Ø3/Ø9/8Ø
                                                   CT TC  DS PS  CR
                LOGON   CNCT S                     >  H>  U> U>  U>
UNO      DATE   TIME    TIME T CRU  TCH   PSU DSU  1K 1ØK 1K 1ØK 5ØØ
-------------------------------------------------------------------
ICN6131Ø!8ØØ3Ø3!2218:43!852Ø! ! 241!Ø7414!  Ø!1519!X !X  !X !   !
ICN61999!8ØØ3Ø3!ØØØØ:Ø7! 183!I! 242!    Ø!  Ø!1167!  !   !X !   !
ICN61999!8ØØ3Ø3!Ø1ØØ:42!  57!I! 142!    Ø! 28!1332!  !   !X !   !
ICN61999!8ØØ3Ø3!Ø212:14! 336! !  48! 6859!2ØØ! 161!  !   !  !   !
IIAØØ914!8ØØ3Ø3!1648:56!127Ø! ! 241!45975! 36! 114!X !X  !  !   !
IIAØØ914!8ØØ3Ø3!1956:59!64Ø6! ! 197!38752! 18! 462!X !X  !  !   !
IPO14ØØØ!8ØØ3Ø3!18Ø4:41!2269! !  57!13986!  Ø!   3!X !X  !  !   !
ICN61777!8ØØ3Ø4!213Ø:14!1Ø58! !  54! 1874!155! 241!X !   !  !   !
ICN61999!8ØØ3Ø4!ØØØØ:Ø3! 156!I! 648!    Ø!  Ø!3Ø39!  !   !X !   !X
ICN61999!8ØØ3Ø4!Ø1ØØ:28! 158!I! 342!    Ø! 28!3325!  !   !X !   !
IIAØØ99Ø!8ØØ3Ø4!1323:26!5357! !  1Ø! 7Ø77!  4!   7!X !   !  !   !
IIAØØ99Ø!8ØØ3Ø4!16Ø6:Ø9!1496! !  16! 41Ø3!  3!   5!X !   !  !   !
IIAØØ99Ø!8ØØ3Ø4!17Ø7:46!1997! !   4!  564!  3!   5!X !   !  !   !
ICN61778!8ØØ3Ø5!2Ø15:32!1485! ! 393! 1674!  3!5Ø14!X !   !X !   !
ICN61778!8ØØ3Ø5!2Ø42:51!1159! ! 136!  596!  3!1692!X !   !X !   !
ICN61778!8ØØ3Ø5!21Ø2:48!3421! ! 742! 2573!  3!9589!X !   !X !   !X
ICN61778!8ØØ3Ø5!1947:5Ø! 9Ø1! !  84!  754!  3!1ØØ9!  !   !X !   !
ICN61999!8ØØ3Ø5!Ø113:39! 254!I! 69Ø!    Ø!  8!3146!  !   !X !   !X
ICN61999!8ØØ3Ø5!214Ø:Ø6! 384!I! 38Ø!    Ø!  8!2ØØ6!  !   !X !   !
ICN61999!8ØØ3Ø5!2215:Ø2! 277!I! 219!    Ø!  8!114Ø!  !   !X !   !
ICN61999!8ØØ3Ø5!223Ø:Ø7! 147!I! 2Ø7!    Ø!  8!1Ø77!  !   !X !   !
ICN61999!8ØØ3Ø5!2245:Ø3! 19Ø!I! 213!    Ø!  8!1128!  !   !X !   !
ICN61999!8ØØ3Ø5!2134:23! 267!I! 268!    Ø!  8!1482!  !   !X !   !
ICN61999!8ØØ3Ø5!22ØØ:22! 142!I! 19Ø!    Ø!  8!1Ø37!  !   !X !   !
ICN61999!8ØØ3Ø5!Ø1ØØ:48! 269!I! 357!    Ø! 28!3343!  !   !X !   !
ICN61999!8ØØ3Ø5!2127:52!1845! !  55! 7647! 78!  26!X !   !  !   !
ICN61999!8ØØ3Ø5!Ø112:44! 931! !  71!19713!2Ø5! 198!  !X  !  !   !
IIAØØ914!8ØØ3Ø5!1418:43!33Ø1! !1474!1259Ø!2Ø3!2527!X !X  !X !   !X
IIAØØ99Ø!8ØØ3Ø5!1355:29!2327! !   4!  8Ø9!  3!   4!X !   !  !   !
IIAØØ99Ø!8ØØ3Ø5!212Ø:Ø5!1114! !   7! 4941!  3!  13!X !   !  !   !
IIAØØ99Ø!8ØØ3Ø5!1335:53! 86Ø! ! 1Ø8!12351!  3!  13!  !X  !  !   !
IFS993ØØ!8ØØ3Ø5!1957:Ø2!1482! !   9! 85Ø7!  Ø!   Ø!X !   !  !   !
ICN61778!8ØØ3Ø6!2Ø24:11!112Ø! ! 329! 1519!  3!4245!X !   !X !   !
ICN61999!8ØØ3Ø6!Ø44Ø:47! 2Ø3!I! 7Ø1!    Ø!  8!323Ø!  !   !X !   !X
ICN61999!8ØØ3Ø6!Ø1ØØ:46! 498!I! 393!    Ø! 28!3665!  !   !X !   !
ICN61999!8ØØ3Ø6!Ø437:23! 995! !  58!126Ø6!158!  64!  !X  !  !   !
IIAØØ99Ø!8ØØ3Ø6!1545:45!874Ø! !  83!12843! 12! 245!X !X  !  !   !
IIAØØ99Ø!8ØØ3Ø6!2119:12!1183! !   4! 1664!  3!   6!X !   !  !   !
ICN61999!8ØØ3Ø7!2352:13!5771! !  3Ø! 447Ø!  2!   Ø!X !   !  !   !
ICN61999!8ØØ3Ø7!ØØØ2:44! 28Ø!I! 61Ø!    Ø!  8!2795!  !   !X !   !X
ICN61999!8ØØ3Ø7!2359:19!62Ø6!I! 43Ø!    Ø!  Ø!2ØØ1!X !   !X !   !
ICN61999!8ØØ3Ø7!Ø1ØØ:37! 371!I! 314!    Ø! 28!3Ø91!  !   !X !   !
ICN61999!8ØØ3Ø7!Ø128:42! 388! ! 384! 66Ø8! 29!218Ø!  !   !X !   !
ICN61999!8ØØ3Ø7!Ø1ØØ:39!15Ø3! ! 438!26961!831! 189!X !X  !  !X  !
IIAØØ914!8ØØ3Ø7!1533:44!1621! !  13! 3188!  2!   Ø!X !   !  !   !
IIAØØ914!8ØØ3Ø7!1445:1Ø!1Ø21! !  11! 6743!  5!   Ø!X !   !  !   !
IIAØØ99Ø!8ØØ3Ø7!16Ø5:Ø3!1354! !  16!11394!  3!  46!X !X  !  !   !
ICN61999!8ØØ3Ø8!ØØØØ:Ø5! 199!I! 643!    Ø!  8!2947!  !   !X !   !X
ICN61999!8ØØ3Ø8!Ø1ØØ:3Ø! 16Ø!I! 346!    Ø! 28!3268!  !   !X !   !
ICN61999!8ØØ31Ø!ØØØØ:Ø6!  51!I! 2Ø7!    Ø!  8!1ØØ1!  !   !X !   !
ICN61999!8ØØ3Ø9!2354:38!  49!I! 221!    Ø!  8!1Ø48!  !   !X !   !
ICN61999!8ØØ3Ø9!Ø1ØØ:3Ø!  56!I! 114!    Ø! 28!113Ø!  !   !X !   !
IIAØØ99Ø!8ØØ3Ø9!1422:39!2Ø57! !   7! 5247!  2!   4!X !   !  !   !
```

```
 H I S T O R Y   R E C O R D S   D R O P P E D
  F O R   L A C K   O F   A C T I V I T Y
                                                     03/09/80
             TO TO
      LAST   TU TS
      UPDATE PD ES   FILES
-----!------!--!--!----------------------------------------
61999!800222! 1! 1!CN61999  BB3XSUB3
-----!------!--!--!----------------------------------------
61999!800222! 1! 1!CN61333  CN61IR   CN61333  LOGGER   CN61500  LASTLOG
     !      !  !  !CN61500  LIST     CN61999  DES      CN61999  NUXXPRNT
     !      !  !  !CN61999  VALCUREV PARL     IR
-----!------!--!--!----------------------------------------
61999!800222! 1! 1!DLYSESLK          SURVEIL4 CN61333  CN61IR
     !      !  !  !CN61333  LOGGER   CN61500  LASTLOG  CN61500  LIST
     !      !  !  !CN61999  DES      CN61999  DLYSESLK CN61999  HISTLIS2
     !      !  !  !CN61999  HISTLIST CN61999  IND      CN61999  REPTCNTL
     !      !  !  !CN61999  XCPPRNT4 CN61999  XCPPRNT6 CN61999  XCPPRNT8
     !      !  !  !CN61999  XCPPRNT9 PARL     IR
-----!------!--!--!----------------------------------------
61999!800223! 1! 1!CN61999  DLYSESLK
-----!------!--!--!----------------------------------------
61999!800223! 1! 1!HISTLIST CN61500  RESP     CN61999  DES
     !      !  !  !CN61999  HISTLIST CN61999  IND
-----!------!--!--!----------------------------------------
61999!800223! 1! 1!HISTLIST          SURVEIL4 CN61333  CN61IR
     !      !  !  !CN61333  LOGGER   CN61500  LASTLOG  CN61500  LIST
     !      !  !  !CN61500  RESP     CN61999  DES      CN61999  HISTLIST
     !      !  !  !CN61999  IND      CN61999  SURVEIL4 PARL     IR
-----!------!--!--!----------------------------------------
```

FIGURE 7

FIGURE 8

```
                    SESSION       RECORDS
                   EXCEPTION   REPORT
                                          Ø3/Ø9/8Ø
                                              CT TC  DS FS  CR  NEW
                TOT  CNCT TOT  TOT    TOT TOT >  H>  U> U>  U>  MAS
JNO      DATE   SES  TIME CRUS TCH    PSU DSU 1K 1ØK 1K 1ØK 5ØØ TER
-------------------------------------------------------------------
ICN6131Ø!8ØØ3Ø3!   1!852Ø! 241!Ø7414!  Ø!1519!X !X  !  !   !   !***
ICN61999!8ØØ3Ø3!   1! 3Ø7!  17! 1243! 28!   7!  !   !  !   !   !***
ICN61999!8ØØ3Ø3!   1! 28Ø!   9! 1767! 17!   7!  !   !  !   !   !***
ICN61999!8ØØ3Ø3!   1! 336!  48! 6859!2ØØ! 161!  !   !  !   !   !***
IIAØØ914!8ØØ3Ø3!   1!127Ø! 241!45975! 36! 114!X !   !  !   !   !
IIAØØ914!8ØØ3Ø3!   1!64Ø6! 197!38752! 18! 462!X !   !  !   !   !***
IPO14ØØØ!8ØØ3Ø3!   1! 936!  78! 1345! 28!   Ø!  !   !  !   !   !***
IPO14ØØØ!8ØØ3Ø3!   2! 924!   6! 1841!  Ø!   6!  !   !  !   !   !***
IPO14ØØØ!8ØØ3Ø3!   1!2269!  57!13986!  Ø!   3!X !   !  !   !   !***
IPS993ØØ!8ØØ3Ø3!   6! 8Ø7!   8!11249!  2!  25!  !   !  !   !   !***
IIAØØ99Ø!8ØØ3Ø4!  1Ø!9686!  54!15434! 31!  47!X !   !  !   !   !
IPO14ØØØ!8ØØ3Ø4!   2! 971!  18! 4739!  Ø!   8!  !   !  !   !   !***
IPS993ØØ!8ØØ3Ø4!  11!1811!  24!2141Ø!  4!  66!X !   !  !   !   !
IPS993ØØ!8ØØ3Ø4!   1!  73!   8! 5272! 18!   Ø!  !   !  !   !   !***
ICN61778!8ØØ3Ø5!   4!6966!1355! 5597! 12!73Ø4!X !   !X !   !   !
ICN61999!8ØØ3Ø5!   1!  12!   1!    Ø!  8!  14!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1! 384! 38Ø!    Ø!  8!2ØØ6!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1! 277! 219!    Ø!  8!114Ø!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1! 147! 2Ø7!    Ø!  8!1Ø77!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1! 19Ø! 213!    Ø!  8!1128!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   2! 4Ø9! 458!    Ø! 16!2519!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1! 454!  2Ø! 1153! 28!   9!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1!1845!  55! 7647! 78!  26!X !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1! 565!  34! 5215! 84!  8Ø!  !   !  !   !   !***
ICN61999!8ØØ3Ø5!   1! 931!  71!19713!2Ø5! 198!  !   !  !   !   !***
IIAØØ914!8ØØ3Ø5!   1!33Ø1!1474!1259Ø!2Ø3!2527!X !X  !  !   !   !***
IIAØØ99Ø!8ØØ3Ø5!  18!4Ø16!  64! 8123! 55!  83!X !   !  !   !   !
IIAØØ99Ø!8ØØ3Ø5!   1!1114!   7! 4941!  3!  13!  !   !  !   !   !***
IIAØØ99Ø!8ØØ3Ø5!   1!  87!   4!  723!  3!   8!  !   !  !   !   !***
IPS993ØØ!8ØØ3Ø5!  13!1735!  24!15419!  7!  62!X !   !  !   !   !***
ICN61999!8ØØ3Ø6!   1!  2Ø!   4!    Ø!  8!   Ø!  !   !  !   !   !***
ICN61999!8ØØ3Ø6!   1! 466!  26! 33Ø9! 19!   8!  !   !  !   !   !***
ICN61999!8ØØ3Ø6!   1! 995!  58!126Ø6!158!  64!  !   !  !   !   !***
IIAØØ99Ø!8ØØ3Ø6!   1!874Ø!  83!12843! 12! 245!X !   !  !   !   !***
IIAØØ99Ø!8ØØ3Ø6!  23!2215!  85!14443! 7Ø! 11Ø!X !   !  !   !   !
IIAØØ99Ø!8ØØ3Ø6!   1!1183!   4! 1664!  3!   6!  !   !  !   !   !***
IPS993ØØ!8ØØ3Ø6!   5! 285!   8! 6425!  Ø!  3Ø!  !   !  !   !   !***
ICN6131Ø!8ØØ3Ø7!   1!  84!  1Ø! 1Ø4Ø!  9!  13!  !   !  !   !   !***
ICN61999!8ØØ3Ø7!   1!5771!  3Ø! 447Ø!  2!   Ø!X !   !  !   !   !
ICN61999!8ØØ3Ø7!   1!62Ø6! 43Ø!    Ø!  Ø!2ØØ1!X !   !  !   !   !
ICN61999!8ØØ3Ø7!   1! 388! 384! 66Ø8! 29!218Ø!  !   !  !   !   !***
ICN61999!8ØØ3Ø7!   1!15Ø3! 438!26961!831! 189!X !   !  !   !   !***
```

FIGURE 9

```
               NEW MASTER RECORDS
                                                     03/09/80

      LAST   TU TS
      UPDATE PD ES FILES
--------------------------------------------------
61310!800303! 1! 1!CN61500  SIRSYST
     !--------------------------------------------
     !     CRU!PSU!DSU !TCH  !CNCT !        LOGON        !
     !TOT:241!  0!1519!07414!78520!       FREQUENCY      !
     !MAX:241!  0!1519!07414!78520!00-06:  0!12-18:   0 !
     !MIN:241!  0!1519!07414!78520!06-12:  1!18-24:   0 !
--------------------------------------------------
61999!800303! 1! 1!CN61333  CN61IR   CN61333  LOGGER   CN61500  LASTLOG
     !             !CN61500  LIST     CN61500  SEND     PARL     IR
     !--------------------------------------------
     !     CRU!PSU!DSU !TCH  !CNCT !        LOGON        !
     !TOT: 17! 28!   7! 1243!  307!       FREQUENCY      !
     !MAX: 17! 28!   7! 1243!  307!00-06:  1!12-18:   0 !
     !MIN: 17! 28!   7! 1243!  307!06-12:  0!18-24:   0 !
--------------------------------------------------
61999!800303! 1! 1!CN61333  CN61IR   CN61333  LOGGER   CN61500  LASTLOG
     !             !CN61500  LIST     CN61999  POWTWO   PARL     IR
     !--------------------------------------------
     !     CRU!PSU!DSU !TCH  !CNCT !        LOGON        !
     !TOT:  9! 17!   7! 1767!  280!       FREQUENCY      !
     !MAX:  9! 17!   7! 1767!  280!00-06:  0!12-18:   0 !
     !MIN:  9! 17!   7! 1767!  280!06-12:  1!18-24:   0 !
--------------------------------------------------
61999!800303! 1! 1!CN61333  CN61IR   CN61333  LOGGER   CN61500  LASTLOG
     !             !CN61500  LIST     CN61999  DES      CN61999  NUXXPRNT
     !             !CN61999  REPTCNTL CN61999  VALCUREV CN61999  XCPPRNT2
     !             !CN61999  XCPPRNT4 CN61999  XCPPRNT6 CN61999  XCPPRNT8
     !             !CN61999  XCPPRNT9 PARL     IR
     !--------------------------------------------
     !     CRU!PSU!DSU !TCH  !CNCT !        LOGON        !
     !TOT: 48!200! 161! 6859!  336!       FREQUENCY      !
     !MAX: 48!200! 161! 6859!  336!00-06:  0!12-18:   0 !
     !MIN: 48!200! 161! 6859!  336!06-12:  1!18-24:   0 !
--------------------------------------------------
```

as a function of the details of the type of audit trails being taken and the kind of monitoring that one attempts to perform on the specific objects.

5. Adapting to SMF Data

5.1 Relevant SMF Records

The principal SMF records of use in performing the kind of auditing discussed in the preceding sections are record types 4, 5, 6, 10, 14, 15, 17, 18, 20, 25, 26, 34, 35, 40, 62, 63, 64, 67, 68, 69, 80 and 81. Ordinarily, these record types would be the records making up the details of a particular job or use of a computer. In producing the audit flow, selection parameters such as user names can be used to extract all audit trail data with that user name associated with it to provide input to the audit record sort step which collects together in one place all record types associated with a particular job or use of a computer. The output of sorted job records is used as input to a job summary or session summary record builder. It is the summary record builder program that would provide the essential information from which the audit history records would be created and maintained.

When dealing with SMF, one is overwhelmed with data, a good deal of it not necessarily useful for security audit purposes. A basic audit history record is shown in Figure 10. This record is the one used in the model program. The individual data items are self-explanatory for the most part. The items indicated in square brackets are additional information available from SMF records that was not available in the accounting data in the model system.

Where the record shows sessions, one could substitute the notion of jobs; aside from that, the history records characterize a particular use of the computer system in which the model was being developed.

FIGURE 10

Data Item	Comments
USERID	
[JOBID]	
File/data set list	List of data sets referred to in this job (session).
[Number of read/writes to each data set]	
Total number of runs (sessions) to date	
Frequency count of logons (job run times) to date	Counted by quarter of day; other distributions are possible.
Date of last update	Used to determine when to purge audit history record.
Total number of updates	
Total to date of:	
. CPU time . I/O operations . Connect time (job turn-around time) . Characters transmitted to terminal	Used to compute mean values: = < parameter >/total sessions
Maximum/minimum to date of:	
. CPU time . I/O operations . Connect time . Characters transmitted	Establishes observed range of values.

NOTE: Items in square brackets ([]) were not available in model system.

FIGURE 10 (continued)

BASIC AUDIT HISTORY RECORD

Data Item	Comments
Sum of the squares of each: . CPU time . I/O operations . Connect time . Characters transmitted	Used to (re)compute standard deviation.
Standard deviation of each: . CPU time . I/O operations . Connect time . Characters transmitted	Computed from: $\sqrt{\frac{\text{Sum sqrs.} <X>}{\text{Total sessions}} - \left(\text{Mean} <X>\right)^2}$
Mean + 2.58 (standard deviation) of each: . CPU time . I/O operations . Connect time . Characters transmitted	Upper bound of distribution.
Mean - 2.58 (standard deviation) of each: . CPU time . I/O operations . Connect time . Characters transmitted	Lower bound of distribution.

Inclusion of the actual standard deviation values and the mean plus or minus 2.58 times the standard deviation of each of the major parameters was to simplify the computation and to make the program run a little faster. It is certainly feasible to compute this data each time it is required; however, with the large number of records, the computation time becomes excessive, and the value of storing it in the record itself becomes a little more apparent.

The accounting data available in the model system does not show the number of read and write operations to each data set that is referred to in the file data set list. If this data were available, the totals, the standard deviations, and the sum of squares information could be augmented by this data to provide a finer grain of detail in the audit history record. It would then be possible to make an exception report for and of those items that exceeded the bounds around the mean for each file rather than treating them in aggregate as shown in this particular format.

5.2 Other Surveillance Tools

It is understood that the customer's SMF data is kept on-line for one day and then written out to tape(s) for longer-term storage. In addition to the standard exception reporting program outlined in this paper, it must be possible for the security officer to look at the detail records associated with a particular user, a particular terminal, a particular job, or a particular file, in order to produce in detail the time sequence of operations actually performed during the job or session. It is not suggested that detailed time sequences of operation be performed for every user at all times; rather, it has been found necessary in order to in greater detail what is going on, to be able to examine the individual accounting records making up a job or a session, particularly for those job sessions which exhibit parameter values outside of the statistical bounds established by the surveillance program.

In the case of the SMF records, it is possible for a user to spawn batch jobs from the VM system. It must be possible for all of the activities of a given user to be traced to the various machines which may be used in accomplishing his or her work. The experience with the model system indicates that it is important that the records making up a session or a job or a unit of work be presented contiguously rather than intermixing the records on the basis of an arbitrary time stamp associated with each record. In practice, this may mean detail entries will be tracked on the VM system to the point where a job is batched to the JES3 job distribution system, then through all the job steps of the batched job, and then back to VM to show the continuation of the activities on the VM in parallel with or while the batch job was running one or more of the batch systems.

In general, there is a requirement to be able to track jobs or sessions based on a variety of kinds of information; for example, terminal identifiers or specific devices referred to and the like. The requirement is to be able to either show all records with the same terminal identifier or the same device address, or sometimes to use the terminal identifier device address or other characteristics to identify the job and then to show all details for that particular job.

For instance, if there is reason to suspect that there is unwarranted file access activity against a particular file, one may wish to examine all details of activity against that file regardless of the individual programs making the references, in which case the fileid would act as a pointer into the first SMF record that contained its identifier. From that record, the job identifier would be obtained and then the detail for the entire job could be displayed or acquired.

5.3 Summary

The computer base security audit and surveillance system can be an effective tool in security control and management of ADP resources. User, data set, and program profiles can provide security personnel with information regarding exceptional use of the system. While it is expected that nearly all such exceptional use will be benign, this approach makes it possible to detect possible misuse of the system. It gives security personnel important automated tools to help provide early detection of unauthorized, [redacted]malicious activity directed against ADP assets.

In the preceding sections, an outline of a system design and the basis for providing statistical detection of abnormal use was developed. The surveillance and detection system is a filter screening out the mass of users of any system who are not doing anything untoward. In general, what constitutes "abnormality" is parametric. It can be set for any given environment. While the bulk of the report focused on the identification of abnormal use by individual users, statistics similar to those described for individual users can be accumulated for the user population as a whole, and the entire population screened for the purpose of identifying potential detailed

With the use of statistical parameters such as those described above, the system can report abnormalities; that is, usage outside of the range of those parameters. This does not mean that a particular episode involves anything wrong; it merely means that something is statistically different from previous accumulated use of the system for that entity; that is, user, file, program, and so forth. If abnormal symptoms do not recur, it is likely that nothing much is happening; however, if the symptoms continue to show up, then the subject involved could be investigated further by more conventional means.

In any real-life situation, computer systems often have thousands of users and tens of thousands of programs in data files. It is necessary to reduce the volume of history data implied by these numbers in various ways. First, if there are individuals whose use of the system is subject to surveillance because of the sensitivity of their jobs or for any other reason, he or she becomes a subject of interest. The selection of job (that is, session, tasks, runs, etc.) records can and should be made on that user's identity to include such individuals. The system designs sketched in the preceding sections indicate the use of such selection functions.

Note that most of the tests applied to systems use are equally applicable to specific files, and, as the section indicated, one could use a pre-pass to collect user's identification for those users referring to a specific named object: file, device, system, and the like.

Rather than attempt to treat all members of a large population with this system, at all times, a sampling technique can be applied to select subsets of the total population for examination either over a particular period of time such as two weeks or for a gross examination against gross parameters established for the population as a whole. Of the two approaches, the detail examination for several weeks appears a priori to be the preferred method.

6. Development Plans

6.1 Introduction

This section outlines a development plan and gives an estimated schedule and level of effort to provide an operationally useful security surveillance system. No serious attempt has been made to estimate computer time or storage cost as this will be affected by the actual system configuration used to implement the design.

The basic system consists of two programs:

- Security Surveillance Subsystem
- Security Trace Subsystem

6.2 Surveillance Subsystem Functional Description

The Surveillance Subsystem will consist of three preparation steps and a series of report formatters. The function of this subsystem is to provide exception reports of "abnormal" system use by specified individuals.

The function of the first step of the surveillance subsystem is to extract from the dump data set all relevant SMF records associated with a list of users making up the (a) "watch list". The selected SMF records are collected in a single data set where they are sorted in time-sequence order by user-id.

The sorted selected records will be processed by the next step to create one record per job or session. The record will be identified by the user-id, and the list of data sets or files referred to as a job/session characteristic.

Detailed measures of time, I/O activity, and the like, associated with the job/session (as described in section 3), will be collected in summary form in the job/session record.

(NOTE: Some of this data was apparently being collected in customer-developed SMF records type 210 in 1978 and 1979. If these records are still being collected, this step may merely be an adaptation of the program that produces the type 210 records.)

The job/session records will then be posted in user-id, job/session characteristic order for the update step to follow.

The update step matches job/session records against history records to:

- determine whether individual job/session records are within statistical "normality";
- accumulate additional data to refine the statistics;
- look for single "abnormal" events (illegal logons, single parameter absolute values exceeding arbitrary thresholds, etc.);
- create "new" history records (existing user, new job/session characteristic or totally new user);

- drop "old" history records for lack of activity.

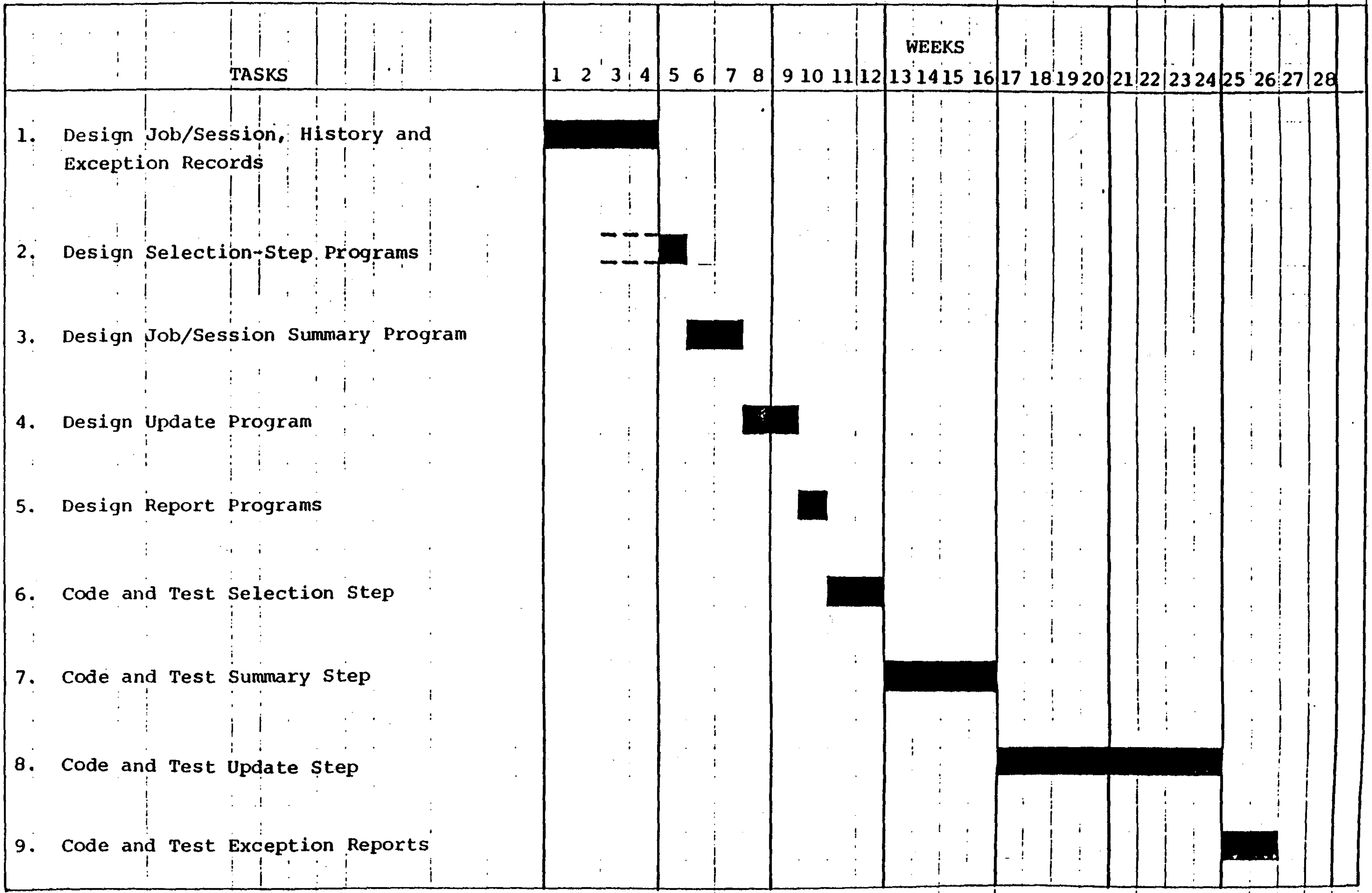
The update step will produce an exception file with all major exceptions reported at least by type (e.g., values exceed absolute limits; values exceed statistical limit; new records added; old records dropped for lack of activity; etc.).

The final step(s) are a set of report formatters that select a particular exception type and edit and format a report for that kind of exception (see Figures 6, 7, 8, and 9 for examples).

6.3 Tasks

	Tasks	Level of Effort (man-weeks)	Elapsed Time (weeks)
I.	Design Job/Session Record, History Record, and Exception Records	4	4
II.	Design Selection Step Program	1	1
III.	Design Job/Session Summary Program	2	2
IV.	Design Update Program	2	2
V.	Design Report Programs	1 (for 4 reports)	1
VI.	Code and Test Selection Step	2	2
VII.	Code and Test Summary Step	4	4
VIII.	Code and Test Update Step	8	8
IX.	Code and Test Exception Reports (approximately 4)	2	2
	TOTALS	26	26

SUMMARY TASK SCHEDULE FOR SECURITY SURVEILLANCE SUBSYSTEMS



6.4 Trace Subsystem Functional Description

The function of the trace subsystem is to produce from the SMF records a detailed, time-sequenced log of activity by (or on) a selected entity.

The Security Trace Subsystem will accept parameters specifying the type of entity and the time scope of the trace. The trace report will be fixed for a given type of entity.

Parameters to the trace should include:

- Type of entity (job-id, user-id, data set, device-id, etc.);

- Time parameters:

  start date (if omitted - today)
  [end date] (if omitted - today)
  start time (if omitted - 00:00:00)
  [end time] (if omitted - 23:59:59)

As long as the times specified are increasing (and not overlapping), it should be feasible to trace multiple time ranges in a single pass of the "raw" SMF data.

Some time parameters might look like:

3/18/80
3/18/80 1600
3/18/80 - 3/20/80 1600
3/18/80 1600 - 1830, 3/20/80 14:30 ...

The trace records will have a standard part, then specific information that is appropriate to the record. A sample trace might look like:

```
                         TRACE FOR USER JONES.J

                        <DATE (OR DATE RANGE)>


    TIME
(HH:MM:SS.hh)           REC. TYPE

 15:23:01.00            JOB INIT   < JOB NAME >

 15:23:02.18            RACF PROC    JOB INIT   <job name>

 15:23:07.46            RACF PROC    ACCESS     <data set name> <type of access>
                                                    OLDF.DATA          READ

 15:23:17.49 ...
                                  .
                                  .
                                  .

 15:26:01.89            STEP TERM  < JOB NAME ><step name>...

 15:26:11.35            JOB TERM   < JOB NAME ><completion code>...
```

6.5 Tasks

	Tasks	Level of Effort (man-weeks)	Elapsed Time (weeks)
I.	Design content of: . user-id trace . job-id trace . device-id trace . .	6	6
II.	Design Trace Program	3	3
III.	Code and Test Trace Program	3	3
	TOTALS	12	12

SUMMARY TASK SCHEDULE FOR
SECURITY TRACE SUBSYSTEM

	WEEKS															
TASKS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. Design contents of:																
a) User-id Trace Report	■	■														
b) Job-id Trace Report			■	■												
c) Device-id Trace Report					■	■										
2. Design Trace Program							■	■	■							
3. Code and Test Trace Program										■	■	■				

6.6 Integration of Subsystems

The scope of this task depends on the system environment in which the security officer subsystems will be placed. If the programs are placed on the VM system, then one or more JCL sets (procedures) can be used to permit the programs to work with current SMF data (SYS1.MANX, SYS1.MANY data sets) or the dump data sets (SMF.DAILY.DATA) or the weekly data sets (SMF.WEEKLY.DATA). Allocation of the correct data sets can be done from the date parameters to the trace programs. There is no particular allocation required for the surveillance subsystem.

If the security officer surveillance subsystem(s) is placed on a stand-alone minisystem (for example), there is some action needed to either copy the entire dump data set to the minisystem (not recommended due to its size) or run the job/session select program on VM to produce a data set that will be brought over to the mini for processing.

Since access to current and recent SMF.DAILY.DATA and SMF.WEEKLY.DATA sets is needed for the trace function, and since at least the surveillance subsystem selection step must access the current SMF.DAILY.DATA, it appears that the security subsystem(s) should be placed in/on VM.

	Tasks	Level of Effort (man-weeks)	Elapsed Time (weeks)
I.	Define Integration Requirements	2	2
II.	Code and Test Procs for Integration	2	2
	TOTALS	4	4

SUMMARY TASK SCHEDULE FOR INTEGRATION OF SECURITY SUBSYSTEM

	WEEKS											
TASKS	1	2	3	4	5	6	7	8	9	10	11	12
1. Define Integration Requirements	■	■										
2. Code and Test Procs for Integration			■	■								