THE DEMISE OF THE UNIX DEFAULT

BY ROBERT S. HANSEL

z/OS v1.13 will be the last release to support BPX.DEFAULT.USER, a feature used by nearly 90 percent of sites that use RACF, according to a recent survey conducted by RSH Consulting. What does this mean for these users? Here, we examine the tasks and considerations related to its replacement.

Many z/OS-based TCP/IP applications use sockets (local IP address plus port number). An example is File Transfer Protocol (FTP). For a z/OS user to access such an application, the user requires a UNIX User Identifier (UID) and the user's logon group requires a UNIX Group Identifier (GID).

In RACF, you can assign a user a UID and a group a GID by adding an OMVS segment to each of their respective records. A segment is an extension of a RACF record that provides attributes required by a specific application, which in this case is z/OS UNIX.

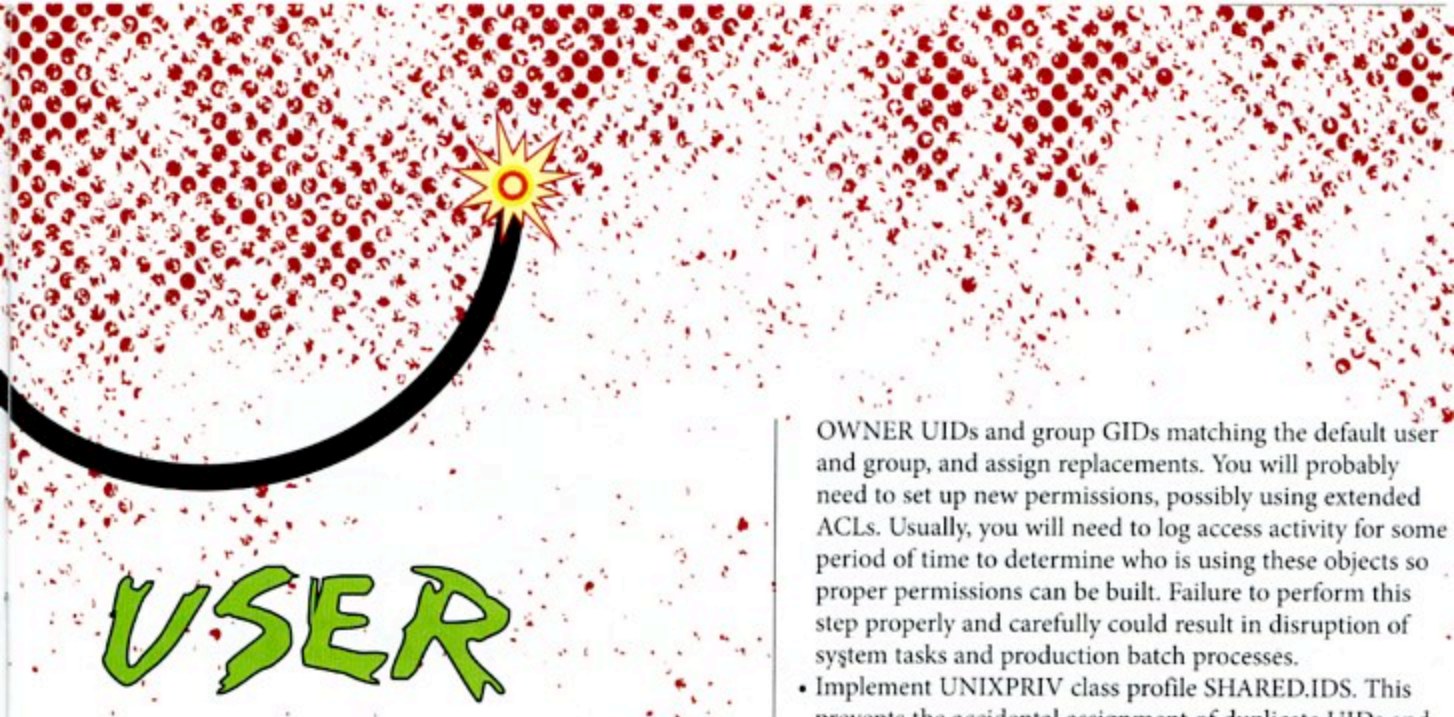
As IBM began rolling out TCP/IP applications in the mid-'90s, it was concerned that clients would balk at having to assign tens of thousands of OMVS segments to their users and groups. To ease the implementation of these applications, IBM introduced a feature to assign a temporary default UID or GID to any user or group lacking a UID. This feature first appeared in OS/390 v2.4 (circa 1997). It was implemented as FACILITY class profile BPX.DEFAULT.USER. The APPLDATA field of this profile identifies a user and group whose respective UID and GID are to be used as the default user and default group. The default IDs were open for use by any and all types of users, including technical support staff, started tasks, customers, production batch IDs, business partner FTP IDs, etc.

Unfortunately, this feature hasn't been a panacea. It essentially creates a shared ID, which security professionals and auditors generally don't consider to be a best practice. UNIX files or directories any user creates using the default user UID are accessible to all such users. Users of the default user UID are unable to use some UNIX functions, and in a few instances, z/OS UNIX can't properly provide users with the default user UID when performing certain tasks. In view of these drawbacks, IBM has decided to discontinue this feature and require that all users and logon groups be assigned individual UIDs and GIDs via OMVS segments.

Now that you know what has to be done and why, here are the steps you can take to replace BPX.DEFAULT.USER with individually assigned UIDs and GIDs. The size of this effort will depend upon the extent you're currently relying on BPX.DEFAULT.USER. The following road map will assist you in making the transition and minimize the possible disruptions this migration might cause.

Step 1: Set Standards

Before launching headlong into the task of assigning UIDs and GIDs, decide what numbers you want to assign. For users, consider establishing ranges of numbers that represent employees, contractors, system IDs (started task and batch IDs), and customers. Make the UID recognizable for ease of administration. For employees, consider



incorporating the person's HR employee number in the UID, assuming it isn't a Social Security Number. For instance, if employee numbers are six digits, prefix them with 10 to form the UID (e.g., employee number 007654 becomes UID 10007654). For groups, set aside different number ranges for default logon groups and access granting groups. For user default logon groups, you might want to incorporate HR department or cost center codes into the GID in a manner similar to what you might have chosen for UIDs.

Step 2: Prepare for ID Assignment

Here's a list of tasks necessary to properly prepare for assigning UIDs and GIDs:

- Verify the RACF database has sufficient space for all new OMVS segments.
- Convert the RACF database to the Application Identity Mapping (AIM) structure. This will let you use some of the new, optional RACF features designed to replace BPX.DEFAULT.USER. (According to an RSH survey, a mere 50 percent of installations use AIM.)
- Activate the RACF class FSSEC to enable use of extended Access Control Lists (ACLs). With extended ACLs, you can give multiple users and groups access to a file or directory. You may find the use of extended ACLs necessary to replace the shared use of the default user. (According to an RSH survey, only 40 percent of installations have activated this class.)
- Search all UNIX file systems for file and directory OWNER UIDs and Group GIDs that no longer have a corresponding RACF user or group; replace these IDs with valid ones. Failure to perform this step could result in assigning a UID to a user that matches the UID on a file or directory created by a former user who had the same UID. In that case, the new user would instantly inherit ownership of these objects.
- Search all UNIX file systems for file and directory OWNER UIDs and group GIDs matching the default user and group, and assign replacements. You will probably need to set up new permissions, possibly using extended ACLs. Usually, you will need to log access activity for some period of time to determine who is using these objects so proper permissions can be built. Failure to perform this step properly and carefully could result in disruption of system tasks and production batch processes.
- Implement UNIXPRIV class profile SHARED.IDS. This prevents the accidental assignment of duplicate UIDs and GIDs; it requires implementing the AIM structure.

Note: There's a performance enhancement feature that lets UNIX make some access decisions on its own (without calling RACF). The feature is implemented by defining FACILITY class profile BPX.SAFFASTPATH. Unfortunately, in bypassing RACF calls, logging of such access is also skipped. If you need to turn on logging to track user activity, delete this profile and deactivate its use with a SET OMVS operator command. Failure to do so might result in crucial access activity being missed and not accommodated with replacement permissions.

There are some additional preparatory tasks that, while not essential to this effort, are strongly recommended. They involve cleaning up the UNIX/RACF environment and tightening security. These tasks mesh nicely with the other preparatory work:

- Eliminate Write (w) permission granted to OTHER. Access granted to OTHER is open to everyone. Write permission to a file lets anyone change the contents of the file. Write to a directory lets anyone create a file or subdirectory or delete an existing file or subdirectory. OTHER Write access to directories opens the door for users of the default user UID to create objects, which only exacerbates the replacement problem. You will need to log activity to analyze and design replacement access permissions, again using extended ACLs.
- Verify existing system IDs are still needed before assigning them UIDs.
- Verify user UNIX home directories are still needed and valid.
- Check the validity of SURROGAT BPX.SRV.userid profiles and their corresponding users.
- Confirm assignments of UID 0 (i.e., Superuser or ROOT authority) are appropriate.
- Verify UPDATE access permissions to FIELD class profiles governing OMVS UID and GID administration are appropriate.

Step 3: Decide How to Assign UIDs and GIDs

You have several options concerning the mechanics of how to assign UIDs and GIDs, including the use of newer features in RACF. Again, you will want to give this some thought before making assignments. To some extent, your choices will be influenced by the UID and GID numbering scheme you've chosen. Here are some considerations:

- Will you assign UIDs to all users or only to those using the default user? System Management Facility (SMF) data generated by RACF can tell you which users currently use the default user. If you plan to assign UIDs based on employee number, consider assigning all your employees a UID, regardless of whether they're using UNIX services. The same may apply to process IDs to prevent failure due to lack of a UID. For customers, perhaps you will assign a UID only as needed.
- What methods will you use to assign UIDs and GIDs? Choices include manual entry, in-house developed automation, sequential assignment by RACF, and automatic assignment by RACF. Manual assignment may be best for process IDs. For employees, developing in-house software that cross-references HR and RACF and adds a UID based on employee number might be best. If certain sets of users and groups are simply to be assigned UIDs and GIDs sequentially in a specific range, the relatively new FACILITY class profile BPX.NEXT.USER can be implemented to define ranges that can be exploited by command keywords AUTOUID and AUTOGID. If you plan to assign UIDs and GIDs only as needed, the newest FACILITY class profile BPX.UNIQUE.USER can be defined to cause RACF to automatically create OMVS segments for users or groups upon their first use of UNIX services. You need not be locked into any one of these methods for all users. For one client, we're employing a combination of all these methods. BPX.NEXT.USER and BPX.UNIQUE.USER will automatically assign UIDs and GIDs to their 500,000 customer users on an as-needed basis.
- What will you do about existing users and groups with UIDs and GIDs that don't conform to the new standard? For a critical process ID, perhaps it's best just to make it an exception and leave its UID intact. If you decide to change any of them, you will need to change OWNERs and extend ACL entries on any files or directories they accessed under their prior UIDs or GIDs.

Note: FACILITY profile BPX.UNIQUE.USER must be used in combination with BPX.NEXT.USER. Defining UNIXPRIV SHARED.IDS is a prerequisite to defining BPX.NEXT.USER. They all require implementing the AIM database structure.

Step 4: Delete BPX.DEFAULT.USER

As with any major change to RACF, the final step is best done during a maintenance period when system activity is minimal. If you don't plan to use profile BPX.UNIQUE.USER, confirm all users and groups using UNIX services have UIDs and GIDs and then simply delete BPX.DEFAULT.USER. If you plan to use BPX.UNIQUE.USER, confirm BPX.NEXT.USER is in place and functioning properly, then just define the profile. Once BPX.UNIQUE.USER is defined, RACF will immediately cease using BPX.DEFAULT.USER and begin automatically creating OMVS segments and assigning UIDs and GIDs sequentially.

You can retain the BPX.DEFAULT.USER profile until you're confident BPX.UNIQUE.USER is functioning as intended. In either case, don't delete the default user and default group referenced in BPX.DEFAULT.USER for a few weeks—just in case you need to reactivate use of BPX.DEFAULT.USER. Performing an Initial Program Load (IPL) of at least one system is advisable to ensure all goes well. Finally, a month or so after implementation, reorganize the RACF database. This will place all the new OMVS segments adjacent to their respective user and group records and let RACF retrieve them both more efficiently.

Summary

BPX.DEFAULT.USER is going away and there's no stopping it. Weaning your system off its use can be a delicate matter, particularly if users have been creating and accessing UNIX files and directories using the default user or default group. This is especially true if some of those users are critical started tasks or process IDs. Don't wait until the eve of your upgrade to a release of z/OS beyond 1.13 to address this issue. The process of examining and updating permissions in all your UNIX file systems is almost certain to take much more time and effort than you might anticipate. Act now! **ETJ**

Robert S. Hansel is lead RACF consultant and president of RSH Consulting, Inc., a RACF professional services firm he founded in 1992. He has worked with RACF since 1986 as a manager, administrator, technician, analyst, auditor, and instructor. He teaches RSH's training seminar "RACF: Securing z/OS UNIX."
Email: R.Hansel@rshconsulting.com; Website: www.rshconsulting.com