Introduction to RACF on z/VM

Bruce Hayden
IBM Advanced Technical Sales Support

March 12, 2014
Session Number 14791

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.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

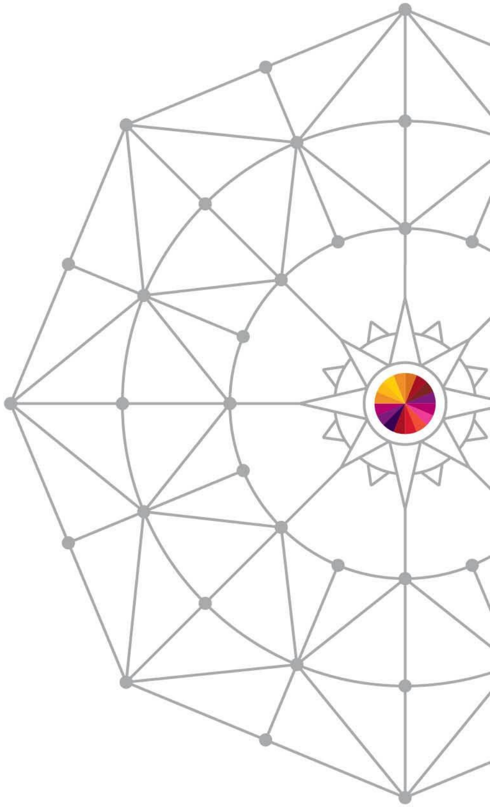
Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT").

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

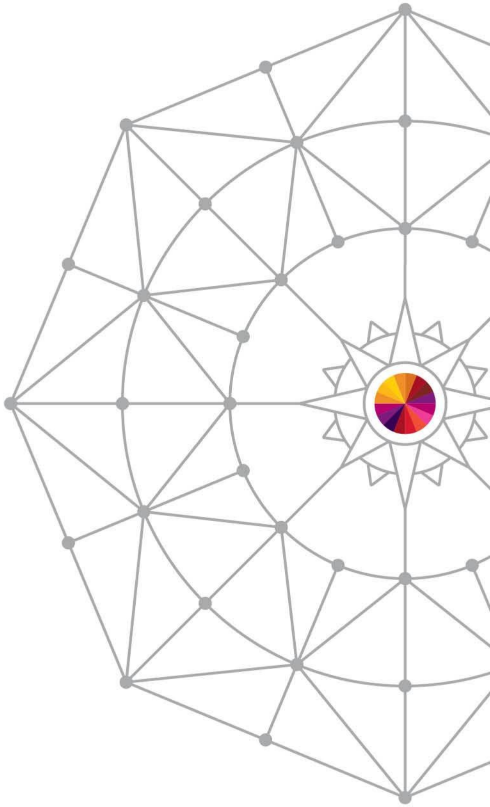
Agenda

- 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



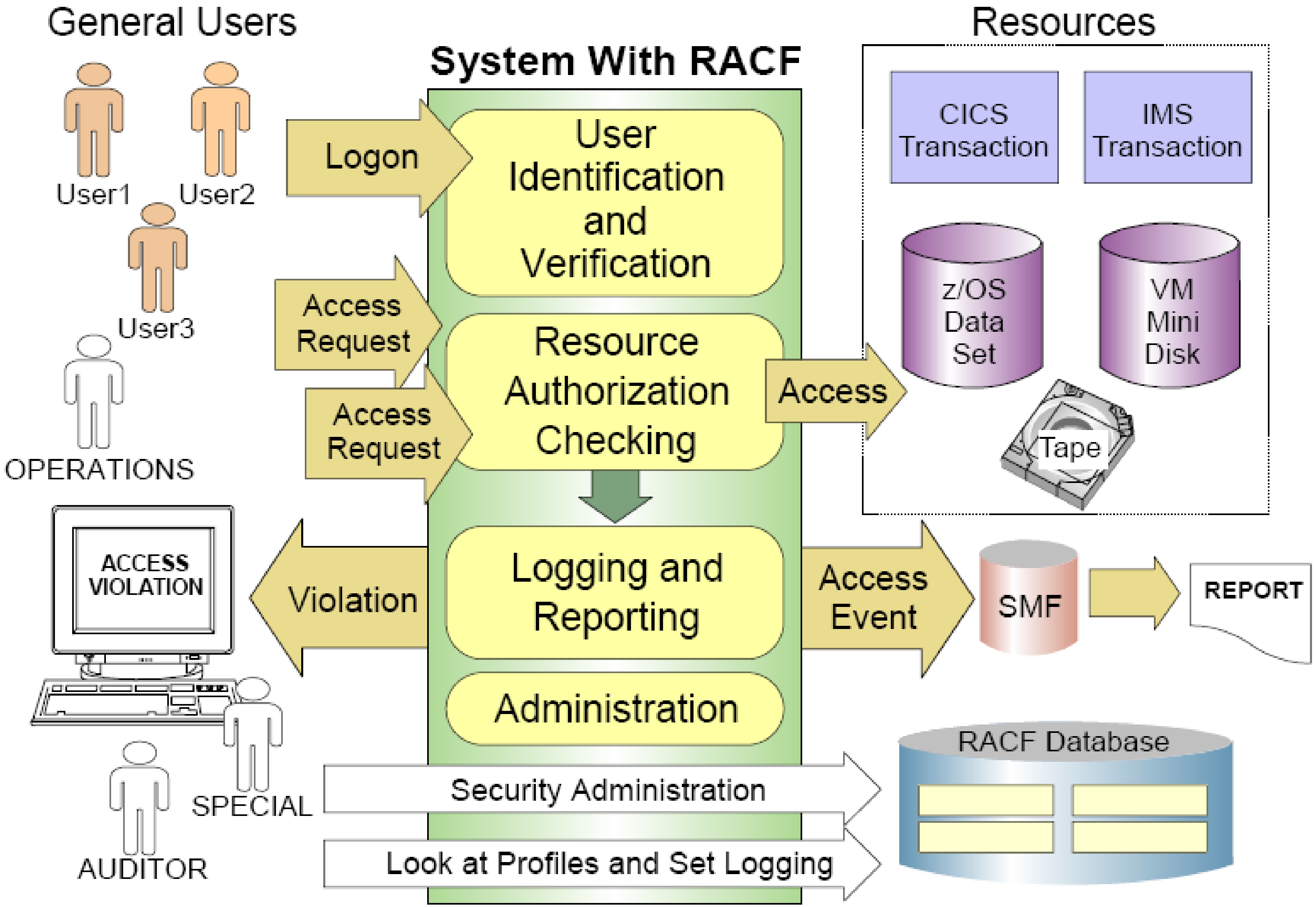
Introduction

The RACF Security Server for z/VM



- A priced, optional, pre-installed feature of z/VM
  - For all current releases - 5.4, 6.2, and 6.3.
- 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

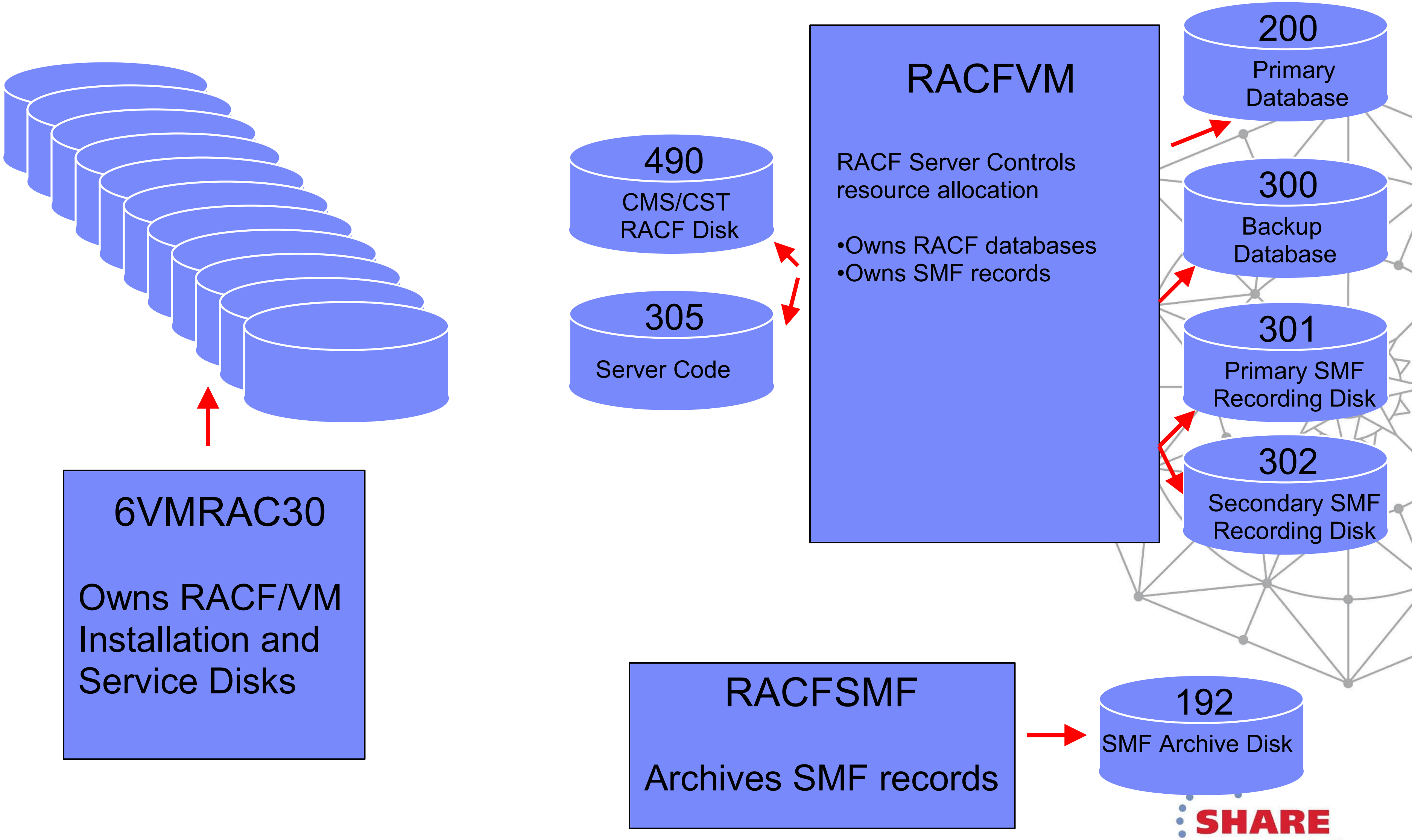
Basic Security Features of RACF



Configuration Guidelines and Best Practices

- Do not always think of “Best Practices” when it comes to security settings!
  - Security settings are determined by Security Policy
  - Every company has a different one
    - Should be determined by the company CIO as a high level guideline for all IT systems
    - Implementation varies on each type of system
  - Systems programmers and administrators implement security policy
    - They do not decide security policy
    - If parts of the policy can't be implemented, then exceptions must be granted, etc.
    - This is all very important for security audits or if a security breech happens!
- There are Best Practices for how to implement security policy
  - As with most systems, there are different ways to implement something
  - With security, it isn't normally about performance
    - Some ways are easier for system administration
    - Some ways are less prone to error, such as inadvertently creating a security “hole”

RACF for z/VM Layout



490
CMS/CST
RACF Disk

305
Server Code

RACFVM

RACF Server Controls resource allocation

- Owns RACF databases
- Owns SMF records

200
Primary
Database

300
Backup
Database

301
Primary SMF
Recording Disk

302
Secondary SMF
Recording Disk

6VMRAC30

Owns RACF/VM Installation and Service Disks

RACFSMF

Archives SMF records

192
SMF Archive Disk

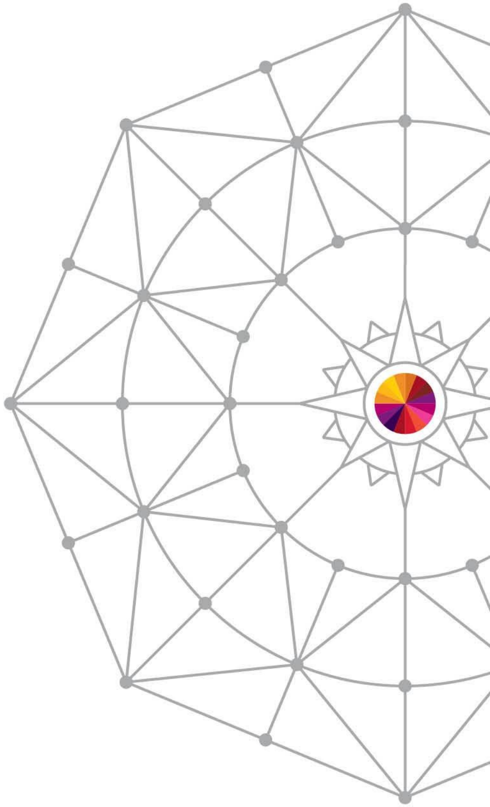
User ids defined for RACF/VM

These are predefined on a new z/VM system installation

- **RACFVM**
  - The main production security server
  - IDENTITY user – runs on every node of an SSI cluster
- **RACMAINT**
  - Configure and test the installation of RACF
  - Test applied service
  - IDENTITY user
- **5VMRAC40, 6VMRAC20, 6VMRAC30**
  - 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 node of an SSI cluster
- **IBMUSER**
  - Used for the initial setup of RACF
- **SYSADMIN**
  - Sample security administration user
- **MAINT** or **MAINTvrm** (MAINT620, MAINT630)
  - Maintenance of all z/VM components
- **BLDRACF**
  - Used to rebuild CST, the modified version of CMS used 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
    - On 6.2: Apply APAR VM65125 for updates to this configuration file
  - Changes the directory are automatically synchronized with RACF
  - On 6.2: Also apply Dirmaint APAR VM65155 for correct handling of SUBCONFIGs

- 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

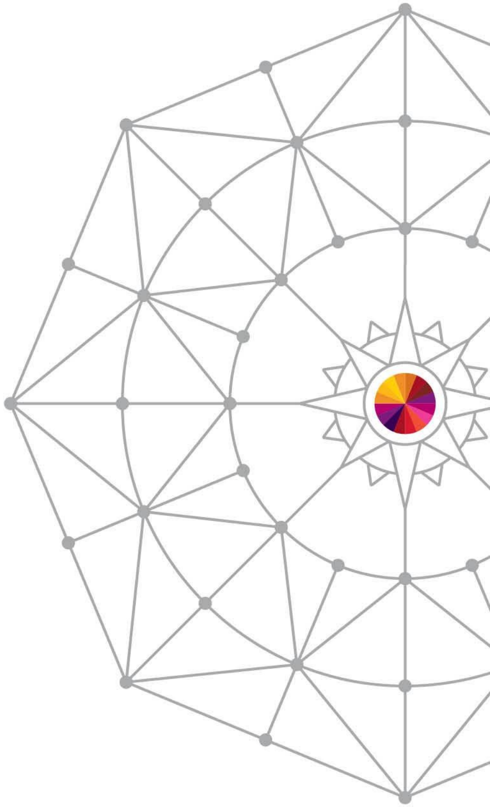
- No need – it is pre-installed!

- 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*

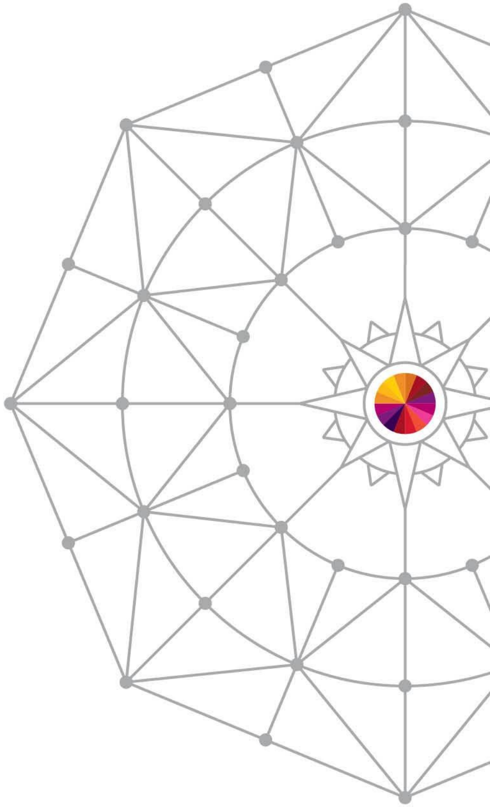
Overview of RACF activation

- 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



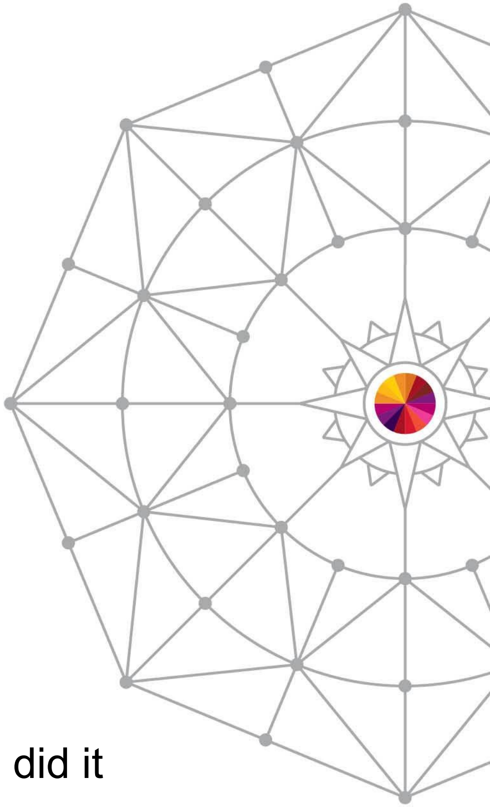
Introduction to RACF on z/VM

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



What are 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

Most common general resource classes on z/VM

VMBATCH	Allows use of DIAG D4 (alternate userid)
VMCMD	Certain CP commands and other requests
VMDEV	Real devices (new in z/VM 6.2)
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 are a classic case
    - FTP server on a modern system
  - The name of the resource is the userid that is the target of the Diag D4

- **VMLAN**
  - Allows virtual machines to connect (couple) to restricted VM LANs
    - VSWITCH and restricted guest LANs
  - CP SET (VSWITCH | GLAN) GRANT commands 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
    - The *vlanid* is only present for VLAN aware VSWITCHes

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

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
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
  - 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 is correct
    - MAINT.2190 is also correct

- **VMDEV**
  - Real devices
  - Resources are named RDEV.*rdevno.sysname*
    - The *rdevno* is the real device number, 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

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 needed on most systems

- **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 user
    - SPOOL PRT TO user
    - TRANSFER TO user
    - CLOSE TO user

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 needed on most systems

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

- **FACILITY**
  - Allows service virtual machines to authenticate directly with RACF
  - This is usually 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, though
  - Permission to a user's surrogate profile also allows you to also use the CP FOR command to that user
    - You must also have Privilege class C or be the secondary user to that id.

Defining resource classes

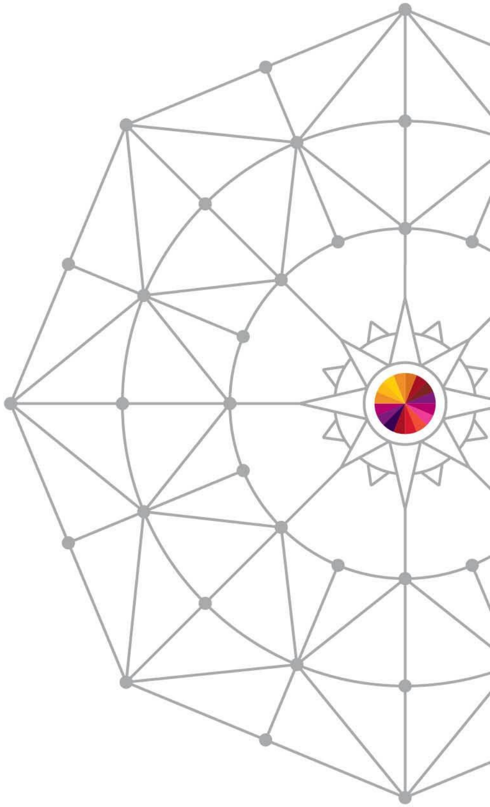
- By default, only 2 resource classes are active:
  - **USER** Allows you to logon to the system
  - **TERMINAL** Allows you to use a terminal to logon

- 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 by default

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
  - Note: Each permission includes all permissions below it

– **ALTER**	Allows full control of the resource
– **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
  - Starting with z/VM 6.2 for the VMMDISK class, this is not true!
    - The documentation has a suggestion for an alternate way to achieve this.

Access permissions details

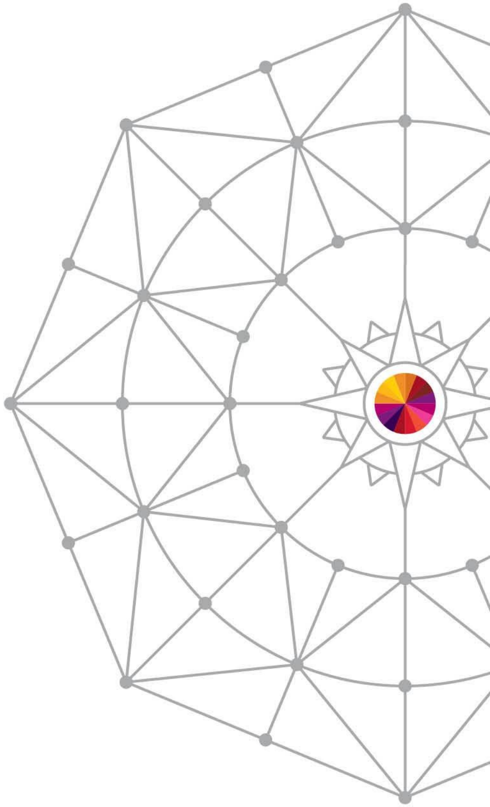
- Details about access permissions for some resources
  - If an access permission isn't listed for a class, it has no additional meaning
- **VMMDISK**
  - **READ**: Link mode R **UPDATE**: Link mode W
  - **CONTROL**: Link mode M **ALTER**: Link mode MW
  - Note: ALTER access for the VMMDISK class is an exception to normal rules
- **VMDEV**
  - **READ**: Attach read only **UPDATE**: Normal read/write attach
  - **CONTROL**: Attach r/w with SYSCTL operand allowed
- **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 your 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 over all RACF profiles in the RACF database
  - Allowed to set RACF options

- **AUDITOR**
  - Allowed to 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)



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) to the system
    - A shared userid that is revoked is not allowed to logon
- **PROTECTED** (new starting with z/VM 6.2)
  - A user without a logon password (NOPASSWORD) or logon phrase (NOPHRASE)
    - Newly added users 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 the argument:
    - `rac permit operator.191 class(vmmdisk) id(maint) access(control)`
  - – Any command output is written to your terminal and to RACF DATA A
- **RACF MODULE**
  - – Starts a RACF command session for multiple RACF commands
  - – Must enter END to leave the session

```
racf
RPITMP001I RACF/VM SESSION ESTABLISHED. TO TERMINATE ENTER "END"
RPITMP002I ENTER RACF COMMAND OR "END" TO EXIT
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 userid 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

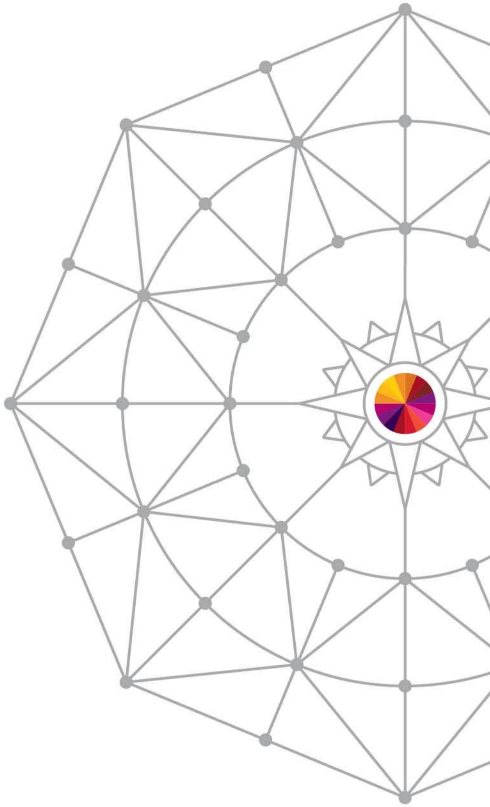
Set RACF options – SETROPTS command

- 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
  - etc.

- 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

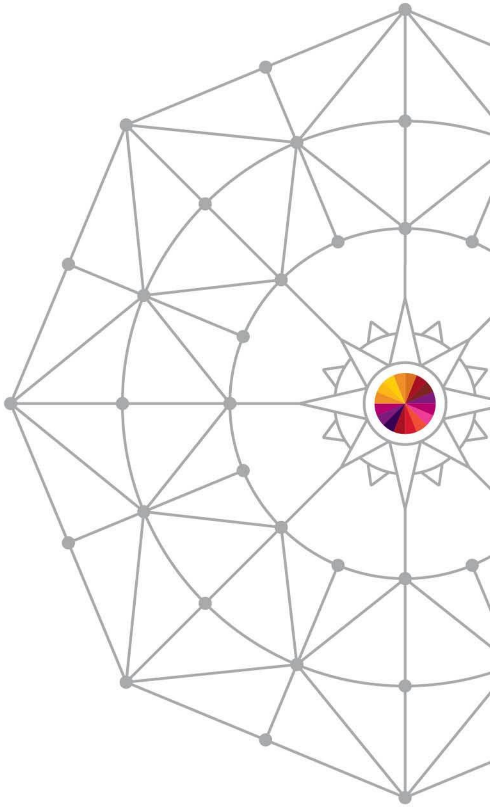


SETROPTS command options

- **CLASSACT**
  - Activates general resource classes
  - SETROPTS CLASSACT(VMMDISK VMRDR)
- **PASSWORD**
  - Sets password rules
    - Maximum change interval (1 to 254 days)
    - Expiration warning (1 to 255 days)
    - History (number of old passwords not allowed to be reused, 1-32)
    - Number of logon attempts before an automatic revoke (1 to 254)
    - Minimum length
    - Rules for types of characters in certain positions
      - *rule1(length(8) alpha(1,8) alphanum(2:7))*
- **RACLIST**
  - Cache selected resource profiles in memory – avoids disk I/O
  - Should only be used for classes with frequently referenced profiles
  - RACLIST(..) REFRESH is used to update the cache
  - Automatically propagated to other SSI members

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
  - i.e. your privilege class or directory options
- Event profiles define the RACF authorization checks that are active
  - Normally only one profile for the entire system
  - Overriding profiles for individual users (overrides system profile)
- By default, RACF checks all of the VM events
  - Listed on the next 2 charts
  - You must customize RACF to remove checking as you require

List of controlled events

COUPLE.G	Couple to restricted guest lan or VSWITCH
FOR.C	FOR command, IBMclass C
FOR.G	FOR command, IBMclass G
LINK	LINK command or directory statement
MDISK	Directory statement or LINK to own minidisk
STORE.C	STORE host memory command, IBMclass C
TAG	TAG command, for RSCS processing
TRANSFER.D	TRANSFER and CHANGE, IBMclass D
TRANSFER.G	IBMclass G spooling commands
TRSOURCE	TRSOURCE command

List of controlled events, continued

APPCPWVL	Used to verify passwords on APPC connect
DIAG088	Use of Diag 88 (Check auth and link minidisk)
DIAG0A0	Use of Diag A0 (Obtain ACI Groupname)
DIAG0D4	Use of Diag D4 (Set Alternate User ID)
DIAG0E4	Use of Diag E4 (Define Full-Pack Overlay)
DIAG280	Use of Diag 280 (Set POSIX security values)
RSTDSEG	Access to restricted saved segments
RDEVCTRL	Attach, Dedicate, or Give of of real devices

Creating event profiles

- To change the VM events checked by RACF, you must create an event profile
- The profiles have a dual purpose
  - Access checking
  - Auditing (not discussed here)
    - ➔ Come to session 14593 on Thursday at 11 AM!
- 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 <u>stop</u> control of selected events
    - By default, all events are controlled

Resource profile for my system

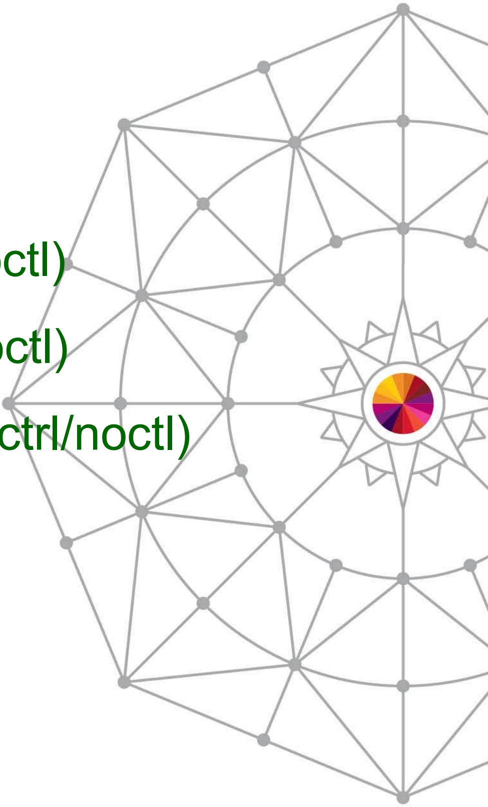
- An example based on my needs for a lab system
    - Note: *Not based on IBM security policy!*

- 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
```



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

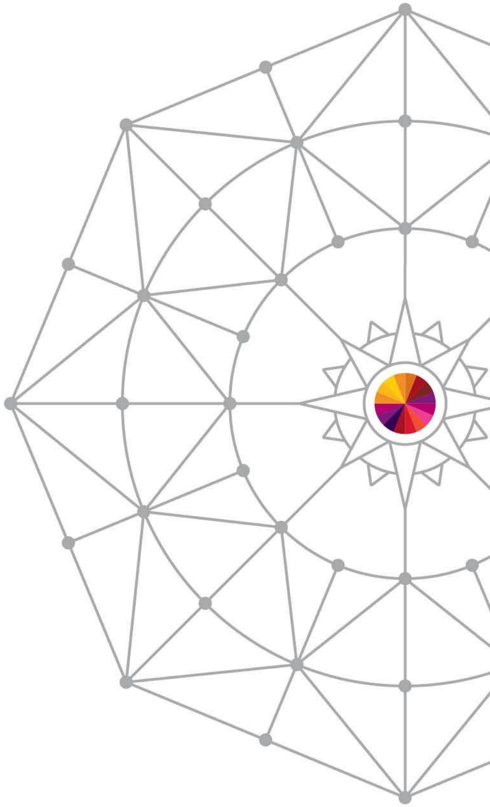
- Profiles can be created to override the system profile for specific users
  - They are named USERSEL.*userid* in the VMXEVENT class

- 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

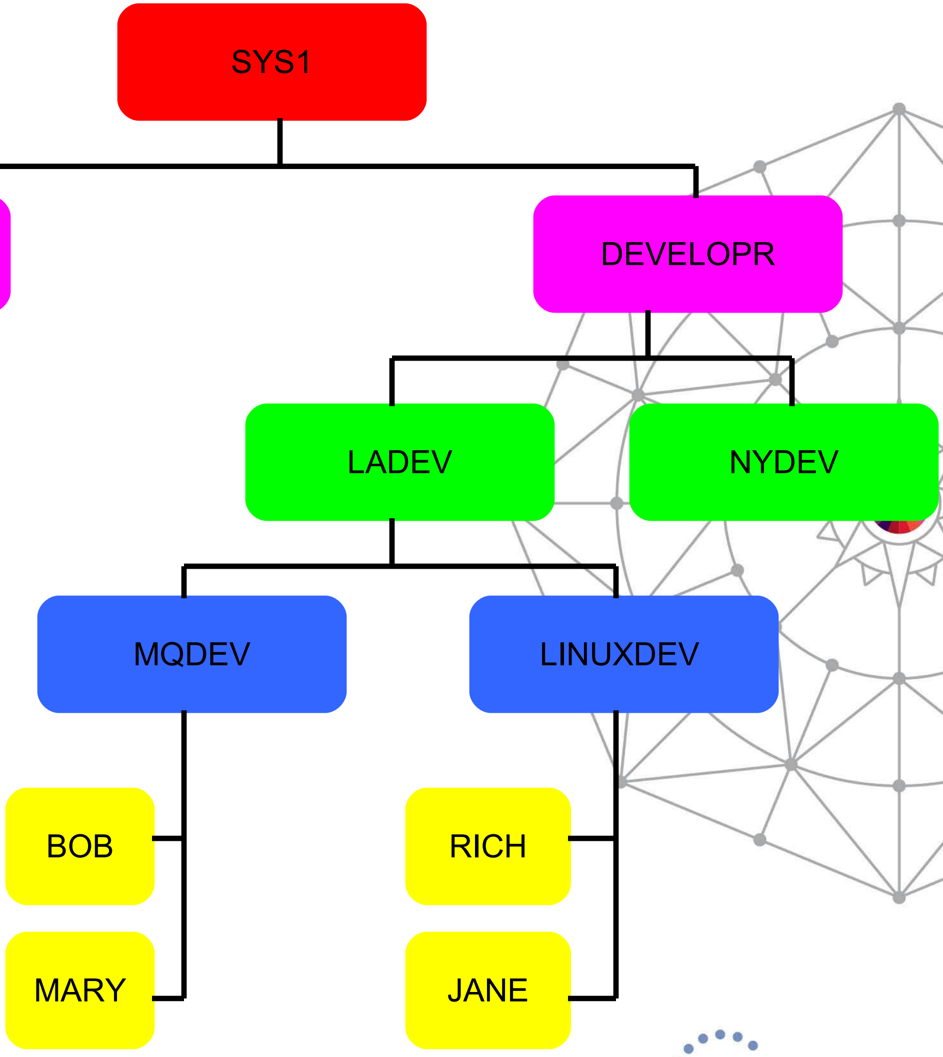
- Groups help with administration of your z/VM system
  - Put user ids with similar roles into groups
    - Linux ids
    - System Administrators
    - Service Virtual Machines (SVMs)
  - New user ids performing the same role just need to be added to the group

- 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

Group Structure

- 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



Using Groups

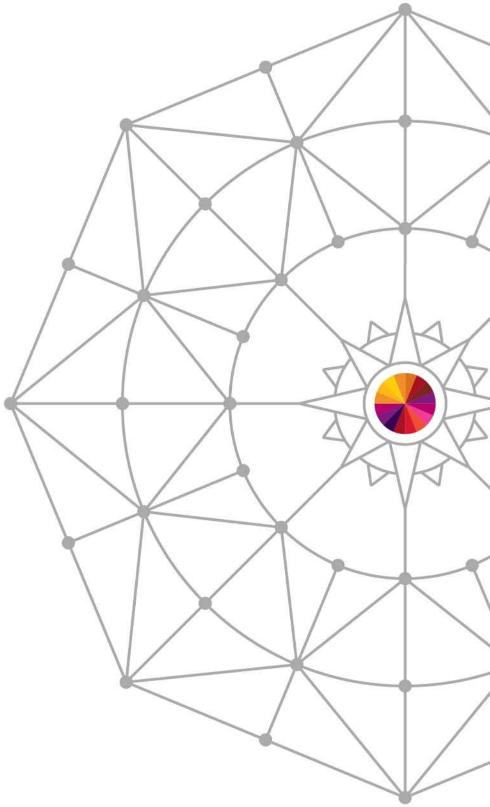
- Becoming a member of a group
  - RACF calls this “connecting” a user to a group
- Naming groups
  - Same “naming space” as user ids – hard to tell them apart!
  - Use a naming convention for groups
    - i.e., start with a special character ($, @, or #), G, end with $, etc.
- Specified user ids can 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 $linux owner(lnxadm) supgroup(sys1)
- Give the LNXADM id authority to connect Linux servers
  - rac connect lnxadm group($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

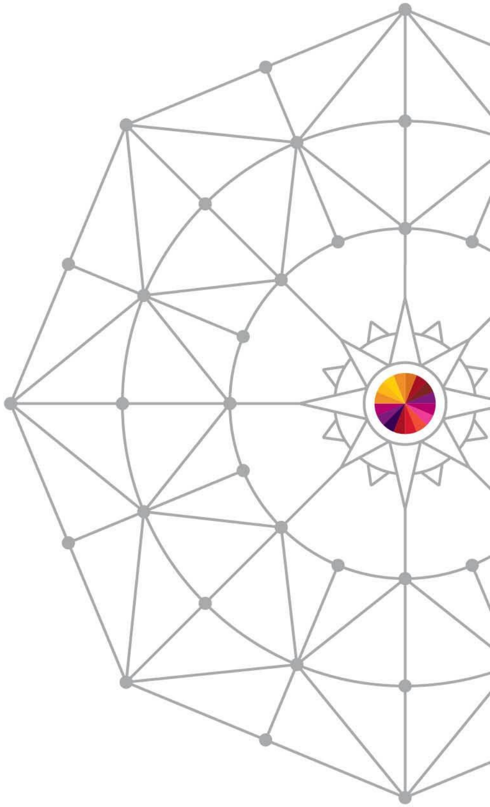


How to use Shared User ids

- Some user ids may need to be shared by multiple users
  - MAINT, MAINTvrm, OPERATOR, TCPMAINT, PERFKIT, etc.
  - Sharing the passwords is not allowed!

- 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
  - Limited to only 8 unshared ids per shared id



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.maint630 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(**$sysprog**) access(read)
  - rac permit logonby.maint630 class(surrogat) id(**$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 04/25/13 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)

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
- **VM documentation centers**
  - http://publib.boulder.ibm.com/infocenter/zvm/v6r2/index.jsp
  - http://pic.dhe.ibm.com/infocenter/zvm/v6r3/index.jsp

The End

Thank you for listening!

Session 14791

Contact information:

Bruce Hayden
bjhayden@us.ibm.com

IBM Plant #1
Endicott, NY