THE RESOURCE FOR USERS OF IBM MAINFRAME SYSTEMS

WWW.ZJOURNAL.COM

APRIL/MAY 2007

Z JOURNAL

INSIDE Z

RACF Self-Assessment: 3 Critical Areas to Examine

A tci PUBLICATION

RACF Self-Assessment: 3 Critical Areas to Examine

By Robert S. Hansel

Most large organizations continue to rely on the z/OS mainframe platform for support of their mission-critical core businesses and internal administration. So the security of the mainframe remains vital. Many z/OS installations rely on IBM's RACF to protect their systems. RACF is a feature-rich product that can fully protect mainframe resources *if* implemented properly.

This article covers three areas of concern with RACF implementation—SURROGAT profiles, storage administration authority, and excessive access granted by default—and offers suggestions for addressing them.

SURROGAT Profiles

Surrogate authority lets a user perform tasks using another user's identity and authority. In RACF, profiles in the SURROGAT class govern the use of surrogate authority. There >

are four types of SURROGAT class resources:

1. ***userid*.SUBMIT** lets a user submit a batch job with the identity and authority of user *userid* simply by coding USER=*userid* on the JOB card and without having to code the ID's password. This authority is typically used to let a process, such as a job scheduler, automated operations task or CICS region, submit jobs with application batch IDs.
2. ***userid*.DFHSTART** lets a user start a CICS transaction that will run with the identity and authority of user *userid*. The *userid* is specified in the EXEC CICS START command in a program. Selection of the ID is under the control of the programmer, not the user. Generally, only certain, designated IDs created specifically for this purpose should be used for started transactions. This check is made only if the CICS Systems Initialization Table (SIT) parameter XUSER= has been set to YES; otherwise, the action is allowed without any checking. The CICS systems programmer is usually responsible for setting this option.
3. ***userid*.DFHINSTL** lets a) a CICS region use user *userid* as its default ID, b) a CICS region use user *userid* to execute its Program Load Table Post-Initialization (PLTPI) programs, c) a CICS administrator assign user *userid* as the pre-set ID for a terminal, and d) a CICS region or a terminal user use user *userid* for Automatic Transaction Initiation (ATI) (a.k.a. trigger transactions) associated with a transient data queue. The CICS default ID and PLTPI ID are assigned in the SIT parameters. Terminal pre-set IDs are defined in the CICS Systems Definition (CSD) file. ATI IDs are assigned in either the CSD or an EXEC CICS SET TDQUEUE program command. Ideally, only certain, designated IDs should be used for these various purposes. Again, these checks occur only if the SIT parameter XUSER= is set to YES. The CICS systems programmer is usually responsible for setting this and most of the other options previously mentioned.
4. **BPX.SRV.*userid*** lets a user perform a z/OS Unix switch user (su) to change identities to user *userid* without having to enter the user's password. The initiating user temporarily becomes user *userid* and acquires all the user's authorities to include both Unix and z/OS access. This is primarily intended for use by z/OS Unix processes in conjunction with various default IDs.

Common problems that occur with SURROGAT profiles are threefold:

1. CICS XUSER= SIT parameter is set to NO (the default value). This enables anyone with control over the CICS commands and parameters to assign any ID of their choosing and acquire whatever authority it might have.
2. CICS-related SURROGAT profiles are too generic. Due to an apparent lack of understanding of their purpose, profiles of *.DFHSTART and *.DFHINSTL are found in many installations.
3. SURROGAT *userid*.SUBMIT profiles let unprivileged terminal users (as opposed to process-type users) use IDs with higher-level permissions and privileges such as OPERATIONS authority. One installation had a single ID with OPERATIONS authority and more than 150 users with permission to submit jobs via a *userid*.SUBMIT profile. This is especially problematic when the installation has defined a profile such as *.SUBMIT or **. The latter could conceivably cover *all* forms of SURROGAT resources. Preferably, terminal users should never be given SURROGAT authority to any other IDs, but instead should perform tasks under their own IDs, which would be given whatever access authority they need. Adhering to this practice also makes it easier to trace the actions of a terminal user, since all work would be attributed to the user's own ID.

Storage and Catalog Administration Controls

Storage administration is the management of data residing on DASD and tape; it entails such tasks as moving, copying, backing up, restoring, defragmenting, and compressing datasets and formatting volumes. Catalog administration involves managing these catalogs and catalog entries. These responsibilities are often assigned to the same individual. Some RACF authorities that allow administrators to perform their duties include:

- **OPERATIONS authority:** The OPERATIONS attribute grants a user ALTER access to any dataset, DASD volumes (DASDVOL), tape volumes (TAPEVOL), or any other resource whose associated class has been defined to honor this attribute *unless* the user has been permitted access at some level less than ALTER. The OPERATIONS attribute can be assigned either to the user's ID, in which case it applies systemwide, or to one or more of the user's group connects, limiting authority to just the scope of those particular groups. OPERATIONS authority was RACF's original mechanism for empowering storage administrators. Unfortunately, OPERATIONS authority lets the user access and manipulate data, not just manage it.
- **DASDVOL class resources:** Profiles in the DASDVOL class allow use of specific functions, such as backup and restore of all the datasets, on entire DASD volumes using certain storage administration utilities such as DFSMSdss's ADRDSSU without requiring access permission to each individual dataset. DASDVOL was introduced early in RACF's evolution. It offered a means of empowering storage administrators without having to grant them OPERATIONS authority. DASDVOL predates the introduction of System Managed Storage (SMS) and doesn't apply to SMS-managed DASD volumes. Note that ALTER access lets you delete any dataset on a non-SMS-managed volume.
- **Catalog dataset profiles:** The catalogs let users find and reference datasets without having to know the DASD volume where they reside. READ access lets users search for and interrogate catalog entries. UPDATE lets them create and delete entries in conjunction with dataset creation and deletion. ALTER access lets a catalog administrator manage the catalogs. Catalog ALTER access provides the ability to delete any SMS-managed dataset *without* needing ALTER access to the dataset itself.
- **FACILITY class STGADMIN resources:** STGADMIN-prefixed resources determine whether a user can perform individual functions associated with the various utility programs used for storage and catalog administration. Introduced in conjunction with SMS, they grant authority over SMS and non-SMS-managed data. These resources also make it

possible to delegate specific administrative functions to various user groups. For example, RACF administrators can be given READ access to STGADMIN.IGG.DEFDEL.UALIAS to define aliases for new Time Sharing Option (TSO) users without having to be given UPDATE access to the master catalog. The number of STGADMIN resources has grown over time to serve as a full replacement for OPERATIONS and DASDVOL authority. As recent as z/OS 1.5 and 1.6, new resources were added for ICKDSF and DFSMShsm, respectively. These new additions have all but eclipsed the need for DASDVOL profiles. You might still need them for controlling Innovations' Fast Dump Restore (FDR) product. Of greatest interest are the resources whose names begin with STGADMIN.ADR. STGADMIN. They govern the use of the ADRDSSU utility ADMINISTRATOR keyword. A user allowed to use this keyword with a specific function can perform the related action on every dataset in the system. This is essentially the equivalent of having OPERATIONS authority and DASDVOL access. For more information on FACILITY Class profiles, see "Security's Multi-Purpose FACILITY Classes" in the April/May 2006 issue of *z/Journal*.

- **ISMF PROGRAM class resources:** Interactive Storage Management Facility (ISMF) is the ISPF menu tool for managing data. It provides capabilities similar to those offered by the data management utilities. However, the programs that comprise ISMF don't check STGADMIN resources or otherwise control who can use them. Instead, the RACF administrator must protect the individual ISMF programs with profiles in the PROGRAM Class. (See the IBM *DFSMS Storage Administration Reference* manual for a list of programs.)

Author's Note: Having to control ISMF with PROGRAM class profiles is cumbersome and error-prone (e.g., failure to cite the correct programs and libraries). Please contact your IBM representative and request ISMF be re-designed to check STGADMIN resources to govern their use.

The robustness of your z/OS security is determined by the effectiveness of your implementation of RACF.

Excessive assignment of OPERATIONS authority almost always tops the list of common problems with RACF implementations. OPERATIONS authority certainly enables systems programmers, storage administrators, and even RACF administrators to do everything they presumably need to do. Unfortunately, it lets them go far beyond their needs with little or no limitation. Being one of the original RACF features, OPERATIONS was assigned to their IDs way, way back when, and once acquired, users are reluctant to relinquish it. OPERATIONS authority, however, is both inefficient and risky. If used exclusively for data management, bulk administrative actions result in a RACF access check for every individual dataset affected and, if OPERAUDIT is enabled as it should be, an SMF record is generated for each access as well. Conversely, a single check to a STGADMIN.ADR. STGADMIN profile may be sufficient to perform the entire task without another call to RACF.

Think of OPERATIONS as a double-edged sword. It will let you modify or delete almost any dataset, including ones you didn't intend to affect. Remember what ALTER access to the catalogs allows; OPERATIONS may give you that ALTER access. It's similar to Unix uid(0): if you don't have it, you can't shoot yourself in the foot.

Replacement of OPERATIONS with STGADMIN profiles should be a priority in every installation. One successful approach is to:

- Grant the primary ID of the OPERATIONS users all the other required forms of storage administrative authority
- Give them an alternate ID with OPERATIONS authority
- Remove the attribute from their primary ID.

This tends to placate them until they discover they don't really need OPERATIONS. Once they're satisfied, these authorities will meet their needs, and the alternate OPERATIONS ID can be eliminated.

Tip: To purposely block the use of OPERATIONS authority, do the following. Create a group (e.g., #NOOPER), connect all the OPERATIONS users to this group, and then permit this group access to the profiles guarding the resources you want to restrict. The access authority of the OPERATIONS users will be capped at whatever access level you specify in the permit. This is particularly effective for guarding catalog and DASDVOL resources.

Implementing STGADMIN profiles is clearly beneficial, but great care and caution are required in setting them up. Default access is frequently set too high, whether Universal Access (UACC) or permission to ID(*), thus giving everyone in the system the equivalent of OPERATIONS authority. The same is true for DASDVOL profiles for installations that use them. Only a few of the STGADMIN profiles are appropriate for granting all users access by default. Contrary to what is found on occasion, these profiles should never be put in WARN mode. Last, don't forget to protect the ISMF programs.

Tip: To perform high-powered ISMF functions, users must make themselves an "Administrator." They do so by changing their "user mode" via a series of ISMF panels that alter the user's profile options. The act of changing oneself into an administrator turns on a bit in the user's own ISPF profile. The ISMF program that turns on this bit is DGTFPF05. If you need to control the ISMF programs and don't yet have the

time to fully protect them, a quick fix is to define a profile to protect this one program. Granted, it doesn't restrict anyone who has already turned this bit on nor would it stop someone from manipulating his or her own ISPF profile (if they could figure out which bit to activate), but it's a start. This is a stopgap measure, not a long-term solution.

Excessive Access Granted by Default

RACF guards only those resources you've told it to protect and only to the extent you specify. Here are some issues to consider in determining if you're giving away too much:

- **Undefined resources (i.e., no profile or inactive class):** With datasets and many resources, no guarding profile means wide-open access. Unless the associated resource class issues a "not-authorized" return code of 8 for an unprotected resource (e.g., JESSPOOL), RACF assumes you don't care if you don't define it. A few resource managers take it upon themselves to protect resources when RACF doesn't. CICS, for instance, forbids the use of transactions not defined to RACF. Some DFSMS utilities don't perform certain functions unless RACF has given explicit approval. RACF administrators can address undefined dataset protection through activation of the SETROPTS PROTECTALL option. This option, however, is somewhat misnamed. It doesn't really require the data to be protected; it merely requires it to be covered by a profile. The profile could have a UACC of ALTER, and this would satisfy PROTECTALL.
- **Universal Access (UACC):** UACC is the default access all users are given—even those users unknown to RACF.
- **ID(*) access:** ID(*) represents all users known to RACF. When permitted access in lieu of using the UACC, a prospective user must have first been authenticated by RACF before being allowed access. This is a minor step up in protection from UACC. **Note to auditors:** Remember when you told the RACF administrators they had to set all the UACCs to NONE? Unfortunately, when they "complied" with your audit requirement, they simply granted ID(*) whatever access level had been set for the UACC. Net change in protection: next to nil.
- **Tape dataset protection (TAPEDSN):** RACF does not, by default, protect tape datasets. You need to tell it to do so by activating the SETROPTS TAPEDSN option. Certain combinations of CA-1 security options will serve the same purpose as having TAPEDSN active. New z/OS 1.8 parameters in the DEVSUPxx member of PARMLIB can ease the transition into full activation of TAPEDSN if it's not active. Once tape protection is implemented, ensure facilities that could be used to circumvent it are controlled. These include Bypass Label Processing (BLP), which can be restricted with FACILITY class profile ICHBLP, and tape management system-specific profiles governing the use of EXPDT=98000 to bypass tape dataset name verification.
- **Global Access Table (GAT):** The GAT consists of a list of resources everyone is permitted to access without requiring a full interrogation of the corresponding RACF profile. The GAT is constructed from profiles in the GLOBAL class. Each profile is itself a class name (e.g., DATASET). In each profile is the list of resources and level of access at which everyone is granted instant access. For more information on the GAT and other RACF performance tuning options, see "10 Ways to Improve RACF Performance" in the October/November 2006 issue of *z/Journal*.
- **WARNING:** WARNING (a.k.a. WARN mode) is a profile option intended for *temporary* use in testing. Profiles in WARNING won't fail a user's access request even if the user doesn't have sufficient access permission. If the user doesn't have permission, RACF allows the access anyway (at up to and including ALTER level) and merely generates a console message and SMF record. Resources covered by profiles in WARNING aren't protected.

Unprotected resources continue to plague most implementations. In particular, classes related to OPERCMDS and JES resources are often inactive. A major area of concern is FACILITY class resources. An amazingly wide array of resources can be protected using this class, but few installations know they exist.

Tip: To see what FACILITY class resources are being used in your environment, try turning on SETROPTS LOGOPTIONS(ALWAYS(FACILITY)). In some installations, doing this may generate an excessive number of SMF records, but most will find it acceptable. What it will reveal is sure to be most enlightening.

In the realm of excessively high UACCs and ID(*), focus most closely on operating system-related datasets. Of particular concern is default access of UPDATE or greater to Authorized Program Facility (APF) libraries—a common problem. The same goes for the linklist libraries, SMF archives, and Job Entry Subsystem (JES) spool and checkpoint datasets, to name a few. Of equal importance is ensuring default access is NONE for the RACF databases and any backups containing a copy.

Whereas tape dataset protection is usually active, effective control of the means of circumvention isn't. Ensure BLP and use of EXPDT=98000 are strictly controlled.

While intended to serve as a performance enhancement feature, the GAT is often a security hole. Access permitted by the GAT overrides restrictions set in the profile. For instance, an entry in the GLOBAL DATASET profile of SYS1.**/READ would allow all users READ access to RACF databases protected by profile SYS1.RACF* with UACC of NONE. GAT entries undercutting profile protections is a common problem, as is assigning WARNING to highly sensitive resources and leaving it on for years. Unless your installation generates and reviews daily monitoring reports of WARNINGs and weekly listings of profiles in WARN mode, you're apt to overlook them.

Conclusion

RACF can provide excellent protection if properly implemented. Common areas of concern can be found in SURROGAT profiles, storage administration authority, and excessive access granted by default. Future articles will cover other areas of concern.

You now have the opportunity to address these issues before they become a problem. Remember, your auditor may have read this article, too, so be prepared for questions. Z

About the Author

Robert S. Hansel is founder and president of RSH Consulting, Inc., a firm specializing in RACF. He has worked with RACF for more than 20 years in the roles of manager, administrator, auditor, technician, instructor, and consultant. He is a frequent speaker on numerous RACF topics and supports several of the RACF Users Groups.

Email: R.Hansel@rshconsulting.com

Website: www.rshconsulting.com