MANEWS Issue Number 26

the Mainframe Audit News

This newsletter tells you stuff you need to know to audit IBM mainframe computers running with z/OS and the MVS operating system. This issue we explore free tools IBM gives us for z/OS audit and security, as well as continuing our structured approach to the audit.

Table of Contents

1) Neat, Free, Often Overlooked Mainframe TCP/IP Security Tool

Most z/OS mainframe computers are connected to both TCP/IP and UDP/IP networks, and very often directly or indirectly to the Internet. We have learned the hard way that you can't rely on the security tools of the distributed network to protect the mainframe. However IBM provides us a free tool for z/OS security with TCP/IP. The tool is named Policy Agent, often abbreviated as PAGENT.

PAGENT provides a large number of security functions for TCP/IP, including:

- Encryption
- Intrusion Detection
- Blocking of Ports
- Packet Filtering
- IPSEC and VPNs (Virtual Private Networks)

What's important about the encryption If sensitive data is flowing across the network, including userids and passwords, then it needs to be encrypted. IBM has recognized that many programs need to encrypt the data they send over the network. Rather than relying on each program to be coded with its own encryption routine, IBM decided to have one, common, encryption routine, and to allow any program to invoke it. This centralizes encryption, reduces administrative overhead, and improves efficiency. (Note: we are talking here about asymmetric encryption, the kind that uses digital certificates. This is used over TCP/IP networks, for example when sending credit card information, or any sensitive data. The other kind of encryption (symmetric) is not commonly used with TCP/IP.)

IBM provides this common encryption routine by means of Policy Agent. It is called **AT-TLS**, for **Application Transparent Transport Layer Security**. (**TLS** is the successor to **SSL**, Secure Sockets Layer.) Any z/OS computer program connected to TCP/IP can use Policy Agent to implement this encryption. Any z/OS computer which does not take advantage of AT-TLS is likely either not encrypting everything it should or else encrypting very inefficiently.

How to address this in your audit. Ask whether Policy Agent is being used, and how. Ask for the output of the **NETSTAT** command to see what programs are using TCP/IP. Ask for the output of the **pasearch** command, which will tell you what policies are being implemented by means of PAGENT. Identify what organizational policies are in place to require encryption over each network connection with sensitive data. If there is a network connection with sensitive data that is not encrypted, this will likely represent significant risk, and often a violation of some regulation.

If the ports are not blocked (so that no program can open a network connection without proper approval), this will likely represent significant risk. The risk is that an unauthorized program could run on the computer and open a port, allowing an uncontrolled path into the system from the TCP/IP network. A real-life exploit using this exposure of uncontrolled outbound paths is described in

http://www.stuhenderson.com/Mainframe%20Audit%20News/MANEWS22.pdf

If PAGENT is not being used at all, even if you don't identify specific risk, you may want to suggest making use of it, in order to provide better security, reliability, and problem avoidance.

2) Automated Healthchecks

Some of the tests we would like to make as auditors are available for free, and automatically. IBM provides us with a set of Healthchecks with z/OS. Each check can automatically test a given condition, at a given time and frequency, and with a given response. For example, one of the checks tests whether system datasets (parmlibs, APF libraries, others) have correct access permissions.

There are sets of checks for RACF, for ACF2, for TopSecret, and for MVS security. (For example, one of the MVS security checks compares the setting of the ALLOWUSERKEYCSA(YES|NO) DIAGxx option to the IBM recommended setting of **ALLOWUSERKEYCSA(NO)**.)

What this means to the audit: Your audit can be simpler if you collect this information early on:

- Who is responsible for deciding use of the Heathchecker?
- What security checks are in place?
- For each check, how often is it made, what is the comparison, and what action is taken?
- Who is responsible for reacting to Healthchecks and what action is taken?

If the Healthchecker is not in use, you may consider making it an audit recommendation (not necessarily a finding though). The Healthchecker has several benefits beyond security, including performance tuning and problem avoidance.

The output of the security Healthchecks can supplement your work papers neatly. In any case, your audit will be more effective if you know what healthchecks the data center is using and what they are doing with the result.

3) More on Structured Audit Approach

In issue 24 of this newsletter (www.stuhenderson.com/Mainframe%20Audit%20News/MANEWS24.pdf) we described how to break IS security audits into pieces which don't overlap, but which add up to a comprehensive audit. This allows you to

scope and budget your audit intelligently, mapping to the available budget and resources.

The first piece described in that article addressed the question "***Can users access the system without being authorized?***". Here's how to break this piece into smaller pieces:

To answer this question, you need to identify all the paths into the system, and then to find out how well each path into the system is secured. To evaluate how well each path into the system is secured, you need to evaluate the methods used to secure, such as userids and passwords, and how well they are administered.

With z/OS, there is a standard list of paths into the system:

- **TSO** (Time Sharing Option, a programmers' workbench)
- **Batch job** (programs running in the background, not connected to a terminal)
- **Started Tasks** (programs running in the background started by operator command, each dedicated to some purpose, comparable to a daemon in UNIX)
- **USS** (UNIX System Services, formerly called OMVS)
- **TCP/IP (**including each of the daemon programs such as FTP)
- **CICS** (Customer Information Control System, where the majority of financial transactions in the world get processed, based on dollar amount)
- **Other Applids (programs with signon screens)**

For all of these paths into the system, the security software (RACF, ACF2, or TopSecret) should control who can use the path. The next page shows how you can break this question into smaller questions, and develop specific tests for each, whether RACF, ACF2, or TopSecret.)

An Example of Breaking a Basic Question into Its Parts

Question: ***Can Anyone Access the System Without Being Authorized?***

You can break this basic question into these parts:

A. Is a valid, approved userid required to access the system?

B. Does the security software reliably verify each userid attempting to access the system?

C. Does the security software control which userids can use each path?

These three questions add up to the basic question, without overlapping. Each can be tested separately. Each can be included in scope or excluded.

Each of these smaller questions can be further de-composed.

Some partial examples: **A.** can be broken into:

A1. Is there a formal definition of "approved" for a userid? ("No one is allowed to have a userid without the approval and annual re-certification of a Supervisor or higher.")

A2. Do the standards and procedures for userid administration) approving, deleting, re-certifying of userids and for password re-sets prevent anyone from having a userid without proper approval?

A3. Is someone responsible for each userid on the system (including its periodic re-certification and its deletion when employment ends?

For example, **C.** can be broken into:

C1. Is there a policy requiring the security software to control every path into the system (no hard-coded lists of userids and passwords permitted)?

C2. Are security software options set to protect each path? (**In RACF**, the APPL resource class for CICS, IMS, TSO, USS, and many other applids; the STARTED resource class for started tasks; BATCHALLRACF and XBMALLRACF for batch jobs; SYS1.UADS for TSO, protected attribute for userids automatically logged on. **In ACF2**, ABORT mode, JOBCHECK for batch jobs, STC OPTION for started tasks, UADS BYPASS, and flags in each user record indicating which paths the user is permitted to. **In TopSecret**, FAIL mode and FACILITY definitions. **Control over one userid submitting work on behalf of another userid** SURROGAT resource class; plus for ACF2: JOBFROM and RESTRICT with PGM and LIB; and for TopSecret: NOSUBCHK and XA ACID= in a user definition)

C3. Is there provision for approved, controlled access without a password (anonymous logon with FTP, PUBLIC in DB2, advertising messages over the Internet with the httpd daemon)?

You can add to these questions, break them down further, modify them to suit your taste, or take a completely different approach. Once you establish the control questions making up your audit, you can test each of them in several ways:

- Organizationally (is someone clearly responsible?)
- Procedurally (are effective procedures in place and followed?)
- Substantively (observations match approvals and baselines?)
- Analytically (what patterns are evident in the SMF data?)

For every potential finding, you will of course want to be able to state a meaningful risk, for example “As a result, it is possible for an unauthorized person to access the system under the authority of someone else and then copy or modify critical data improperly.”

Checklists, including the **STIGs** and **800-53** (see below to get copies), can help you to develop a complete list of tests and risks. These will help you to break the questions down even further, and to develop practical tests for each. These tests might for example include checking password options in the security software. Other checks you’ll see there include: checking to see: that all passwords and sensitive data sent over the network are encrypted, that: essential resource classes are active, and that USS security is reliable.

Appendices: Seminar Information and Miscellanea

Appendix A) >>>>Seminar Information

Henderson Group seminars are available for in-house as well as public sessions.

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

- How to Audit **z/OS with MVS, RACF, ACF2, TopSecret, CICS, DB2, and MQ Series Security** (May 10-13, 2016 in Raleigh, NC and Sept. 19-22, 2016, in Chicago)
- How to Audit **z/OS with USS, TCP/IP, FTP, and the Internet** (November 15-17, 2016 in Bethesda, MD), a logical follow-on to the previous course
- How to Audit **UNIX and Windows Security** (October 24-27, 2016 in Bethesda, MD)

To learn more about them, please go to

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

Appendix B) >>>>This Issue's Proverb of the Day

"*A test can only be interpreted sanely in the context of prior probabilities.*"
— "***The Laws of Medicine***" by Siddhartha Mukherjee

Appendix C) >>>>Useful Information

Here are more useful information sources to help you audit more effectively:

1. Free webinar on effective passwords: ***"Why Complex Passwords Should Be Fact and Not Fiction***" by Richard Faulhaber (presented and recorded before press time)
   To get the handouts or hear the recording:
   http://www.newera-info.com/RF1.html

New Era offers free webinars by top speakers, and free books to help you audit mainframes better. You can see the seminar schedule and get handouts from previous sessions at
http://www.newera-info.com/The-z-Exchange.html

To get the free books: http://www.newera-info.com/eBooks.html (Revised)

Book topics include:

- AE2 - z/Auditing Essentials - Volume 2 - The Taming of SETROPTS
- AE1 - z/Auditing Essentials - Volume 1 - zEnterprise Hardware - An Introduction for Auditors
- CICS Essentials - Auditing CICS - A Beginner's Guide
- What's New in z/OS V2R2

2. The NIST STIGs (Security Technical Information Guides) for various types of computer, including mainframes
https://web.nvd.nist.gov/view/ncp/repository

3. Useful guidelines for knowing that your InfoSec is comprehensive (Note especially Publication 800-53):
http://csrc.nist.gov/publications/PubsSPs.html#800-53

4. The current release of z/OS is 2.2. The previous releases are z/OS 1.13 and z/OS 2.1. The end of support for z/OS 1.13 is September 30, 2016. You can monitor end of support dates for IBM software at
http://www.ibm.com/software/support/lifecycle/index_z.html

5. An additional source of free, practical information on mainframe security and auditing, from a variety of sources:
http://www.stuhenderson.com/XINFOTXT.HTM

6. IBM z/OS manuals (including Healthchecker under "z/OS System-Level:)
http://www-03.ibm.com/systems/z/os/zos/library/bkserv/v2r2pdf/

7. IBM Multi-Factor Authentication Manual
http://publibz.boulder.ibm.com/epubs/pdf/azfug100.pdf

Appendix D) >>>>About the Mainframe Audit News; 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**, **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 **httpd daemon** software which connects a mainframe to the Internet. (Note, we expand each of these acronyms and explain how the software works in past and future issues.) The MA News is for auditors who are new to IBM mainframes, and also for experienced MVS auditors who want to keep up to date with the latest developments. We will not make the list of subscribers available to anyone else for any reason.

To Subscribe or Unsubscribe Click on
http://www.stuhenderson.com/subscribe.html .

To see Back Issues: www.stuhenderson.com/Newsletters-Archive.html

Feel free to contact us at (301) 229-7187 or
stu@stuhenderson.com.