IBM Software Group

Converting IMS/SMU Security to RACF with IMS Version 9

DB2 Information Management Software

Alan Cooper
EMEA Technical Sales
alan_cooper@uk.ibm.com

Overview

- ***IMS Version 9 will be the last release of IMS to support SMU***
- **Version 9 introduces new RACF* facilities**
  - **All SMU usage can now be replaced with RACF security**

This presentation:

- **Considers the SMU facilities that previously had no directly corresponding RACF facilities**
- **Explains the corresponding RACF options in Version 9**

* In this presentation, “RACF” should be interpreted as “RACF or equivalent product”

RACF Security Before IMS Version 9

Most IMS security can already be implemented with RACF

- **Sign-On user validation and verification**
  - Check user is known
  - Check password is correct
- **Terminal Security**
  - User v. physical terminal
- **IMS System Access Security**
  - User v. IMS ID
- **Transaction Security**
  - User v. Trancode
- **Command Security**
  - User v. IMS Command in Control Region
  - User v. IMS Command in Operations Manager
- **AOI Type2 ICMD Call Security**
  - User v. IMS Command
- **IMS Data Set Access Security**
  - Controls access to DBs and system datasets
- **DB Data Access Security – used with DL/1 AUTH call**
  - User v. DB Record
  - User v. Segment
  - User v. Field
- **PSB Access Security - For ODBA and CPI-C**
  - User v. PSBname
- **Connection Access Control**
  - IMS Connect, CQS, CSL address spaces, etc

Security Enhancements in IMS V9

- **Version 9 introduces enhancements to the RACF interface to support:**

1. **Application Group Name (AGN) security**
2. **Type 1 and Type 2 Automated Operator Interface (AOI)**
3. **Terminal security for Time-Controlled Operations (TCO)**
4. **MSC link receive security**
5. **/LOCK, /UNLOCK and /SET commands**
6. **Signon verification**

- **Benefits**
  - Overcomes limitations that previously prevented migration from SMU

Since this is the last release that will support SMU security, the release is considered a migration release where customers can migrate off of SMU security and to RACF (or equivalent product). With the additional enhancements provided in IMS V9 over those provided in previous releases, the use of SMU security should no longer be required.

Resource Access Security
(Replaces AGN Security)

Resource Access Security with SMU

- **Uses Application Group Name (AGN) security**
  - IMS Version 9 is the last release to support AGN security
- **Objectives of AGN Security**
  - Check at Program Scheduling Time that the resources involved (PSB &/or TRANcode &/or LTERM) are authorised to be used by the Dependent Region
- **Predominantly used for BMPs, but actually applies for all dependent regions and connecting threads (DRA/CCTL/ODBA)**

AGN Security Requirements

- **THREE Required Elements**

1. AGN defined in SMU
   - A named group of
   - PSBs &/or Transaction Codes &/or LTERMnames
2. RACF (optional – can alternatively use DFSISIS0 Exit)
   - Define AGN in AIMS resource class
   - Permit userids to use AGN
3. Dependent Region JCL must contain AGN=xxx execution parameter
   - Would also contain USERID

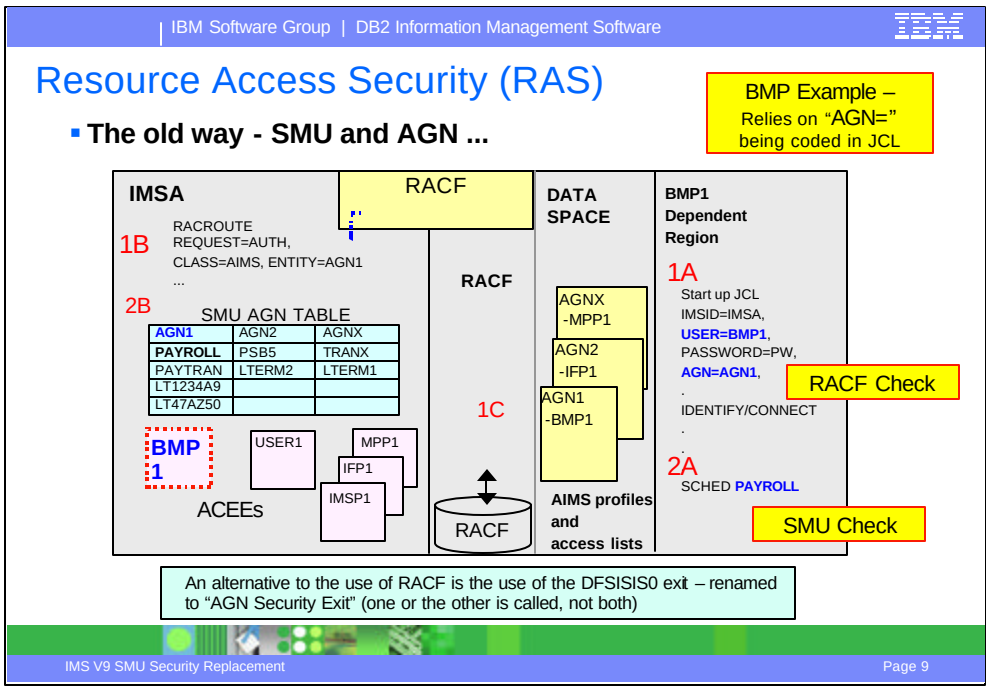
AGN Security Checks

- **At Dependent Region Startup**
  - AGN name (**if** specified in JCL) is authorised for use by **Region's USERID**
    - RACF or DFSISIS0 (Resource Access Security Exit)

Mostly, in practice, AGN security is only used with BMPs

- **At Program Scheduling Time**
  - Check (performed by SMU) that required IMS resource(s) are in the AGN group for this region
    - MPP / JMP : check TRAN in AGN*
    - Message Driven BMP : check TRAN and PSB in AGN*
    - NMD-BMP / IFP / JBP : check PSB in AGN
    - NMD-BMP with OUT= : additionally check output LTERM / TRAN in AGN

* If LTERMs are included in AGN for a message driven region, then use of the region will be limited to LTERMs in the group



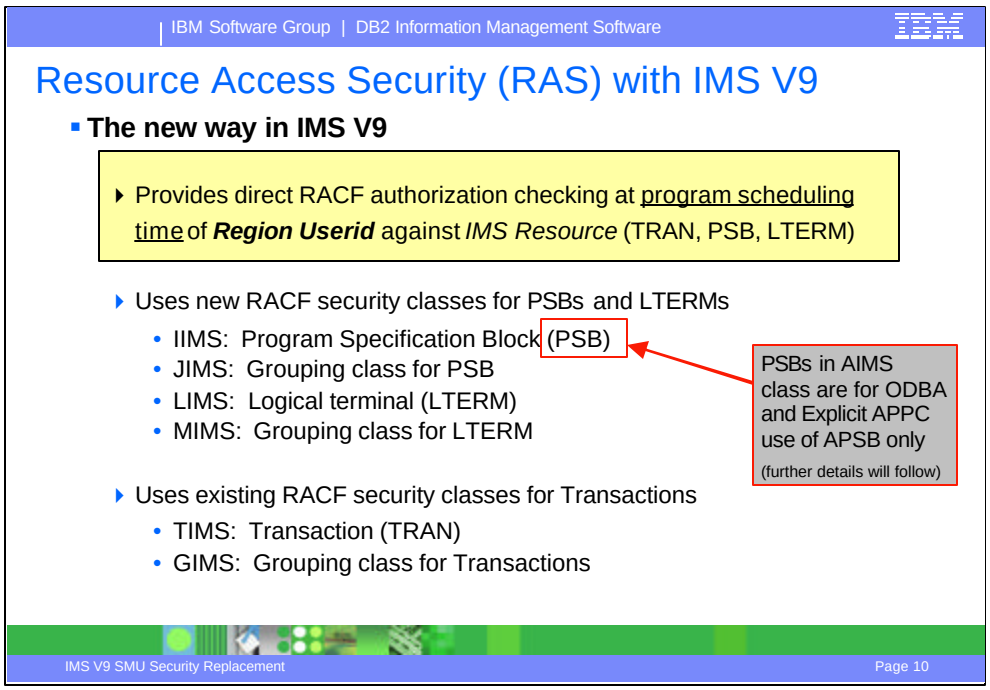
This visual describes the use of SMU and AGN as a review to better understand the changes in IMS V9. Note that systems migrating to IMS V9 can continue to use this process until a migration to RACF or equivalent product can be completed.

In a SMU and AGN environment, the checking process is a two-step process. The first check involves RACF. The second does not.

1. Each dependent region has a userid associated with it. Before a dependent region can connect to IMS, IMS performs a RACF authorization check to see whether the dependent region userid can use the application group name specified in it's AGN= startup parameter. IMS performs this check using the RACF class name (AIMS) and the name of the AGN passed to it in the EXEC statement parameter list. If the region's userid is not PERMITted to use the application group name, RACF returns a "not authorized" return code, and IMS does not allow the dependent region to connect. If RACF returns an "authorized" return code, the connection is made.

2. The second part of the two-step process is an IMS function only. IMS checks the name of the transaction or PSB or logical terminal that is being requested by the dependent region against the entries in the AGN Table and allows or disallows use depending on whether the name is in the entry for the application group name.

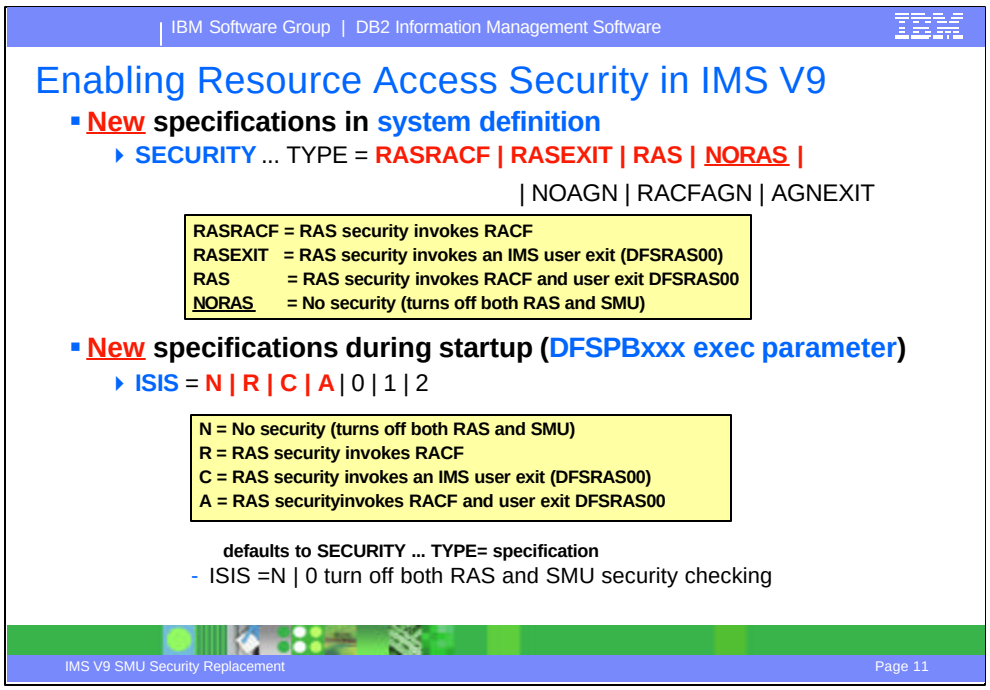
For message processing regions and fast path regions, application resource security is somewhat like class scheduling in that transactions can be scheduled only in regions whose application group name allows them.



In IMS V9, the two-step process described on the previous visual is now a direct check of the userid authorization against the specific resource. To accomplish this and ensure that the new RAS capability in IMS V9 provides equivalent protection, four new security classes for PSBs and LTERMs have been added.

The four new RACF security classes will be added to the RACF product as default classes. If your environment does not have the level of RACF that includes the new classes as defaults then they must be added to the installation-defined class descriptor table. Macro ICHERCDE is used to define a new class. The procedure for adding classes to this table is described in "Adding Installation-Defined Classes" in the z/OS Security Server RACF System Programmer's Guide. This is the same procedure that is used to change the default class names, e.g., TIMS to Txxx where xxx is more meaningful to a specific environment.

The existing TIMS and GIMS classes continue to be used to protect transactions and groupings of transactions. APSB security using the existing AIMS security class will also continue to be done as in prior releases. The AIMS class will only be used for APSB security when SMU with AGN security is not selected for use.



The new support in IMS V9 is called Resource Access Security. It is no longer called AGN. RACF, a new user exit, or both are used to provide protection. Either AGN security, or the new RAS security can be used, but not both. If both are specified, then RAS is used. Resource Access Security is requested by using the system definition SECURITY macro with TYPE=RASRACF|RASEXIT|RAS|NORAS or by specifying the execution parameter ISIS=N|R|C|A. In both cases, the new values are provided in addition to the values that are already supported.

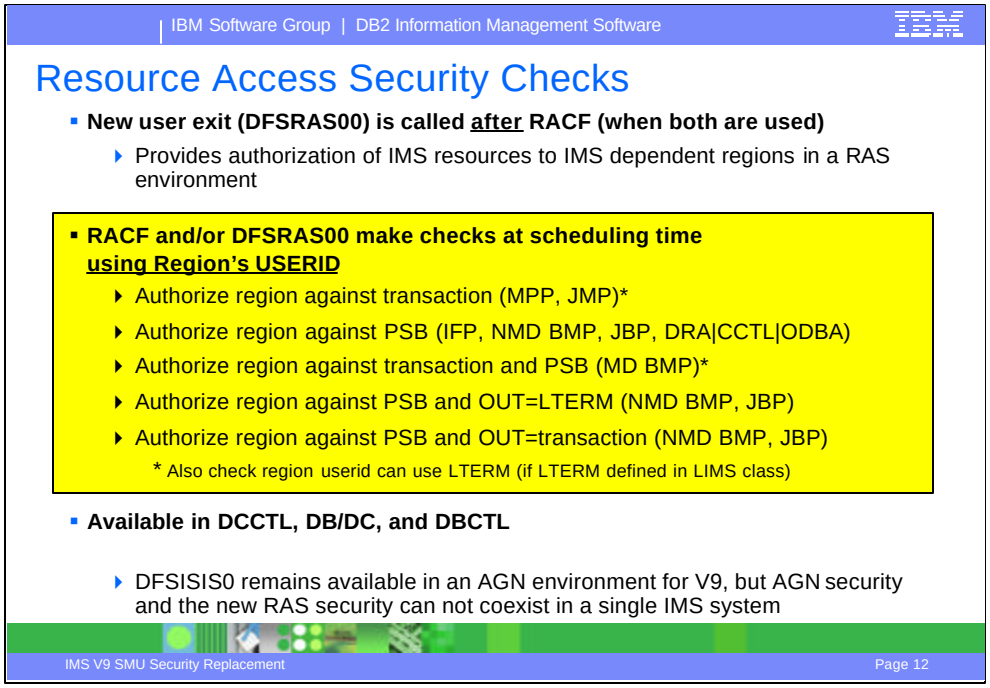
ISIS continues to be used for AGN and RAS security specifications. The parameters for AGN and RAS are unique, and can therefore correctly define the type of security desired. If ISIS=N is specified, there will be no authorization checking for the use of transactions, PSBs, and LTERMs by dependent regions. SMU will not be used either, because ISIS=1 or 2 must be specified (if the SECURITY macro has no specification) for SMU to be used for AGN. If ISIS is not specified, the default is the specification on the TYPE parameter of the IMS system definition SECURITY macro.

The new values for the ISIS parameter are documented in the visual. The values that already existed include:

ISIS=0 deactivates AGN security .

ISIS=1 activates AGN security using SMU and RACF.

ISIS=2 activates AGN security using SMU and the Resource Access Security Exit Routine (DFSISIS0).



A new user exit interface called the Resource Access Security exit routine (DFSRAS00) provides authorization of IMS resources to IMS dependent regions. Although this exit replaces the use of the DFSISIS0 exit routine in a RAS environment, it is not quite a direct replacement.

DFSISIS0 was called to authorize a user to an AGN name. The AGN table with that AGN name contained the IMS resources that were indirectly authorized for use.

The new IMS V9 exit DFSRAS00 is called after calling the SAF interface (RACF or equivalent product) to directly authorize the user to the actual IMS resources (transaction, PSB, and/or output LTERM name). The SAF call and user exit call are made based upon the specification of the parameters on the system definition SECURITY macro and/or the startup parameter ISIS=. This new user exit is available for the DB/DC, DCCTL, and DBCTL environments. Neither this new exit nor DFSISIS0 supports Callable Services.

Resource Access Security and APSB Security

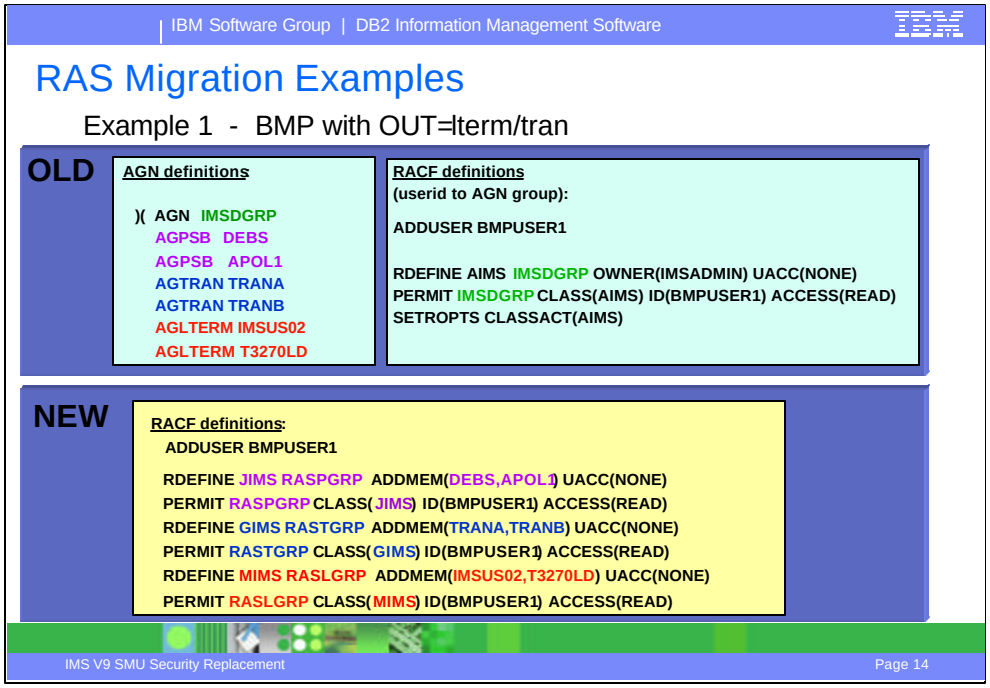
- **When RAS is enabled**
  - RAS check is made at every MPP/JMP program schedule using region's userid
  - RAS check is made at every BMP/IFP/JBP program schedule using region's userid
  - RAS check is made at every CICS/DBCTL program schedule using userid of CICS address space
    - Completely separately, CICS can perform check of terminal user against PSB
- **RAS checking takes place at a program schedule**
  - PSB defined in IIMS RACF class

APSB security checking takes place for an "APSB Call"

  - PSB defined in AIMS RACF class

IMS will never use both checks for the same schedule!

- **ODBA APSB call**
  - Exec parameter "ODBASE=Y" means use APSB security
  - With ODBASE=N, RAS (or AGN) security will apply (if enabled)
- **Explicit APPC (CPI-C) APSB call**
  - If APSB security is performed (with caller's userid), RAS check will not be made
  - If APSB security is not performed, RAS check (if enabled) will be performed using region's userid



The next three visuals provide examples of converting from the use of SMU to RACF security.

RAS Migration Examples ...

Example 2 - AGN name with access to all entities of a particular resource type

OLD

AGN definitions:

```
)( AGN  ALLGRP
    AGPSB  ALL
    AGTRAN  ALL
```

In RACF, generic resource definitions can be used

NEW

RACF definitions:

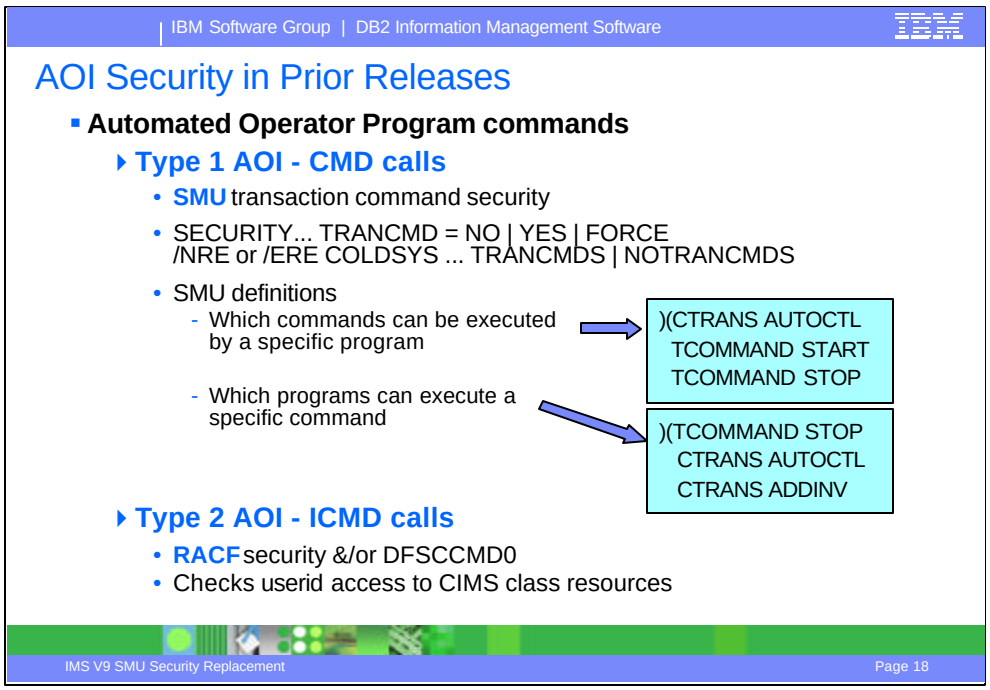
```
ADDUSER DRAINBMP

RDEFINE JIMS ** UACC(NONE)
PERMIT ** CLASS(JIMS) ID(DRAINBMP) ACCESS(READ)
RDEFINE TIMS ** UACC(NONE)
PERMIT ** CLASS(TIMS) ID(DRAINBMP) ACCESS(READ)
```

Migrating Off SMU

- **Define all AGN resources to RACF in the appropriate classes**
- **Define <u>all</u> region ids as RACF users**
  - BMPs, MPPs, IFPs, etc.
- **Permit region ids to access appropriate resources**
- **Change SECURITY macro to specify RAS**

  and/or
- **Change ISIS= parameter in DFSPBxxx to specify RAS**
- **If needed, add ODBASE=Y to DFSPBxxx**
- **Restart IMS**
- **When safe, remove SMU definitions**

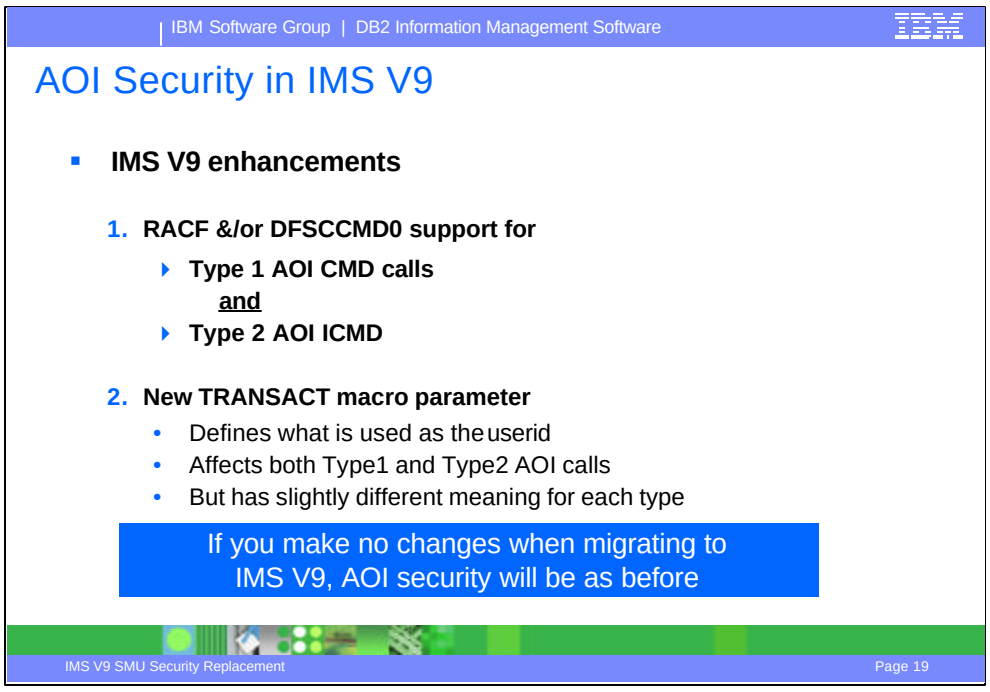
AOI Security



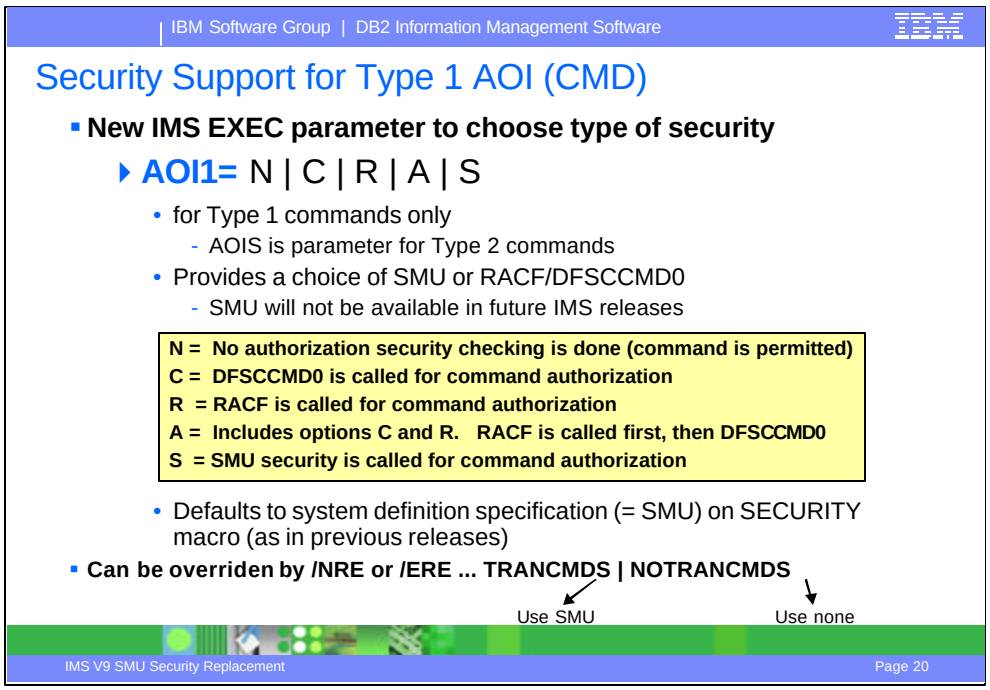
Prior to IMS V9, commands issued by Type 1 AO programs were secured using SMU transaction command security profiles. AO programs were restricted to issuing only the commands defined in the SMU profiles. When the Type 1 AOI CMD call was issued, IMS performed the security checking using the tables/matrixes loaded during restart.

On the other hand, commands issued by Type 2 AO programs were secured using command profiles stored in the RACF CIMS Class. Type 2 programs were also optionally secured using a user exit routine (DFSCCMD0) which could perform command authorization independently or in conjunction with RACF. When security checking was required for command keywords, the exit provided a way to implement more granular levels of command security.

IMS V9 provides several enhancements in this area. Type1 AOI CMD calls are now secured using the RACF interface in a fashion similar to that which was already provided for the Type 2 ICMD calls. Additionally, a new parameter on the TRANSACT macro provides a greater level of granularity that can make the RACF security check closer to that which was provided for SMU.



IMS V9 provides several enhancements in this area. Type1 AOI CMD calls are now secured using the RACF interface in a fashion similar to that which was already provided for the Type 2 ICMD calls. Additionally, a new parameter on the TRANSACT macro provides a greater level of granularity that can make the RACF security check closer to that which was provided for SMU.

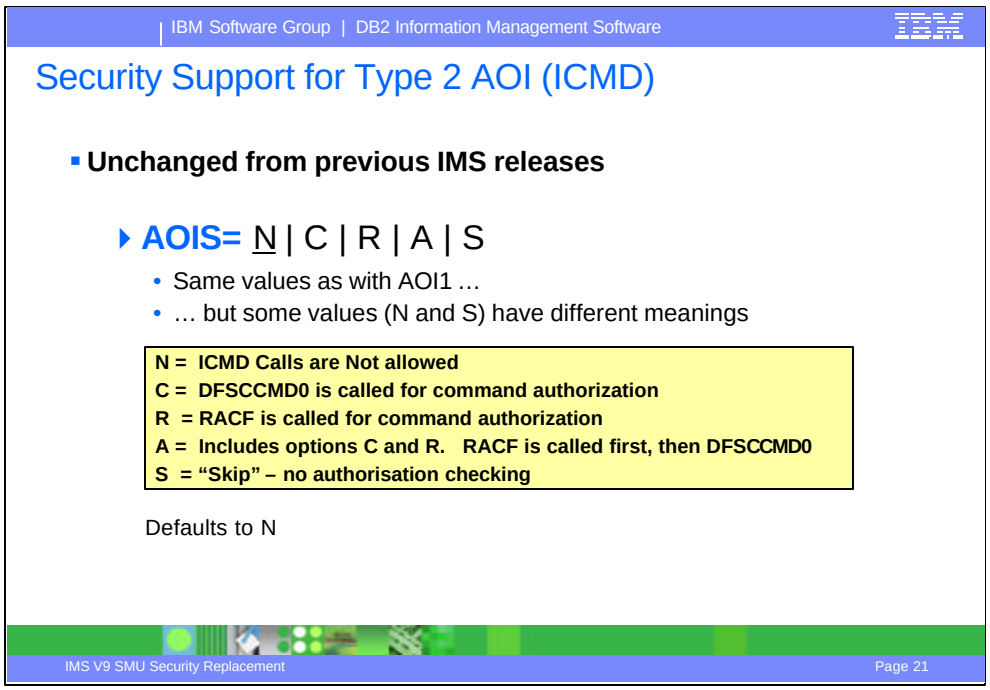


The ICMD call of type 2 AOI (DFSAOE00) already uses the SAF interface and the Command Authorization exit (DFSCCMD0) for security. The CMD call of type 1 AOI (DFSAOUE0), as mentioned on the previous visual, used SMU security in prior releases. In IMS V9, this has been enhanced to provide the option of using SAF (RACF or equivalent product) and the DFSCCMD0 user exit.

A new startup parameter in IMS V9, AOI1=A|N|C|R|S can be specified to indicate which security product is to be used for authorization of commands for type 1 AOI, and the level of security. If no value is specified, then IMS uses the specification in the SECURITY macro as defined in the system definition process.

Because the AOI1 specification is not included in a checkpoint record, the AOI1 value can be changed each time IMS is initialized.

During an IMS restart, the /NRE or /ERE can specify a value of either TRANCMDS or NOTRANCMDS. This only applies to TYPE 1 AOI security. If neither value is specified on a restart then IMS uses the AOI1 startup specification or what has been defined in the SECURITY macro. A specification of TRANCMDS causes SMU to be used for Type 1 AOI security and overrides TRANCMD=N in system definition and AOI1 in startup. NOTRANCMDS results in no TYPE 1 AOI security, unless prevented by the specificaiton of TRANCMD=F during system definiton and a further override by AOI1=R|C|A|N.

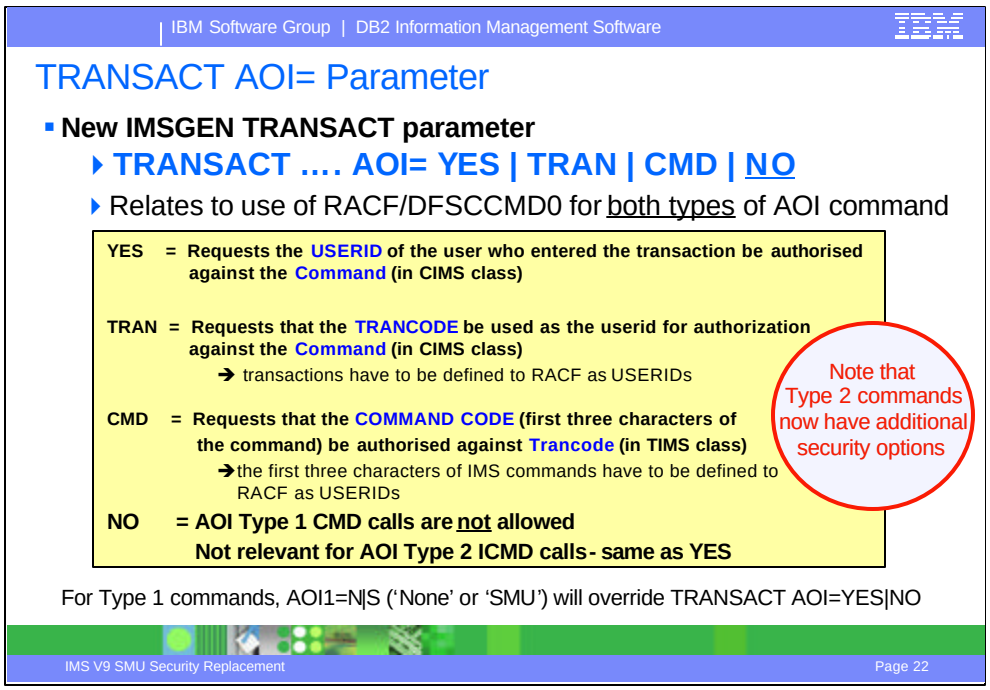


The ICMD call of type 2 AOI (DFSAOE00) already uses the SAF interface and the Command Authorization exit (DFSCCMD0) for security. The CMD call of type 1 AOI (DFSAOUE0), as mentioned on the previous visual, used SMU security in prior releases. In IMS V9, this has been enhanced to provide the option of using SAF (RACF or equivalent product) and the DFSCCMD0 user exit.

A new startup parameter in IMS V9, AOI1=A|N|C|R|S can be specified to indicate which security product is to be used for authorization of commands for type 1 AOI, and the level of security. If no value is specified, then IMS uses the specification in the SECURITY macro as defined in the system definition process.

Because the AOI1 specification is not included in a checkpoint record, the AOI1 value can be changed each time IMS is initialized.

During an IMS restart, the /NRE or /ERE can specify a value of either TRANCMDS or NOTRANCMDS. This only applies to TYPE 1 AOI security. If neither value is specified on a restart then IMS uses the AOI1 startup specification or what has been defined in the SECURITY macro. A specification of TRANCMDS causes SMU to be used for Type 1 AOI security and overrides TRANCMD=N in system definition and AOI1 in startup. NOTRANCMDS results in no TYPE 1 AOI security, unless prevented by the specificaiton of TRANCMD=F during system definiton and a further override by AOI1=R|C|A|N.



The TRANSACT macro has a new parameter, AOI=YES|NO|TRAN|CMD which specifies whether or not a particular transaction is allowed to issue the AOI command (CMD) call. In prior releases, this information was derived during IMS restart from the SMU matrix tables.

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.

When AOI=NO is specified, no authorization is permitted. Type 1 AOI CMD calls can not be issued.

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 being kept for use by future authorization checks. In this case, the Type 1 AOI transactions have to be defined to RACF (or equivalent product) as a user. The transactions must also be specified on RACF PERMIT statements for each command they are allowed to issue from a Type 1 AOI transaction.

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 Type 1 AOI transaction that is allowed to issue them.

Note that for Type 1 CMD commands, the AOI1= value of None or SMU will take precedence over a TRANSACT AOI value of Yes or No

RACF Replacement for Type 1 AOI (CMD) SMU Security

OLD

```
)(CTRANS AUTOCTL                    )(TCOMMAND STOP
  TCOMMAND START                      CTRANS AUTOTRAN
  TCOMMAND STOP                       CTRANS ADDINV
```

NEW

RACF definitions:

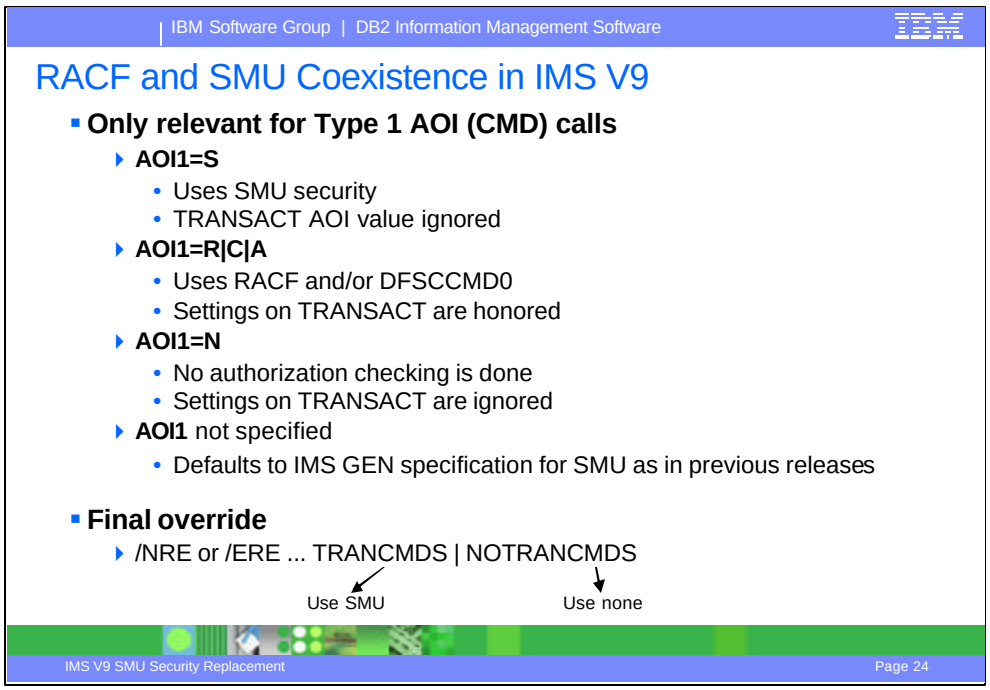
```
ADDGROUP AOCMDS
ADDUSER STO DFLTGRP(AOCMDS)
ADDUSER STA DFLTGRP(AOCMDS)

RDEFINE TIMS AUTOCTL  UACC(NONE)
 PERMIT AUTOCTL CLASS(TIMS) ID(AOCMDS) ACCESS(READ)
```

TRANSACT CODE=AUTOCTL
AOI=CMD

```
ADDUSER AUTOTRAN
ADDUSER ADDINV

RDEFINE CIMS STO  UACC(NONE)
 PERMIT STO CLASS(CIMS) ID(AUTOTRAN, ADDINV) ACCESS(READ)
```

TRANSACT CODE=AUTOTRAN
AOI=TRAN

Specify TRANSACT macro AOI= parameter in IMS definitions



The support of the new parameter AOI on the TRANSACT macro is dependent on the specification of the new startup parameter AOI1:

If AOI1=S, SMU is used and the AOI settings defined during system definition in the TRANSACT macro are ignored.

If AOI1=R|C|A, SMU for AOI is ignored, RACF and/or DFSCCMD0 are used for authorization and the settings defined during system definition in the TRANSACT macro are honored.

If AOI1=N, No authorization checking is done and the settings in the TRANSACT macro are ignored.

The final override for Type 1 AOI security, as mentioned earlier, is specified through the /NRE or /ERE specification of TRANCMDS or NOTRANCMDS. Although this was documented previously, the explanations is repeated here for clarification. This only applies to TYPE 1 AOI security. If neither TRANCMDS or NOTRANCMDS is specified on a restart , IMS uses the AOI1 startup specification or what was defined in the SECURITY macro. A specification of TRANCMDS causes SMU to be used for Type 1 AOI security and overrides TRANCMD=N in system definition and AOI1 in startup. NOTRANCMDS results in no TYPE 1 AOI security, unless prevented by the specificaiton of TRANCMD=F during system definiton and a further override by AOI1=R|C|A|N.

Migrating Off SMU

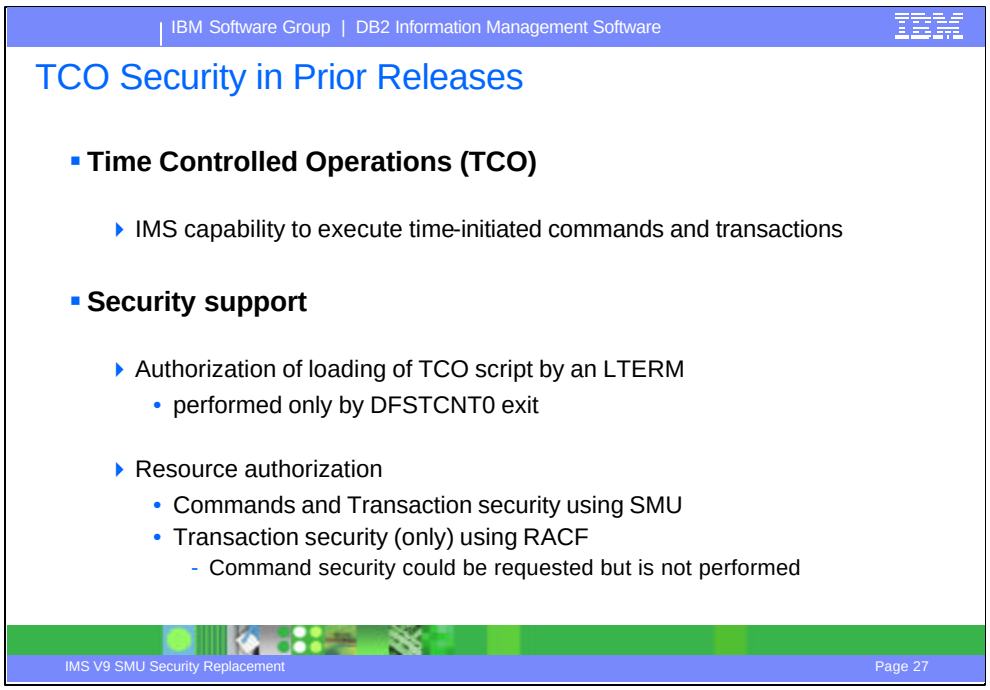
Type 2 (ICMD)

- **No action needed, but now have choice of what userid to use**

Type 1 (CMD)

- **Initially, code AOI1=S or use default (SECURITY macro) value to get SMU security**
- **Set up required RACF definitions for type 1 commands**
- **Add AOI=value to TRANSACT macros in IMSGEN**
  - Can use online change
  - Will be ignored for type 1 commands while AOI1= indicates SMU security
- **Change (or add) AOI1=R to DFSPBxxx**
- **Restart IMS**
- **When safe, remove SMU definitions**

Time Control Option (TCO) Security



Time Controlled Operations (TCO) is an IMS capability to execute time-initiated commands and transactions. TCO can generate any IMS input that an IMS operator can, except for the IMS restart commands, /NRESTART and /ERESTART. Additionally, it cannot initiate conversational transactions, full-function response mode or Fast Path input transactions.

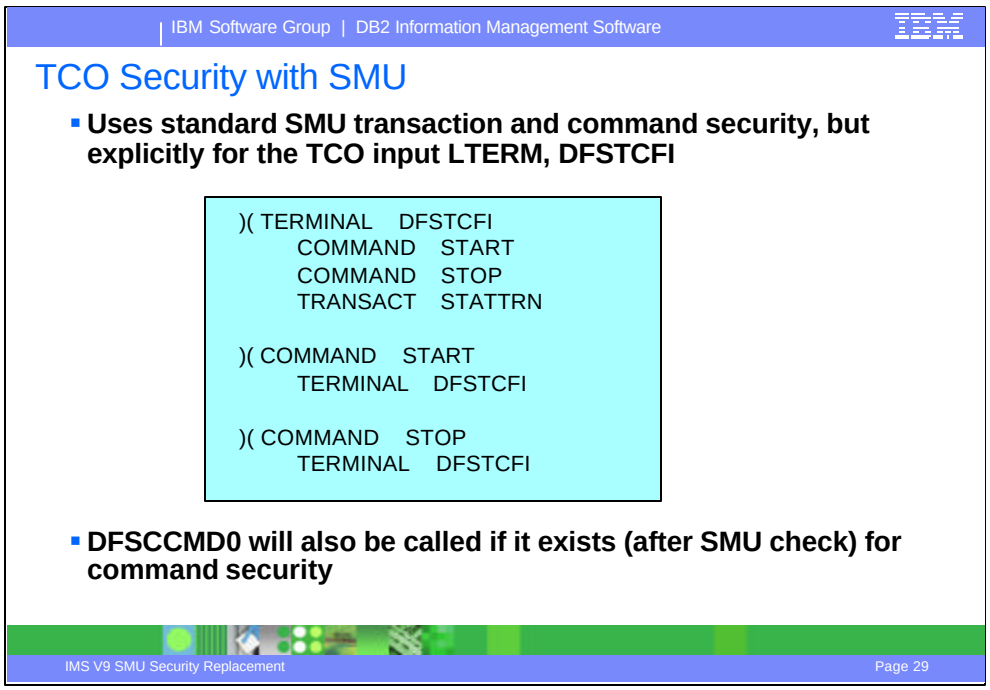
The scripts used by TCO to execute the time-initiated commands, transactions, and message switches are stored in IMS.TCFSLIB. This is the time controlled facility's script library.

A new script can be loaded by a program, a user exit or by a “Load command” from an LTERM (isues message switch, “DFSTCF LOAD script-name …”

TCO support provides security at two different levels. One capability restricts which LTERMs can load TCO scripts. The other restricts which IMS commands or transactions can be accessed.

TCO Security in IMS V9

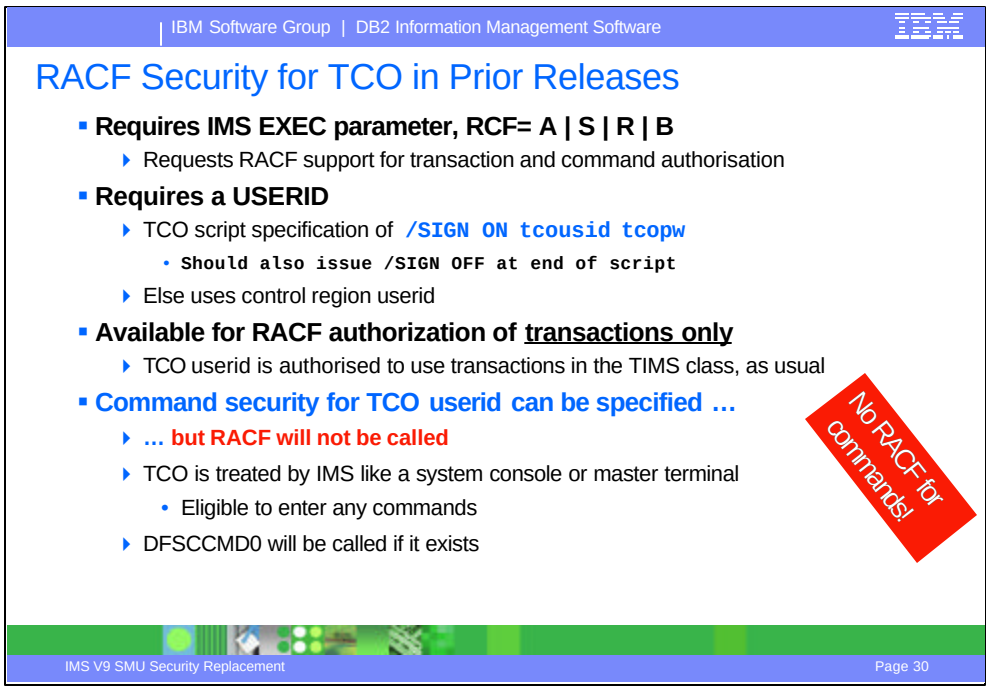
- **Loading of TCO scripts**
  - No change - performed only by DFSTCNT0 exit
- **Resource Security**
  - Command and Transaction security with SMU
    - Last release of IMS to provide this
  - Command and Transaction security with RACF



In prior release, IMS commands and transactions issued by TCO had the choice of using either SMU or RACF for authorizaiton.

The use of SMU security for TCO command and transaction authorization in prior releases was predicated on defining the TCO LTERM names DFSTCF and DFSTCFI in the SMU definitions. More specifically, the SMU definitions were used to define which resources could be issued by a TCO script running under this LTERM name.

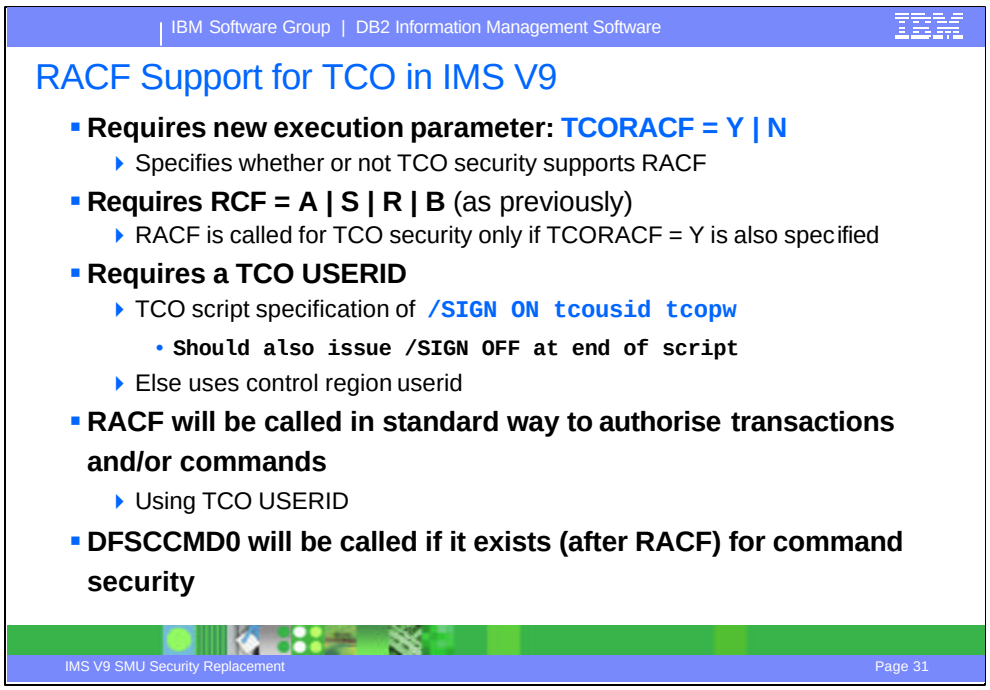
On the other hand, when the IMS startup parameter RCF was specified, SMU authorization was not used. If the script issued a /SIGN ON command, the associated userid was available to be used in RACF calls to authorize access to transactions.



In prior release, IMS commands and transactions issued by TCO had the choice of using either SMU or RACF for authorizaiton.

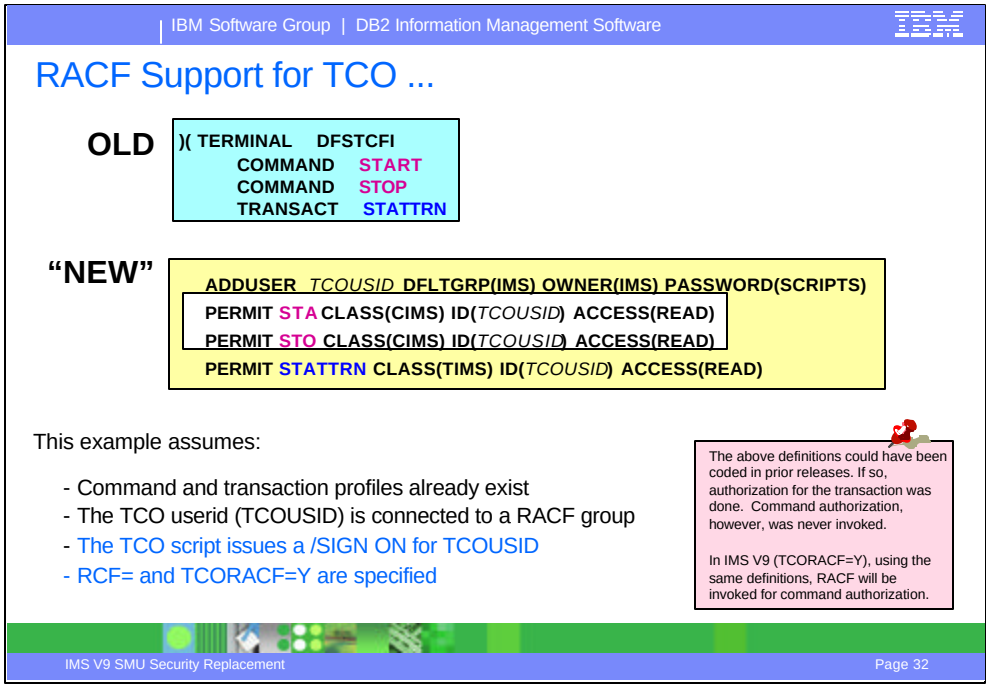
The use of SMU security for TCO command and transaction authorization in prior releases was predicated on defining the TCO LTERM names DFSTCF and DFSTCFI in the SMU definitions. More specifically, the SMU definitions were used to define which resources could be issued by a TCO script running under this LTERM name.

On the other hand, when the IMS startup parameter RCF was specified, SMU authorization was not used. If the script issued a /SIGN ON command, the associated userid was available to be used in RACF calls to authorize access to transactions.



In IMS V9, a new execution parameter, TCORACF=, is used to indicate whether (Y) or not (N) the RACF interface should be called to perform an authorization check of commands from a TCO script. The value can be changed each time IMS is initialized. Therefore, specification of RCF has additional criteria. Only if both the IMS startup parameter RCF=A|S|R|B and TCORACF=Y are specified, is the RACF interface invoked to call RACF or an equivalent product. The Command Authorization Exit Routine (DFSCCMD0), if it exists, will also be called. Coding a /SIGN ON command at the beginning of the TCO script can provide the userid which will be used for authorization. The userid defined in the /SIGN ON is signed on to IMS and not signed off until a /SIGN OFF at the end of the script. Any commands after the /SIGN ON are checked by RACF for being authorized for use by the signed on user. The commands are also passed to the DFSCCMD0 exit. DFSCCMD0 can check the input CNTNAME to see if the input is from TCO (DFSTCFI), and if it is, allow the /SIGN command without checking the userid. For other commands, the exit can authorize the signed on user to enter that particular command.

A note on providing a userid/group in the /SIGN ON at the beginning of a script: The userid/group provided must have been previously been defined to RACF. If it has not, RACF returns a RC=4. As a result of the RC=4, IMS issues a FASTAUTH using the IMS CTL region's ACEE. If the CTL region's userid/group is authorized to the command then the TCO script can also access the command. If the CTL region's userid is not authorized to the command then an RC=8 is issued to deny access.



The example assumes that RACF security profiles already exist for the START and STOP commands and for the STATTRN transaction.

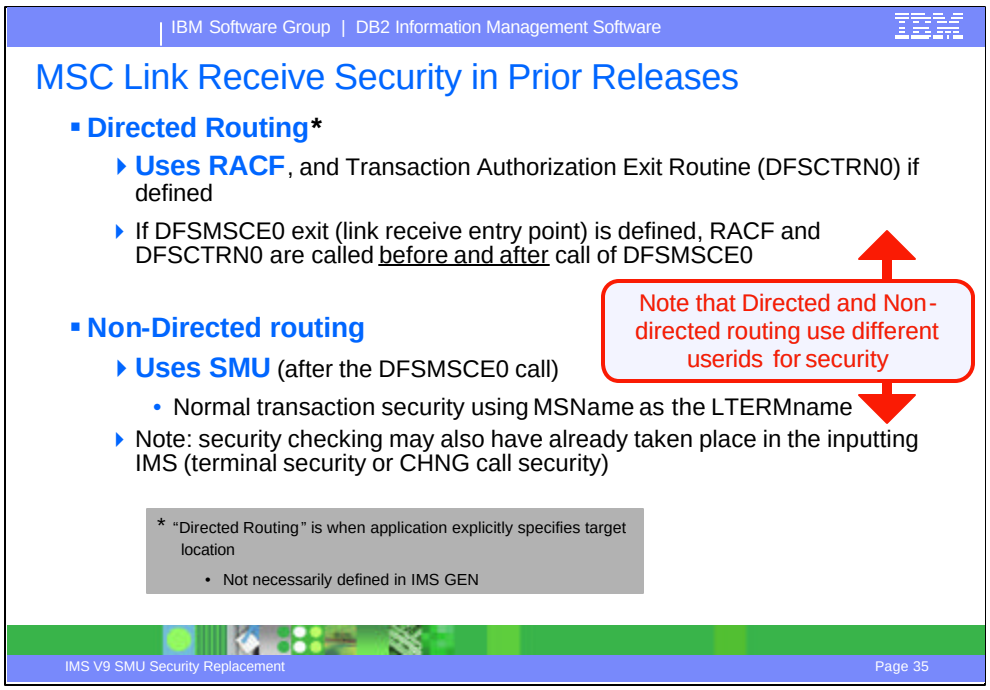
The TCO userid, TCOUSID, is added to the access lists for each of the commands and for the transaction.

Although this definitions could be provided in prior releases, it is not until IMS V9 that IMS will invoke the CIMS profile to authorize access for a TCO script.

Migrating Off SMU

- **Prerequisite is that RACF is used for command / transaction security**
  - RCF= A | S | R | B
- **Define TCO userid and permissions in RACF**
- **Add /SIGN ON (and /SIGN OFF) to all TCO scripts**
- **Add TCORACF=Y to DFSPBxxx**
- **Restart IMS**
- **When safe, remove SMU definitions**

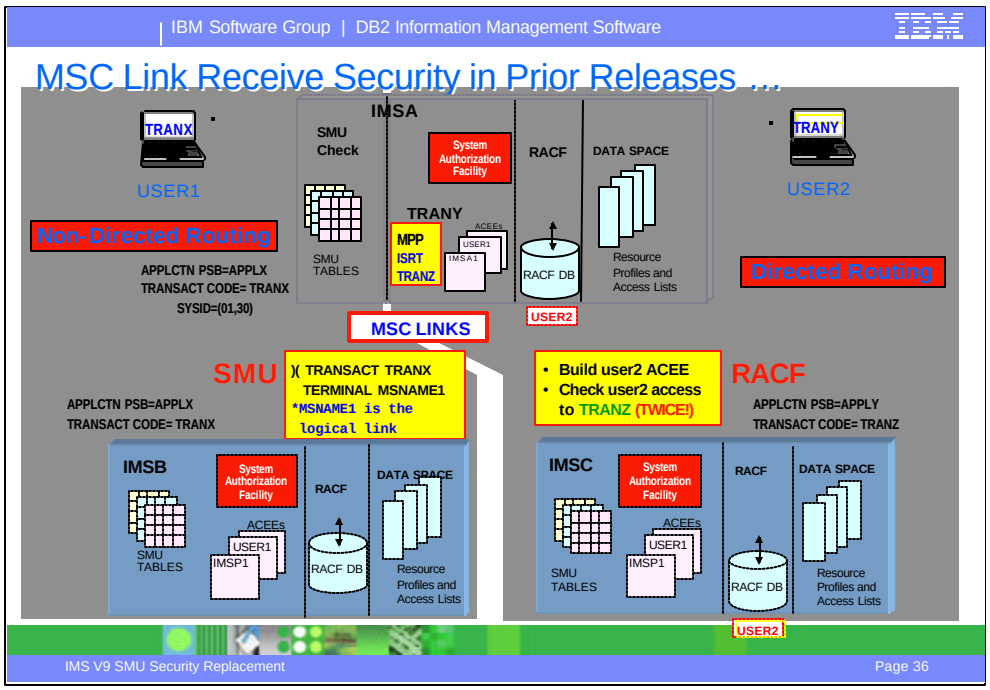
MSC Link Receive Security



MSC directed routing is a function of IMS that allows for the routing of messages to other IMS systems without the target resources having to be defined as remote resources in the sending IMS. This capability relies on the application, in the sending IMS, to specify the logical link path (MSNAME) as the destination of the message. Additionaly, the sending application provides the actual target LTERM or transaction name in the first few bytes of the IOAREA. When the message is received at the destination, the LTERM or trancode is removed from the message by the link receive entry point of the DFSMSCE0 exit and the message enqueued for the proper destination.

Non-directed routing, on the other hand, relied on all target destinations in the remote IMS to be defined as remote resources on the sending system.

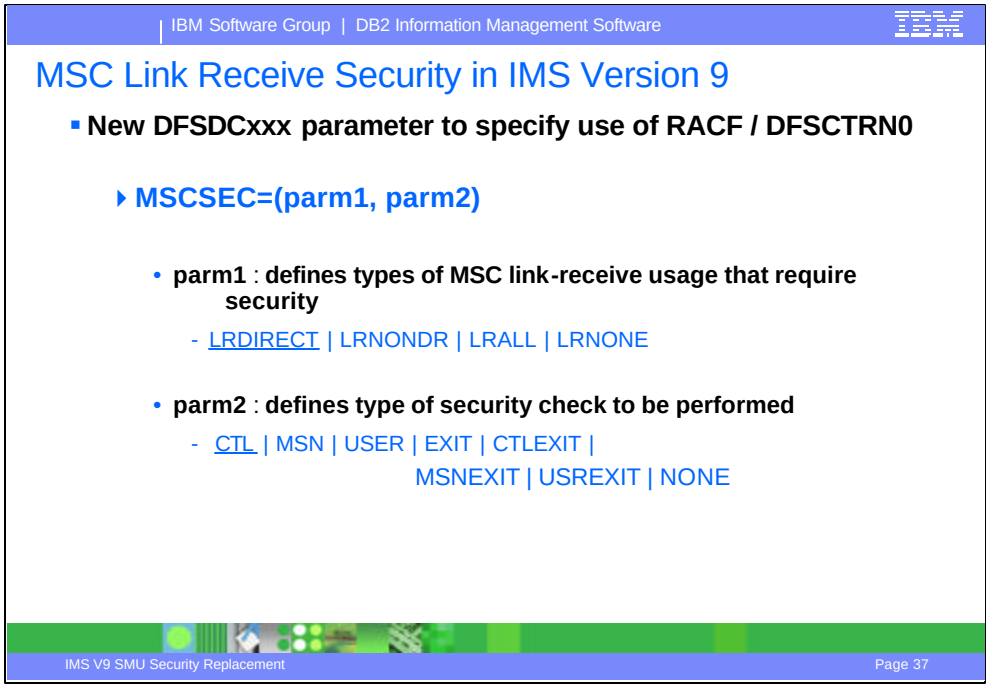
For releases prior to IMS V9 where MSC link input to transaction destinations used directed routing, SAF(RACF) and the DFSCTRN0 user exit were called before and after the DFSMSCE0 user exit call. Otherwise, for non-Directed Routing, SMU security was called after the DFSMSCE0 call.



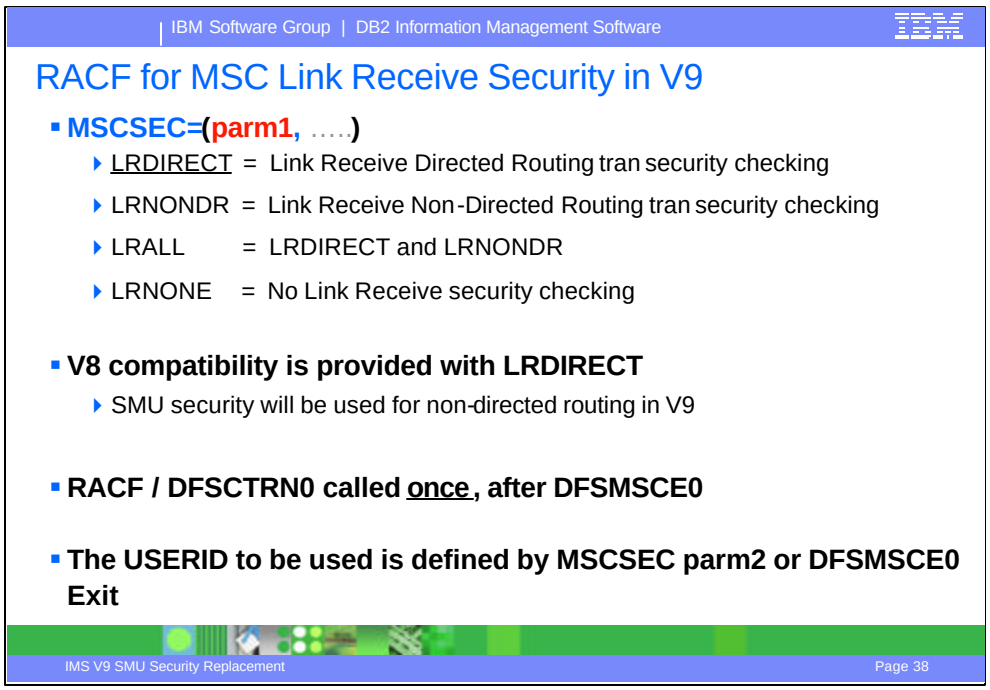
This visual shows MSC link receive security as it exists today. Note that this does not address the other points of security for MSC which are already adressed by RACF interfaces.

On the left side of the page, the example shows non-directed routing. In this environment, remote destinations are defined in the front-end IMS. When the message is received in the back-end system, SMU definitions are used to determine whether a transaction can be accessed using the MSNAME associated with the MSC link.

For directed routing, RACF calls along with an optional call to DFSCTRN0 are made before and after calls to DFSMSCE0. DFSMSCE0 is the exit routine that contains the entry point for the Link Receive code.

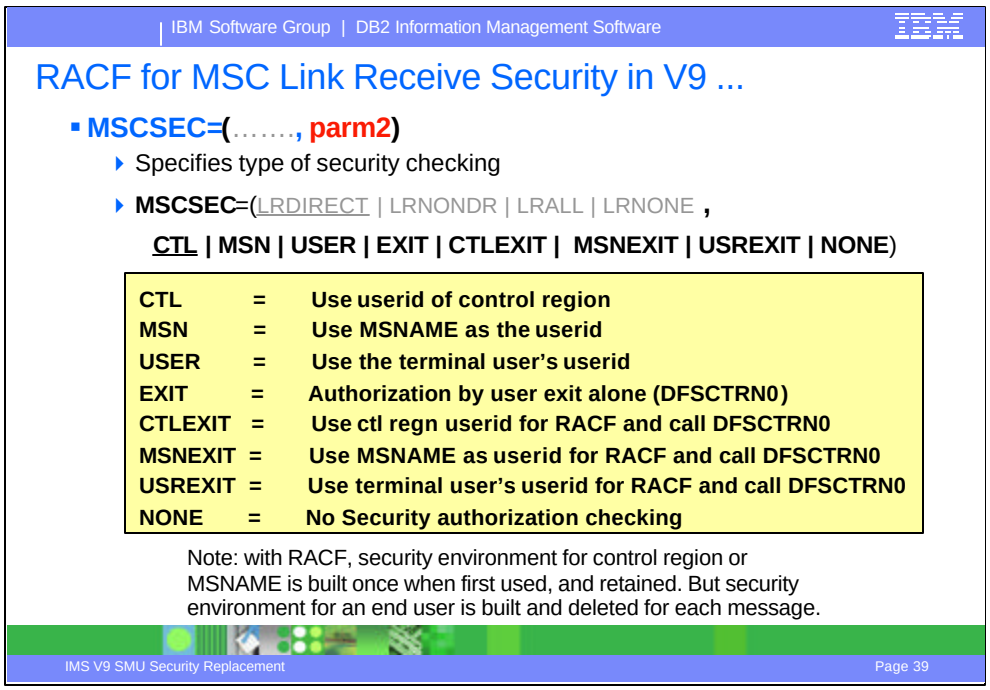


For IMS V9, a new startup parameter defines the type of security required for MSC. In the DFSDCxxx proclib member, the new parameter is MSCSEC=(parm1,parm2).

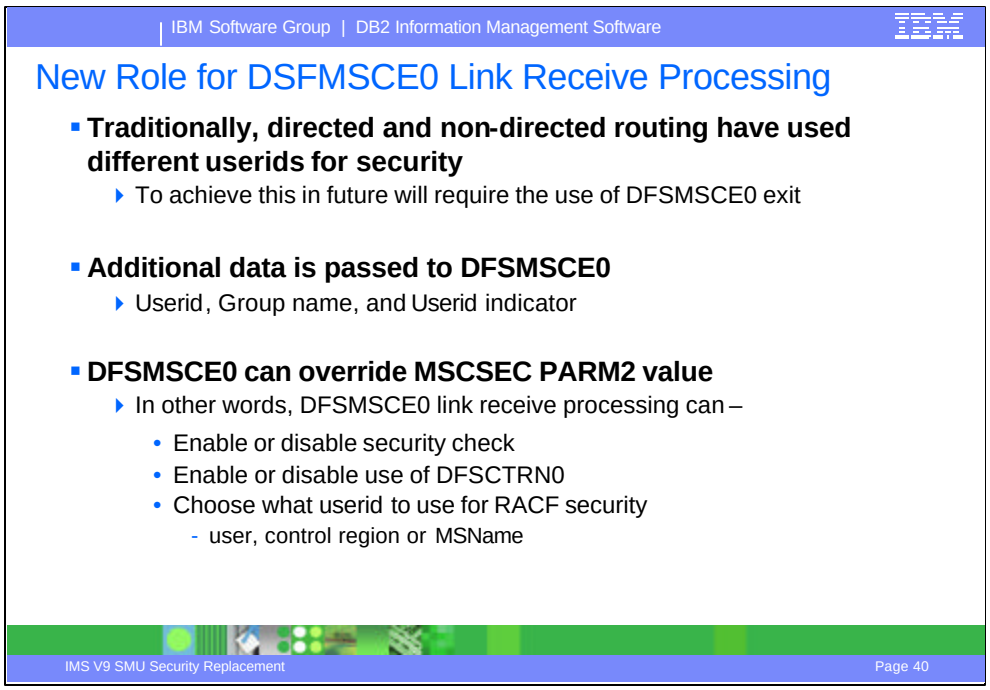


SMU security can still be invoked when non-directed routing is used and MSCSEC=LRNONDR is not specified.

There will be no longer be calls to RACF and the DFSCTRN0 user exit prior to calling the DFSMSCE0 user exit during Link Receive processing. DFSMSCE0 can set the level of authorization checking. Calls to RACF and DFSCTRN0 are made after DFSMSCE0.



The security authorization options that can be returned by the DFSMSCE0 user exit in field MSLRFL3 are also provided as a startup option. These new options are specified as the second value in the MSCSEC= keyword in the DFSDCxxx proclib member.



Additional data will be passed to the DFSMSCE0 user exit including: userid, group name, and userid indicator. The userid indicator clarifies the value in the userid field as being one of: Userid, LTERM name, PSB name, MSNAME, or Other.

The DFSMSCE0 user exit call during Link Receive processing has the ability to control the level of authorization checking. The user exit response in field MSLRFL3 specifies the level of authorization as one of:

Authorization by MSNAME where the ACEE is dynamically created for the first authorization, then reused.

Authorization by CTL address space security.

Authorization by userid of inputting terminal where the ACEE is dynamically created and deleted for each authorization.

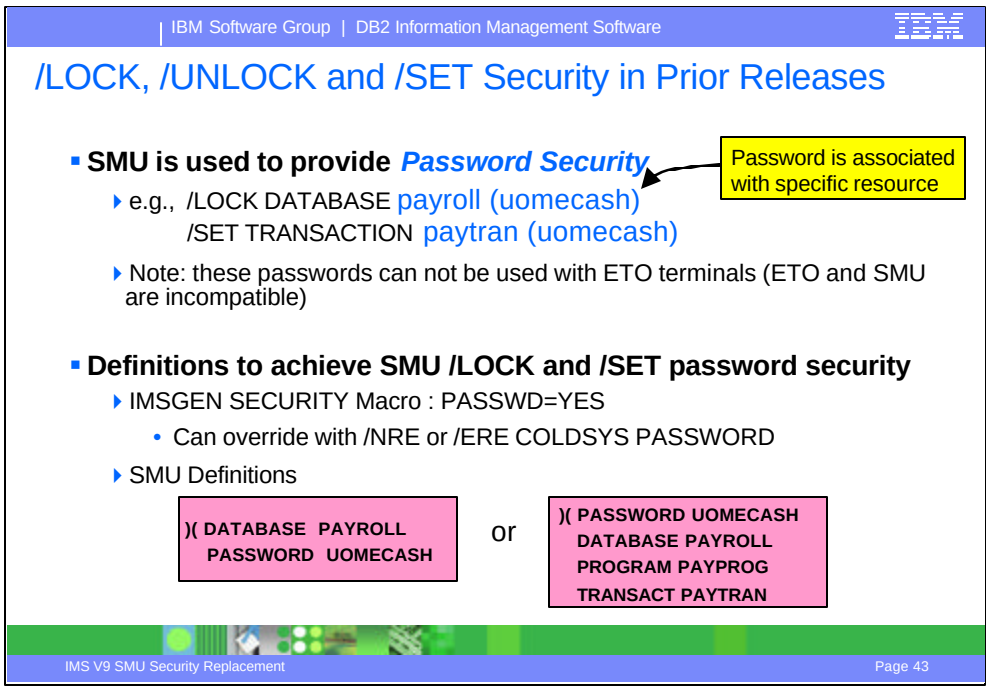
Authorization by user exit DFSCTRN0.

No security authorization checking.

Migrating Off SMU

- **When migrating to IMS V9, add to DFSDCxxx**
  - MSCSEC=(LRDIRECT,USER)
    - or authorise control region for transaction execution, and take default MSCSEC values (LRDIRECT,CTL)
- **Decide what type of userid to use for directed and non-directed routing**
  - Easier when both the same, but can be different
- **Update RACF to include new userids (MSNAMEs and Ctl Rgn) if necessary, and grant their access to transactions**
- **If using two types of userid, code DFSMSCE0 accordingly**
- **Change DFSDCxxx to include**
  - MSCSEC=(LRALL,USER |MSN |CTL)
- **Restart IMS**
- **When safe, remove SMU definitions**

/LOCK, /UNLOCK and /SET Security



The /LOCK and /UNLOCK LTERM|DATABASE|PROGRAM|TRANSACTION|NODE|PTERM commands also support specification of password security for the defined resource. In prior releases, this was accomplished with the use of SMU. If the resource was not defined with password protection in SMU, the check was not made and the password ignored.

Password protections was by the specification of the PASSWD= parameter on the SECURITY macro along with the appropriate SMU definitions.

There were, however, certain restrictions:

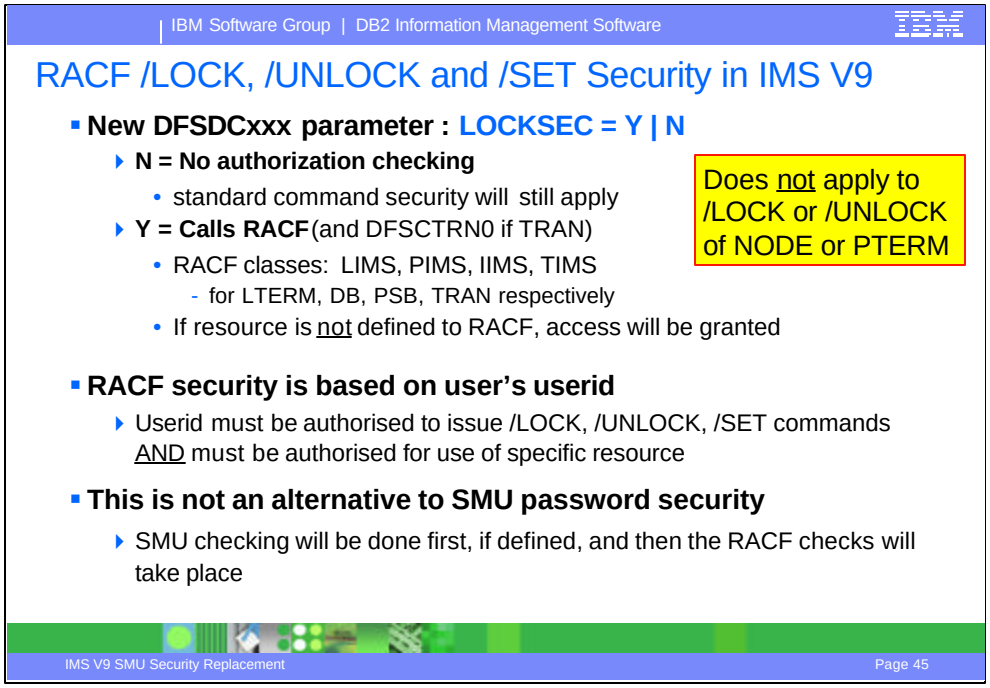
The /LOCK DATABASE, /LOCK PROGRAM, and /LOCK TRANSACTION commands are only valid when entered from: a master terminal or system console; TCO script; or AO program.

The /LOCK TRANSACTION command cannot be used for CPI-C driven programs.

The /LOCK LTERM, /LOCK NODE, and /LOCK PTERM commands apply only to the entering physical terminal.

Use of /LOCK, /UNLOCK and /SET Security

- **An "end user manager" can LOCK and UNLOCK his users' LTERMs**
  - One or more LTERMs for a physical terminal
  - Only he knows the password to do this (when using SMU)
- **Similarly he can SET the destination transaction code for a terminal**
  - Only he knows the password to do this (when using SMU)
- **Senior operators can LOCK and UNLOCK DBs, programs and transactions**
  - Only they know the passwords to do this (when using SMU)
- **In IMS V9 with RACF, these "special people" are explicitly authorised to LOCK, UNLOCK and SET specific resources**



In IMS V9, the security of the LTERM, DATABASE, PROGRAM, and TRANSACTION parameters include the use of SAF (RACF) and/or the Transaction Authorization Exit, DFSCTRN0. The resources are defined to RACF in the classes: LIMS, PIMS, IIMS, and TIMS respectively. Because this RACF and exit call were not made in prior releases, the call cannot be made unconditionally beginning in IMS V9. Therefore, a new startup option LOCKSEC=Y|N (where N is the default) has been added to the DFSDCxxx IMS.PROCLIB member. If LOCKSEC=Y is specified, the new RACF and user exit calls are made after the optional call to SMU security. If the resource is not defined to RACF, or is defined and is authorized to the user, the command is processed. If the resource is defined to RACF but is not authorized for use, the command is rejected with a new message DFS3689W.

DFS3689W USE OF resourcename BY LOCK/UNLOCK REJECTED

Explanation: The transaction, LTERM, database, or program resource entered as a parameter on the /LOCK or /UNLOCK command is not authorized for use by the user entering the command.

No RACF call made for /LOCK and /UNLOCK NODE | PTERM. Protection of VTAM nodes and BTAM terminals relies on the use of RACF TERMINAL|GTERMINAL support

Migrating Off SMU

- **Define to RACF all resources that need to be LOCKed or SET**
  - LTERMs, DBs, Programs (PSBs), and Transactions
- **Grant authority for using these resources to the appropriate userids**
- **Add LOCKSEC=Y to DFSDCxxx**
- **Restart IMS**
- **When safe, remove SMU definitions**
- **Inform users that passwords are no longer needed**

Sign On Verification Security

Signon Verification Security

- **SMU method for static terminal Signon Verification**
  - Defines which static (non-ETO) terminals must /SIGN ON

```
)( SIGN
        STERM TERM1
        STERM TERM2      }  or    STERM ALL
        STERM TERM3
        ...
```

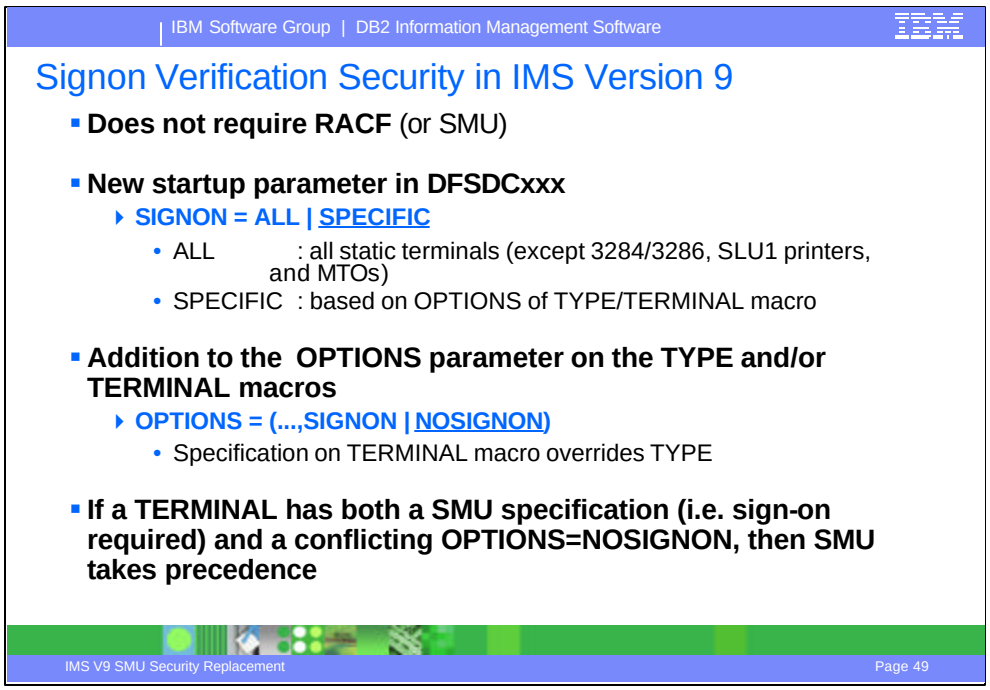
  - Requires
    - SECURITY SECLVL=SIGNON or FORCSIGN
  - … and typically requests RACF verification of userid/password with
    - SECURITY TYPE=RACFTERM

In previous releases, statically defined terminals were only required to signon if the terminal was defined in SMU as being required to do so. This was done with the)(SIGN control statement. SMU, however, was limited to supporting:

- a maximum of 65535 LTERM definitions

- a maximum of 32767 different patterns for sets of commands and transactions (a pattern is unique set of commands and/or transactions that can be issued by terminals)

- a maximum of 32767 terminals to require sign-on



In IMS V9, a new initialization parameter, SIGNON, is provided in the DFSDCxxx proclib member to request that static terminals be required to signon. The format and the options for SIGNON are: SIGNON=ALL | SPECIFIC.

Specification of ALL requests that all static terminals be required to signon. This is equivalent to the SMU definition of)(SIGN STERM ALL. Note that like the SMU support, SIGNON=ALL will not set sign on required in LU6.1, 3284/3286, SLU1 (when printer-only device), and master terminal devices.

The use of SPECIFIC defines that individual static terminals may be required to signon. These terminals will be specified in either system definition with the OPTIONS=SIGNON parameter on the TYPE and/or TERMINAL macros, or via SMU definitions with the)(SIGN statements.

Migrating Off SMU

For "ALL"

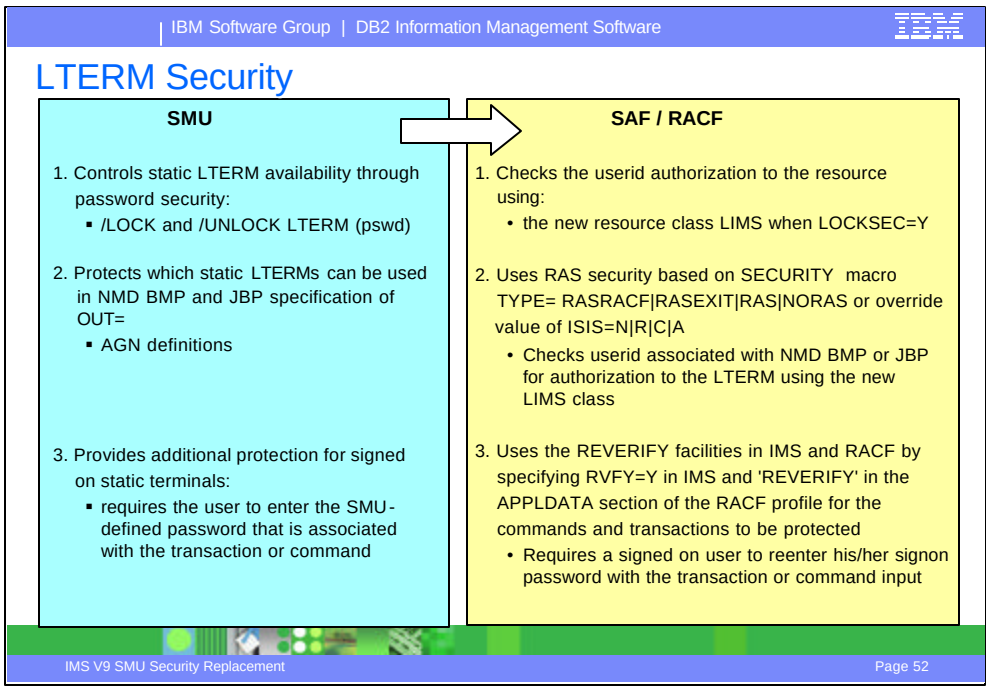
- Add SIGNON=ALL to DFSDCxxx
- Restart IMS

For "SPECIFIC"

- Add OPTIONS=(…SIGNON…) for all TERMINALs which currently have an explicit SMU signon requirement
- Add SIGNON=SPECIFIC to DFSDCxxx
- Restart IMS

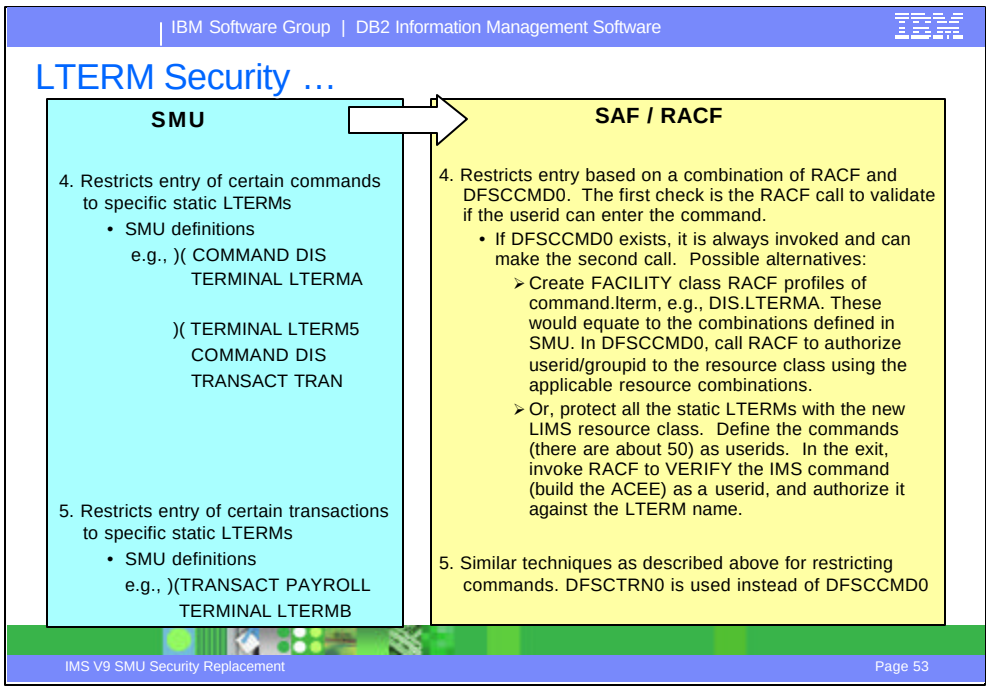
- When safe, remove SMU definitions

Other Considerations



The next two visuals show the considerations when migrating LTERM-based security from the use of SMU to the use of SAF/RACF.

The first three areas show specific SAF/RACF solutions to assist in LTERM migration considerations.



These two areas are a little more complex since there is no direct RACF equivalent .

To migrate LTERM-based command security, consider using RACF in conjunction with the Command Authorization Exit Routine (DFSCCMD0). Normal RACF command authorization checking based on userid/group may be used. If it exists, the Command Authorization Exit DFSCCMD0 is always invoked after RACF for commands entered from IMS terminals. The sample DFSCCMD0 exit is coded to determine if a terminal is ETO or STATIC. If DFSCCMD0 determines that the command is from a static terminal, rather than accept the command, the exit could be changed to reinvoke RACF to do one of the following:

- check a special user-defined RACF profile. For example, the exit could obtain the name of the static LTERM used to enter the command and construct a RACF profile name from a combination of the command verb and inputting LTERM name. Finally, the exit could invoke RACF to determine if the userid/group is authorized to the command/LTERM profile.

- invoke the command as a userid. This alternative approach would be to RACF protect all the static LTERMS that were defined in SMU using the new RACF LIMS resource class. Additionally, since there are only about 50 commands, they could be defined to RACF as userids. DFSCCMD0 could then verify (build the ACEE) for the IMS command as a userid and authorize it against the LTERM name.

Similar techniques as those described above for restricting commands could be used for restricting transactions. The difference would be that the Transaction Authorization Exit (DFSCTRN0) would be used instead of DFSCCMD0. Note, however, that DFSCTRN0 is not called for any further security checking if the userid is not authorized to the transaction. As with the command examples, you could also do all of the following to accomplish the same thing:

- Protect all of the static LTERMs in the new LIMS resource class provided by IMS V9

- Define the affected IMS transactions as RACF userids

- Use RACF's VLF ACEE cache so the transaction ACEEs are always in storage (eliminates I/O to the RACF DB)

- Invoke RACF to VERIFY (build the ACEE) for the IMS transaction name as userid if one did not already exist

Migration Considerations

- AOI considerations
  - CMD has new status code and new return/reason (AIB) codes
  - ICMD has new return/reason codes
- Log record (type X ‘10’) has new error codes
- New and changed Exits
  - DFSRAS00, DFSCCMD0, DFSISIS0, DFSMSCE0
- New RACF security classes
  - IIMS, JIMS, LIMS, MIMS
    - Predefined in z/OS 1.6
- Changing RCF from “N” to something else requires a cold start

Summary

- **Prior to IMS V9, there are six security functions that are only possible with SMU**
  - They can still be implemented with SMU in IMS V9
  - But this will not be so in the follow-on release of IMS
    - **IMS V9 is last release to support SMU**
- **IMS V9 introduces new facilities that enable these six security functions to be implemented with RACF (or equivalent product)**
  - Also adds some new security functions
- **Recommend: all users of SMU should install IMS Version 9 and then migrate all SMU functions to RACF while on this release**