

A good team RACF and CICS

In the following slides I will describe how you can secure CICS transactions with RACF

Is the combination of both unbeatable?



WHATS IS:

RACF:

- RACF stands for Resource Access Control Facility.
- It's an IBM product.
- RACF can be used as a External Security Manager (ESM) for other
- IBM products DB2, MQ, CICS.
- Those products has to invoke RACF as External Security Manager.
- So RACF can check if a user has the proper authorisation to use the other product in this case a CICS profile and CICS transaction.

CICS:

- CICS stands for Customer Information Control System
- CICS® Transaction Server, often called simply *CICS*, is a powerful, mixed-language application server that runs on the IBM® mainframe operating system called z/OS®.

An application server provides an environment to host applications.

It can provide services to solve many concerns, such as security, transactionality, or exchanging data between new and existing applications. Developing custom enterprise-grade solutions for these issues is difficult and can take time away from focusing on what the application is intended to do for the business. Importantly, CICS can provide these services to applications that are composed of components written in different programming languages.

CICS is used as an application server by a host of different companies in numerous industries, such as banking, transportation, retail, and insurance. The kind of applications that run in CICS often form the core of the owning business and are critical to their success.

If these applications fail, unexpected downtime or errors can result in significant revenue loss and impact on reputation.

This significance means that they need to run on a highly available, reliable, and robust platform such as CICS.

A good team RACF and CICS

Some RACF basics

RACF	Explanation
Userid	Userid (6:8 characters) related to personal enduser
Group	Defines a group of users
ADDGROUP	Command of R-A to define a group in RACF
RDEFINE	Command of R-A to define a RACF-resource
PERMIT	Command of R-A to allow a user using RACF resource
REFRESH	Command of R-A to refresh RACF class in mainstorage. This command is always necessary after changing CICS permissions.

The following RACF definitions specifies OCTAVIA has access to run the CEMT transaction:

```
RDEFINE 1 TCICSTRN 2 CEMT 3 UACC(NONE) 4 PERMIT CEMT  
CLASS(TCICSTRN) ID(OCTAVIA) ACCESS(READ) 6
```

The order is important. First you define a resource profile in a class using the RDEFINE command. Then you permit users to access it using one or more PERMIT commands.

TCICSTRN is a class that contains a set of resource profiles representing resources.

The RDEFINE command explicitly uses the transaction name as the resource profile. CEMT is the resource profile that protects the transaction CEMT.

UACC(NONE) means that no one has access unless they are explicitly permitted to have access. UACC other than none means that everyone needs some access for their job, such as a transaction that displays help information.

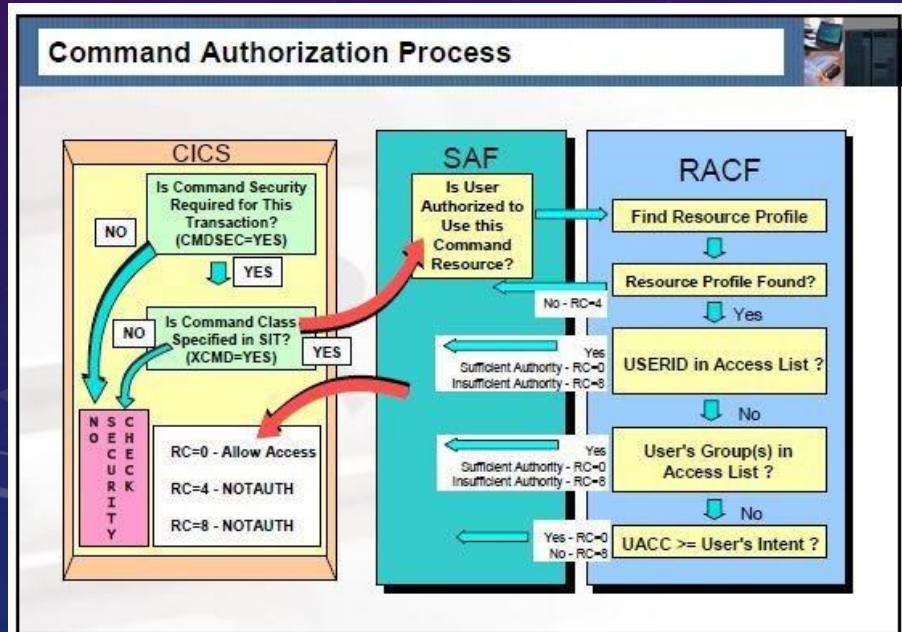
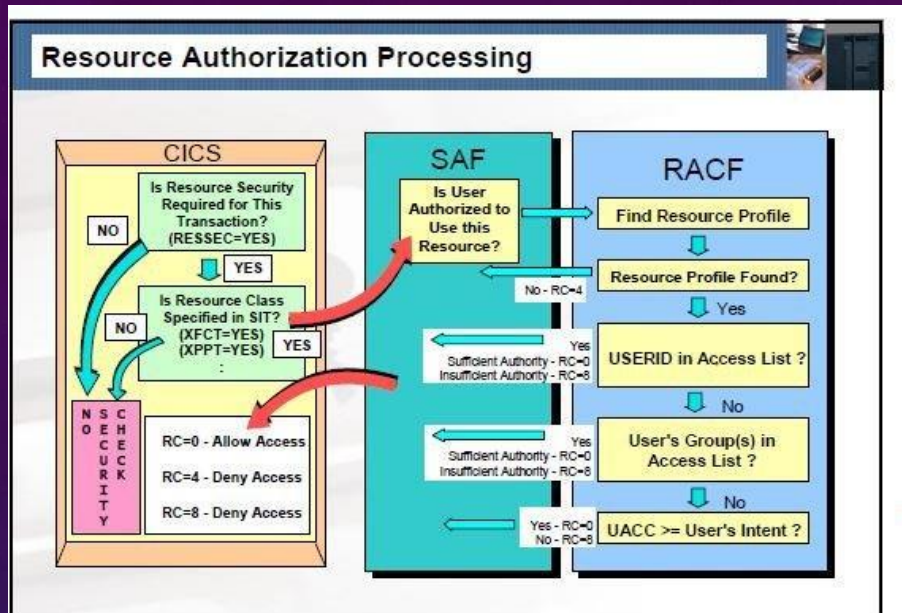
The format of the RDEFINE and PERMIT commands are different, but the TCICSTRN class and the CEMT resource profile are the same in both.

The user ID OCTAVIA is given READ access to the CEMT profile in the TCICSTRN class. To CICS, READ means OCTAVIA can run the CEMT transaction.

SOME CICS BASICS

CICS part	Explanation (In this situation ESM is used (SEC=YES))
CICS region	A single CICS region is a CICS region that is not part of any CICSplex
CICS resource	Specific part inside CICS defined by C-A (Terminal, Files, Programs, Transactions enz) for example CEMT. A CICS resource must be defined in the Resource Definition Attributes where you can select the specific parameters DB2, MQ, WEB, and so on....
CICS Program	
CICS Profile	Specific CICS profile defined in RACF
CICS Permission	READ, UPDATE
XTRAN SIT parameter	CICS SIT Parameter which defines transaction classes (GCICS/TCICS) (to use RACF Resource Classes GCICSTRN/TCICSTRN). SIT stands for System Initialisation Table where you define XTRAN=YES to check the transaction by RACF.
CICS Transaction	A transaction is an item of processing that is initiated by a single request. This request might be from an end-user at a terminal, a web page, a remote workstation program, or an application in another CICS system, or triggered automatically at a predefined time. A single transaction consists of one or more application programs that, when run, carry out the processing needed.
CICS MEMBER/transaction	The self-defined transaction in CICS. You define the transaction in CICS by Transaction Resource definition. Unlike other transaction architectures, CICS can maintain a transaction over the execution of programs that are written in different programming languages. A transaction (task) can execute several programs in the course of completing its work.

In What way is CICS using the External Security Monitor (RACF)



- In both pictures is using CICS RACF for security control
- So in the CICS SIT table = security YES
- Via SAF call is CICS checking the security
- First finding the CICS Resource Profile

• When it is found => It is checking if the **User** is on the Access list. (YES => RC 0) (NO => RC 4 or RC 8)

• When the user is NOT found => It is checking if the user have The autorisation to use a certain **GROUP**. (YES => RC 0) (NO => RC 4 or RC 8)

• When the user is NOT found => It is checking the **UACC** of the Resource profile. (YES =>RC 0) (NO => RC 4 or RC 8)

The outcome of the security control in RACF is given back to CICS. Returncode (RC)=0 means Access. RC=4 or 8 means NO Access.



```

set mirror object to m
mirror_mod.mirror_object

operation = "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation = "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation = "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
("Selected" + str(modifier_ob
mirror_ob.select = 0
= bpy.context.selected_obj
data.objects[one.name].sel

print("please select exactly

-- OPERATOR CLASSES ----

types.Operator):
on X mirror to the selected
object.mirror_mirror_x"
mirror X"

context):
context.active_objer

```

DIFFERENT TYPE OF TRANSACTIONS

Different Categories of CICS transactions

Category	Program/ Table	Category / Group	Cat description/ Sec category	Like transaction	Security Recommendation
Category 1 CICS transactions	Table 3	Cat.1	Category 1 transactions are for CICS internal use only and must not be started from a user terminal	CATA	Only CICS region IDs allowed to use
List of CICS sample transactions	Table 4	Cat.2	Category 2 transactions are initiated by CICS users or are associated with CICS users . For explanations of the subcategories, see Transactions in CICS . Like CICS administration transactions.	CEDA CEMT	Key technical personnel only
Cat.3 CICS transactions		Cat.3	Category 3 transactions are available to all users, whether signed-on or not.	CESN	No security needed or appropriate

SECURITY ADVICES:

- CICS is very fast and security by ESM (RACF) is highly effective.
- Beware what your CICS transactions can do!!
- Security requirements for all CICS supplied transactions are documented in CICS-RACF Security Guide.
- CICS security is for teamplayers because it can handle a lot of things and can work with a lot of objects like WAS, JAVA,DB2,MQ.
- All these things can be secured by RACF.
- To achieve the best security all application builders, programmers, technicians and security administrators has to work together.
- Document what your transactions can do.
- Mind security breaches and analyse all possibilities to minimize Security Risks.

WHICH SITES, BOOKS OR PDF I HAVE USED?

Sites:

<https://www.ibm.com/docs/en/cics-ts/6.x?topic=cics-how-it-works-securing-racf>

<https://www.ibm.com/docs/en/cics-ts/6.x?topic=racf-security-administration-tasks-commands-run-them>

