Introduction to MulitLevel Security

Jun Ogata / Poughkeepsie, NY
ogata@us.ibm.com
Phone #: (845) 435-7680

Trademarks

The following are trademarks or registered trademarks of the International Business Machines Corporation:

- IBM
- MVS/ESA
- RACF
- VTAM

Agenda

- What is Multilevel Security?
- The Road to Multilevel Security
- Defining a SECLABEL
- Dominance and Equivalence
- DAC vs. MAC
- MAC Scenario
- SECLABEL-related options
- Special SECLABELs
- Q & A

What is Multilevel Security?

Multilevel security is:

- The ability to mix different categories and classes of information within the same computing environment in a controlled manner without compromise

- A combination of hardware, software, and operational procedures

- Valuable anytime there is a need to isolate data, such as:
  - When there is truly sensitive data
  - As a way of complying with evolving regulatory environment

Why Multilevel Security

- **Traditional access control mechanisms allow the resource owner to control who has access to data**
  - The data owner has the discretion to grant access, hence the term ‘discretionary access’ mechanism

- **Data classifications, if present are assigned by the data owner**
  - Data owners could misclassify data by opening a data set at one level and then writing it to another level

- **Multilevel security formalizes the classification of data and enforces a data access policy that is set by the security administrator, not the data owner**

The Road to Multilevel Security

RACF’s support for Multilevel security has evolved since the mid-80s:

- 1985: RACF 1.7 - Assignment of levels and categories to users and data objects
- 1990: RACF 1.9 - Multilevel (“B1”) support
  - SECLABELs
  - Console logon
  - NJE, RJE, JES controls
- 2004: z/OS R5 – Multilevel support
  - Extends existing Multilevel controls to TCP/IP, UNIX System Services, and DB2

The Road to Multilevel Security…

1985 / RACF 1.7: Levels and Categories:

- **Security Level** (SECLEVEL defined in the SECDATA class)
  - A name that corresponds to a level of security
  - Hierarchical relationship (higher level, more secure)
- **Security Categories** (CATEGORY defined in the SECDATA class)
  - A name that represents a nonhierarchical characteristic of data

- Levels and categories are assigned to users and data objects
  - When a user access a resource which has a SECLEVEL or security category, the user must have an equal or higher SECLEVEL and all of the categories that are associated with the resource

The Road to Multilevel Security…

1990 / RACF 1.9: security label or SECLABEL:

- This included enhancements to MVS/ESA 3.1.3, RACF, JES2, JES3, TSO, VTAM, DFP, and PSF

- A security label or SECLABEL consists of two parts:
  - **Security Level** + (zero or more) **Security Categories**

- SECLABELs are defined in the **SECLABEL** class

The Road to Multilevel Security…

1990 / RACF 1.9: security label or SECLABEL:

- In a fully-operational multilevel security environment, all users and data objects must have SECLABELs

- SECLABELs can be assigned to users (including started task and batch users), data resources, and to other security-related objects (such as terminals) using RACF commands

Defining a SECLABEL

1. Create SECDATA profiles

- RDEFINE SECDATA SECLEVEL UACC(NONE)
- RALTER SECDATA SECLEVEL ADDMEM(*seclevel-name/seclevel-number*)

- RDEFINE SECDATA CATEGORY UACC(NONE)
- RALTER SECDATA CATEGORY ADDMEM(*category-1 category-2* ...)

NOTE: It is not necessary to activate the SECDATA class

Defining SECLABELs *(continued)*

2. **Define SECLABEL profiles using the data defined in the SECDATA profiles**
   - RDEFINE SECLABEL *security-label* SECLEVEL(*seclevel-name*) ADDCATEGORY(*category-1 category-2* ...)

3. **Setup USER to have authority to SECLABEL(s)**
   - PERMIT *security-label* CLASS(SECLABEL) ACCESS(READ) ID(*user-id-1 user-id-2* ...)

   - ALTUSER *user-id-1* SECLABEL(*secuirty-label*)

     NOTE: This will define the user's default SECLABEL

Defining SECLABELs *(continued)*

4. **Define/Alter resource profiles to have a SECLABEL**
   - ALTDSD ‘*dataset-profile*’ SECLABEL(*secuirty-label*)
   - RALTER *class resource-profile* SECLABEL(*secuirty-label*)

5. **Activate and RACLIST the SECLABEL class**
   - SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)

     or
   - SETROPTS RACLIST(SECLABEL) REFRESH

Dominance and Equivalence

Dominance

- For SECLABEL **A** to dominate SECLABEL **B**
  - The Security Level of **A** is equal to or greater then the Security Level of **B**
  - **A** has at least all the Categories that define **B**

- Sometimes we will say that SECLABEL **A** is greater then SECLABEL **B**
- This is OK, except that one must keep in mind the possibility of disjoint relationships between SECLABELs
  - Where **A** will have Categories **C1** and **C2** while **B** will have Categories **C2** and **C3**

Dominance and Equivalence *(Continued)*

Equivalence

- For SECLABEL **A** to be equivalent to SECLABEL **B**
  - The Security Level of **A** is equal to the Security Level of **B**
  - Both **A** and **B** have the same set of Categories

One may also think of equivalence as follows:

IF SECLABEL A is equivalent to SECLABEL B, **THEN**

– SECLABEL A dominates SECLABEL B

AND

– SECLABEL B dominates SECLABEL A

DAC vs. MAC

- **DAC** = Discretionary Access Checking
  - Standard access lists manages this type of access
  - User decides access to data

- **MAC** = Mandatory Access Checking
  - SECLABELs manages this type of access
  - Object sensitivity decides access to data

- MAC will occur first, then DAC
  - Or DAC only, if the SECLABEL class is not active

MAC

MAC authorization in a fully operational Multilevel Security environment:

- To pass a **R/O Test** one needs:
  - The target/user to **DOMINATE** the object's SECLABEL
- To pass a **R/W Test** one needs:
  - The target/user to be **EQUIVALENT** to the object's SECLABEL
- To pass a **W/O Test** one needs:
  - The object to **DOMINATE** the target/user's SECLABEL

NOTE: SETROPTS options allow these rules to be slightly different allowing for a more robust security environment

MAC *(Continued)*

- For some classes it is necessary for the opposite to be true, that is for a **R/O** Test, one will want to **OBJECT** to dominate the **USER** or **TARGET SECLABEL**
- As a result certain classes defined in the Class Descriptor Table (CDT) have the option Reverse MAC (RVRSMAC) on

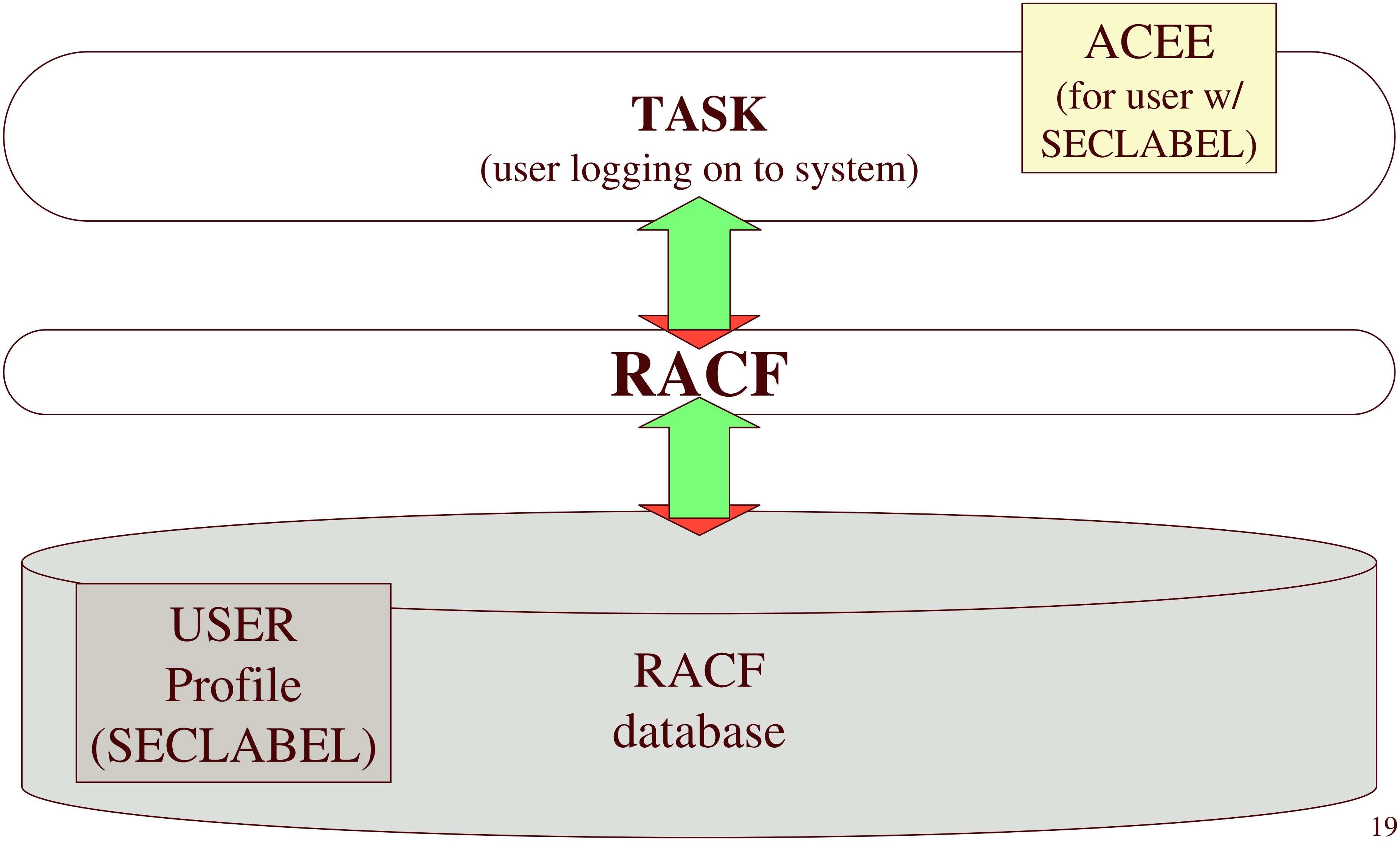
- With this set on in the CDT:
  - To pass a **R/O Test** one needs:
    - The object to **DOMINATE** the target's SECLABEL
  - To pass a **R/W Test** one needs:
    - The object to be **EQUIVALENT** to the target's SECLABEL

MAC *(Continued)*

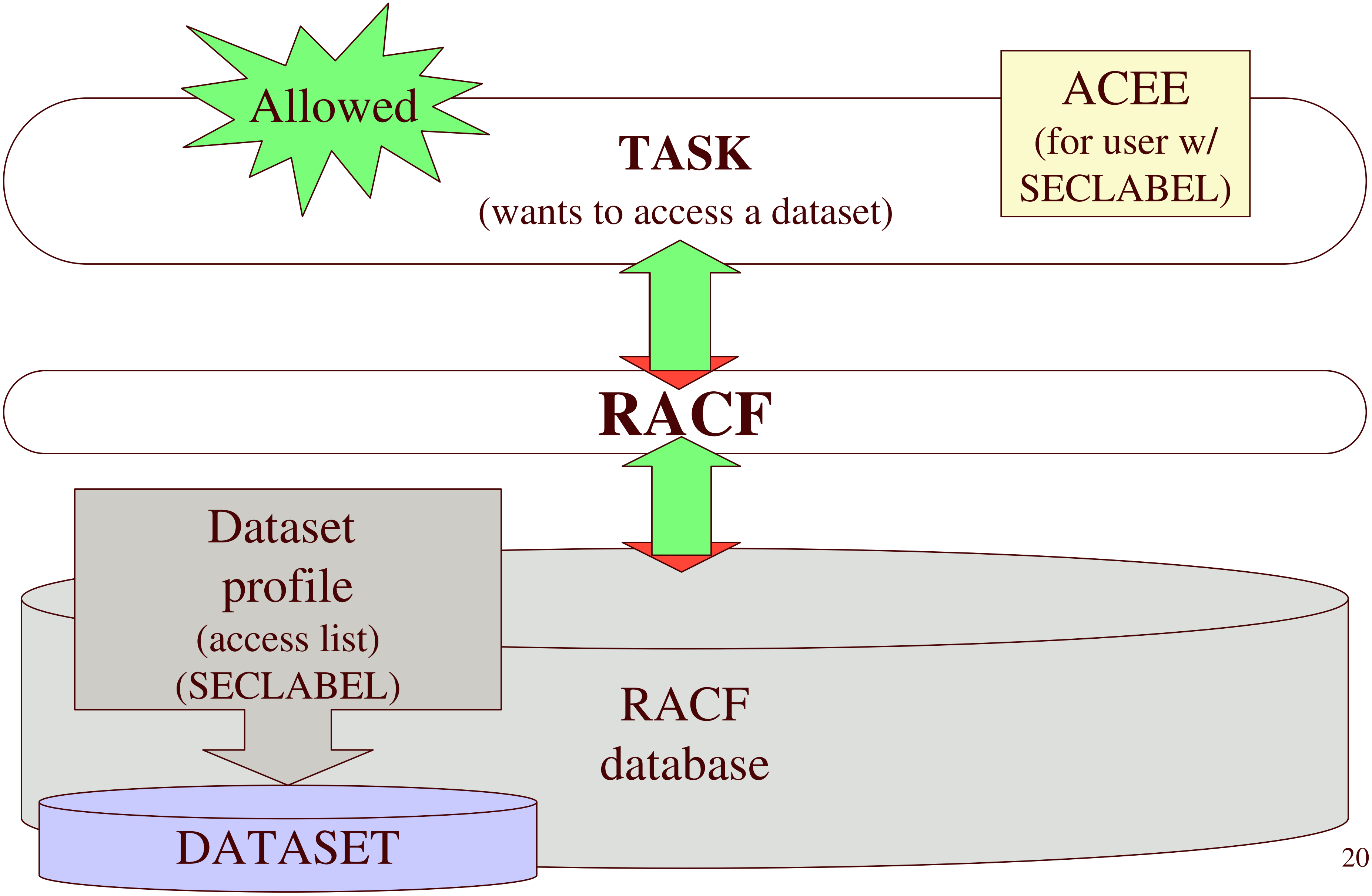
- For some classes it is necessary for **OBJECT** to always be equivalent to the **USER** or **TARGET SECLABEL**
- As a result certain classes defined in the Class Descriptor Table (CDT) have the option Equal MAC (EQUALMAC) on

- With this set on in the CDT:
  - To pass any **MAC Test** one needs:
    - The object to be **EQUIVALENT** to the target's SECLABEL

MAC Scenario (user logon)



MAC Scenario (access attempt)



SECLABEL-related options

- Activating SECLABEL processing
- SETROPS MLACTIVE
- SETROPS MLS
- SETROPS MLSTABLE
- SETROPS MLQUIET
- SETROPS SECLABELCONTROL
- SETROPS COMPATMODE
- And more ….

- **NOTE:** These options are system wide. So turning on or off any of these options <u>will</u> effect the entire system

Activating SECLABEL processing

- By activating the SECLABEL class and RACLISTing it, one activates SECLABEL processing
  - **SETR CLASSACT(SECLABEL) RACLIST(SECLABEL)**

- This alters the access check path:
  - If the both the user and the object have a SECLABEL then the user's SECLABEL is compared to the object's (MAC & DAC Test)
  - If the object has a SECLABEL and the user does not, then the access check fails
  - If the user has a SECLABEL and but the object does not, then the access check continues with the DAC check *

* The request will fail should MACLTIVE be on, and the class of the object has the SECLABEL required bit on.

SETROPTS MLACTIVE

- RACF will require that all resources for classes with SECLABEL=REQUIRED in the CDT have SECLABELs

- This option is activated by issuing the command:
  - **SETR MLACTIVE**

- There is a WARNING and FAILURE modes for this option

SETROPTS MLS

- **With SETR MLS in effect, RACF enforces the write-down property**
  - Subjects are prevented from writing down to a “lower” SECLABEL
- Prevents improper declassification of data
  - To pass a **R/O Test** one usually needs:
    - The target/user to **DOMINATE** the object’s SECLABEL
  - To pass a **R/W Test** one usually needs:
    - The target/user to be **EQUIVALENT** to the object’s SECLABEL
  - To pass a **W/O Test** one usually needs:
    - The object to **DOMINATE** the target/user’s SECLABEL

SETROPTS MLS *(Continued)*

- This option is activated by issuing the command:
  - **SETR MLS**

- There is a WARNING and FAILURE modes for this option

- When MLS is off
  - To pass a **R/O** or **R/W Test** one usually needs:
    - The target/user to **DOMINATE** the object’s SECLABEL
  - To pass a **W/O Test** one usually needs:
    - The target/user to **DOMINATE** the object’s SECLABEL

      OR

    - The object to **DOMINATE** the target/user’s SECLABEL

MLS & MLACTIVE WARNING mode

- If either MLS and/or MLACTIVE are in warning mode, RACF will pass a MAC test and generate a ICH408I warning message if and only if
  - The request would have passed if the option was off

    and
  - The request will fail with the option on

- This can be done by placing WARING after the SETROPTS MLS or MLACTIVE:
  - **SETR MLS(WARNING) MLACTIVE(WARNING)**
- This may be something useful when one first turns on these options to make sure all the correct profiles have been created with the correct SECLABELs

SETROPTS MLSTABLE

- Ensures that SECLABELs won’t change while someone is in the process of using them by:
  - Preventing changes of SECLABELs definitions
  - Preventing changes of SECLABELs assigned to a RACF profile
- Must set MLQUITE to allow such changes to occur while MLSTABLE is active
- This option is activated by issuing the command:
  - **SETR MLSTABLE**

SETROPTS MLQUIET

- Allows changing of SECLABEL definitions and SECLABELs within a RACF profile

- Overrides (and only needed if) MLSTABLE is active

- Only SPECIAL, TRUSTED, or console operator can logon or access resources protected by RACF profiles

- This option is activated by issuing the command:
  - **SETR MLQUIET**

SETROPTS SECLABELCONTROL

- Prevents non-SPECIAL users from setting or changing a resource SECLABEL

- Without SECLABELCONTROL, a user who can create or modify a RACF profile, can also modify the SECLABEL assigned to the profile

- This option is activated by issuing the command:
  - **SETR SECLABELCONTROL**

SETROPTS COMPATMODE

- A migration mode that allows certain users running WITHOUT a SECLABEL to access resources protected by RACF profiles that have a SECLABEL
  - RACF will check all SECLABELs the user has authority to, to verify their access to the resource

- Applies **ONLY** to applications that issue RACROUTE REQUEST=VERIFY to create the user ACEE without specifying any RACF 1.9.0 or later keywords

- This option is activated by issuing the command:
  - **SETR COMPATMODE**

Special SECLABELs

- **SYSHIGH**
- **SYSLOW**
- **SYSNONE**
- **SYSMULTI**

SYSHIGH

- **Combines the highest Security Level with all categories**

- SYSHIGH should be restricted to:
  - special system-level address spaces such as consoles
  - system programmers
  - system operators
  - system administrators

- SYSHIGH will dominate all SECLABELs

SYSLOW

- **Combines the lowest Security Level and has no categories**

- Used for data without a security classification,
  - Such as data IBM supplies as part which most users only need to read from
- Can also be used by customers for any data they create that has no need for classification

- SYSLOW will be dominated by all SECLABELs

SYSNONE

- **Combines the lowest Security level and has no categories, with an extra feature**

- Like SYSLOW, but allows write-down of data, when SETR MLS is in effect
- Should **ONLY** be used for data the user does not write to directly
  - Data whose access (for writing) is mediated by another program that will ensure no classified content is written (for example: System Catalogs)

- Should not be assigned to real users

SYSMULTI

- **This SECLABEL will always test to be EQUIVALENT to any other SECLABEL**

- It is intended for use by:
  - Daemons or Servers to be able to perform work for users running with different SECLABELs

- Should not be assigned to real users

Got YA!!!!

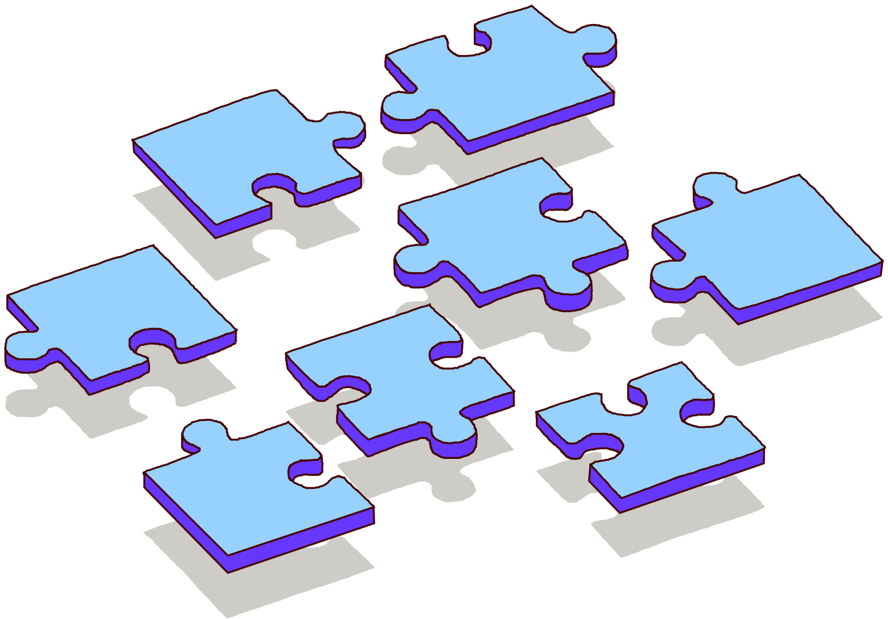
- Do not attempt to enable a multilevel security environment unless you have an accepted and well-defined data classification policy

- It’s very important to keep in mind that all this MAC, DAC, etc. security checking can be skipped if you place the object in the global access table

- If MLS **and** MLACTIVE are **BOTH** in FAIL mode, then any user that has the SPECIAL attribute **AND** is logged on with **SYSHIGH** is treated as though they are in **WARNING** mode
  - Useful to know if you get into any trouble

Q & A

- Any final questions?



Reference

- **RACF Security Administrator’s Guide**
  - http://publibz.boulder.ibm.com/epubs/pdf/ichza750.pdf
  - Chapter 4 – Classifying User and Data
  - Appendix F – In the section called:
    - “Security Label Authorization Checking”

- **Planning for Multilevel Security and the Common Criteria**
  - http://publibz.boulder.ibm.com/epubs/pdf/e0z2e111.pdf

- **MVS/ESA Planning: B1 Security**
  - http://publibfp.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/IEA5F600/CCONTENTS