MVS SECURITY AND THE USE OF SAS* SOFTWARE TO GATHER SMF DATA

Bruce Blank, Empire Blue Cross & Blue Shield
Michael Mirabile, First Boston Corporation

INTRODUCTION

With the increasing complexity of data processing equipment, programs and operating systems, assurances and verifications of the MVS system components are routinely necessary.

The purpose of this document is to identify and highlight the important MVS system's security aspects. This paper does not serve as a "cure all" for security breaches by any means, but rather as an informative look at the internal components of MVS and where security breaches can result. SAS* Software is the tool that can be utilized to review many of the aspects relating to assurance and review of the MVS system security components. Additionally, SMF (System Management Facility) provides the major input to our SAS* Software code for assurance and review purposes, since SMF collects and records a wide variety of related information.

"The protection of data against persons deliberate or accidental access by unauthorized persons is rapidly becoming a major problem. Ultimately, the security of data depends on some combination of locks, or access-control measures, for which certain users possess the keys. No such combination is completely secure. For the intruder, the effectiveness of security measures is really only a matter of the cost of breaking the combination of locks as compared to the value (to the intruder) of obtaining data in this way. Conversely, for one wishing to devise and implement a combination of locks on the data must be small relative to the cost of a breach of security".1

Information security then can be looked at as the protection of data against unauthorized usage, whether accidental or purposeful.

Obvious, there are two major reasons why security issues should be reviewed more closely:

A. Asset Protection - The information processed by data processing equipment (as well as the equipment) represents a major corporate asset. The ongoing heart beat of an organization requires that data processing equipment be protected (as well as data stored in the equipment).

B. Legal Requirements - Legislation places the responsibility on management to guarantee the accuracy and integrity of an organization's data and ensure the privacy of customer and employee files.2

As a result, MVS system security with the use of SAS* Software to review vulnerabilities, should receive a high level of management attention.

In highlighting IBM philosophy, there are seven ways to implement a total security package that can protect systems and data; they are:

Item #1: Risk Analysis and Acceptance

Determining which data and data processing equipment must/should be protected and how.

Item #2: Security Policies and Practices

Defining where in the organizational structure the responsibility for security lies and developing a security policy, with guidelines, instructions, and recommendations for implementing such a policy.

Item #3: Physical Security

Selecting secure sites for data processing equipment/system, controlling physical access to the system, and planning for controlling potential damage to the system (backup/recovery).

Item #4: System Integrity

Insuring that the system resists compromise of its control through misuse or manipulation and that it has barriers to ensure that its resources can be accessed only through established access routes.

Item #5: Access Control

Restricting the use of the systems resources to authorized persons only, and limiting those persons to using only the system resources they require to do their jobs (without causing undue impact on their productivity).

Item #6: Information Transmission Protection

Where physical access control cannot be established, making sensitive data unintelligible, using some means such as cryptography.

Item #7: Auditor Review

Examining, verifying and demonstrating the adequacy of the security controls of the data processing system, as well as verifying the accuracy and completeness of the results.
Additional attention should be focused on Items #4, "System Integrity", and Item #7, "Auditor/Review", because this is where the review process establishes a basis.

An operating system should have integrity; that is, it should be capable of preventing one program from interfering with or modifying another (system or user) program's execution unless it is authorized to do so. How do we define and how can we verify that the above is true?

An operating system that has integrity has the ability to protect itself against unauthorized access to the extent that security controls cannot be compromised. It is important to recognize the distinction between an authorized program and an unauthorized program. The operating system recognizes an authorized program. Other control mechanisms are needed to help insure that only an authorized user submits a job to execute an authorized program.

MVS integrity provides a base on which an installation can add an access control mechanism (such as ACF2 or RACF). The protection provided by security software packages cannot, however, be more absolute than the integrity of the operating system under which it is running. Access controls are needed to provide reasonable assurance that MVS integrity is intact. A periodic review of the controls is therefore necessary. Identification of who can gain access to the system, and what resources can be affected is important and should be reviewed periodically. An access control mechanism while it prevents unauthorized access, should not prohibit authorized users from doing their jobs. (I.e., it should detect and log violations when activity contrary to management's direction is observed). SAS* software can be used to complement security mechanisms. For example, SAS Software can be generated to determine who opens or closes critical system data sets.

A major significant factor which affects the total scope of every security package (I.e., ACF2 or RACF) is the use of local exits to insert local code. These exits can alter or circumvent the security decision processes. Exits pass control to local code in a supervisory state. SMP and JES exits are also important when reviewing the total scope of security software. All exits should be identified and evaluated for their significance when reviewing the total ramifications of MVS - security/integrity.

While it is important to cover miniscule levels of detail when reviewing the MVS system, a pragmatic approach for this report would be to cover the important areas of MVS integrity and to enlighten the reader as to the recognized major security elements of an MVS system.

Therefore, this next section describes and highlights security concerns for the following system components, utilities, aids, etc.

I. Authorized Program Facilities (APF)
II. Sensitive System Services
III. Program Properties Table (PPT)
IV. System Modification Program (SNM)
V. Job Entry Subsystem (JES)
VI. System Management Facility (SMF)
VII. SMF Reporting with SAS* Software

I. AUTHORIZED PROGRAM FACILITY

The Authorized Program Facility (APF) is a facility used to identify system or user programs that are permitted to use sensitive system functions. An authorized program is one that executes in supervisor state, with a Program Status Word (PSW) key of 0-7, or with APF authorization.

APF allows an MVS installation to restrict use of restricted Supervisor Calls (SVC's) and authorized programs. Uncontrolled, these system functions can bypass security authorization checking without an audit trail. They may also be allowed access to other than a process that is authorized when a program or process executes in any of the following states;

a. Supervisor state (bit 15 of the PSW is zero).

b. A system protection key (bits 8-11 of the PSW are in the range of 0-7).

c. As part of an APF authorized job step task (bit "JSCW1M" in the Job Step Control Block is 1).

APF, then, works on the assumption that if the first module in a program sequence is authorized, then authorized individuals have determined the flow of control to all subsequent modules in that sequence. MVS considers these modules authorized to access sensitive system functions as long as they come from authorized libraries.

Assurances should satisfy:

- That all programs that run as authorized adhere to the guidelines set by security administration as well as conform to MVS integrity requirements.

- Which programs are stored in the authorized libraries.

- Examples of MVS authorized libraries are: SYS1.LINKLIB, SVCLIB, LPALIB, IMACELIB. There are many more APF.LIBRARIES included in an installation's APF which need to be identified.
Authorized libraries are protected via (ACF2 or RACF) so that only the selected users can store or alter programs in these libraries.

That no two load modules with the same name exist across the set of authorized libraries. Two modules with the same name could lead to accidental or deliberate mixup in module flow, possibly introducing an integrity exposure.

That only the first load module in program sequence is link edited with the authorization code of 1 (AC=1). If subsequent load modules are called with the authorization code, they become APP-authorized, and possibly bypass validity checking and/or critical logic flow.

That the IEAAPF (contained in SYSL.PARM.LIB) does not contain the names and volume serial numbers of data sets that no longer exist. If it does, a user could assign his own data sets with the same names on the same volumes and cause his own libraries to become authorized.

That the volumes specified in the IEAAPFX members are mounted at IPL time and are permanently resident. This can be done by using the VAILSTDFX member of SYSL.PARM(LIB). Otherwise a user could supply his own BASD volume.

With the above combination of "Assurances", you will have a better security grip on APF. You will be assured that the APF restrictions apply to load modules so that authorized programs cannot access any load module that is not in an authorized library.

II. SENSITIVE SYSTEM SERVICES (SVC)

By definition,'any computer program that has the capability to execute in supervisor state is an extension of the operating system and is, therefore, considered as system software.' This statement reflects the reality at most installations.

Supervisor calls (SVC's) are the primary method by which programs communicate with the operating system. SVC's provide specific services, such as handling I/O operations, and opening and closing of files. Executing programs issue SVC's, which cause control to pass to the operating system and the requested function to be performed. Most SVC's are provided with MVS, but additional user SVC's can be developed at each installation, or added by other subsystems or program products.

SVC's run in supervisor state and have the power to use privileged instructions. This means that SVC's also have access to all the main storage of the computer. Therefore, the user-developed SVC's should be monitored to assure that they cannot access/c tamper with important system data.

You should therefore review the following:

The number and names of the SVC's.
The origin and type of each SVC.
Where each SVC resides in storage (ie., LPA or nucleus).

III. PROGRAM PROPERTIES TABLE (PPT)

MVS recognizes that some programs need special properties in order to run as efficiently and as securely as possible. A list of these programs, along with their special properties, is maintained in MVS and is known as the Program Properties Table (PPT). SVC's in the PPT can be given the power by bypassing security mechanisms such as those employed by ACF2 or RACF. They are also capable of bypassing password protection and ignoring data set integrity. Programs residing in the PPT can be non-swappable and potentially affect system performance.

You should review and monitor:

What programs are in the PPT
What special privileges each program has
Identity of the library in which each program resides

It is important to assure that programs in the PPT are only given the privileges they require to function and that all of these programs are properly protected from unauthorized usage.

IV. SMP/E - SYSTEM MODIFICATION PROGRAM

The SMP is the program that provides facilities for the installation of IBM software products and maintenance. It controls changes to the system at the element (module/entry) level. From a large number of potential changes, SMP selects the proper levels of elements to be installed, invokes system utility programs to accomplish the installation, and maintains records of the changes and of the function and service level of every element in the system. The precise control of element levels is the primary system integrity function provided by SMP.

SMP/E control information is maintained in VSAM data sets, known as the Consolidated Software Inventory (CSI). These inventory data sets contain information on system structure, system function and service
levels, and other control information required for SMP processing. SMP/E uses functions inherent to VSAM to maintain the integrity on SMP data.

Review of who reads, writes and changes SMP data sets is necessary to verify that authorized individuals are using this facility.

SMP/E runs as an authorized program under MVS. Use of this program should be controlled by access control software.

V. JES: JOB ENTRY SUBSYSTEM

The MVS Job Entry Subsystem controls the scheduling and submission of jobs in the operating system. JES makes a clear distinction between the functions that:

a. Schedule jobs for execution
b. Supervise the execution of jobs
c. Remove job output from the system

The supervision of jobs is done by MVS itself. The scheduling and removing of jobs from MVS is accomplished by JES.

You should be able to review JES parameters, such as, logon, started task, and batch conversion because of the implications that these important parameters have in controlling the way JES functions. In essence, JES parameters specify how, and with what restrictions, jobs will run on the system.

VI. SMF: SYSTEM MANAGEMENT FACILITIES

Reviewing the access to controlled resources is a must for efficient control of an installation's sensitive data. A complete record of unauthorized attempts to access a controlled resource may prove useful in discovering security leaks. ACF2 or other access control software can provide these reporting features, however MVS by itself provides logging facilities that SAS Software can use to provide detailed analysis. Therefore, another dimension of security review needs to be administered and verified.

System Management Facilities is perhaps the most important of all MVS logging facilities because it provides the primary audit trail in an installation.

Control of the SMF facility is critical. The SMF system options chosen for any installation determine which SMF records are recorded, which SMF exits are in effect and whether SMF parameters may be changed at IPL time. SMF files contain all SMF records collected on the system. This includes a journal of all events occurring in the system, as well as records logged by access control software. The integrity of this audit trail can be jeopardized by the loss of SMF records when files are not dumped at appropriate times. The possibility also exists to alter or delete records from files and destroy system integrity. It is also possible to bypass any writing to SMF for a particular TSO user ID.

SMF exits are IBM supplied routines which have the ability to take control from a job at either the beginning of a job or before the processing of a job step. These exits can be tailored (by programmers) to suit an installations needs. All SMF exits are APF authorized. SMF exits have the ability to modify all JCL, including changing file names, program names, job accounting information, and alteration of job priorities. SMF exits can be used to cancel programs and jobs, and cancel SMF records before they are journaled.

Examination of SMF records/exits is crucial when looking at security implications, and SAS Software is the tool that can effectively be used.

The following list provides some examples of important security related records that may be useful.

| Rec Type | List all IPLs which have occurred 00 |
| List job steps that abended with particular ABEND codes 04 |
| List job steps executing particular program (ie., AMASP2AP) 04 |
| List jobs and job steps executed with a privileged storage protection key (ie., 0-7) 04/05 |
| List jobs terminated by SMF exits 06/05 |
| List jobs run by a particular user 05 |
| List all output generated for a particular form 06 |
| List all occurrences of lost SMF recording 07 |
| List all accesses to a particular data set group 14/15 |
| List all scratchs/renames for a particular data set group 17/18 |
| List all TSO sessions for a particular user group 34 |
| List uses of specific TSO commands by a particular user 32 |
Rec Type
List all denied accesses to VSAM clusters 11
List RACF exception events 80
List RACF initialization parameters 81
List occurrences of SETSMF or SET SMF commands 90

VII. SMF REPORTING WITH SAS

We can now highlight how the SAS* Software product can be utilized to depict, analyze, and produce formatted reports from SMF raw data.

Sources of existing SAS* Software code to simplify the coding process will be identified.

SAS* Software Characteristics

Structure of a SAS Library
- Similar to a PDS with numerous members
- Records become observations
- Fields become variables
- Self-defining data set labels and formats are stored with data

SAS Data Step
- Typical use is to create or modify one or more data sets. Also used to create reports
- Uses a PL/I-like programming language
- Performs routine housekeeping for the user
- Interactive
- Reads SMF records to create a SAS* data set

SAS Procedure
- Compiled program (PL/I, BAL, etc.) which processes SAS* Software data sets
- Standard syntax

Characteristics of an SMF Data Set

What is SMF
- SMF is part of MVS - Event driven
- SMF formats information into a number of record types
- Variable spanned record formats
- Data formats vary
  - Bit values, binary, packed decimal, zone decimal, alphanumeric
- Multiple record types

Decoding SMF data

Merrill's Book and Code
- Book provides a very well written description of the data content of the various SMF record types
- Code decodes SMF records into a SAS* Software structure

- Provide basic coding techniques for handling SMF data
- Create your own approach
  - Use IBM manuals
  - Decode what you specifically need
  - Combine SAS* data sets

Here are two examples of SAS* Code used by Quality Review to complement and verify existing control functions.

Example 1.: Identifies users of restricted utilities.

```sql
1 DATA;
2 INFILE SMF;
3 INPUT 62 ID PIB1. 8;
4 IF ID=4;
5 INPUT 855 FGMNAME $2. 8;
6 IF FGMNAME = 'AMASZAP' OR
7 FGMNAME = 'VSAMZAP' OR
8 FGMNAME = 'LIBRARY' OR
9 FGMNAME = 'AMBLIST';
10 INPUT 815 JOBNAME $8.
11 @63 STEPNAM $8.
12 @39 STEPIFM PIB1.
13 @31 USEPID $8.
14 @23 JOBSTART SMFSTAMP8.
15 @52 CJNJCDE PIB2.
16 @84 TECODE PIB1.
17 @11 SYSID $4. ;
18 FORMAT JOBSTART DATETIME.
19 CONDICE HE63.
20 TERMDEC HEX22.
21 PROC SORT;
22 BY FGMNAME JOBSTART;
23 PROC PRINT;
24 TITLE JOBS EXEC A RESTRICTED UTILITY;
```

Example 2.: Identifies users who access 'SYS1.' data sets

```sql
1 DATA;
2 INFILE SMF;
3 INPUT 62 ID PIB1. 8;
4 IF ID=14 OR ID=15;
5 INPUT 855 DSNNAME $44. 8;
6 IF (ID=14 OR ID=15) AND
7 DSNNAME='SYS1.UADS' OR
8 DSNNAME='SYS1.PARMLIB' OR
9 DSNNAME='SYS1.DAE' OR
** Program continued on next page **
```
In conclusion, a review of MVS system security refers to the features and characteristics needed to examine, verify, or demonstrate controls to achieve an accurate and complete look at MVS. SAS® Software is a comprehensive software package that automates many aspects of a technical review of system's hardware and software environment. SAS® Software is capable of performing a variety of data gathering tasks, and can help monitor the integrity of the MVS environment. Most issues addressed in this document can be serviced by this tool with uncompromising accuracy. Again, the review process involves capturing data, documenting it, and feeding it back to management in such a way that it can be used to make decisions about the adequacy or inadequacy of security and integrity of your installation's operating environment.

References

IBM OS/VS2 Systems Programming Library:
System Management Facilities (SMF)
IBM Publication GC28-1135 (3A)
Merrill's Guide to Computer Performance

ACF2 (Access Control Facility) is developed by SKX Inc., Rosemont, IL. ACF2 is a registered trademark of SKX, Inc.

RACF (Resource Access Control Facility) is an IBM program product, Armonk, New York

MVS/Extended Architecture System Programming Library
Initialization and Tuning, IBM Publication GC29-1149-3

AIC II Conference Proceedings, "SMF Data Integrity and Archival", Jason G. Shane

*SAS is the registered trademark of SAS Institute Inc., Cary, NC, USA

************************************************************
Special appreciation goes to my family.

Also, special thanks goes to Gloria Thomas, who edited and retyped this report so many times that she probably has the text memorized by now.

************************************************************

For additional information contact:
Bruce Black
Empire Blue Cross and Blue Shield
622 Third Avenue
New York, NY 10017
Quality Review/Dept. 280/31st Floor

230