

MAINFRAME SECURITY:

BACK TO THE FUTURE?

BY BARRY SCHRAGER

In the early '70s, mainframe access control became a frequent topic of discussion. The SHARE Security Project was formed in 1972, and its membership was, interestingly enough, comprised of mainly universities, service bureaus, and Department of Defense installations. Big business and financial institutions were noticeably absent.

But this group of installations had a pressing problem: to segregate data between persons or groups and to control how that data was disclosed or updated. Up until then, there really was no usable method for achieving this—dataset passwords were the only control and they were difficult to administer and enter for access permission.

So the SHARE Security Project met over a period of a year and came to the conclusion that to provide true data security, the operating system (at that time it was OS/MVT) must provide an assurance of system integrity; this was defined as the inability of a user to bypass the formal interfaces of the operating system to >



Upgrade Your IBM z/OS Security Now! Secure Mainframe Access with SSH Tectia

The only enterprise-class SSH for:

- Securing transparently any TN3270 connection
- Securing file transfers with full MVS support
- Optimizing encryption with hardware acceleration

... from the original developers of SSH.

IBM Server *Proven*[™]

For further information, go to www.ssh.com/zjournal/



Original. Secure. Supported.

obtain access to data, alter the operation of the computer system, etc.

In late 1973, IBM announced OS/VS2, which included an operating system integrity statement that let the project concentrate on data access control issues and develop requirements for IBM such as:

- A centralized installation replaceable security system through which all system and application delivery systems such as CICS could call for authorization requests (implemented by IBM as its external security manager)
- Dataset protection by default (first introduced by ACF2 in 1978)
- Algorithmic grouping of users and resources (implemented by ACF2 pattern masking and RACF generic profiles)
- Protection of logical resources (CICS transactions, etc.)
- Designated interface programs (RACF PADS [Program Access to Data Sets], ACF2 Program Pathing)
- Secure journaling facility (SMF)
- Support for additional identification processes (OIDCARD, secure ID cards, etc.).

When RACF was introduced in 1976, many of these requirements were lacking. ACF2 was developed and introduced in 1978 in order to provide for these features, and eventually, RACF also incorporated this functionality. Top Secret was introduced in 1981. Currently, both ACF2 and Top Secret are owned by CA and are named CA-ACF2 and CA-Top Secret.

These security systems provide a centralized service that all system and application services can invoke for authorization and authentication requests. With all invokers using this centralized service instead of providing their own security mechanisms, there is only one security image that has to be administered and controlled for the entire z/OS system.

These systems established the precedent for today's data security standards and set extremely high standards as to what level of security could be achieved.

Unfortunately, IT has changed over the last 25 to 30 years. Back then, these systems had hundreds of users, thousands of datasets, and hundreds of megabytes of data storage. Mainframes were the data processing vehicle for companies. Now, many systems have more than 100,000 users, millions of

datasets, and terabytes of data storage. In many companies, there are thousands of Linux, Unix, and Windows servers both processing information and passing requests through to the mainframe processors.

Not too long ago, some believed the mainframe was quietly going away. Obviously, that isn't the case. Mission-critical applications still run on the mainframe and are servicing e-commerce. In fact, according to recent IBM financials, mainframes are experiencing greater than a 15 percent growth in installed capacity. However, as we detail in the following paragraphs, there are many issues facing today's security officer:

No one remembers why a security officer or administrator did something 20 years ago: There's no built-in documentation or incident support in existing security systems, and these people have retired or moved on to other jobs in other companies. Nothing can be done about the past, but mainframes aren't going away. What assurance is there that this problem won't continue to persist?

Incident support should include security authorization requests, and incidents should be easily created for follow-up on a security event such as an access violation or logging. The security officer's follow-up notes can then be made part of and associated with the security event.

It should be easy for a security officer or administrator to open an incident, add information about the request, authorizations, include emails, etc., and relate it to a specific authorization or administrative change for security incident follow-up, etc. For incidents that can be related to categories such as the Health Information Portability and Accountability Act (HIPAA), Sarbanes-Oxley Act (SOX), Gramm-Leach-Bliley (GLB), etc., there should be a way to categorize the incident so when the auditor requests all incidents relating to one of these categories, the request can be easily satisfied.

Security access definitions are a mess and need to be cleaned up: Security officers and administrators are afraid to try cleaning them up and make them follow company policy. They're afraid the business will stop because they didn't allow a critical job or transaction to execute, but current regulations require review and cleanup of these definitions. Security systems must provide the ability to test new security

definitions in the production environment, so the security officer can feel comfortable revising security definitions without risking production interruptions.

Security systems must monitor new security definitions to ensure they follow company policy: Security systems must provide an autonomic capability to raise the quality of the security definitions; for example, by recognizing conflicting or undercut access permissions and new privileged user attributes and periodically bringing them to the attention of the security officer. Also, as the security officer cleans up the access permissions, or the security administrators add new access permissions, the autonomic portion of the security system must recognize changes that will introduce new exposures and either immediately raise them with the security officer or even automatically roll them back.

Security systems also must diagnose and assess security issues based on usage and periodically provide the security officer with suggestions on how to fix them. For example, it would be nice to know that a RACF group hasn't been used for access in the last three months, or that a list of specific users who are connected to the group hasn't used it for access in the last three months. Then the security officer can decide to clean up the access permissions or leave them alone awhile longer.

There are many security violations and logging messages: The number of security violations and loggings used to be small. But, as the number of datasets, transactions and users has skyrocketed, so has the number of violations. Today's security officer can't review them over his or her first cup of coffee each morning and often faces the equivalent of an eight-inch thick report each morning. One person (or even a few generalists) can't adequately monitor the security events happening in today's environments.

Security events must be able to be categorized based upon importance in addition to category or regulatory—such as PAYROLL, NUCLEAR, SOX, HIPAA, GLB, etc. Security systems must be able to produce and distribute security reports via email to interested parties based on their events of interest. The interested parties know what events pose an imminent danger.

Security officers and other interested parties must be notified immediately when a critical event occurs:

Security systems must be able to alert interested parties via email or SMS message to their cell phones immediately when a critical event occurs. For example, this could include everything from a security violation in attempting to access the nuclear waste shipment schedule to two or more consecutive invalid password attempts by a user with special privileges. Authorized individuals must be able to subscribe to these kinds of events and indicate the mechanism for alerting them.

Security administrators need a user-friendly graphical interface that leads them along and helps them avoid mistakes: Initially, the security administrator position wasn't required; the administration was handled by the security officer, who normally came from an applications development or operations position. Now, the security officer is assisted by several security administrators who handle the routine tasks of accepting requests for additional users or access, obtaining proper approvals and entering the required new access permissions. These new security administrators almost always come from clerical positions. Unfortunately, the native interface to current security systems is a terminal emulator, emulating a terminal that's no longer being manufactured and using a programmer interface rather than a clerical interface.

The number of security administrators familiar with the mainframe 3270-style interface has significantly dwindled, but the pool of potential security administrators who understand a graphical, Web-based interface is massive. So, security systems must offer a graphical user interface that helps prevent mistakes by providing drop-down lists, type-ahead technology, hover help text, etc., to help security administrators do their job effectively and with few errors.

Security systems must provide support for routine security administrator or help desk tasks: Password reset calls to security administrators or help desks are costly. Security systems must provide an interactive, secure password reset capability.

Relational databases are normally used for reporting but the current products don't directly support them: Current security systems recommend that control and event data be unloaded from their proprietary file formats and SMF and then reloaded into a relational database for reporting. Then, these databases can be used for reporting.

The benefits of using relational databases can be vast, especially if users and events are categorized by security category (e.g., HIPAA, SOX, etc. and level of importance (e.g., critical, important, etc.). Then, simple SQL or a multitude of reporting tools can be used to extract and report on what's been happening in the enterprise's security. The often unrecognized advantage of using a relational database is that the mainframe database packages support Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) access. So, from a server or even the security officer's PC, the database can be interactively interrogated to extract the information needed.

An example might be a security officer who notices a violation attempt by an individual against sensitive data. The interactive capability provided by server or PC access would easily allow the security officer to, with a few mouse clicks, inquire about all the access violations and loggings generated by that individual over the last week. If this is significant, the interactive report could easily be packaged into an Adobe Acrobat PDF file and attached to an email sent by the security officer to the individual's supervisor, asking for an explanation. This process is so easy to do via the supporting server or PC, and so powerful that all security systems should be providing this capability.

Unfortunately, using the current system's unload and reload process means the data is almost immediately out-of-date. Security systems must either use the relational database as their primary data store or find a mechanism to keep a relational database in synch with its proprietary database and generated SMF event records.

Mainframe-style security controls must be extended to the enterprise: It's not unusual for the mainframe to be a part of a complex with thousands of Linux, Unix or Windows servers. Although analysts say that about 75 percent of critical business data still resides on the mainframe, a significant amount of access to that data is via these distributed servers. A mechanism must be created to administer security as a single or small number of images using similar or, better yet, the same concepts and controls.

Some large organizations are moving their distributed security administration under their mainframe security organizations because mainframes have had unsurpassed security implementa-

tions for decades and mainframe security officers fully understand the controls necessary from an organizational perspective. However, the mainframe-style security controls aren't available on the distributed servers, so these organizations lack the tools they're familiar with and which have worked successfully for them.

This situation isn't dissimilar to what existed 25 years ago, before IBM's implementation of the centralized External Security Manager. Each delivery system had its own security system and database. There was no consistency in security, there were a multitude of different security definitions and databases, and it was difficult to even understand what the overall security was. The implementation of the External Security Manager provided the capability for a single security image. Something similar is needed for today's enterprise collection of both mainframes and many other servers. Of course, since this situation involves multiple systems, any solution has to take into account data caching to reduce network requirements and improve response time and provide for database redundancy and failover.

So, even though the mainframe security systems have worked well over the last 25 years, there are issues facing today's security officers that have been brought on by the:

- Massive increase in the size of today's systems
- Size of the distributed, non-mainframe, security environment
- Departure of the people who knew the system well and understood the security systems
- Influx of new people who really did not grow up in the mainframe environment.

There's still much to be done to keep mainframes, and the rest of the enterprise, as secure as they were 25 years ago. Security officers will continue to have a great challenge to keep everything in control. **Z**

About the Author

Barry Schrage founded the SHARE Security Project in 1972 and has been a leader in data security since. In 1978, after he felt IBM didn't deliver a data security product, RACF, that met the SHARE requirements, he founded SKK, Inc., and designed and was the primary author of ACF2. He previously was vice president of JME Software, where he led the design and development of the Deadbolt security product. He is currently chief security architect for Vanguard Integrity Professionals. Email: Barry.Schrage@go2vanguard.com