RSH CONSULTING

Replacing BPX.DEFAULT.USER

SHARE - 13393 - August 2013

Robert S. Hansel Lead RACF Consultant R.Hansel@rshconsulting.com 617-969-9050

Robert S. Hansel

Robert S. Hansel is Lead RACF Specialist and founder of RSH Consulting, Inc., an IT security professional services firm he established in 1992 and dedicated to helping clients strengthen their IBM z/OS mainframe access controls by fully exploiting all the capabilities and latest innovations in RACF. He has worked with IBM mainframes since 1976 and in information systems security since 1981. Mr. Hansel began working with RACF in 1986 and has been a RACF administrator, manager, auditor, instructor, developer, and consultant. He has reviewed, implemented, and enhanced RACF controls for major insurance firms, financial institutions, utilities, payment card processors, universities, hospitals, and international retailers. Mr. Hansel is especially skilled at redesigning and refining large-scale implementations of RACF using role-based access control concepts. He has also created elaborate automated tools to assist clients with RACF administration, database merging, identity management, and quality assurance.

Contact and background information:

- 617-969-8211
- R.Hansel@rshconsulting.com
- www.linkedin.com/in/roberthansel
- www.rshconsulting.com

Announcement 211-007 - Feb. 15, 2011

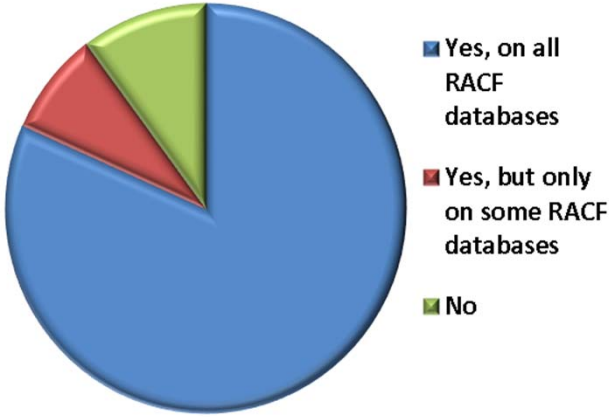
- Statement of Direction from Preview: z/OS Version 1 Release 13 and z/OS Management Facility Version 1 Release 13 are planned to offer new availability, batch programming, and usability functions

- *z/OS V1.13 is planned to be the last release to support BPX.DEFAULT.USER. IBM recommends that you either use the BPX.UNIQUE.USER support that was introduced in z/OS V1.11, or assign unique UIDs to users who need them and assign GIDs for their groups.*

RSH RACF Survey - March 2012

Is FACILITY class profile BPX.DEFAULT.USER defined in your RACF database?

Responses	Count	Percent %
Yes, on all RACF databases	40	81.6%
Yes, but only on some RACF databases	4	8.2%
No	5	10.2%
Total	**49**	**100%**



UNIX Default User & Group

- Use of Unix services and TCP/IP socket applications (e.g., FTP) requires a UID and GID
- Unix Default User and Group
  - Introduced in OS/390 2.4 (1997) to ease administrative burden
  - Created as alternative to assigning each user and group an OMVS segment
- FACILITY BPX.DEFAULT.USER APPLDATA('*userid*/*groupid*')
- Create matching ID and group with OMVS segments

```
ADDGROUP @OEDGRP OMVS(GID(999999))
ADDUSER $OEDUSR DFLTGRP(OEDGRP)
     OMVS(UID(999999) [ HOME(---) PROGRAM(---) ] )
RDEFINE FACILITY BPX.DEFAULT.USER APPLDATA('$OEDUSR/@OEDGRP')
```

- Default UID and/or GID are assigned when the user and/or group, respectively, has no OMVS segment
  - Overridden by assigning UID or GID via an OMVS segment
  - Assigning an OMVS segment with no UID or GID specified nullifies Unix identity

UNIX Default User & Group - Issues

- Shared ID - accountability difficult to establish - frequent audit finding
- UID becomes OWNER of File System objects created by a user using the Unix Default User
  - All users using Unix Default User have ownership rights
  - Giving OTHER "write" (w) permission to directories allows object creation
  - More likely to find when Default User's OMVS segment does not specify blocking values for HOME and PROGRAM (e.g., HOME('/none') PROGRAM('/bin/echo'))
- UID becomes OWNER of File System objects when " chown " specifies a USERID that does not have an OMVS segment
  - All users using Unix Default User have ownership rights
  - Can happen inadvertently when Tech Support sets up a home directory for a user and executes " chown " without informing Security that the user needs a UID
- Cannot use kill(), sigqueue(), _pid_affinity(), or ptrace
- Default UID and GID cannot be acquired in certain processing situations

Replacement Objectives

- Establish standards and processes for assigning UIDs to users
  - Employees & contractors
  - Systems IDs - Started Tasks, Batch, FTP, Training
  - Business partners
  - Customers & clients
- Establish standards and processes for assigning of GIDs to groups
  - User default groups
  - STARTED profile STDATA segment groups
  - Access granting groups (use in Extended Access Control Lists (ACLs))
  - File & directory resource owner groups (use for FSP GROUP)
- Assign unique UIDs to all users who require Unix services
  - Multiple USERIDs assigned to a single user can have a common UID
- Assign unique GIDs to all default and STARTED STDATA groups

Preparatory Tasks

- Verify RACF database has sufficient space for all the new OMVS segments
- Verify PARMLIB(BPXPRMxx) MAXUIDS is set to handle additional UIDs
- Convert RACF database to Application Identity Mapping (AIM) structure
  - Enables use of new features related to replacing BPX.DEFAULT.USER
  - Improves UID and GID lookup performance
- Activate general resource class FSSEC to enable use of Extended ACLs
- Search all Unix File Systems for objects whose OWNER UID or GROUP GID no longer has a corresponding RACF USERID or group and assign replacements
- Search all Unix File Systems for objects whose OWNER UID and GROUP GID is the Unix Default User and Group and assign replacements
  - May be necessary to use ACLs to grant multiple replacement users access
  - May be necessary to audit access to identify actual user - chaudit -a +s *path*
    - Temporarily remove FACILITY BPX.SAFFASTPATH if defined
- Implement UNIXPRIV profile SHARED.IDS

Implement UNIXPRIV Profile SHARED.IDS

- UNIXPRIV SHARED.IDS (must be Discrete)
- Prevents assignment of existing UIDs or GIDs to other users and groups
  - IRR52174I or IRR52185I message issued when an attempt is made to assign an existing ID
- Intended to keep UIDs and GIDs unique
- Profile acts as a switch to activate this feature
- Requires AIM be at Stage 2 or 3
- Can be overridden using the SHARED keyword with commands creating or altering the OMVS segment (e.g., assign UID(0) to multiple system IDs)
  - Example: ALTUSER *userid* OMVS(UID(0) SHARED)
  - Authority to specify SHARED keyword
    - System-SPECIAL
    - READ access to SHARED.IDS

Preparatory Tasks - Optional

- Eliminate OTHER "write" access to files and directories
  - May be necessary to use ACLs to grant multiple users access
  - May be necessary to audit access to identify users - chaudit -a w+s *path*
    - Temporarily remove FACILITY BPX.SAFFASTPATH if defined
  - Ignore file types SYMLINK, EXTLINK, CHAR, and FIFO, and directories /tmp and /dev
- Verify existing System IDs are still needed before assigning UIDs
- Verify HOME directories in Unix (e.g., /u/userid) have a corresponding RACF user and are specified in the user's OMVS segments
- Verify existing SURROGAT BPX.SRV.*userid* profiles and corresponding USERIDs are still valid
- Verify assignments of UID(0) are valid and replace if not required
- Verify permissions to FIELD class profiles governing OMVS segment administration are appropriate

Assigning UIDs

- Issue#1: Will you assign UIDs to all users or only to those users using the Unix Default User
  - Use RACF SMF data to identify users dubbing with Unix Default User
    - SMF Unload - INITOEDP record - IOEP_DFLT_PROCESS field value "YES"
- Issue#2: How will you assign UIDs
  - Manually
  - Automated (e.g., create UID from employee number)
  - Sequentially - BPX.NEXT.USER
  - Automatically - BPX.UNIQUE.USER
- Issue#3: What will you do about existing users whose UIDs do not conform to the new standard
  - Leave as is
  - Change UID
    - Will need to find and change Unix File System OWNER or ACL entries
    - Option - for users not used in OWNER or ACL entries, delete OMVS segment and let BPX.UNIQUE.USER recreate

Assigning GIDs

- Issue#1: Will you assign GIDs to all logon default groups and STARTED profile STDATA segment groups or only to those groups whose users are using the Unix Default Group
  - Use RACF SMF data to identify users dubbing with Unix Default Group
    - SMF Unload - INITOEDP record - IOEP_DFLT_PROCESS field value "YES"
- Issue#2: How will you assign GIDs
  - Manually
  - Automated (e.g., create GID from department number)
  - Sequentially - BPX.NEXT.USER
  - Automatically - BPX.UNIQUE.USER
- Issue#3: What will you do about existing groups whose GIDs do not conform to the new standard
  - Leave as is
  - Change GID
    - Will need to find and change Unix File System GROUP or ACL entries
    - Option - for default and STARTED groups not used in GROUP or ACL entries, delete OMVS segment and let BPX.UNIQUE.USER recreate

FACILITY Profile BPX.NEXT.USER

- Automatically selects UIDs and GIDs for assignment to ensure uniqueness
- Profile's APPLDATA field specifies starting numbers and/or number ranges
  - APPLDATA(*starting-uid-value*|*range*/*starting-gid-value*|*range*) - e.g., (10/50-200)
  - If an APPLDATA value is null or specifies 'NOAUTO', no assignments for that type of id are done - e.g., APPLDATA(NOAUTO/2200)
  - UIDs and GIDs are assigned sequentially
  - RACF confirms UID or GID is not already assigned before making an assignment
  - RACF automatically increments the APPLDATA number after each assignment
- Invoked with AUTOUID and AUTOGID keywords in commands administering OMVS Segment - e.g., ALU *userid* OMVS(AUTOUID)
- Requires UNIXPRIV SHARED.IDS be defined and AIM be at Stage 2 or 3
- If using RRSF, explicit UID/GID is passed - use different ranges for each node to avoid risk of two nodes assigning the same ID simultaneously

FACILITY Profile BPX.UNIQUE.USER

- Automatically adds OMVS segments with UIDs and GIDs to users and groups
  - Invoked during execution of callable services initUSP, getUMAP, and getGMAP
  - Ignored if user or group already has an OMVS segment
  - Facilitates creation of OMVS segments for IDs created and maintained by Group administrators who do not have UPDATE permission to FIELD USER.OMVS.UID
- Profile acts as a switch to activate this feature
  - Once defined, RACF will cease using BPX.DEFAULT.USER
- Requires BPX.NEXT.USER be defined and AIM be at Stage 3
- (Optional) APPLDATA(*userid*) - ID with model OMVS segment attributes
  - Automatically fills in segment fields when a user's OMVS segment is created
  - Can be used to override defaults of HOME(/) and PROGRAM(/bin/sh)
  - With APAR OA42554, can assign HOME using &RACUID or &racuid (e.g., /u/&racuid)
  - To block use of telnet and command OMVS, set PROGRAM(/bin/echo _or_ /bin/false)
  - Assign RESTRICTED, PROTECTED, OMVS(NOUID), and no permissions to model ID

Deleting BPX.DEFAULT.USER

- If implementing BPX.UNIQUE.USER, keep BPX.DEFAULT.USER and the Unix Default User and Group temporarily as a fall-back
  - If BPX.UNIQUE.USER does not appear to be functioning properly, simply delete it to resume use of BPX.DEFAULT.USER
  - Keep BPX.DEFAULT.USER until after at least one IPL has been performed
- Consider deleting only after all preparatory work has been completed
  - All users [using Unix services] have a UID /or/ BPX.UNIQUE.USER is defined
  - All default and STARTED groups [containing users using Unix services] have a GID /or/ BPX.UNIQUE.USER is defined
  - SMF data shows no recent use of the Unix Default User UID or Group GID
- Delete during a system maintenance period, preferable involving an IPL to test all UIDs and GIDs
  - Prior to deletion, confirm ability to logon at a console and make emergency profile changes, including SETROPTS REFRESH
  - Consider running IRRUT400 to align profile segments and fix any index problems

Implementation Toolkit

- IRRIRA00 - AIM Conversion
- IRRDBU00 - UID and GID assignments
- IRRHFSU - File System object FSPs and ACLs
  - Requires READ (r) and Search (x) authority to all directories
- SYS1.SAMPLIB(IRRICE) - 'UIDS' and 'GIDS' ICETOOL samples
- BPXCHECK - REXX EXEC - checks requirements needed for BPX.UNIQUE.USER
  - Available via RACF downloads webpage
  - Requires access to FACILITY IRR.RADMIN.*command* profiles
- z/OS Health & Migration checks - APAR OA37164