Introduction to RACF on z/VM

Bruce Hayden

IBM Washington Systems Center

bjhayden@us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.
Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- Introduction
- RACF on your z/VM system
- Resource classes in RACF
- Permissions
- User Attributes
- RACF options
- VM events controlled by RACF
- Groups
- Shared User ids

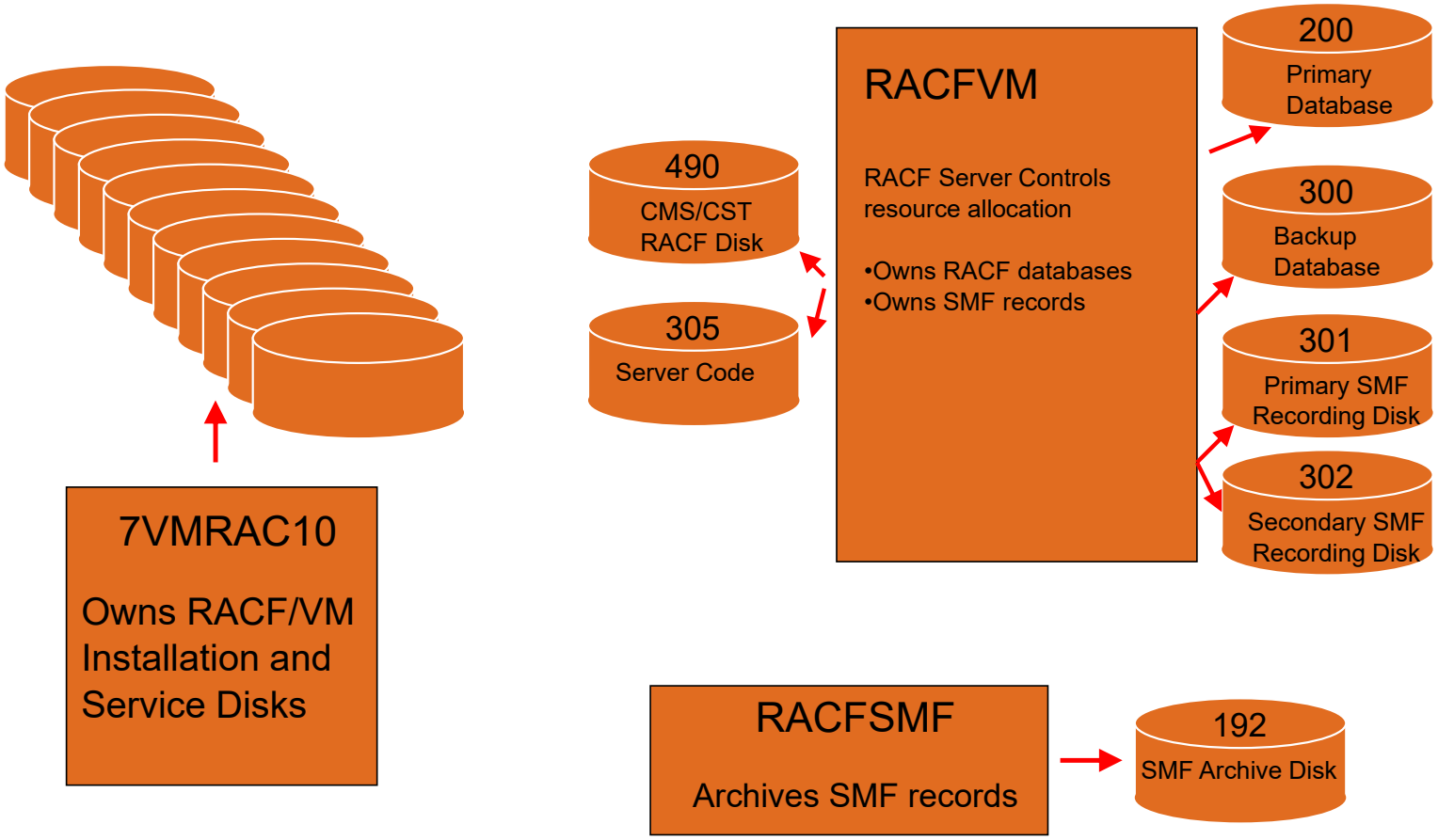
Introduction

- **The RACF Security Server for z/VM**

- A priced, optional, pre-installed feature of z/VM
- Licensed under International Program License Agreement (IPLA) terms and conditions
- Pricing is based on engine-based Value Units and is available for both IFL and standard processor configurations.

- RACF releases are specific to the release of z/VM
  - The level of RACF and CP must be the same

RACF for z/VM Layout



User ids defined for RACF/VM

These are predefined on a new z/VM system installation

- **RACFVM**
  - The main production security server
  - IDENTITY user – runs on every member of an SSI cluster
- **RACMAINT**
  - Configure and test the installation of RACF
  - Test applied service
  - IDENTITY user
- **6VMRAC40, 7VMRAC10**
  - Name is derived from the z/VM version and release
  - Owns all the minidisks that hold RACF code
  - For the sake of this presentation, they are interchangeable

User ids defined for RACF/VM

- **RACFSMF**
  - Management of RACF audit log files
  - IDENTITY user – Runs on every member of an SSI cluster
- **IBMUSER**
  - Used for the initial setup of RACF
- **SYSADMIN**
  - Sample security administration user
- **MAINTvrm** (MAINT640, MAINT710)
  - Maintenance of all z/VM components which includes RACF
- **BLDRACF**
  - Used to rebuild CST, the modified version of CMS used only by RACF

RACF and DIRMAINT

- DIRMAINT can be configured to automatically update RACF
  - This is done via IBM supplied exits in DIRMAINT
  - A DIRMAINT configuration file is provided
  - Changes the directory are automatically synchronized with RACF
  - z/VM 6.4 (and later) synchronizes more directory statements with RACF, such as LINK and NICDEF

- You can activate RACF either before or after you activate DIRMAINT
  - I prefer to activate and configure RACF first on a new system
  - Some people may prefer activating DIRMAINT first
  - Either way will work!

- Limitation on characters in VM user ids
  - No dash (-), plus (+), colon (:), or underscore (_)
  - This applies even if you're not using DIRMAINT

RACF/VM Installation

- No need – it is pre-installed!

- But, it is disabled by default
  - You enable it if you have bought a license

- The program directory is the main guide to configuration
  - Unfortunately, it can be a bit confusing with a lot of choices
    - After this presentation, I hope you know what choices you will need!

  - More background about configuration in the RACF documentation
    - See *z/VM: RACF Security Server Security Administrator's Guide*

Overview of RACF activation

- Prepare your system for RACF
  - Use RACF utilities to migrate definitions from the CP directory
- Enable RACF
  - This will create a new CP Nucleus with RACF enabled
- Shutdown and IPL z/VM from parm disk 2
  - Must be the only SSI member running
  - See the Service Guide on how to IPL a test level of CP
- Start RACF in “test” mode on user RACMAINT
- Load your initial database
- Configure RACF
  - This step takes the longest
- Run PUT2PROD
- Start RACF in production mode and perform testing
- Perform a normal IPL of your system

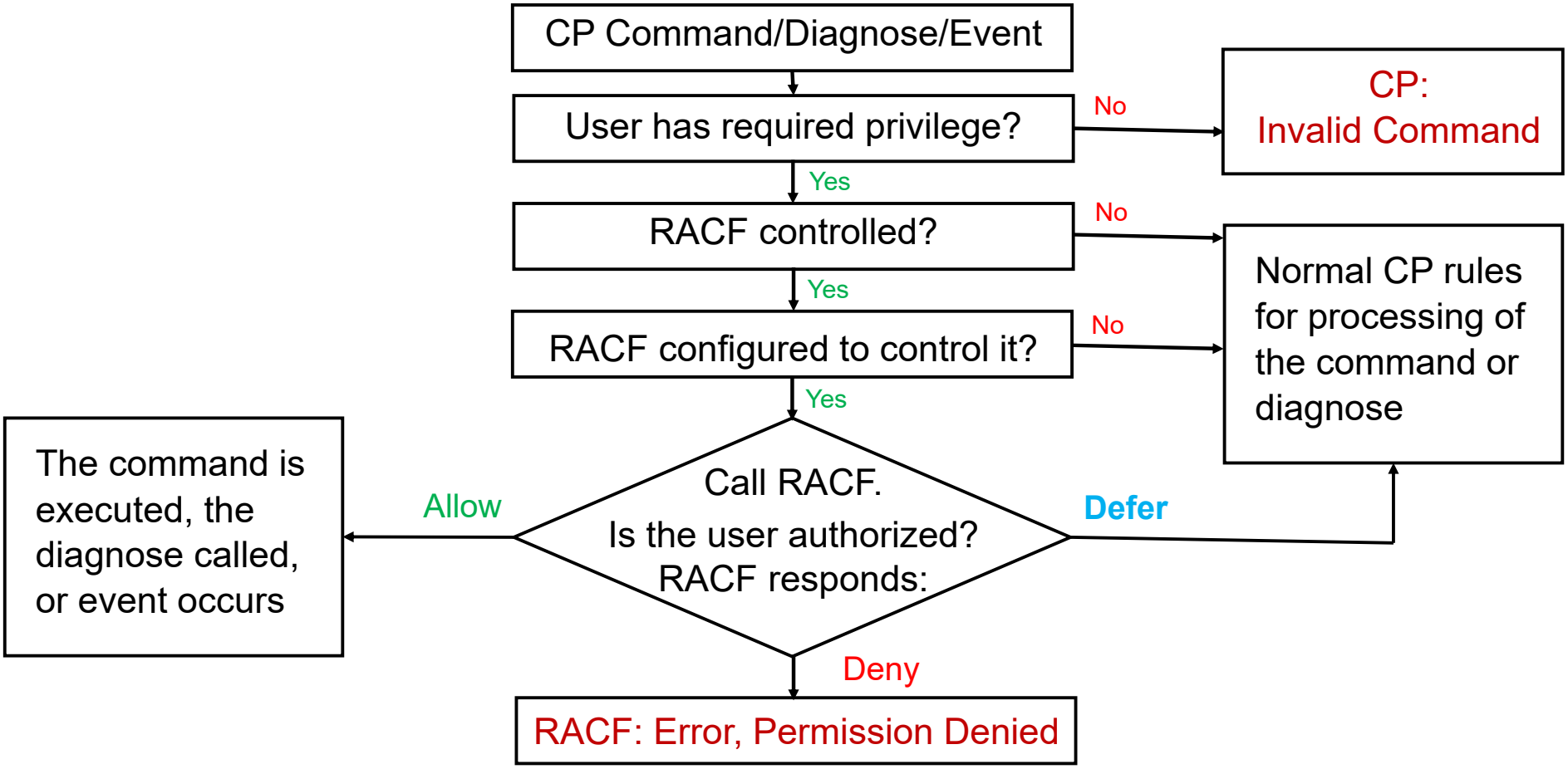
Introduction to RACF on z/VM

RACF Basics

What does RACF do?

- RACF controls user logon to the system
  - Defines passwords and controls
  - Protects terminals
- RACF protects resources
  - So... what is a resource?
  - Stay tuned!
- RACF allows you to grant permissions to resources
  - You can't use a resource unless you have permission
  - This is the PERMIT command
- RACF provides an “audit trail”
  - A log of what happened on the system and who (which user id) did it

Permission flow for RACF and CP



What does CP do?

- Validates that a user exists in the directory
- Assigns the user's privilege class (and any keywords in the OPTIONS statement)

- When a command is entered or a diagnose instruction executed:
  - Validates it is allowed by comparing the privilege class of the command with the privilege class of the user.
  - For a subset of commands and diagnoses:
    - Check that CP has been configured to call RACF for this command or diagnose.
    - If so, call RACF to also validate the user's permission to use the command/diagnose or use the resource referenced by it.
    - RACF responds with "Allow", "Deny", or "Defer to CP"
    - If CP is not configured to call RACF for the command/diagnose, also "Defer to CP"
  - If response is "Defer to CP", use normal CP rules to allow or deny the command or diagnose.
    - Examples: Prompt for a minidisk password, check for GRANT permission

What are RACF resources?

- RACF defines resources this way:
  - Places in the system where data resides (such as minidisks or real devices)
  - Places in the system where data passes during data processing (such as terminals or network interfaces)
  - The functions by which users work with data (such as commands)

- RACF protects resources so that only authorized users can access a resource in approved ways

- A general resource class defines a name for a collection of similar resources
  - Such as VMMDISK for minidisks or VMLAN for virtual LANs
  - There are many general resource classes
    - A lot only apply to z/OS, but they are listed in the z/VM documents
    - I'll only discuss the ones that are most often used on z/VM
  - The following charts describe each one and what it controls

Common z/VM general resource classes

VMBATCH	Allows use of DIAG D4 (alternate userid)
VMCMD	Certain CP commands and other requests
VMDEV	Real devices
VMLAN	Permission to connect to VSWITCH and Guest LANs
VMMDISK	Minidisks
VMNODE	Allows you to target other VM nodes via RSCS
VMRDR	Allows you to target other users via spooling commands
VMSEGMT	Allows access to restricted (class R) saved segments
VMXEVENT	Event profiles for commands and auditing
FACILITY	Allows a virtual machine to use the RACROUTE interface.
SURROGAT	Allows LOGON BY and FOR to another user

General Resource Classes on z/VM

- **VMBATCH**
  - Allows virtual machines to use Diagnose D4 – “set alternate user”
  - Useful for virtual machines that do things on your behalf. “Batch” worker machines is the classic case.
    - FTP server on a modern system
    - Other uses in automation (Operations Manager workers, for example)
  - The name of the resource is the user ID that is the target of the Diag D4
  - CP event for this class: DIAG0D4
- **VMLAN**
  - Allows virtual machines to connect (couple) to restricted VM LANs
    - VSWITCH and restricted guest LANs
  - CP SET (VSWITCH | LAN) GRANT commands or directory authorizations are ignored
  - Resources are named userid.lanname.vlanid
    - For a VSWITCH, the “userid” is SYSTEM, lanname is the name of the VSWITCH or guest lan
    - The vlanid must be 4 digits, such as 0014. It is only present for VLAN aware VSWITCHes
  - CP event for this class: COUPLE.G

General Resource Classes on z/VM

- **VMCMD**
  - Controls certain CP commands, diagnoses, and system events
  - The list is small – only those with critical security concerns or controls
  - VM65930 on 6.4 is required for support of XAUTOLOG.ON.*user.sys* (or 7.1)
  - CP events for this class: STORE.C, TRSOURCE, DIAG088, DIAG0A0, DIAG0E4

VMCMD Profile Name	What It Protects
STORE.C	STORE HOST command
TRSOURCE	TRSOURCE command
DIAG0E4	Diagnose code X'E4' (Minidisk query and define)
XAUTOLOG.*userid*	XAUTOLOG command by a class G user
XAUTOLOG.ON.*user.sys*	XAUTOLOG ON command to *user* on *sys* (new)
DIAG088	Diagnose code X'88' (all subcodes) (DMSPASS)
DIAG0A0.HRTSTORE	Diagnose code X'A0' Subcode X'34' (security labels)
DIAG0A0.QUERYSEC	Diagnose code X'A0' Subcode X'30' (query label)
DIAG0A0.RACONFIG	Diagnose code X'A0' Subcode X'50' (read config)
DIAG0A0.VALIDATE	Diagnose code X'A0' Subcodes X'04' and X'3C' (Validate userid and password or pass phrase)
RAC	RAC command processor
RACF	RACF command session

General Resource Classes on z/VM

- **VMMDISK**
  - Minidisks, which are MDISK statements in the user directory
  - Minidisk passwords in the user directory are ignored (they do not have to be removed)
  - OPTION LNKNOPAS is also ignored
  - Resources are named userid.vdev
  - Leading zero on a 4 digit vdev is not used
    - MAINT.0190 is incorrect; MAINT.190 and MAINT.2190 are correct
  - CP events for this class: MDISK, LINK
- **VMDEV**
  - Real devices
  - Resources are named RDEV.*rdevno.sysname*
    - The *rdevno* is the real device number (*nnnn*), or SYSASCII for the HMC ASCII console
    - The *sysname* is the system identifier
    - Generic resource definitions can be used to authorize a device across multiple systems
  - May be useful depending on how you manage your real devices.
  - CP event for this class: RDEVCTL

General Resource Classes on z/VM

- **VMNODE**
  - Permission to send spool files to remote systems via RSCS (RSCS does not interface with RACF)
  - Resource name is the node id of the remote system
  - The CP TAG command is checked for the node id read by RSCS
    - For example: CP TAG DEV PUN nodeid userid
  - Not useful or needed on most systems
  - CP event for this class: TAG.G
- **VMRDR**
  - Permission to send a spool file to another user
  - Resource name is the user id that will receive the spool file
  - All CP spooling commands are checked
    - SPOOL PUN TO *userid* and SPOOL PRT TO *userid*
    - TRANSFER TO *userid*
    - CLOSE TO *userid*
  - Not needed on most systems or use generic resources for control and exceptions
  - CP events for this class: TRANSFER.D, TRANSFER.G

General Resource Classes on z/VM

- **VMSEGMT**
  - The ability to use a restricted (class R) saved segment or NSS
    - Use of normal class A segments is not controlled by RACF
    - The NAMESAVE record in the directory is ignored
  - Resources are named NSS.*segmentname* or DCSS.*segmentname*
  - Not useful or needed on most systems
  - CP event for this class: RSTDSEG

- **VMXEVENT**
  - Special class that holds event profiles, which is used to define the CP and auditing interface to RACF
  - Will be discussed later

- **FACILITY**
  - Allows service virtual machines to authenticate directly with RACF (known as the RACROUTE interface)
  - Also used for other “miscellaneous” authorizations

General Resource Classes on z/VM

- **SURROGAT**
  - Note: it is the “surrogate” class, but specified with just 8 characters

  - Allows a user id to use its password to logon to another id
  - For example: `LOGON MAINT BY BRUCE`
    - I enter the password for BRUCE at the logon prompt, but I am logged on to MAINT
  - Resources are named LOGONBY.*userid*
    - The userid is the user that will be logged onto
    - In the above example, MAINT, so the resource is LOGONBY.MAINT
  - LOGONBY statements in the directory are ignored
  - When a LOGONBY.*userid* profile is defined for a user, direct logon to that user is not longer allowed
    - You can override this behavior, but normally this is what you want.
  - Permission to a user's surrogate profile also allows you to also use the CP FOR command to that user
    - You must also have CP Privilege class C or be the secondary user to that id.
  - CP events for this class: FOR.C and FOR.G.
  - When RACF is active, LOGON BY cannot be deferred to CP

Defining resource classes

- By default, only 2 resource classes are active:
  - **USER** Allows you to logon to the system
  - **TERMINAL** Allows you to use a terminal to logon

- You can choose which resource classes to activate
  - This is the CLASSACT option on the SETROPTS (Set RACF options) command (discussed later)

- The RDEFINE (resource define) command defines actual resources in a class
  - For example, to define MAINT's 191 minidisk:
    - `RDEFINE VMMDISK MAINT.191 UACC(NONE)`
    - VMMDISK is the general resource class for minidisks
    - UACC is the default access type, for “universal access”
      - NONE is the default, but it is often specified in the command
    - With NONE, no users have access to this resource unless explicit permission is granted

Introduction to RACF on z/VM

Permissions and User Attributes

Giving permissions to resources

- This is the PERMIT command
  - If a resource is defined with a universal access of NONE, you must be given permission to access it.

- Syntax: PERMIT *resource options*
  - *resource* is the name of the resource from the RDEFINE command
  - *options* are specified as KEYWORD(VALUE)
  - Required options (they can be in any order)
    - **CLASS( )** The resource class, such as VMMDISK or VMRDR
    - **ID( )** The user id that is allowed to access
    - **ACCESS( )** The permission, such as READ
    - **DELETE** Delete permission, specified instead of “ACCESS()”
  - These can be abbreviated – but automation should use the long form
    - For this command, the first letter is all that is needed.
  - Example: Allow MAINT read/write access to TCPMAINT 198
    - PERMIT TCPMAINT.198 CLASS(VMMDISK) ID(MAINT) ACCESS(CONTROL)

Access permissions

- The keywords allowed on ACCESS or UACC. Each permission includes all permissions below it
  - **ALTER** Allows full control of the resource, including changing the access list*
  - **CONTROL** Read/write and possibly more control
  - **UPDATE** Read/write access
  - **READ** Read only access
  - **NONE** No access allowed
- Each general resource class defines what these permissions mean for resources in that class
  - More detail on the next chart
- ALTER permission also allows you to change the access list
  - Which means you are allowed to PERMIT others to the resource, even if you do not own the resource
  - *This is not true for the VMMDISK class! (changed in z/VM 6.2)
    - The documentation has a suggestion for an alternate way to achieve this.

Access permissions details

Details about access permissions for some resources

If an access permission isn't listed for a class, it has no additional meaning

- **VMMDISK**
  - READ: Link mode R
  - CONTROL: Link mode M
  - UPDATE: Link mode W
  - ALTER: Link mode MW
  - Note: ALTER access for the VMMDISK class is an exception to normal rules for the ALTER permission
- **VMDEV**
  - READ: Attach read only
  - CONTROL: Attach r/w with SYSCTL operand allowed
  - UPDATE: Normal read/write attach
- **VMLAN**
  - UPDATE: Normal couple
  - CONTROL: Promiscuous Mode
- **VMCMD**
  - READ: Allows the user to execute the command
- **VMRDR**
  - UPDATE: Allows you to send or transfer a spool file to another user
- **VMBATCH**
  - CONTROL: Allows the user to set your user id as an alternate user
- **SURROGAT**
  - READ: Allows the id to be used to logon to the shared user id

RACF User Attributes

- A VM user may have one or more of these attributes

- **SPECIAL**
  - Security administrative authority – allowed to issue any RACF command
  - Full control (add, delete, modify, permit) over all RACF profiles in the RACF database
  - Allowed to set RACF options
  - New options to delegate some authority (usually called “Help desk” functions)
    - Authority to list user profiles (all profiles or selected profiles)
    - Authority to reset passwords (all users or selected users)
- **AUDITOR**
  - Allowed to view and set RACF auditing options and controls
    - Note: SPECIAL without AUDITOR is not allowed to set auditing options
  - Allowed to run the DSMON program (Data Security Monitor)
- **ROAUDITOR**
  - Only allowed to list or view auditing controls and run DSMON – not allowed to make any changes
    - z/VM 7.1 or APAR VM65930 for 6.4 required to enable this.

RACF User Attributes

- **OPERATIONS**
  - Default authorization to access resources in certain classes
    - VMBATCH, VMCMD, VMMDISK, VMNODE, and VMRDR only
  - Authorization to a resource can be overridden with a specific permit
    - For example:
      Don't allow MAINT, with the OPERATIONS attribute, access to the RACF database:
    - PERMIT RACFVM.200 CLASS(VMMDISK) ID(MAINT) ACCESS(NONE)
- **REVOKE**
  - User is not allowed to access (i.e. logon or authenticate) to the system
    - A shared user id that is revoked is also not allowed to logon
- **PROTECTED**
  - A user without a logon password (NOPASSWORD) or logon phrase (NOPHRASE)
    - Newly added users have no password so are Protected until a password or phrase is assigned
  - User can't be used to logon to the system
    - However, the id can be logged on using a shared (surrogate) permission
  - User will not be automatically revoked from inactivity or invalid logon attempts

Introduction to RACF on z/VM

RACF Commands

Entering RACF commands

- **RAC EXEC**
  - The preferred way
    - Propagates certain commands to other SSI members automatically
  - Enter a single RACF command as arguments:
    - rac permit operator.191 class(vmmdisk) id(maint) access(control)
  - Any command output is written to your terminal and to the file RACF DATA A
    - GLOBALV variables can be used to affect the output – see the documentation and RACOUTP EXEC

- **RACF MODULE**
  - Starts a RACF command session for multiple RACF commands
  - Must enter END to leave the session
  - The output is not saved

```
racf
RPITMP001I RACF/VM SESSION ESTABLISHED. TO TERMINATE ENTER "END"
altuser maint special
RPITMP002I ENTER RACF COMMAND OR "END" TO EXIT
permit operator.191 cl(vmmdisk) id(maint) acc(control)
RPITMP002I ENTER RACF COMMAND OR "END" TO EXIT
end
RPICMD003I RACF/VM COMMAND SESSION COMPLETE
```

Working with user profiles

- Add a new user profile: **ADDUSER**
  - rac adduser linux name('Master Image') password(new4you)
  - The password is expired and must be changed during logon
  - You can add a user profile that is not in the CP directory!
- Delete a user: **DELUSER**
  - rac deluser linux
  - This does not delete the user ID from the VM user directory
- Change a user: **ALTUSER**
  - To set a new temporary password:
    - rac altuser maint password(temp4you)
  - To set a new password that is not expired:
    - rac altuser maint password(sup3rusr) noexpire
  - To change a user attribute, such as if a user is revoked:
    - rac altuser maint resume

Introduction to Generic resources

- Single resources are defined to RACF
  - For minidisks – a specific virtual address owned by a user
  - For a virtual lan – a specific vswitch and vlan
  - In RACF terms – a “discrete profile”

- RACF also supports generic resources
  - A lot like wildcard matching of file names
  - More restrictive definitions and discrete profiles have priority

  - Example: Only allow the OPERATOR user id to log on to a local terminal
    - Allow any user to log on to any terminal and also local terminals (named LOGNrdev)
      - RDEFINE TERMINAL * UACC(READ)
      - RDEFINE TERMINAL LOGN* UACC(READ)
    - Only allow OPERATOR to log on to local terminals by not allowing it to log on to “all terminals”
      - PERMIT * CLASS(TERMINAL) ID(OPERATOR) ACCESS(NONE)

Generic resources

- Enabled via RACF options (The SETROPTS “Set RACF Options” command)
  - **GENCMD**(classes)
    - Allows generic profiles to be specified in commands
    - You can create generic profiles before making them active
  - **GENERIC**(classes)
    - Activates generic profile checking for specified classes
    - Also allows generic profiles in commands
- Not enabled on any classes by default
  - Due to extra searching, and not part of old RACF systems
- This can make managing your system easier!
  - Fewer resources to define and manage
  - Some resources only need controls for the exceptions
- The next slide shows how to use this for the VMBATCH class
  - In the backup slides at the end are examples for the VMRDR and VMCMD classes

Using Generic Resources with VMBATCH

- **VMBATCH (set alternate user)** – How to use with FTPSERVE
  - Allows FTPSERVE to access your resources on your behalf – e.g. when you “log in” via an FTP client
    - Instead of giving FTPSERVE explicit permission to your resources
  - FTPSERVE uses Diag D4 to ask CP to set its alternate user to your user id when you log in
    - If FTPSERVE has permission from RACF to your VMBATCH resource, CP allows it to be set
    - Now FTPSERVE can access any resource you have permission for
  - Define a generic resource for VMBATCH
    - The default permission is no access
      - RAC RDEFINE VMBATCH * UACC(NONE)
    - Allow the FTP server to be an alternate user to any id
      - RAC PERMIT * CL(VMBATCH) ID(FTPSERVE) ACCESS(CONTROL)
  - Exceptions for critical users such as MAINT can be defined
    - A discrete permission (PERMIT) overrides a generic permission or universal access (UACC)
    - An access permission of NONE overrides any higher permission
      - RAC RDEFINE VMBATCH MAINT UACC(NONE)
      - RAC PERMIT MAINT CLASS(VMBATCH) ID(FTPSERVE) ACCESS(NONE)

Set RACF options - SETROPTS

- The SETROPTS command
- Allows you to dynamically set system-wide RACF options related to resource protection and auditing
- Many options use NO as a prefix to invert the selection
  - CLASSACT( ) or NOCLASSACT( ), GRPLIST or NOGRPLIST, and so forth
- Current settings displayed with **SETROPTS LIST**

- Both audit and system security settings
  - Users with only SPECIAL cannot alter the audit settings
  - Must have AUDITOR attribute to change audit settings

- Some settings must be propagated to other SSI members
  - This is done automatically for the commands that require it
  - The RAC command must be used
  - Duplicate output from other members is suppressed unless there is an error

SETROPTS command options

- **CLASSACT**
  - Activates general resource classes
  - SETROPTS CLASSACT(VMMDISK VMRDR)
- **RACLIST**
  - Cache selected resource profiles in memory – avoids disk I/O
  - Should only be used for classes with frequently referenced profiles
  - RACLIST(..) REFRESH updates the cache
  - Automatically propagated to other SSI members
  - Not allowed on all resource classes
- **ADDCREATOR**
  - Adds the creator of a resource to the access list
  - This is what older releases always did
  - You probably want NOADDCREATOR
  - NO is the default for newly created databases

- **PASSWORD**
  - Sets password rules
    - Maximum change interval (1 to 254 days)
    - Expiration warning (1 to 255 days)
    - History (reuse old passwords, 1-32)
    - Logon attempts before revoke (1 to 254)
    - Minimum length
    - Rules for types of characters in certain positions
      - *rule1(length(8) alpha(1,8) alphanum(2:7))*
    - Minimum change interval
    - Expanded special characters
  - ALGORITHM – z/VM supports the KDFAES enhanced AES-based encryption of passwords

Introduction to RACF on z/VM

VM Events and RACF control

VM events controlled by RACF

- VM calls RACF for authorization checking of certain z/VM events
- It is not a long list
  - Most authorization in z/VM is still controlled by normal CP rules
  - This is your CP privilege class or options in your directory entry
- Event profiles define the RACF authorization checks that are active
  - Normally there is only one profile for the entire system
  - Also overriding profiles for individual users (overrides system profile)
- By default, RACF checks all of the VM events (see the next 2 charts)
  - The check can be turned off for most events
  - You must customize RACF to remove checking as you require
  - Events for LOGON and XAUTOLOG are always enabled

List of controlled events

COUPLE.G	VMLAN	Couple to a restricted guest lan or a VSWITCH
FOR.C	SURROGAT	FOR command, IBMclass C
FOR.G	SURROGAT	FOR command, IBMclass G
LINK	VMMDISK	LINK command or LINK directory statement
MDISK	VMMDISK	Directory statement or LINK to a user's own minidisk
STORE.C	VMCMD	STORE host memory command, IBMclass C
TAG	VMNODE	TAG command, for RSCS processing
TRANSFER.D	VMRDR	TRANSFER and CHANGE spooling command, IBMclass D
TRANSFER.G	VMRDR	IBMclass G spooling commands
TRSOURCE	VMCMD	TRSOURCE command

List of controlled events, continued

APPCPWVL	USER	Used to verify passwords on APPC connect when SECURITY(PGM) is specified
DIAG088	VMCMD SURROGAT	Use of Diag 88 (Check authority and link minidisk)
DIAG0A0	VMCMD	Use of Diag A0 (Obtain ACI Groupname)
DIAG0D4	VMBATCH	Use of Diag D4 (Set Alternate User ID)
DIAG0E4	VMCMD	Use of Diag E4 (Define Full-Pack Overlay)
DIAG280	VMPOSIX	Use of Diag 280 (Set POSIX security values, used by the VMPOSIX class)
RSTDSEG	VMSEGMT	Access to restricted (class R) saved segments
RDEVCTRL	VMDEV	Dedicate, CP Attach or Give commands of real devices

Creating event profiles

- To change the VM events checked by RACF, you must create an event profile
  - This is strongly recommended, so that the controlled events match the resource classes you activate
- The profiles have a dual purpose
    - Access checking
    - Auditing (not discussed here)

- Create a resource profile in the VMXEVENT class
  - The name can be anything you choose
  - More than 1 system profile can exist
    - Normally, only 1 is active
    - Separate system profiles for audit and access are possible, but not recommended.
  - Members are added to stop control of selected events
    - By default, all events are controlled

Resource profile for my system

- An example based on my needs for a lab system
  - Note : *Certainly not based on IBM security policy!*

- I want RACF control of everything, except:
  - FOR command
    - Controlled by the SURROGAT profile. I only want to use SURROGAT for logon to shared user ids
  - TAG command
    - I do not have RSCS active, no need to control TAG
  - Restricted segments
    - I will use the NAMESAVE authorization in the directory instead
  - User's own minidisks (in directory or via link command)
    - If it is yours, then I have no need for RACF to check your own access
  - Real devices
    - No need to control them

RACF commands for my profile

- Create profile EVENTS1 in VMXEVENT
  - Remember that you can choose any name for this profile

```
rac rdefine vmxevent events1
rac ralter vmxevent events1 addmem(for.c/noctl for.g/noctl)
rac ralter vmxevent events1 addmem(tag/noctl mdisk/noctl)
rac ralter vmxevent events1 addmem(rstdseg/noctl rdevctrl/noctl)
rac setropts classact(vmxevent)
rac setevent refresh events1
```

- List the current settings and profile name with rac setevent list
  - You'll need 6.4 APAR VM65930 or 7.1 to see the current profile name in the output

Output from creating an event profile

- When profile is activated, default members are made active

```
rac setevent refresh events1
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: LINK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: STORE.C
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.D
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRANSFER.G
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: TRSOURCE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG088
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0A0
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0D4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG0E4
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: DIAG280
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
```

- You can explicitly define these members in the profile for completeness

```
ralter vmxevent events1 addmem(couple.g/ctl link/ctl store.c/ctl trsource/ctl)
```

Event profiles for specific users

- Profiles can be created to override the system profile for specific users
  - They are named USERSEL.*userid* in the VMXEVENT class
- If a user profile exists, none of the system profile is active for that user
  - Make sure you create a complete user profile
  - It must include both control and audit settings
- They are created just like the system profile
  - rac rdefine vmxevent usersel.datamove
  - rac ralter vmxevent usersel.datamove addmem(link/noctl tag/noctl mdisk/noctl)
  - rac setevent refresh usersel.datamove

Introduction to RACF on z/VM

RACF Groups

RACF Groups

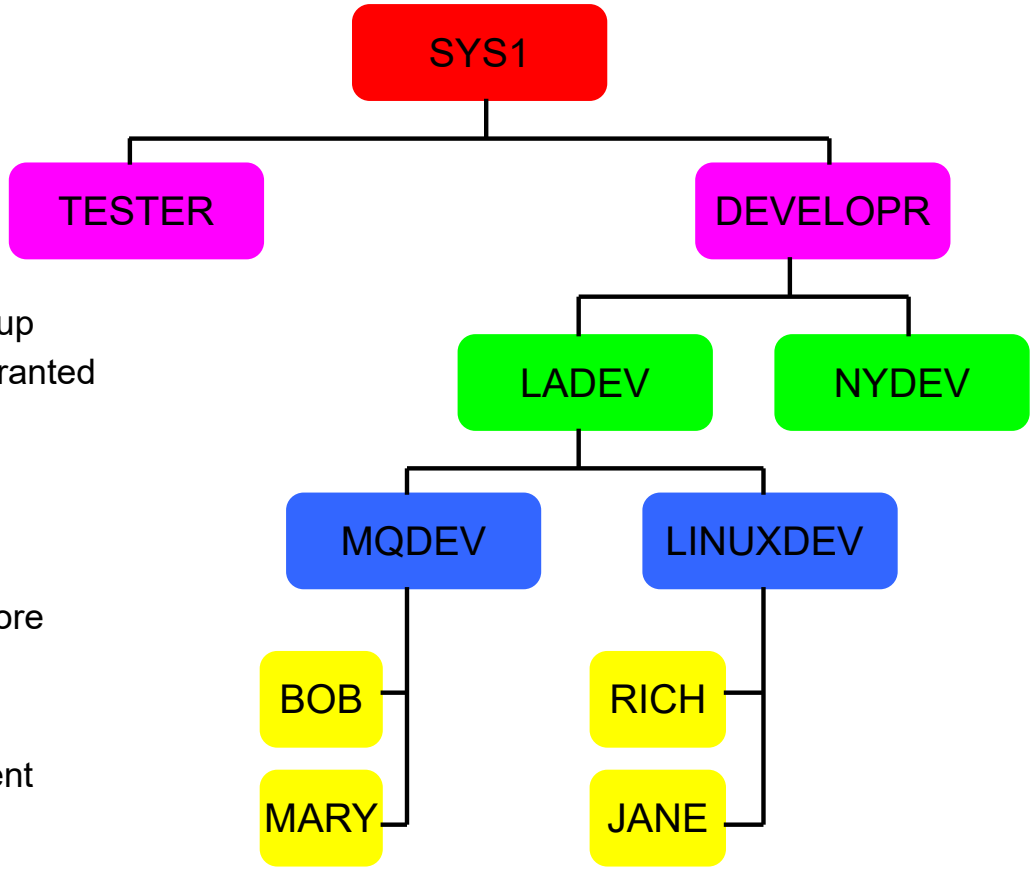
- Groups help with administration of your z/VM system
  - Put user ids with similar roles into groups
    - Linux ids
    - System Administrators
    - Service Virtual Machines (SVMs)
  - New user ids performing the same role just need to be added to the group

- RACF defines groups as a hierarchy
  - The intent was to be able to map the management of the group structure to an organizational structure
  - Such as: A system support group subdivided into system programmers, storage management, and security.

- But – RACF groups can just be used as lists of user ids
  - Examples
    - All ids that need access to a set of resources
    - All ids that have a related role

RACF Groups

- Give access rights to a group
  - Note: rights are not granted to lower groups in the hierarchy
- Connect users to one or more groups
- Delegate group management
- Reduce administration effort



Using Groups

- Becoming a member of a group
  - RACF calls this “connecting” a user to a group
- Naming groups
  - Same “naming space” as user ids – hard to tell them apart!
  - Use a naming convention for groups, such as start with a special character ($, @, or #), G, end with $, etc.
- Specified user ids can be designated as the administrator of a group
  - The ability to connect (add) or remove users
- Be sure to enable RACF option GRPLIST
  - Enables checking all groups the user is connected to for authority
    - Otherwise, only the user's current connect group is checked
    - This is required if a hierarchy of groups is not used
  - RAC SETROPTS GRPLIST

Using Groups – Examples

- Creating a Group for Linux servers
  - rac addgroup $linux owner($linux) supgroup(sys1)
- Give the LNXADM id authority to connect Linux servers
  - rac connect lnxadm group($linux) owner(lnxadm) authority(connect)
- Connecting a new Linux server to the group
  - rac connect linux01 group($linux) owner(linux01) authority(use)
- Granting permission to a resource for all Linux servers
  - rac permit lnxadm.291 class(vmmdisk) id($linux) access(read)
- Removing a user
  - rac remove linux01 group($linux)
- Deleting a group
  - Remove all users first
  - rac delgroup $linux

Introduction to RACF on z/VM

Sharing User IDs

How to use Shared User ids

- Some user ids may need to be shared by multiple users
  - MAINT, MAINTvrm, OPERATOR, TCPMAINT, PERFKIT, etc.
  - Sharing the passwords is not allowed! (Your security policy states that, right?)

- Use the SURROGAT class and groups to allow multiple people to access these user ids
  - Allows logon “by” (or using) a personal id and its password
  - There is no limit on the number of sharing users

- CP also has native LOGON BY support
  - Defined in the user directory using the LOGONBY statement
  - Limited to only 8 unshared ids per shared id
  - z/VM 7.1 defines user IBMVM1 for shared logon to many default system user ids.

Shared User ids – Examples of defining

- Activate the SURROGAT class
  - rac setropts classact(surrogat)
- Define a resource for each user id that is shared
  - rac rdefine surrogat logonby.operator uacc(none)
  - rac rdefine surrogat logonby.maint uacc(none)
  - rac rdefine surrogat logonby.maint710 uacc(none)
  - rac rdefine surrogat logonby.tcpmaint uacc(none)
  - rac rdefine surrogat logonby.perfsvm uacc(none)
- Give permission to groups
  - rac permit logonby.operator class(surrogat) id(**$sysprog**) access(read)
  - rac permit logonby.operator class(surrogat) id(**$opergrp**) access(read)
  - rac permit logonby.maint class(surrogat) id(**$sysprog**) access(read)
  - rac permit logonby.maint710 class(surrogat) id(**$sysprog**) access(read)
  - rac permit logonby.tcpmaint class(surrogat) id(**$sysprog**) access(read)
  - rac permit logonby.perfsvm class(surrogat) id(**$sysprog**) access(read)
- Give permission to specific user ids
  - rac permit logonby.maint class(surrogat) id(bruce) access(read)

Shared User ids – Using

- Logging on a shared id
  - logon maint by bruce
  - Operator console shows:
    - `GRAF vdev LOGON AS MAINT USERS = nnn BY BRUCE`
  - Query who is logged on to MAINT
    - query byuser maint
    - `The BYUSER for MAINT is BRUCE`
  - The “byuser” is retained when you disconnect, updated on reconnect
- Direct logon is no longer allowed when SURROGAT resource is defined for a user
    - LOGON MAINT

```
RPIMGR066A User ID MAINT is defined as a shared user ID that may not be logged onto directly
LOGOFF AT 16:24:31 EDT THURSDAY 08/25/18 BY SYSTEM
```

  - Allowed if you permit the shared user id read access to its own profile
    - permit logonby.maint class(surrogat) id(maint) access(read)

The End

Thank you for listening!

- Contact information:
- Bruce Hayden

bjhayden@us.ibm.com

IBM Washington Systems Center

- Mississippi River Office

Sabula, IA “Iowa’s Island City”

References

- **VM home page**
  - http://www.vm.ibm.com
- **z/VM Security and Integrity Resources**
  - http://www.vm.ibm.com/security
- **z/VM Statement of Integrity**
  - http://www.vm.ibm.com/security/zvminteg.html

- Recent RACF for z/VM service updates
  - APAR VM66278 RACF for z/VM Usability Enhancements - ”Fix Pack 1”
    - Available June, 2019, for z/VM 7.1 only