z/VM RACF
The Right Way

3rd Edition

Alan Altmark, IBM
Senior z/VM Engineer and Consultant

Alan_Altmark@us.ibm.com

Notes

References to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe on any of the intellectual property rights of IBM may be used instead. The evaluation and verification of operation in conjunction with other products, except those expressly designed by IBM, are the responsibility of the user.

Agenda

- Problem statement
- New approach
- Summary

Right v. Wrong

- The default RACF configuration created when you enable RACF for the first time is … suboptimal
- The books tells you to customize the configuration “to meet your installation’s requirements”, but there is no guidance to provide a workable starting point
  - You end up with configurations that are indefensible
  - You are very likely to fail a competent audit by someone familiar with z/VM and RACF
- The default DIRMAINT-RACF Connector configuration is affected, too

Right v. Wrong

- The default RACF and DIRMAINT configurations aren't actually wrong, *per se*, but…
  - They introduce unneeded complexity
  - They make RACF administration more difficult and time consuming
- On the positive side, they illustrate the full power of RACF
  - "With great power comes great responsibility."
  - "Just because you *can* doesn't mean you *should*."
- Material that follows represents the opinion of the author
  - Working to integrate these suggestions into the product - TBD

What's wrong?

- RACF is protecting things that don't need protection in most cases
  - Spool files
  - Restricted segments
  - VSWITCH and Guest LANs
- No group/role-based access controls (RBAC)
- No generic profiles
  - Makes administration more time-consuming
- The RACF database doesn't use KDFAES
- There is no defined AUDITOR

- It's easy to add controls in the future, but painful to remove unneeded controls
- The profile you don't create doesn't have to be managed!

A refresher on RACF database initialization

- RPIDIRCT analyzes the CP directory and generates RACF profiles that provide the same set of access rights
  - Reads USER DIRECT
  - Creates RPIDIRCT SYSUT1
- Modify RPIDIRECT SYSUT1 to meet your needs
  - RPIDIRCT hasn’t been updated in a while, so what it generates doesn’t reflect what we need today
- Once RACF is enabled in the CP kernel and SYSTEM CONFIG, IBMUSER populates the database using the **RPIBLDDS** command
  - E.g. RPIBLDDS RPIDIRCT

A New Approach

- Make needed changes to a private **copy** of the directory before running RPIDIRCT
- Use PROLOG and EPILOG SYSUT1 files to supplement RPIDIRCT SYSUT1
  - **PROLOG SYSUT1** has RACF commands that need to be issued before users and their resources are defined.
  - **PRIME SYSUT1** is a modified version of **RPIDIRCT SYSUT1** that defines the users and their resources
    - Don’t change the RPIDIRCT file itself, just in case you need to start over.
  - **EPILOG SYSUT1** takes care of anything that is dependent on specific user IDs
- Use groups and generic profiles
- Use LOGON BY for most things
  - Remove unneeded passwords

Directory – Remove passwords

- Change most passwords to LBYONLY
- Do not change
  - Personal IDs of the admins (e.g. IBMVM1)
  - Service accounts (e.g. CSMWORK)
  - IDs that have AUTOONLY or NOLOG as their password
    - RPIDIRCT knows what to do!

```
XEDIT
 zone 1 4
 case m i
 shadow off
 all /iden/ | /user/
 zone 1 *
 (make changes)
 file
```

Directory – Remove ACIGROUP

- ACIGROUP will cause two things:

    1. User default group will be set
       e.g. ADDUSER … DFLTGRP(*acigroup*)

    2. Certain profiles (e.g. VMMDISK) will have "*acigroup*." prefix
       e.g. VMMDISK *acigroup*.ALAN.191

- Remove, but remember

- Set groups in PRIME or EPILOG SYSUT1

Prolog

- Auditor role required to change certain settings
- Turn on generics
- Don’t add the profile creator to the access list of the profiles
- Search all groups that the user is connected to determine access rights

```
ALTUSER IBMUSER AUDITOR

SETROPTS GENERIC(*) GENCMD(*) NOADDCREATOR GRPLIST
```

- NOADDCREATOR is the default for new databases, but good to set on old databases

Prolog - Passwords

- Encrypt password and password phrase tokens with AES
- Require password change every 30 days
  - Only admins have access to z/VM
- Limit password changes to once a day and remember last 30 to prevent reuse
- Revoke user after 4 bad passwords
  - You specify the number of “free tries” you get. The next time, it’s for all the marbles.

```
SETROPTS PASSWORD( ALGORITHM(KDFAES) )
SETROPTS PASSWORD( MINCHANGE(1) REVOKE(3) INTERVAL(30) HISTORY(30) )
```

Prolog – FTP servers

- Create a group that gives FTP servers permission to use RACROUTE and to change their identity via DIAGNOSE 0xD4

```
ADDGROUP $FTP

RDEFINE FACILITY ICHCONN UACC(NONE)
PERMIT ICHCONN CL(FACILITY) ID($FTP) ACC(READ)

RDEFINE VMBATCH ** UACC(NONE)
PERMIT ** CL(VMBATCH) ID($FTP) AC(CONTROL)
```

Prolog – Set up protected event list

- Define the system default VMXEVENT profile
  - It will stop messages from being issued at LOGON and when RACF starts

```
RDEFINE VMXEVENT EVENTS1 OWNER(SYS1) UACC(NONE)

RALTER VMXEVENT EVENTS1 ADDMEM( FOR.C/CTL FOR.G/CTL )
RALTER VMXEVENT EVENTS1 ADDMEM( APPCPWVL/CTL STORE.C/CTL )
RALTER VMXEVENT EVENTS1 ADDMEM( LINK/CTL DIAG0D4/CTL )

RALTER VMXEVENT EVENTS1 ADDMEM( COUPLE.G/NOCTL MDISK/NOCTL )
RALTER VMXEVENT EVENTS1 ADDMEM( DIAG088/NOCTL DIAG0A0/NOCTL )
RALTER VMXEVENT EVENTS1 ADDMEM( DIAG0E4/NOCTL DIAG280/NOCTL )
RALTER VMXEVENT EVENTS1 ADDMEM( RSTDSEG/NOCTL RDEVCTRL/NOCTL )
RALTER VMXEVENT EVENTS1 ADDMEM( TRSOURCE/NOCTL )
RALTER VMXEVENT EVENTS1 ADDMEM( TRANSFER.D/NOCTL TRANSFER.G/NOCTL)
RALTER VMXEVENT EVENTS1 ADDMEM( TAG/NOCTL )

SETEVENT REFRESH EVENTS1
```

Prolog – LOGON BY default

- Create a group that gives z/VM sysadmins the ability to LOGON BY to any user in the system that does not have a discrete LOGONBY profile in the SURROGAT class

```
ADDGROUP $VMADMIN

RDEFINE SUROGAT LOGONBY.** UACC(NONE)
PERMIT LOGONBY.** CL(SURROGAT) ID($VMADMIN) AC(READ)
```

- A more specific profile will override LOGONBY.**

Prolog – LOGON BY subset group

- Create a group that gives Linux sysadmins the ability to LOGON BY to any user in the system whose user ID begins with “LNX”

```
ADDGROUP $LNXADM

RDEFINE SURROGAT LOGONBY.LNX* UACC(NONE)
PERMIT LOGONBY.LNX* CL(SURROGAT) ID($LNXADM) AC(READ)
```

- Use RACFVARS if the server names don’t follow a pattern

```
RDEFINE RACFVARS &LNXGRP1 UACC(NONE) ADDMEM(DB2QC001 WAS00005)
RALTER  RACFVARS &LNXGRP1 ADDMEM(WAS0008)
RDEFINE SURROGAT LOGONBY.&LNXGRP1 UACC(NONE)
PERMIT LOGONBY.&LNXGRP1 CL(SURROGAT) ID($LNXADM) AC(READ)
```

PRIME SYSUT1 – LOGON BY changes

- Delete all LOGONBY.*userid* profiles **except** those associated with personal IDs
  - When you delete the profile, also delete any PERMIT commands that reference that profile – they're not located near each other ("all /SURROGAT/")

- Add the following for each **personal** ID and **service account**:

```
RDEFINE SURROGAT LOGONBY.userid  UACC(NONE)
PERMIT LOGONBY.userid CL(SURROGAT) ID(userid) AC(READ)
```

- Any ID without a discrete LOGONBY profile can be accessed by members of the $VMADMIN group
  - Granted via the generic LOGONBY.** profile

PRIME SYSUT1 - Passwords

- Personal IDs will have both a LOGONBY profile and a password
- IDs to be accessed by groups other than $VMADMIN will be covered by a different LOGONBY profile
- Everything else will have neither a LOGONBY profile nor a password
- Don’t revoke IBMUSER
  - Remove its password
  - Let z/VM admins LOGON BY to IBMUSER to make RACF changes

PRIME SYSUT1 – Unused classes

- Delete all statements in PRIME SYSUT1 that refer to classes
  - VMBATCH (Diagnose 0xD4 – these will be replaced in the Epilog)
  - VMRDR (TRANSFER)
  - VMNODE (TAG)
  - VMLAN (COUPLE.G)
  - VMSEGMT (RSTDSEG)
  - VMDEV (ATTACH)

```
XEDIT PRIME SYSUT1
       ALL /VMBATCH/ | /VMRDR/ | /VMNODE/
       DELETE *
       ALL /VMLAN/ | /VMSEGMT/ | /VMDEV/
       DELETE *
       FILE
```

PRIME SYSUT1 – Assign groups

- Update ADDUSER to statement for those users who had an ACIGROUP
  - ADDUSER xxxxx DFLTGRP(acigroup)

Epilog

- Explicitly turn on the classes we expect to use and turn off the others
- Add FTPSERVE and CSMSERVE to the $FTP group
- Add the sysadmins to the $VMADMIN group
- Make DIRMAINT system-SPECIAL

```
SETROPTS CLASSACT( FACILITY SURROGAT VMXEVENT RACFVARS )
SETROPTS CLASSACT( VMMDISK VMBATCH VMCMD VMPOSIX )

SETROPTS NOCLASSACT( VMRDR VMNODE VMSEGMT VMLAN )

CONNECT (FTPSERVE CSMSERVE) GROUP($FTP)

CONNECT (admin-personal-ids) GROUP($VMADMIN)

ALTUSER DIRMAINT SPECIAL
```

Epilog

- For each user in the directory that has the LNKNOPAS option, add the following

```
ALTUSER userid OPERATIONS
```

- If you have any additional RACF administrators, identify them now.

```
ALTUSER userid SPECIAL AUDITOR
```

- Any user who needs to see, but not change, RACF settings

```
ALTUSER userid ROAUDIT
```

Populate the RACF database

- Now you can run RPIBLDDS
  1. RPIBLDDS **PROLOG**
  2. RPIBLDDS **PRIME**
  3. RPIBLDDS **EPILOG**

- Examine the output report after each run and correct any errors

DIRMAINT

- CONFIG**RC** DATADVH has the DIRMAINT-RACF Connector configuration
- Stop DIRMAINT from creating discrete profiles that we don't want or need by changing the CONFIG**RC** DATADVH:

```
RACF_RDEFINE_SURROGAT_DEFAULTS=
RACF_RDEFINE_VMRDR_DEFAULTS= OWNER($VMADMIN)
RACF_RDEFINE_VSWITCH_LAN= NO
```

- When a new personal ID is added, you must manually create a LOGONBY profile

```
RAC DEFINE SURROGAT LOGONBY.userid  UACC(NONE)
RAC PERMIT LOGONBY.userid CL(SURROGAT) ID(userid) AC(READ)
```

Summary

- While the procedure to run RPIDIRCT and “modify to suit” sounds simple enough, it takes some experience to understand what that really means

- Use the techniques shown here to make a quantum leap forward
  - This doesn't address auditing requirements!

- Your system will be more secure and easier to manage

Contact Information

Alan Altmark
Senior z/VM Engineer and Consultant

z/VM Development
IBM Infrastructure

IBM

1701 North Street
Endicott, NY 13760

Mobile 607 321 7556

Email: Alan_Altmark@us.ibm.com