MANEWS 03

===
===

M A News
Mainframe Audit News
August, 2002
Issue Number 03

===
===

Table of Contents

I N T H I S I S S U E
===
===

(YOU CAN USE the Find function of your email software to jump directly to a given section. For example, to jump directly to the Websites section, you would Find on "7)" or on "Websites".)
==
==

1) Introducing the Mainframe Audit News

This is the third edition of the Mainframe Audit News, a vehicle for sharing information about auditing IBM mainframe computers.

The MA News is a free, email, newsletter for auditors who need (or suspect that they will need) to be auditing IBM mainframe systems (primarily MVS, OS/390, z/OS, and the system software associated with them). This software includes: CICS, DB2, JES, VTAM, MQSeries, TSO, USS (UNIX System Services), TCP/IP, and others. It also includes the Websphere server software which connects a mainframe to the Internet. (Note, we will expand each of these acronyms and explain how the software works over the course of the next several issues.)

We expect to have a new issue at least every four months.

The MA News is meant for auditors who are new to IBM mainframes, as well as for experienced MVS auditors who want to keep up to date with the latest developments from IBM. We will not make the list of subscribers available to anyone else for any reason.

To Sign Up for the Mainframe Auditors' Newsletter, see section 9) below.

==
==

2) What Are ACF2, RACF, and TopSecret? (Security Software)

You need one of these three products for complete security on MVS (or OS/390 or z/OS) computers. They each answer the two basic security questions:

Q1) Who is this user? (often answered by means of a userid and password)

Q2) Can this user do x? (where x can be anything you want to specify)

Unlike UNIX, Windows 2000, and many other operating systems, MVS does not provide answers to these two basic questions. So any MVS shop will need to implement one of these products. RACF (from IBM) was created first. At the time, it was so awful that many users complained to IBM, and made specific suggestions on how to improve it. When IBM (at that time) was not responsive to these comments, some of the complainers decided to make their own competing product to do the job right. (Why whine when it's more fun to get rich while proving yourself right!) This product was ACF2, which is now offered by Computer Associates. The third product, TopSecret, was developed for similar reasons, and is also offered by CA.

Eventually, IBM started to improve RACF, to the point where it now leads in market share, and seems to be growing. CA has also improved both ACF2 and TopSecret. It is now a fair statement that all three products do a good job of providing computer security when they are properly implemented. There are differences between them in terms of features and architecture, but these differences seem small to our editorial staff. It would be difficult to demonstrate convincingly that one of the three products does a significantly better job than the other two of providing good security.

All three products use similar approaches to answering Q1 and Q2. When a user logs on at a terminal, the software she connects to will ideally call the security software to verify her userid and password. This function is called **RACINIT**, and the result of a successful verification will be a control block in memory named the **ACEE**. The ACEE has most of the information from the security software's definition of the user.

Later, when the user tries to open a file or access a protected resource, the appropriate software ideally calls the security software to ask Q2, can this user access this dataset or resource. (A resource is anything other than a dataset you want to control access to, for example, an on-line transaction, a terminal, a program, or literally anything.) This access-permission-checking function is called **RACHECK**. One of its inputs is the ACEE control block describing the user, which was created by the RACINIT function. The other input will be a dataset or resource rule stored by the security software in its rules database. The rule will specify who is permitted to read, write or, otherwise access the dataset or resource. (TopSecret functions a little bit differently from this, but the effect is the same.)

We will discuss each product briefly, showing you differences in names of things and basic data gathering for an audit.

3A) >>>>RACF (Resource Access Control Facility)

RACF maintains its security rules in a disk dataset called the RACF database. The records include: user records, group records, dataset records, and resource records. (A record is also called a rule, a definition, a profile, and an entity.) User records include the user's: userid, name, password, privileges, and other information. Group records list the users who are considered members of the group. (Grouping users into groups simplifies administration and supports "role-based" security.) Dataset rules describe datasets, which users and groups can read and write them, the default access permission (called a UACC or UNIVERSAL ACCESS), and other information. Resource rules have similar information describing resources.

Userids in RACF can have privileges, such as the SPECIAL privilege which lets the user issue any RACF command, and the OPERATIONS privilege which (with some exceptions) lets the user do anything to any dataset.

To prepare for a RACF audit, you will want to get two printouts: SETR LIST and DSMON. To get an explanation of each field in these printouts and of how to address each field in an audit, please check our website www.stuhenderson.com and click on ARTICLES.

You will also want to get a userid with the AUDITOR privilege, which will let you list any rule you want.

3B) >>>>ACF2

ACF2 stores its rules in three databases: the LOGONID database (user definitions), the Access Rules database (dataset rules), and the Infostorage database (resource rules and other information). User records are called LID records, short for LOGONID records.

ACF2 installations each define a UID string, that is a collection of fields from the LID record which are to be concatenated (treated as if they are stuck together in one sequence). These fields might include the logonid, the user's department, division, or other fields. For each ACF2 audit, you should learn what fields constitute the UID string.

The UID string is compared to the dataset and resource rules to determine whether a user can access a dataset or resource.

LIDs can have privileges in ACF2, for example the NON-CNCL privilege permits a LID to access any dataset or resource and READALL permits a LID to read every dataset.

To prepare for an ACF2 audit, you will want to get several printouts, including: the FDR (Field Definition Record) which defines the UID string, a SHOW ALL command which lists option settings, and a LID with the AUDIT option so that you can list rules.

3C) >>>>TopSecret

In TopSecret, userids are called ACIDs (Access Identifiers). This is pronounced "Ay-Sids", not "Acids", because the TopSecret developers were Canadian.

ACIDs can be grouped in Profiles to simplify security administration.
A Profile in TopSecret is something like a group in RACF. This makes for great fun when converting from TopSecret to RACF, since "profile" in RACF means any type of rule.

ACIDs can have privileges, such as NODSNCHK which allows total access to any dataset and NOSUBCHK which allows submission of any batch job.

To prepare for a TopSecret audit, you will want to get certain printouts including: TSSAUDIT and TSS STATUS. You will also want an ACID with the AUDIT privilege, so that you can list any rule you want.

==
==

3) What's the Difference Between VTAM, SNA, TCP/IP, and APPC?

VTAM (Virtual Telecommunications Access Method) is IBM's communications software for MVS computers. It controls all accesses by programs to terminals. A program which wants to read from and write to a terminal will use VTAM to make the requests.

SNA (System Network Architecture) is the hardware and software architecture IBM has defined for its computers. SNA is the architecture VTAM executes on. SNA consists of formal definitions of message formats and possible responses to messages.

TCP/IP (Transmission Control Protocol/Internet Protocol) is the communications standard for UNIX computers, for the Internet, and increasingly, for any connection between two different vendors' computers. When you mix a TCP/IP network with an SNA network, you have two choices: You can take the TCP/IP messages, put them inside an SNA envelope, send the SNA envelope to its destination, and have the destination take the TCP/IP message out of the envelope and process it as a TCP/IP message. This is called TCP/IP under SNA tunneling. The second option is called SNA under TCP/IP tunneling, and you can guess how it works by reversing the first option.

APPC is IBM's Advanced Program to Program Communication. It is a standard to let two programs, each on a different type of computer, share information with each other. APPC is comparable to, and competes with, TCP/IP. (APPC is losing the race, by the way.) APPC is a subset of VTAM, or may be considered a protocol that executes on top of VTAM.

==
==

4) What Is JCL (Job Control Language) and How Does It Work?

JCL (Job Control Language) is the way you tell an MVS computer how to run a program. JCL specifies the name of the program and the files which are to be made available to it, and what other programs are to be run as part of the same sequence.

JCL was originally designed to fit on 80-column punched cards which were read into the computer by a (now-obsolete) machine called a card reader. Now JCL usually consists of disk files which are edited by programmers at terminals. JCL still retains its old, 80-column, card image format.

Auditors need to know at least the basics of JCL, both to interpret JCL as part of the audit, and also to be able to execute data gathering programs themselves.

Each batch job is a unit consisting of one or more programs to be executed and the files associated with each program. Each job is defined as a series of one or more job steps, each of which includes one program to be executed and the files it will read or write.

Almost every JCL card has a common format: a "//" (slash-slash) in columns 1 and 2 (the left-most positions on the card), followed by a name, followed by a space, followed by the card type (one of "JOB", "EXEC", or "DD"), followed by a space, and then additional information. To show you a sample skeleton, imagine a job named STUSJOB with two job steps. The first jobstep is for the program MYBIGPGM, and uses two files called BIGIN and BIGOUT. The second jobstep is for the program SMALLPGM, which uses two files named SMALLLST and OUTPUT. The JCL might look something like this:

```
//STUSJOB     JOB ...info describing the job
//STEP1       EXEC PGM=MYBIGPGM ...info describing the program
//BIGIN       DD ...info describing the file
//BIGOUT      DD ...info describing the file
//STEP2       EXEC PGM=SMALLPGM ...info describing the program
//SMALLLST    DD ...info describing the file
//OUTPUT      DD ...info describing the file
```

So when you need to read JCL to understand what a job does, start by breaking it into jobsteps (one per EXEC card). Then see what each program does and what files it uses.

JCL consists of three main types of control card:

-- **JOB cards**, each of which describes one job, including its userid, accounting and other information. The accounting information can be used to charge back computer resource usage to the department sponsoring the job.

-- **EXEC cards**, each of which defines a program to be executed as part of the job. Information might include the name of the program, information to be passed to the program, and how much memory it will need.

-- **DD (Data Definition) cards**, each of which defines a file or dataset which is to be made available to the program in the immediately preceding EXEC card. Information might include the dsname (dataset name) of the dataset, what type of device (tape or disk) it resides on, how big its records are, and what is to be done with the file when the jobstep is over.

You can learn more about the details of each of these card types by consulting the IBM manuals "MVS JCL User's Guide" Number SA22-7598 and "MVS JCL Reference" Number SA22-7597.

==
==

5) Question and Answer Column

(Readers are invited to send Questions to stu@stuhenderson.com, along with an indication of whether we should print your name. If you need an immediate reply to your question, please indicate this in your email.)

Q) What is the difference between VSAM and a database like IMS or DB2?

A) VSAM (Virtual Sequential Access Method) is a way of organizing data in disk datasets. It is MVS-specific: you won't find it on any other operating system. VSAM makes it possible to read the records in a file one after the other. It also makes it possible to find a specific record quickly. It does this by maintaining an index. This index is like the index in the back of a book: it tells you where to find the information for a given name. The VSAM index points to a place within the the VSAM file instead of pointing to a page in the book. The pointer is a six-byte hexadecimal number specifying the number of bytes from the beginning of the dataset that you will find the record you want. The pointer is called an RBA or Relative Byte Address. VSAM maintains the index as a separate file, which is tightly associated with the data file.

VSAM is an access method, that is it is considered one of the standard ways to read and write data. It is part of the MVS operating system at no extra charge. Database software like IMS and DB2 (discussed in the last issue) is considered to be a separate program with additional costs. Database software generally provides much greater functionality than VSAM in managing access to data.

==
==

6) What is MQ Series?

MQ Series is software from IBM that permits a program on one computer easily to share information with a different program on a different computer, even if the second computer is a completely different type from the first.

MQ Series does this by defining a "queue", that is a place where the first program can store the data and where the second program can retrieve it at its convenience. (If you ever took a computer science course describing FIFO and LIFO queues, this is one of those. [FIFO is of course, First In First Out; and LIFO is Last In, First Out.])

IBM has versions of the MQ software that run on just about any type of computer you could run into, including the mainframe. MQ series is popular with companies re-engineering old, legacy programs by re-writing them in more modern languages on more modern platforms, because it lets the old legacy program share the current version of the master file easily with the new program. It also lets organizations link together applications developed on different computers at different times.

IBM sells MQ software in large quantities. Since it is an easy path for data to flow from one computer to another, auditors should consider it when planning the audit schedule. On the mainframe, MQ Series has excellent interfaces with security software such as RACF, ACF2, or TopSecret. Evaluating the way the security software interfaces are used is an excellent subject for an audit.

We will describe MQ Series in 3 sections:

6A) How Does a Program Invoke MQ Series?
6B) How Does Security Work with MQ on Mainframes?
6C) What Should an Auditor Do Now?

6A) >>>>How Does a Program Invoke MQ Series?

MQ Series has five main functions a program will invoke. They are:

1. Connect to an MQ sub-system (there might be, for example, test and production sub-systems)
2. Open a Queue
3. Put Data to a Queue
4. Get Data from a Queue
5. Disconnect from a Queue

You can imagine how easily these calls could be added to an old legacy program with logic like this:

1. Housekeeping
    ? Open files
    ? Initialize counters
    ? Print headings on reports
    ? [Connect to MQ and Open a Queue]

2. MainLine
    ? Read Next Record, at End GoTo WrapUp
    ? [Put Record on a Queue]
    ? Do Something to the Record
    ? GoTo MainLine

3. WrapUp
    ? Print totals on report
    ? Close files
    ? [Disconnect from MQ]

6B) >>>>How Does Security Work with MQ on Mainframes?

MQ security uses security software rules to determine:

- What security checking is to be done for each sub-system
- Who can connect to which sub-systems
- Who can access which queues
- Who can issue which MQ commands
- Who can access MQ resources
- How users are identified

To determine how these questions are answered, you will want to list the security software rules for MQ series. Most of these are in resource classes whose names beginn with MQ-, for example MQADMIN.

6C) >>>>What Should an Auditor Do Now?

In deciding when and whether to audit MQ series, you need to know where it is used in your installation. You might want to collect this information for your standing files, to be used in planning future audits:

- Where do we use and plan to use MQ (between what computers)?
- What applications use MQ on these computers?
- What data is stored on these computers?
- How many MQ sub-systems do we have on each mainframe computer? (There might be, for example, a test MQ sub-system and a production sub-system.) And what are their names?

You can gather this data by means of interviews with system programming staff.

==
==

7) Websites for Mainframe Auditors, Six "How To Audit" Seminars, and the Proverb of the Day

7A) >>>>Websites for Mainframe Auditors

Here are more websites useful to mainframe auditors. We'll let you judge them for yourself, rather than trying to explain what each one does. We do not endorse (nor disparage) any of their products, nor the quality of their websites, since we have not had an opportunity to evaluate them. However, we find them interesting, and hope you will too.

http://www.techweb.com/search/advancedSearch

http://search390.techtarget.com/home/

http://www.share.org/

7B) >>>>Six "How to Audit" Seminars

Six new "How to Audit..." courses are available for IT auditors:

1) How to Audit **Cross-Platform Applications**
2) How to Audit **Mainframe/Internet Connections**
3) How to Audit **TCP/IP**
4) How to Audit **CICS**
5) How to Audit **RACF**
6) How to Audit **MVS**

To learn more about them, please go to http://www.stuhenderson.com/XAUDTTXT.HTM

7C) >>>>This Issue's Proverb of the Day

(Paraphrased from Peter Drucker) "*Too often we spend time and energy optimizing something we actually should be eliminating*."

==
==

8) Tell Us What You Think

We'd love to hear from you, in particular on these topics:

- What do you like/not like about the MANEWS?
- What websites do you know that you want to share with other auditors?
- What topics and/or columns would you like to see in future issues?
- Is your mainframe connected to the Internet and do you plan to audit the security implications of this connection?

Please email your comments to stu@stuhenderson.com. Thanks.
==
==

9) How to Subscribe/Unsubscribe

This section shows you how to:

9A) Subscribe,
9B) Unsubscribe,
9C) Request back issues,
9D) Take advantage of our free technical support for mainframe auditors.

9A) >>>>To Subscribe to the Mainframe Auditors' Newsletter (MA News)

Send an email to: stu@stuhenderson.com
with the subject field set to: MA News
and in the body of the email just this word: SUBSCRIBE

9B) >>>>To Unsubscribe from the Mainframe Auditors' Newsletter (MA News)

Send an email to: stu@stuhenderson.com
with the subject field set to: MA News
and in the body of the email just this word: UNSUBSCRIBE

9C) >>>>To Request Back Issues of the Mainframe Auditors' Newsletter (MA News)

Send an email to: stu@stuhenderson.com
with the subject field set to: MA News
and in the body of the email just this phrase
(for example to request issues 1 and 2): Back Issues: 1, 2

9D) >>>>To Get Questions Answered from the Mainframe Auditors' Newsletter (MA News)

Send an email to: stu@stuhenderson.com

with the subject field set to: MA News
and in the body of the email the word: Question: (followed by
your question, and tell us whether you want your name included or not if we
decide to publish your question and answer)

Whether we print your question and answer or not, we will try to email you
back an answer to your question within five business days. If you need a
faster answer, please phone your question to (301) 229-7187, leaving the
question on the machine. Please repeat your phone number slowly and clearly.

==
==

10) Feature Article: Three Types of Work: Batch Jobs, Started Tasks, and Online

To understand what is happening in an MVS computer, you need to understand these three types of work. Everything you audit on an MVS computer beyond the operating system itself will be one of these three. Your audit approach will depend upon which of the three. We will describe each of them, along with the audit issues each raises.

10A) >>>>Batch Jobs

Batch jobs are "background" jobs, that is they represent programs which are not connected to any terminal. They are defined by means of JCL (Please refer to section 4 of this issue for more on JCL), Job Control Language.) Originally, batch jobs consisted of punched cards which were read by a card reader to get them into the computer. Nowadays, punched cards are obsolete, IBM no longer makes card readers, and batch jobs are submitted by other programs.

For example, a programmer might be logged onto a terminal using the TSO (Time Sharing Option) software. TSO is a software tool that lets programmers edit, compile, execute, and test programs.

The programmer might use an editor program within TSO to create the JCL (Job Control Language, described in section 4 above) to execute a job. The JCL specifies what programs are to run as part of the job, and which files are to be made available to each of these programs. The JCL would be saved in a disk file. The programmer would then issue a SUBMIT command at the terminal to tell TSO to submit the job for execution.

The SUBMIT hands the JCL to software called JES. (There are two variants of this software named JES2 and JES3. Since they perform the same functions, we will just says "JES" to refer to either one of them.) JES translate the JCL to its own internal code and then schedules the job for execution.

Batch jobs can enter the system from the TSO SUBMIT command, from the card reader (obsolete), and from the "internal reader". The internal reader is the part of JES which accepts batch jobs. Programs can submit batch jobs for execution by "writing them to the internal reader". In this case the internal reader is treated like a file. A program writes the JCL to the file which is the internal reader. Writing JCL for a batch job to the internal reader is the same thing as SUBMITing the JCL from TSO.

The most common use of the internal reader is by job scheduling software. This is software which determines when production batch jobs should execute, and then writes them to the internal reader. For example, in your installation perhaps the payroll jobs are scheduled to execute every Thursday at 7PM. The job scheduling software maintains a calendar with this information. When Thursday 7PM comes around, the job scheduling program wakes up, consults the calendar, and write the JCL for the first payroll job to the internal reader.

Audit Implications:

Auditors need to evaluate the way userids are assigned to batch jobs, the controls over the ability to write to the internal reader, and the use of the internal reader as a POE or Port of Entry.

Batch jobs are assigned userids defined to the security software, just like users logged on at terminals. If the JCL does not specify a userid for a job (on the JOB card, or for ACF2 on the LOGONID card), then the job inherits the userid of whoever submitted it. In some installations, production batch jobs do not have userids specified in the JCL. They then all inherit the same userid, that is: the userid of the job scheduling software. This means that for example, a payroll job looks like a marketing job to the security software. (We will give a Henderson Group golf shirt to the person who sends us the best example we receive by December 1, 2002 of how a marketing programmer could abuse this to give himself a raise without being caught. Please specify either Large or XXL shirt size and mailing address with your submission.) There will be better control if each application (Payroll, General Ledger, Marketing, etc.) has its own userid for all its batch jobs (specifying USER=PAYROLL for example on the JOB card). The security software can authorize the job scheduling software to submit batch jobs for USER=PAYROLL without specifying the password. Auditors will want to address how userids are assigned and how the security software supports this authorization.

Controlling the ability to write to the internal reader is the same as controlling who can submit batch jobs for execution. The security software will have rules to control this ability, as well as the ability to issue the TSO SUBMIT command. The auditor will want to evaluate these rules and how they are administered.

The internal reader is considered a POE, or Port of Entry, that is, a place where work can enter the system. It is possible to permit a userid to read or write a dataset only when that userid has entered the system through a specified POE. You might for example permit a userid to write a dataset, but only when coming through the internal reader. While this is not a common practice, the auditor should be aware that is possible, and occasionally is an elegant solution to a security administration problem.

10B) >>>>Started Tasks

Started tasks are similar to batch jobs: they have JCL specifying what programs to run and what files to make available to each program. However, instead of being written to the internal reader, started tasks are started by the operator command START. When the operator at a console in the computer room types START GEORGE, and hits the ENTER key, the operating system finds the JCL named GEORGE and executes it, similar to the execution of a batch job.

Since started tasks usually do not have JOB cards, the security software will have tables or other means of assigning userids to them.

The START command can be issued by programs too. For example, JES is a started task. When MVS starts up ("IPL"s or "boot"s), MVS issues a START JES command, along with several other START commands. With this in mind, you can now understand the sequence of events when the MVS system starts up. MVS starts executing, and then issues START commands to get other programs executing as started tasks. These started tasks include: JES, the security software, the job scheduling software, DB2 (described last issue), VTAM (described below in section 10C, and last issue), and others.

Audit Implications:

Auditors will want to consider controls over the following:

- Use of the security software to control who can issue the START command
- Assignment of userids to started tasks
- The ability to update the datasets where the started JCL is stored

10C) >>>>Online

Online work is work entering the system through a terminal. The key to understanding online work on an MVS computer is to understand that ALL communication through terminals is controlled by the VTAM (Virtual Telecommunication Access Method, described in Issue Number 02) software. Your terminal is not able to talk to any program unless VTAM permits it by defining a "bind" between the terminal and the program. Each program which VTAM permits to talk to terminals is called an "applid", short for "application identifier". Each applid must be defined to VTAM, and each terminal must be defined to VTAM before VTAM will allow them to talk to each other.

Audit Implications:

Auditors will want to evaluate:

- Whether every applid is required to call the security software to verify userids and passwords (as opposed to using hard-coded lists of userids and passwords which are not changed nor deleted when someone terminates employment)
- What controls VTAM provides over binds between terminals and applids
- The security implications of TCP/IP connections and the Internet

Stu Henderson, (301) 229-7187, stu@stuhenderson.com, www.stuhenderson.com