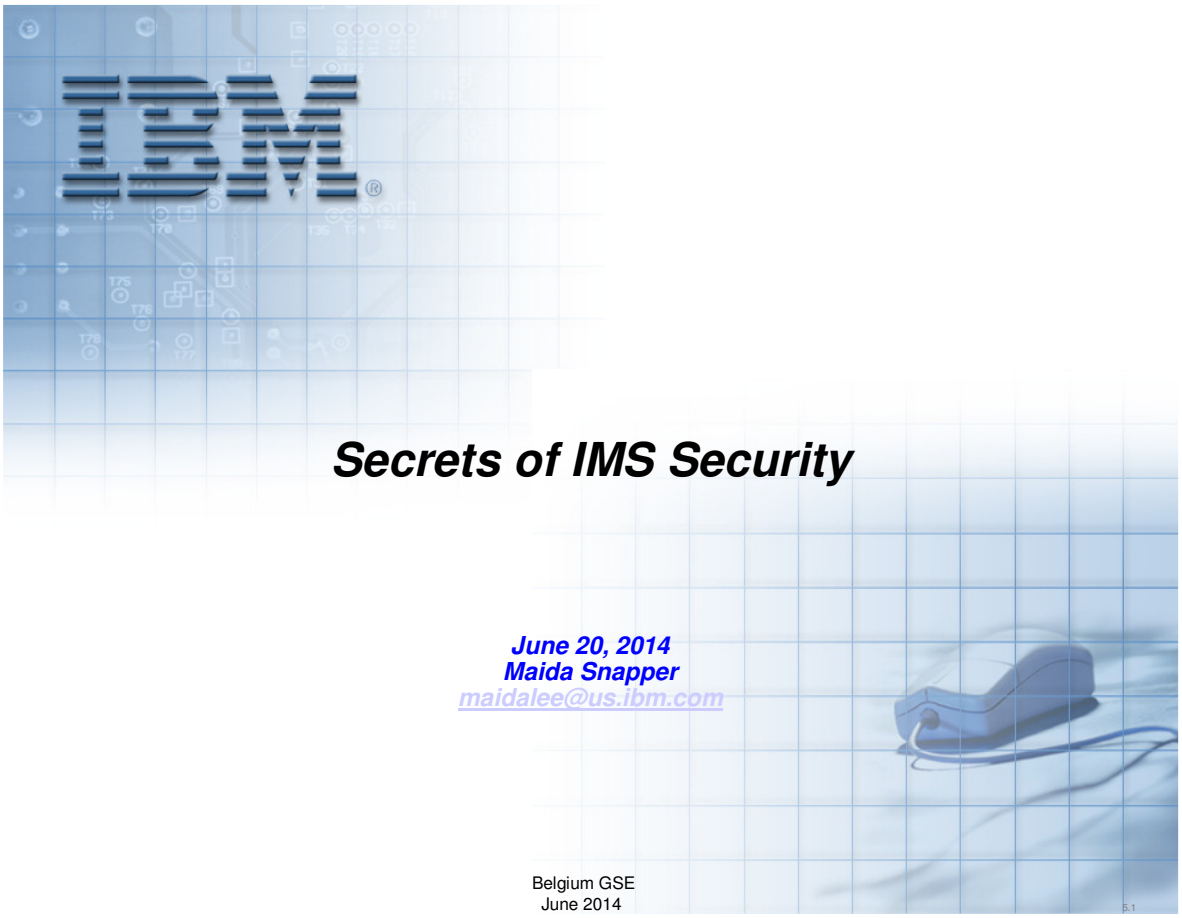


Secrets of IMS Security

June 20, 2014
Maida Snapper
maidalee@us.ibm.com

Belgium GSE
June 2014

Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

Agenda

What is IMS?

When and how does IMS talk to RACF?

How do you set up RACF profiles to protect IMS resources?

How can you tell if IMS is secure?

What is IMS?

IMS Today is a powerful DB and TM system with significant systems services, built on and exploiting z processors and operating systems.

IMS DB can be used with IMS TM or CICS applications, with DB2 stored procedures, WebSphere ejbs, etc.

IMS TM can access IMS DB, DB2, or other data.

IMS Systems Services includes support for industry-standard Java applications and database connectivity.

It also includes the integrated IMS Connect function, which provides open connectivity support to IMS applications and operations.

IMS applications and data can also use and be accessible across platforms to a wide variety of other environments.

Interoperability is provided between Java, Cobol and PL/I applications, and between DB2 and IMS databases.

DB2 Stored procedures and WebSphere ejbs can access IMS TM applications, as well as IMS DB data.

IMS also provides storage/retrieval of decomposed, or intact, XML data in IMS DB databases, and provides support for XQuery, the new standard XML interface.

The IMS SOAP Gateway is a lightweight XML-based solution. It enables access of IMS applications as Web Services for non-WebSphere environments, like Microsoft .Net.

It can assist with modernization, Application development, and Business Integration.

DATABASE Manager

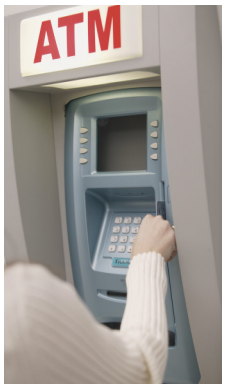
and

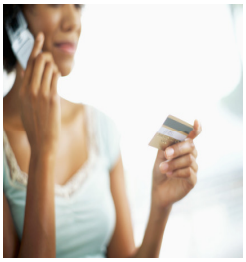
TRANSACTION Manager

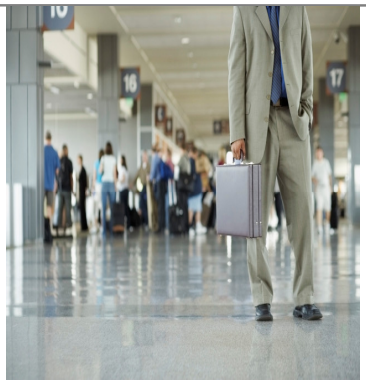
The World Depends on IMS

IMS is a part of everyday life……..

The world depends on IMS.
Chances are you're using IMS when you turn on a light, make a call, use an ATM, send packages, run a business, and much more.
IMS is the product of choice for critical on-line operational applications and data –
providing ultra-high availability, performance/capacity, integrity, and low cost


ATM






FRAGILE

How can you protect these important business functions?

Security Facilities IMS Can Use

- **RACF (or other SAF product)**
- Encryption
- IMS default security
- Program Specification Block (PSB)
- VSAM password protection
- Application-based security
- Physical security
- IMS Exits

At least one form of protection is available for each resource.

The protection for each IMS resource may be provided by one or more security facilities, such as RACF and/or user exit routines.

How and when does IMS talk to RACF?

IMS talks to RACF through the z/OS SAF interface. The security product does not have to be RACF.

IMS can talk to any security product that conforms to the rules of the SAF interface including ACF2, Top Secret, or a product you write yourself.

The SAF Interface

- IMS calls RACF through the SAF interface
  - RACROUTE call
- RACF builds Accessor Environment Element (**ACEE**) for each signed on user
  - Constructed by RACF when user signs on
  - Deleted when user signs off
  - Contains a description of the user's security environment

z/OS Security Server RACF RACROUTE Macro Reference
z/OS Security Server RACF Data Areas (for description of ACEE)

SAF is part of the base z/OS Operating System.

SAF usually works with an External Security Manager (ESM) such as RACF.

SAF is invoked at many points within the z/OS Operating System where security decisions have to be made.

SAF is invoked to perform Authentication, Resource Access control, and Auditing.

SAF can be invoked by other software

An accessor environment element (ACEE) is a control block that represents the security environment for the user.

An ACEE is constructed for IMS users during sign on and provides a description of the current user, including userid, current connect group, user attributes, and group authorities.

Each user in a group requires a level of group authority for that group. If a user is connected to several groups, the user has a level of group authority for each group.

Basically, a user's authority to access a resource while operating in a RACF-protected system at any time is determined by a combination of user's identity,

user's attributes, user's group authorities, security classification of the user and the resource profile, and access authority specified in the resource profile.

For applications, such as IMS, that permit multiple users per address space, IMS can request RACROUTE REQUEST=VERIFY CREATE to return a pointer to the ACEE that represents a user.

IMS keeps track of user/ACEE relationships and passes the appropriate ACEE pointer to RACROUTE REQUEST=AUTH and RACROUTE REQUEST=VERIFY (CHANGE or DELETE).

When IMS Comes Up and Initializes

- IMS calls RACF to load general resource profiles into data spaces
  (DATASET, Group, User profiles not eligible)

  RACROUTE REQUEST=**LIST**,GLOBAL=YES

- RACF builds ACEEs for IMS user ID (and DL/I, DBRC)

.

When an application like IMS RACLISTs a class using RACROUTE REQUEST=LIST,GLOBAL=YES, the RACLISTed profiles are stored in a data space.

The data space can be shared by many applications.

Applications that issue a subsequent RACROUTE REQUEST=LIST,GLOBAL=YES for the same class simply access the data space built by the first application.

When all applications have relinquished their access to the data space by issuing a RACROUTE REQUEST=LIST,ENVIR=DELETE request,

the data space can be deleted by issuing a SETROPTS NORACLIST(classname) command.

The SETROPTS NORACLIST command processes not only the class specified by classname, but also all valid classes that share the same POSIT value as classname.

If you issue a SETROPTS RACLIST for that class, RACF rebuilds the data space from the RACF database profiles and replaces the existing data space.

User and group profiles are not loaded into a dataspace. An I/O to the RACF database is required to access a user/group profile.

DATASET profiles are not loaded into the dataspace but a RACF caching scheme is used to reduce or eliminate I/O.

When A User Signs On

- IMS calls RACF for user ID verification

```
RACROUTE REQUEST=VERIFY,
                ENVIR=CREATE
                USERID=
                GROUP=
                PASSCHK=YES/NO
                PASSWRD=
                APPL=sapplid
                TERMID=
                ACEE=addr…..
```

- RACF verifies user ID, password, group, physical terminal, **application**
  - *sapplid* defaults to *imsid*
- RACF builds ACEE
- RACF returns ACEE address and SAF return code to IMS
- IMS logs x'16'

z/OS Security Server RACF Macro Reference

RACF user verification is invoked when an IMS/ESA user enters the /SIGN ON command. RACF verifies:

- *Is this a valid user (defined to RACF)*
- *Is the password correct for this userid (USERID)*
- *Is user authorized to connect to the specified group (GROUP)*
- *Is the user entering a new password (PASSWRD)*
- *Is the user authorized to sign on to this IMS (APPL)*
- *Is the user authorized to use this physical terminal (TERMID)*
- *Is the sign on a being done during the authorized times*

TERMID is the physical terminal being used. If it is protected in the RACF TERMINAL/GTERMNL class, then the user must be authorized in the access list.

APPL is the application the user is accessing. IMS passes the value of *sapplid* specifed in DFSDCxxx member of PROCLIB. *sapplid* defaults to *imsid* .

When applications like IMS, TSO, CICS, etc. are protected in the RACF APPL class, the user must be authorized in the access list.

If IMS RAS security is activated (ISIS=R|A), all dependent region user IDs must also be authorized in the access list of the *imsid* in the APPL class (see next slide)

When A Dependent Region Connects to IMS

- Only if IMS RAS security is active:
  - RACF verifies user ID, password, group, **application**
  - Application is *imsid* (not *sapplid)*

If IMS RAS security is activated (ISIS=R|A), all dependent region user IDs must also be authorized in the access list of the *imsid* in the APPL class.

When A Resource Is Accessed

- IMS calls RACF to check authorization
  - IMS passes ACEE, CLASS, ENTITY, ATTR

```
Example:
RACROUTE REQUEST=FASTAUTH,LOG=ASIS,
          ACEE=addr,
          CLASS=CIMS,
          ENTITY=DIS,
          ATTR=READ
```

- RACF sends SAF return code to IMS
  - 0 user is authorized, IMS grants access
  - 4 resource has no profile, **IMS grants access**
  - 8 user is not authorized
    - IMS **may** deny access and log x'10'
    - RACF issues ICH408I message and logs SMF TYPE 80

Who is attempting the access? What is the class of the resource? What is the name of the resource? What type of access is being requested (e.g. READ or WRITE)?

RACF passes return code and reason code information back to SAF and SAF returns control to IMS with its own return and reason codes, and with RACF's return and reason codes.

IMS can examine the Return Code and Reason Code values and make a decision on the basis of those codes.

Depending on details about the user, system settings, and resource, logging may take place.

RACF calls SMF to perform the logging. SMF is responsible for placing the log records onto the SMF log datasets.

LOG= describes the auditing options

ASIS - RACF performs auditing if its authorization check results in success (RC=0) or failure (RC=8), and determines whether auditing is necessary based on the following conditions:

The user's UAUDIT setting

The AUDIT, GLOBALAUDIT, and WARNING options in effect for the resource

If SETR SECLABELAUDIT is in effect, then the AUDIT options in the resource SECLABEL profile

The pre- or postprocessing installation exit's indication of whether or not to do auditing.

NOFAIL - If the authorization check fails, the attempt is not recorded. If the authorization check succeeds, the attempt is recorded as in ASIS.

NONE - The attempt is not recorded. LOG=NONE suppresses both messages and SMF records regardless of MSGSUPP=NO.

For FASTAUTH, the default is LOG=NONE; IMS (as of version 10) always specifies LOG=ASIS

If A User Is Not Signed On

- If a USER ID is not available, IMS passes zeroes in the ACEE field.
- IMS calls RACF to check authorization

Example:

```
RACROUTE REQUEST=FASTAUTH,LOG=ASIS,
          ACEE=00000000,CLASS=CIMS,ENTITY=DIS,ATTR=READ
```

- RACF uses the ACEE of the “home” address space
  - usually home is IMS control region
  - in some cases home is dependent region

IMS requires all ETO terminals to sign on.

Static terminals are not required to sign on unless specified by IMS parameters (SIGNON parameter in DFSDCxxx member of PROCLIB).

When A User Signs Off

- IMS calls RACF to delete the user's ACEE

  RACROUTE REQUEST=**VERIFY**,ENVIR=DELETE,ACEE=addr…

- IMS logs x'16'

When IMS Shuts Down

- IMS calls RACF to deregister interest in the resource classes
- RACF deletes the ACEE for IMS user ID
- **GLOBAL=YES data spaces are not deleted**

When IMS terminates (and “signs off”), RACF does not delete the data space containing the IMS resource profiles.

When IMS comes up again and issues a RACROUTE REQUEST=LIST,GLOBAL=YES for the same class, RACF will access the data space that is already there.

That is why is it necessary to do RACF REFRESH in order for any changes made to the RACF database to take effect in the RACF data space.

When all applications have relinquished their access to the data space by issuing a RACROUTE REQUEST=LIST,ENVIR=DELETE request,

the resources in the data space can (optionally) be deleted by issuing a SETROPTS NORACLIST(classname) command.

Summary: IMS Calls RACF when…….

- When IMS comes up:
  - RACLIST
- When user signs on
  - VERIFY (CREATE)
- If IMS RAS security is active, when dependent region connects
  - VERIFY (CREATE)
- When user accesses a resource
  - FASTAUTH, AUTH
- If IMS RAS security is active, when dependent region accesses a resource
  - FASTAUTH, AUTH
- When user signs off
  - VERIFY (DELETE)
- When IMS comes down

How do you set up the RACF definitions for IMS users and resources?

Setting Up RACF

- Define Resource Classes in Class Descriptor Table (CDT)
  - Static and Dynamic
- Activate Resource Classes
  - CLASSACT
- Populate the RACF database
  - Add group & user profiles
    - ADDUSER
    - ADDGROUP
  - Connect users to groups
    - CONNECT
  - Define resource profiles
    - RDEFINE
  - Create access lists
    - PERMIT

The is a list of high level functions that need to be performed to implement RACF security for IMS resources.

Installations that are currently using RACF will probably already have defined groups, users, and connected users to the groups.

To secure IMS resources using RACF, the installation will need to define resource security requirements:

- *In existing or default RACF resource classes*
- *Or create new, installation-defined RACF classes*

Once the security definitions have been created and stored on the RACF database, you will need to refresh the security information in storage to have the new security definitions take effect.

IMS Resources RACF Can Protect

- IMS itself
- Commands
- Transactions
- Datasets
- Databases
  - records, segments, fields
- Programs (PSBs)
- Terminals (Logical, Physical)
- Coupling Facility Structures
- IMSPlex and XCF group membership

At least one form of protection is available for each resource.

In many cases multiple security facilities may be used to protect a single resource.

A resource is identified by

Resource Class + Resource Name

A resource is defined by the combination of resource class and resource name.
For example, an IMS command like DISPLAY belongs to a class that contains IMS commands.
The default class for commands is CIMS.
The name of an IMS command resource is always its first 3 characters.

IMS Resource Classes

Default IMS RACF General Resource Classes

RACF default resource classes used exclusively by IMS (RCLASS=IMS)

CIMS | DIMS	Commands
TIMS | GIMS	Transactions
IIMS | JIMS	Application programs (PSBs)
LIMS | MIMS	Logical terminals (LTERM)
AIMS	APSB (Allocate PSB) for CPIC-PSB and ODBA
RIMS	Asynchronous hold queues for RESUME TPIPE call
PIMS | QIMS	Databases (for AUTH call)
FIMS | HIMS	Database fields (for AUTH calls)
SIMS | UIMS	Database segments (for AUTH calls)
OIMS | WIMS	Other (information in RACF for AUTH calls)

These RACF resource classes are used exclusively by IMS. The default classes are classes provided by IBM in the RACF static Class Descriptor Table.

Note that the AIMS and RIMS classes do not have a grouping classes associated with them.

RACF General Resource Classes

These RACF resource classes are also used by IMS

TERMINAL | GTERMINL
APPL
DATASET
FACILITY
OPERCMDS
STARTED
VTAMAPPL
APPCPORT
APPCLU
APPCTP

In addition to the RACF classes used exclusive for IMS resource security definitions, a number of other RACF classes may be used to secure access to IMS resources.

The other classes may be used by other subsystems, such as CICS, TSO, and other MVS subsystems.

If your RACF database is shared,

can IMS systems

have different security rules

for the same resources?

You may want to allow access to a resource in a test IMS but deny access to that resource in a production IMS.

Yes!

Because a resource is identified by

Resource Class + Resource Name

You can define your own class.

You may want to give a user different authorization to the same resource name (an IMS command, for example).
You may want to allow access in a test IMS but deny access in a production IMS.
If test and production share the same RACF database you need to find a way to differentiate the resource when it is accessed in test versus production.
If the resource cannot be differentiated by its name (an IMS command, for example), then you can differentiate it by its class. You can create installation-defined classes.

IMS General Resource Profiles

IMS resource	Resource class singular/grouping	Resource name
Transaction	T*IMS* / G*IMS*	transaction code
Command (type 1)	C*IMS* / D*IMS*	first 3 characters of command
DBRC command	FACILITY	*safhlq*.command_verb.qualifier.modifier
Command (type 2)	OPERCMDS	IMS.*plxname*.command_verb.command_keyword
Program (PSB)	I*IMS* / J*IMS*	program name
Logical terminal	L*IMS* / M*IMS*	logical terminal name (lterm)
CF structure	FACILITY	CQSSTR.*structure_name* or IXLSTR.*structure_name*
IMS Control Region	APPL	*imsid*
IMSPlex (CSL)	FACILITY	CSL.*imsplexname*
XCF group (Client bid)	FACILITY	IMSXCF.groupname.*membername*
Dataset	DATASET	*dataset name*

The portion of a resource class or name that is shown in blue italics on this chart is the part you can change to make a resource unique.

Member class profile names must conform to the rules shown in this chart.

Grouping class profile names can be any 1-8 alphanumeric characters you choose.

Notice that there are 2 kinds of IMS commands: type 1 and type 2. Type 2 commands are newer and can only be entered through the Operations Manager address space.

They are sometimes called plex commands.

IMS points to its own set of security rules

using the IMS RCLASS parameter

RCLASS = position 2-8 of the resource class

The RCLASS parameter can be specified in the DFSPBxxx or the DFSDCxxx member of PROCLIB.
If specified in both places, DFSPBxxx overrides DFSDCxxx.

Default RACF Resource Classes

RCLASS defaults to ***IMS*** when not specified

T*IMS* **G***IMS*	**C***IMS* **D***IMS*	**I***IMS* **J***IMS*	**L***IMS* **M***IMS*
transactions	*commands*	*programs*	*logical terminals*




The boxes represent classes. They are “empty” until you define some profiles for the resources in the class.
It’s OK to have an empty box with no profiles defined.
The 4 classes shown with their corresponding grouping classes are defined by IBM
and delivered with the RACF product. You can use these or you can define your own.

Sample Installation-defined RACF Resource Classes

Example of some installation-defined resource classes when RCLASS=***IMSTEST***

T*IMSTEST*	**C***IMSTEST*	**I***IMSTEST*	**L***IMSTEST*
G*IMSTEST*	**D***IMSTEST*	**J***IMSTEST*	**M***IMSTEST*
transactions	*commands*	*programs*	*logical terminals*

If you define your own classes as shown here, you should model them on the default classes on the previous slide.

Defining a New IMS RACF Resource Class

- Class name 1-8 alphanumeric characters
  - First character must be the same as its corresponding default class:
    - C, D, T, G, I, J, L, M, A, R, etc.

- You must define both the singular and its grouping class.

- **Model new classes on the corresponding default class**
  - Optionally can change the POSIT value
  - Do not change MAXLNTH

- Activate new resource classes
  SETR CLASSACT(*classname*)

The singular and grouping class are a pair and both must exist.

If you are adding a new installation-defined class, be sure you also add its grouping class.

RACF Resource Class

- Class Descriptor Table (CDT)
  - entries can be defined statically (IPL) or dynamically (no IPL)
  - maximum 1024 entries
    - 256 default classes delivered with RACF
    - 768 can be installation-defined
  - loaded at IPL by merging static, then dynamic class descriptors
    - dynamic entry replaces static of the same name
    - if merge reaches 1024, RACF warns entries are being ignored
  - CDT processes a paired member and grouping class together.
- There is no need to update the RACF Routing Table
  - ACTION=RACF is the default

IBM supplied CDT entries are documented in Appendix C of the z/OS Security Server RACF Macros and Interfaces

There are two ways to define resource classes:

Define them in the dynamic class descriptor table, using RDEFINE and RALTER commands.

For information on how to do this, see z/OS Security Server RACF Security Administrator's Guide.

Define them in the static class descriptor table, using the ICHERCDE macro

where each installation-defined class entry becomes a CSECT in load module ICHRRCDE.

Guideline: Define your classes in the dynamic class descriptor table, to avoid the need to re-IPL.

Sample IMS Resource Class Description for Transactions

TIMS

POSIT=4
OTHER=ALPHANUM
MAXLNTH=8
DFTRETC=4
DFTUACC=NONE
GROUP=GIMS
OPER=NO
ID=9
FIRST=ALPHANUM

GIMS

POSIT=4
OTHER=ALPHANUM
MAXLNTH=8
DFTRETC=4
DFTUACC=NONE
MEMBER=TIMS
OPER=NO
ID=10
FIRST=ALPHA

The TIMS and GIMS definitions on this chart was copied from z/OS Security Server RACF Macros and Interfaces Appendix C

IMS resource classes have a default return code of 4.

Secret: *Bigger is not better*

If you define a new IMS resource class,

use the same MAXLNTH

as the corresponding default IMS resource class.

Be careful when specifying MAXLNTH.
IMS passes up to an 8 character resource class name in the call to RACF
and RACF reads MAXLNTH number of characters.
If MAXLNTH is greater than what IMS passes to RACF,
RACF may pick up “junk” for part of resource class name and results can be unpredictable.

POSIT Values

You can specify POSIT values 19–56 and 128–527.

POSIT values 0–18, 57–127, and 528–1023 are reserved for IBM use and should not be used for your installation-defined class entries unless you intend to share SETROPTS options with an IBM supplied class**.**

There are 1024 possible numeric POSIT values. You can specify POSIT values 19–56 and 128–527.
POSIT values 0–18, 57–127, and 528–1023 are reserved for IBM use
and should not be used for your dynamic class entries **unless you intend to share SETROPTS options with an IBM supplied class.**
If you use a reserved POSIT number that is not currently used for an IBM supplied class,
be aware that in the future IBM might create a supplied class with this POSIT number.
If this conflict occurs, processing results for your class will be unpredictable.
Classes with the same POSIT value are administered as a single class when you specify a class option, such as CLASSACT or RACLIST.
You would add a new class with a unique POSIT value
when you want to administer it separately from any other class.

Secret: ***You might accidentally deactivate a resource class***

CLASSACT | NOCLASSACT

affect all resources with shared POSIT value.

Others: AUDIT, STATISTICS, GENERIC, GENCMD, GLOBAL, LOGOPTIONS, RACLIST, etc.

When a POSIT value is shared between two or more classes,
certain RACF processing options are controlled in the same manner (simultaneously) for all classes with the shared POSIT value.
Any of the following SETROPTS affect the resources or profiles with shared POSIT value:
CLASSACT, AUDIT, STATISTICS, GENERIC, GENCMD, GLOBAL, LOGOPTIONS, RACLIST and ALTUSER *userid* CLAUTH

If you deactivate a class using SETROPTS NOCLASSACT,
RACF deactivates all classes in the class descriptor table that have the same POSIT value as the class you specify.
For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries.
If you deactivate any one of these classes, you deactivate all of them.
For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

Secret: ***If a required class is undefined or inactive, IMS will abend.***

- When IMS comes up:
  - IMS calls RACF to load general resource profiles into data spaces

RACROUTE REQUEST=**LIST**,GLOBAL=YES

Define and Activate classes or IMS may abend U0166

Example: to activate CIMS
(and all other classes with the same POSIT value as CIMS):

SETROPTS CLASSACT(CIMS)

IBM-supplied classes need to be activated too.

When you install a new RACF system, initially only a few RACF classes are active (for example USER, GROUP, and DATASET).

When you specify RACF, required classes must be active even if you have not defined any resource profiles in those classes.

For example, if RAS security is activated either by any RAS specification on the SECURITY macro or by ISIS=R|A then classes IIMS, LIMS and TIMS must be active or IMS will ABENDU0166 at initialization.

.

Profiles

RACF Profiles

- **Group** profile

Defines group name, group authority, subgroup, ...

- **User** profile

Defines individual user ID, password, user attributes, connect groups, ...

- **Resource** profile

Defines Universal Access and authorized users (access list)

  - Discrete
  - Generic
  - Fully Qualified Generic

Profiles are data that describes users, groups, and resources.

Group profiles define groups of users. You may think of a group as a department in an organization.

User profiles define an individual user. The profiles contains the userid, password, and user attributes such as whether the user has group special, special, or operations authority.

Resource profiles define the resources that are secured, such as transactions and commands. They also specify the default access level for unauthorized users.

RACF commands are used to add, alter, and delete security profiles in the RACF database.

Who/what are IMS users?

Why do they sometimes need to have strange user IDs that don't conform to your installation standards?

Secret: An IMS User Isn't Always a Person

A user ID can represent a...

- Person
- Job, Started Task (BMP, utility, etc.)
- Transaction
- Command
- Logical terminal (LTERM, Master, WTOR, TCO)
- Program (PSB)
- TCO (Time Controlled Operations) script
- IMS Master terminal or system console WTOR

How an IMS Transaction or Command can be a "User"

A programmer writes a program that issues an IMS command.
When the program runs, RACF checks to see if the program is authorized to issue the command. RACF needs a user ID.

The three choices for user ID in this case are:

1) User ID of the person who entered the transaction that invoked the program
   - Resource is the command
   - **This choice allows the person to also enter the IMS command directly**

2) User ID is the transaction code
   - Resource is the command
   - Recommend NOPASSWORD
     - IMS calls RACF to VERIFY the ID with PASSCHK=NO

3) User ID is the command
   - Resource is the transaction code
   - Recommend NOPASSWORD and RESTRICTED
     - IMS calls RACF to VERIFY the ID with PASSCHK=NO

If an IMS transaction issues commands (CMD call), then we call it an AOI transaction (Automated Operator Interface).
SMU security authorized this by building a table (matrix) of valid transaction/command pairs. To replace this SMU security with RACF requires a user ID.

The IMS TRANSACT macro has a new parameter, AOI=YES|NO|TRAN|CMD.
The main purpose is to define the type of check – what is used as the "userid" (user's userid, trancode or command code), and what it is authorised against (command or trancode).

When AOI=YES is specified, the authorization of the commands for the CMD calls issued by the transaction is done using the userid of the user who entered the transaction.
For some environments, if a Get Unique call has not yet happened, then the program name rather than the userid is used for the authorization.

The TRAN specification is similar to that of YES, but requests that the transaction code, be used instead of the userid of the user who entered the transaction.
Use of the transaction code provides authorization checking more like that provided by the SMU transaction-command security.
When a transaction is defined with AOI=TRAN, the first authorization check done for AOI for the transaction results in the security environment (ACEE) being built and kept for use by future authorization checks.
In this case, the AOI transaction codes have to be defined to RACF (or equivalent product) as userids.
The transactions must also be specified on RACF PERMIT statements for each command they are allowed to issue from an AOI transaction.
IMS calls RACF with PASSCHK=NO and we recommend defining these user IDs with NOPASSWORD option.

The CMD specification is also similar to that of YES, but requests that the command code (first three characters of the command), be used instead of the userid for the authorization check.
Use of the command code provides authorization checking more like that provided by the SMU transaction-command security.
When a transaction is defined with AOI=CMD, the first authorization check done results in the security environment (ACEE) being built, and being kept for use by future authorization checks.
In this case, the IMS command codes (first three characters of IMS commands) have to be defined to RACF (or equivalent product) as a user.
The command codes must also be specified on RACF PERMIT statements for each AOI transaction that is allowed to issue them.
IMS calls RACF with PASSCHK=NO and we recommend defining these user IDs with NOPASSWORD option.
If CMD is chosen as userid, note that any transaction that is not defined to RACF will also be authorized to issue the command unless the user ID is defined as RESTRICTED.

Access Lists

RACF Access Authority

- User or Group Access Authority (ACCESS) can be:
  - NONE
  - EXECUTE
  - READ
  - UPDATE
  - CONTROL
  - ALTER
- Maximum entries in the access list of a profile is 5957
  - access list of each profile is limited to 65535 bytes
  - each user or group in the access list uses 11 bytes
- **READ is sufficient for most IMS general resources**
- UPDATE is required for some IMS general resources
  - Some Type 2 commands
  - CQS access to CF structures (SMQ and RM)
  - Registering with SCI to join an IMSplex

NONE -Specifies that the user or group not be permitted to access the resource.

READ -Specifies that the user or group be authorized to access the resource for the purpose of reading only.

This is the level of access needed to access most IMS resources. In order for users to execute commands and transactions, they need to be permitted with an access level of 'READ'.

EXECUTE -Specifies that the user or group can run programs but not read or copy them.

UPDATE -Specifies that the user or group be authorized to access the resource for the purpose of reading or writing.

CONTROL -Is used only for VSAM data sets and specifies that the user or group have access authority that is equivalent to the VSAM control password.

ALTER -Specifies that the user or group have full control over the resource.

The other items in the security profile (security classification, auditing options, warning options, and notification options) are not covered in the class.

See your RACF security administrator or the RACF manuals for more information on these options.

How much authority does IMS itself need?

- IMS needs access to its datasets
  - JCL defined
  - Dynamically allocated
- IMS does not normally need to access transactions or commands
  - If a user ID is not available, RACF uses the “home address space” user ID for authorization
    - IMS user ID
    - Dependent region user ID
- IMS does not need to be defined as privileged or trusted

Secret: RACF Resolves Conflicts

What happens if there is conflicting information in the RACF database?

RACF uses
the most restrictive UACC
the most permissive ACCESS

see z/OS: Security Server RACF Security Administrator's Guide: Resolving Conflicts among Multiple Profiles

Secret: Undefined IMS resources are authorized

What happens if the resource is not defined to RACF?

IMS allows access.

RACF sends a return code of 4 when a resource is not defined to RACF

IMS treats return code 4 and return code 0 the same.

Making RACF Changes

- To update RACF security definition
  - update the RACF database
  - refresh the RACF data space from the database by issuing
    SETROPTS RACLIST(*classname*) REFRESH
- RACF refreshes all classes with the same POSIT value
- specify the *member* classname not the grouping classname
  for example, specify CIMS not DIMS
- REFRESH must be entered on the LPAR with the dataspace
- REFRESH must be entered on all members of a SYSPLEX unless RACF is configured for SYSPLEX communication

You added a profile to RACF to protect the /STA command with UACC(NONE).

Why can everyone still issue /STA?

Did you REFRESH ?

Once the security definitions have been created and stored on the RACF database, you will need to refresh the security information in storage to have the new security definitions take effect.

Make sure you issue the REFRESH on the LPAR with RACF and its dataspace.

Refresh the RACF Dataspace(s)

- Updating a RACF resource profile updates the RACF *database*.
- REFRESH the RACF *dataspace* for the update to take effect.

For example
SETR RACLIST CLASS(CIMS) REFRESH
SETR GENERIC (CIMS) REFRESH

This brings in a new copy of all profiles in the CIMS class.
It also refreshes any other classes with the same POSIT value as CIMS.

Recycling IMS does not refresh IMS resource definitions in the RACF dataspace.

SETROPTS RACLIST(classname) REFRESH causes RACF to reload resource profiles from the database into the data space.
The scope of a RACLIST REFRESH command is the class named on the command plus any other classes sharing the same POSIT value.
See z/OS Security Server RACF Security Administrator's Guide for further information.

If your installation has two or more systems sharing a RACF database, you must issue the REFRESH on all systems to have the results effective on all systems,
unless RACF is enabled for sysplex communication.
However, if you do not perform a refresh on a system sharing a RACF database and that system needs to re-IPL, a fresh copy of the RACLISTed profiles is read from the database at IPL time.
When RACF is enabled for sysplex communication, it propagates the REFRESH command to each of the systems in the data sharing group.

SETROPTS GENERIC(classname) REFRESH
When you specify GENERIC and REFRESH, you also specify one or more classes for which you want RACF to refresh in-storage generic profile lists.
This causes all of the in-storage generic profiles in the specified general resource class (except those in the global access checking table) to be replaced with new copies from the RACF database.
The following example refreshes in-storage generic profiles for the CIMS and TIMS classes.
SETROPTS GENERIC(CIMS TIMS) REFRESH

If you specify SETROPTS GENERIC(*) REFRESH, RACF refreshes profile lists for the DATASET class and all active classes
except resource grouping classes and classes defined with the GENERIC(DISALLOWED) attribute.

For the DATASET class:
RACF caches a list of generic profiles which begin with the HLQ of a data set for which an authorization check has been done.
For example, if you open 'IMSP1.RECON1'
RACF loads the names of all the generic profiles which begin with IMSP1
To refresh the list: SETR GENERIC(DATASET) REFRESH
This purges all of the data set profiles that were cached by RACF.

Secret: ***You rarely have to recycle IMS for RACF changes***

Two rare cases when IMS has to be recycled for a RACF change to take effect:

1) For DATASET resource: if new access is given to a GROUP and IMS was not previously connected to that GROUP you have to recycle IMS.

2) For general resource: if you activate a new IMS class you have to recycle IMS to get it loaded into a RACF dataspace

Help!!!

Why is IMS discarding the RECON datasets?

Secret: RECONs Come in Sets of Three

- Each IMS has 3 RECON datasets
  - 2 copies plus a spare
- Each of the 3 RECON datasets might have a different high level qualifier
  - That's an IBM recommendation
- Users must have the same RACF access to all 3 RECON datasets
  - If VSAM open gets RACF violation, IMS discards the RECON

When IMS resources like the RECON are protected by RACF, then the user ID of DBRC needs access.

User IDs can be assigned to started tasks and jobs using the RACF STARTED class. Users who run batch DBRC jobs also need access.

If you protect the RECONs in RACF, be sure users have authorization to all 3 RECON datasets.

If the VSAM open for a RECON fails because of a RACF security violation, VSAM tell IMS the RECON could not be opened and IMS interprets the VSAM open failure as an I/O error.

IMS discards that RECON dataset. If the RECON is being accessed READONLY, the user's job will fail and the RECON will not be discarded.

RACF caches a list of generic profiles which begin with the HLQ of a data set for which an authorization check has been done.

For example, if you open 'IMSP1.RECON1'

RACF loads the names of all the generic profiles which begin with IMSP1

To refresh the list: SETR GENERIC(DATASET) REFRESH

This purges all of the data set profiles that were cached by RACF.

RACF Access Authority for the RECON dataset

- READ is sufficient for readers
  - they must specify the READONLY parameter
- UPDATE is sufficient for all accesses except DELETE and DEFINE
- ALTER required for DELETE and DEFINE
- CONTROL is never required anymore (since IMS10)

How can you tell if IMS is secure?

Determining the Security in Effect

The security in effect for a given input message is determined by ...

- IMS system definition (IMSGEN)
- IMS JCL overrides
- IMS PROCLIB
  - DFSPBxxx
  - DFSDCxxx
  - CSLOIxxx
  - DFSCGxxx
- IMS commands and restart options
  - Example: /SECURE APPC FULL
- Source of the input message
- RACF definitions
- Exits
- Program Specification Block (PSB)
- Database Definition Block (DBD)
- IMS Connect security setting

IMSGEN macros

DBD may specify encryption

Summary

What is IMS?

When and how does IMS talk to RACF?

How do you set up RACF profiles to protect IMS resources?

How can you tell if IMS is secure?

Write to me!

Maida Snapper
maidalee@us.ibm.com