MANEWS 10

the Mainframe Audit News

August, 2008 Issue Number 10

==
==

Table of Contents

===
===

1) Scoping and Planning a Mainframe Security Audit

Planning and scoping an audit is always easier if you have a model to work with, and if the model breaks easily into two separate parts. For mainframe security audits, we have such a model, and it makes auditing much easier to organize.

Our security model consists of two parts:

A) the **Trusted Computing Base** (TCB), which is the security of the MVS operating system itself and

B) the **Q1Q2 Mechanism**. (which answers the two questions: **Who is this User**? And **Can He Do x?**) On mainframes, the Q1Q2 mechanism is always one of the three security software packages (RACF, ACF2, or TopSecret)

For this current issue, we will set aside the Q1Q2 mechanism (the part that checks userids and passwords and that controls who can access a given dataset) to concentrate on the TCB.

We will show you how the TCB works, what a "backdoor" is and why they are necessary, and then how to audit the TCB, You may think of the TCB as the "operating system security", as opposed to the Q1Q2 mechanism.

(We will address the Q1Q2 mechanism in a later issue, first explaining how it works, then giving you an audit program for it. The two parts (TCB and Q1Q2 depend on each other, so both need to be addressed eventually. It makes sense to audit one or the other part, not both at once.)

The TCB consists of two parts: the hardware controls and the MVS operating system itself.

There are two types of hardware control: some restrict what memory you can touch, others restrict what instructions you can execute.

On the mainframe (IBM, z/OS type computer), the two hardware controls that restrict what memory you can touch are **Protect Keys** and **Address Spaces**. The hardware control that restricts what instructions you can execute is called **Supervisor State**. (Supervisor State is comparable to kernel mode in UNIX or Windows. For you hardware geeks, to Intel ring 0.) Both UNIX and Windows use address spaces like the mainframe, but neither has anything comparable to Protect Keys (yet another advantage of the mainframe).

We will skip the details of how these work for another time. You should understand that the operating system (MVS, which is part of the package called z/OS) uses the hardware controls to build a virtual cage around each program executing on the computer. This prevents the programs from interfering with each other, and from interfering with MVS itself. IBM is so sure of the integrity of this cage that they document it (please see section 5 of this issue).

Occasionally it is necessary to give some program the privileges of MVS itself, in order to let the program perform some specialized function. You would want of course to have some means of knowing that the programs which have these privileges are “safe”, that is, that they don’t introduce security exposures to the system. Programs with these privileges are called “**backdoors**”.

There are a variety of techniques that a system programmer can use to grant a program such privileges, turning the program into an additional backdoor. Examples of such techniques include: exits, user SVCs, APF-authorization, functional subsystems, I/O appendages, and the program properties table. We will not describe these in further detail, but you should recognize the terms when you see them as the names of types of backdoors.

Backdoors are not a problem by themselves, so long as they are properly controlled. This control is the subject of your audit.

You job as an auditor is not to ensure that the hardware controls work, nor to ensure that the backdoors are safe. You are probably not competent to do either. Rather your job is to evaluate the controls available to systems programming management for them to know, and for them to be able to demonstrate, that the backdoors on the system are all:

- Safe
- Approved
- Protected Against Improper Modification

These form the basis of your control objectives. To prepare for your audit, you will need to identify the backdoors on the system. You will also need to list the security software rules describing who can update the datasets where the backdoors are defined, and whether successful updates to these datasets get logged.

To scope your mainframe audit, you need to know how many copies of the MVS operating system are in the data center. The common term for a copy of MVS is an "**MVS image**". If you ask how many CPUs (central processing units) are in the data center, the reply will often be something like "Oh, we have 5 CPUs, but they're configured into 15 LPARs and two sysplexes." An **LPA or logical partition** is a way that the hardware pretends that one real CPU is one or more "pretend" or "virtual" CPUs. This is sort of like VMWARE, but done in the hardware, and invented long before VMWARE. (Did we mention that the mainframe had both hardware and software virtualization before anyone else?) Each LPAR is like one pretend CPU, so it will correspond to one MVS image. A **Sysplex** is a collection of CPUs that are connected by fiber optic cable in such a way that they can exchange information, mirror databases, and back each other up. So if there are 15 LPARs, then there are 15 MVS images. We suggest that you limit your scope (at least at first) to one MVS image, usually an important, production one.

To learn the backdoors on that one image, you will need to execute special software like CA-EXAMINE (there are others available) OR to browse the parmlibs, that is the datasets where the backdoors are all defined.

So before your audit starts, get the names of the parmlibs, and browse access to them, or to a copy of them. Learn whether the security software is RACF, ACF2, or TopSecret, and get privileges that let you list dataset rules.

In a future issue, we will show you how to go about this in more detail. But for now, you have an idea of how the security works, your role in a mainframe security audit, and how to prepare for it.

<table>
<tr><th colspan="6">Mainframe Security Audit Model</th></tr>
<tr><td colspan="4">Trusted Computing Base
(this issue of this newsletter)</td><td colspan="2">Q1Q2 Mechanism
(future issues)</td></tr>
<tr><td colspan="2">Hardware Controls, Which Restrict:</td><td colspan="2">MVS Operating System Use the Hardware Controls for Two Purposes:</td><td>Q1 (Who is this User?) based often on userid and password</td><td>Q2 (Can this user do x?) where x is open a dataset, issue a transaction, or anything you want</td></tr>
<tr><td>What Memory You Can Access</td><td>What Instructions You Can Execute</td><td>Keep Users from Interfering with Each Other</td><td>Keep Users from Interfering with MVS</td><td colspan="2">Always One of These Three Software Packages: RACF, ACF2, or Top Secret</td></tr>
</table>

IBM’s Integrity Statement (please see section 5) is their assurance that the code delivered by IBM has no security exposures. So your audit just has to address the software added by the system programmers: the backdoors. You don’t review the added code: you evaluate the controls management has for them to know and to demonstrate (SOX, of course), that the added coded is safe.

==
==

2) Two Approaches to Mainframe Auditing

There are two extreme examples of approaches to mainframe auditing: checklist based and evaluation of adequacy of controls.

A) Checklist Based The auditor:

- Has a checklist,
- Does not understand the architecture of what he is auditing,
- Does not have nor understand control objectives for the audit,
- Collects data and interviews staff
- Does not have standards against which to audit
- Is not able to discuss how the controls (comparison to a standard) support the control objectives, nor how adequate the controls are
- Often was assigned the audit at the last minute, against his will, and over his objections that he is not qualified to perform it
- Still, continues to do his best on a very difficult assignment, working with what he has available, and trying as hard as he can to do good work

B) Evaluation of Adequacy of Controls The auditor:

- Understands the architecture,
- Knows where the controls are located on the architecture,
- Knows for each control what standard (if any) is being used
- Knows for each control what objective it is meant to support
- Knows how to get technical advice when he needs it
- Often was assigned to the audit because he has some general knowledge of the subject area
- Continues to do his best, working with what he has available, trying as hard as he can to do good work.

If the first type of auditor contributes less than the second, and is less happy, should we blame the auditor or his management?

==
==

3) Seminar Information and Miscellanea (Useful Articles, Proverb, Interesting Products)

3A) >>>>Seminar Information

The Henderson Group offers these "How to Audit..." courses :

- How to Audit **MVS, RACF, ACF2, TopSecret, CICS, DB2, and MQ Series Security** (November 17-20, 2008 in Clearwater, FL, and May 4-7, 2009 in Raleigh, NC)
- How to Audit **z/OS with USS, TCP/IP, FTP, and the Internet** (June 9-11, 2009 in Bethesda, MD) (a logical follow-on to the previous course)

To learn more about them, please go to

http://www.stuhenderson.com/XAUDTTXT.HTM

3B) >>>>Useful Articles

How to Secure Mainframe FTP
is described in an article at http://zjournal.tcipubs.com/issues/zJ.Dec-Jan08.pdf in the **zJournal**.

21 RACF Tips

RACF is of course the security software for mainframe computers (it competes with ACF2 and TopSecret.) The handout from a recent ISACA presentation **"21 Things You Didn't Used to Know About RACF"** is available at:

www.stuhenderson.com/XARTSTXT.HTM

If you are involved in RACF audits, you should take a look. If you're involved in ACF2 or TopSecret audits, then you should look too, since the concepts apply (with different buzzwords) to those software tools as well.

Tape Security

To learn more about why tape security requires special efforts, you might want to read this article titled "**Full Tape Security from Security Software and Tape Management Software**":

www.stuhenderson.com/TAPESEC1.PDF

3C) >>>>This Issue's Proverb of the Day

"There is ALWAYS another possibility right in front of us which we could see and seize, if only we would open our eyes. There is ALWAYS something going on with the person in front of us which we could see, if only we would open our eyes."

3D) >>>>Interesting Products

While we seldom recommend products, we think these two software products are worth your becoming familiar with. Both provide ways of monitoring changes to system datasets, including the datasets where the backdoors are defined. You might consider them as audit tools, as well.

- **Image Focus** from **NewEra Software** at http://www.newera.com or telephone (408) 201-7000 AND
- **eventACTION** (formerly **Change Action**) and **ussACTION** from Action Software at http://www.actionsoftware.com or telephone (905) 470-7113

==
==

4) What Mainframe Auditors Can Learn from the Credit Card Scandal

You have probably read about a number of retailers from whom thieves stole customer credit card information, with resulting publicity and possible liability. You may have wondered whether this applies at all to your mainframe audits. The thieves in these cases apparently used two techniques: listening to wireless (unprotected) transactions transmitted from a cash register to its server computer AND installing a sniffer program on the server computer to learn everyone's userid and password.

But this was with cash registers and servers, so it doesn't affect a mainframe audit, does it? Yes it does. Almost everyone logging onto a mainframe logs on from a personal computer and in one of three ways:

- through the Internet,
- by means of a dial-up modem, or
- through a LAN (Local Area Network).

The LAN is managed by a server computer and is subject to sniffer programs if no protection is implemented. It is also possible that one or more of the personal computers on the LAN has a wireless connection, or can be physically accessed by someone who wants to install a sniffer.

It seems ironic that one of the greatest security risks for mainframes is the Windows based LANS they connect to. And you should either address this in your mainframe audit, or specifically exclude it from your scope while suggesting that it be addressed in a future audit.

You address this risk by inquiring whether the LANs make use of the Kerberos or SSL (Secure Sockets Layer) or have some other means of protection. Ask your Windows and network auditors to help you test whatever controls are in place.

===
===

5) **IBM’s Integrity Statement**

Here is the text of IBM’s integrity statement for the z/OS system, including the MVS operating system, as found at

http://www-03.ibm.com/servers/eserver/zseries/zos/racf/zos_integrity_statement.html

This is IBM’s assurance that the code they deliver can be trusted, but with the exception of code added by system programmers (either home grown, or purchased software).

z/OS Statement of Integrity

> First issued in 1973, IBM’s MVS ™ System Integrity Statement, and subsequent statements for OS/390® and z/OS, has stood for over three decades as a symbol of IBM’s confidence in and commitment to the z/OS operating system. IBM reaffirms its commitment to z/OS System Integrity.
>
> IBM’s commitment includes design and development practices intended to prevent unauthorized application programs, subsystems, and users from bypassing z/OS security – that is, to prevent them from gaining access, circumventing, disabling, altering, or obtaining control of key z/OS system processes and resources unless allowed by the installation. Specifically, z/OS “System Integrity” is defined as the inability of any program not authorized by a mechanism under the installation’s control to circumvent or disable store or fetch protection, access a resource protected by the z/OS Security Server (RACF®), or obtain control in an authorized state; that is, in supervisor state, with a protection key less than eight (8), or Authorized Program Facility (APF) authorized. In the event that an IBM System Integrity problem is reported, IBM will always take action to resolve it.
>
> IBM’s long-term commitment to System Integrity is unique in the industry, and forms the basis of z/OS’ industry leadership in system security. z/OS is designed to help you protect your system, data, transactions, and applications from accidental or malicious modification. This is one of the many reasons IBM System z™ remains the industry’s premier data server for mission-critical workloads.

==
==

6) About the Mainframe Audit News; How to Subscribe/Unsubscribe

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.)

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 Subscribe, Unsubscribe, or Request Back Issues for 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 the phrase you want: SUBSCRIBE or UNSUBSCRIBE or BACK ISSUES: 1, 2