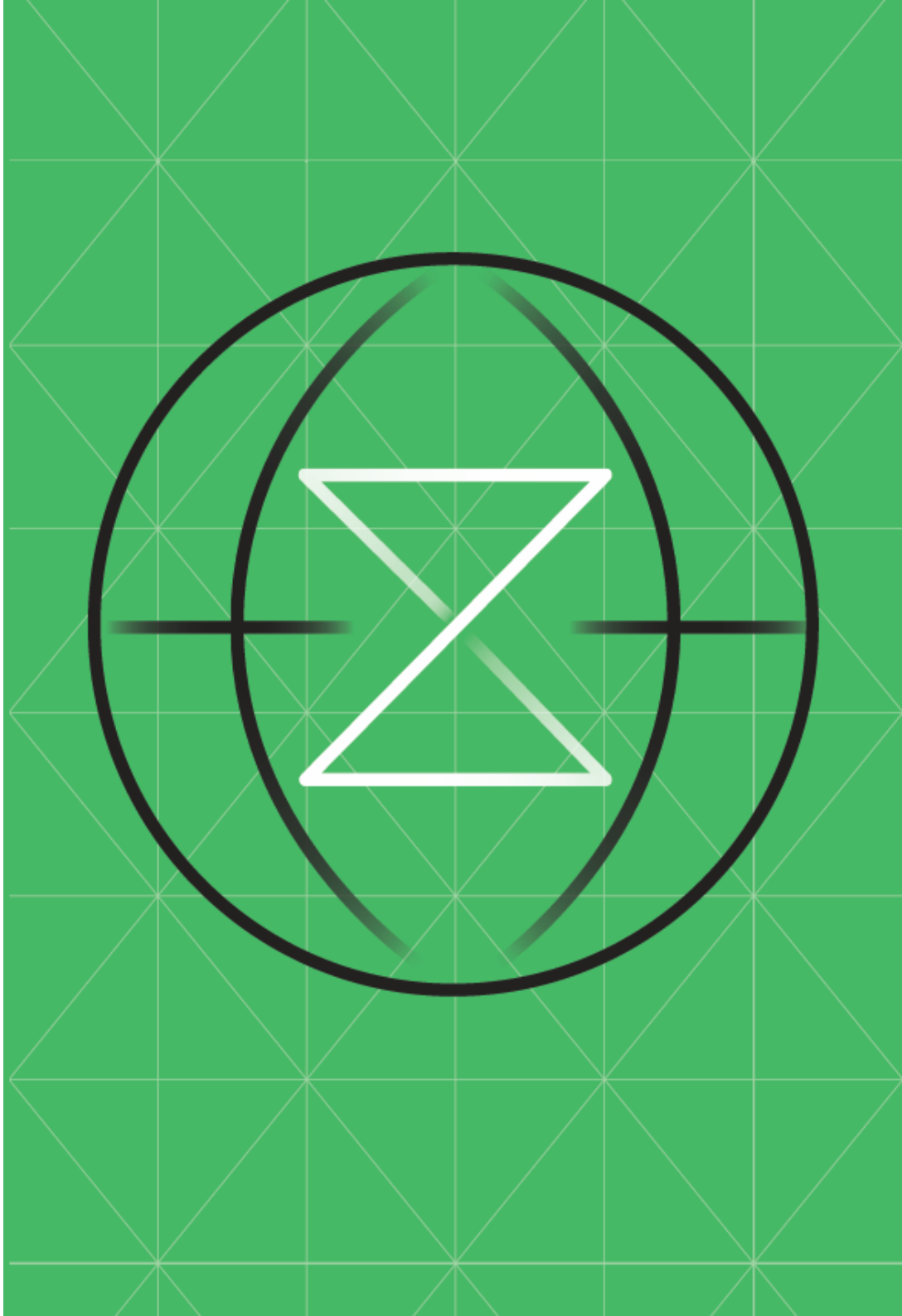
Extended

230222-1423

RACF1

zOS Security with RACF



Z Xplore

SECURITY, WITH RACF

Getting started with the basics of RACF and understanding key terminology.

THE CHALLENGE

RACF is one of the external security managers (ESM) that protects the z/OS resources on IBM zSystems. There is a lot that encompasses RACF. In this challenge, we will be introducing you to the terminology of RACF and dip our toes in the different RACF commands.

BEFORE YOU BEGIN

You will be using a 3270 terminal for this challenge. If you have not completed the TSO challenges, please complete those first.

INVESTMENT

Steps	Duration
10	90 minutes

RACF1|230222-1423

1 SUBJECTS VS. OBJECTS

The key to understanding RACF is understanding the terminology.

First and foremost, when dealing with security, we are dealing with *subjects* and *objects*.

- Objects: Things that are protected. This can be documents, profiles, data, etc.
- Subjects: Want to use and gain access to the objects. This can be a software, user identification, and more.

In RACF:

- Subjects access objects
- Objects are protected

Throughout the next few challenges, we will be dealing with subjects and objects concurrently.

RACF1|230222-1423

1.1 SUBJECTS AND OBJECTS ARE FLUID

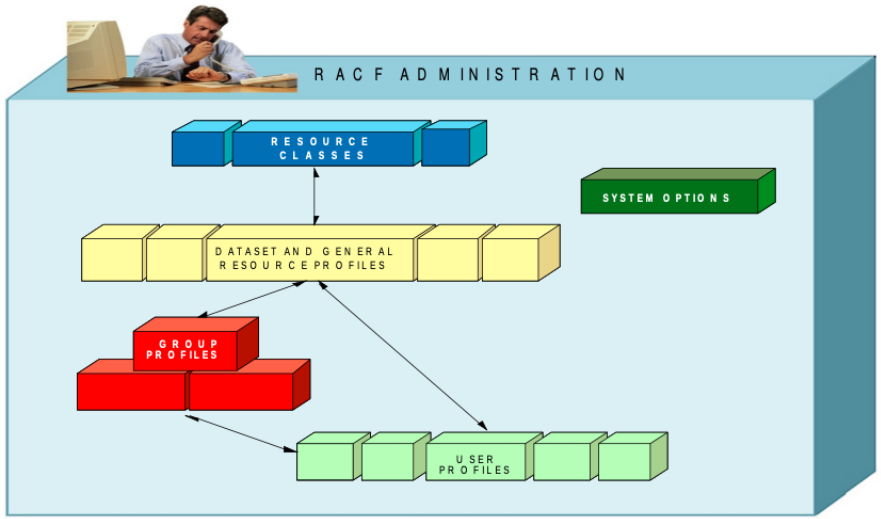
z/OS consists of resource managers, like TSO. TSO is capable of being an object that is protected while also being a subject permitted to get to other objects.

Example of Subjects/Objects Using TSO:

- When TSO is the object:
  When you log on to TSO, TSO is the object where your assigned ID (zxxxxx) is the subject. The datasets that you want to access are objects (e.g. zxxxxx.jcl.new).
- When TSO is the subject:
  If TSO wants to access a certain software or resource, like Workload Manager (WLM), TSO then becomes the subject and WLM becomes the object.

2 INTRODUCTION TO RACF TERMINOLOGY

RACF uses profiles and the RACF database contains information needed to control access to resources.



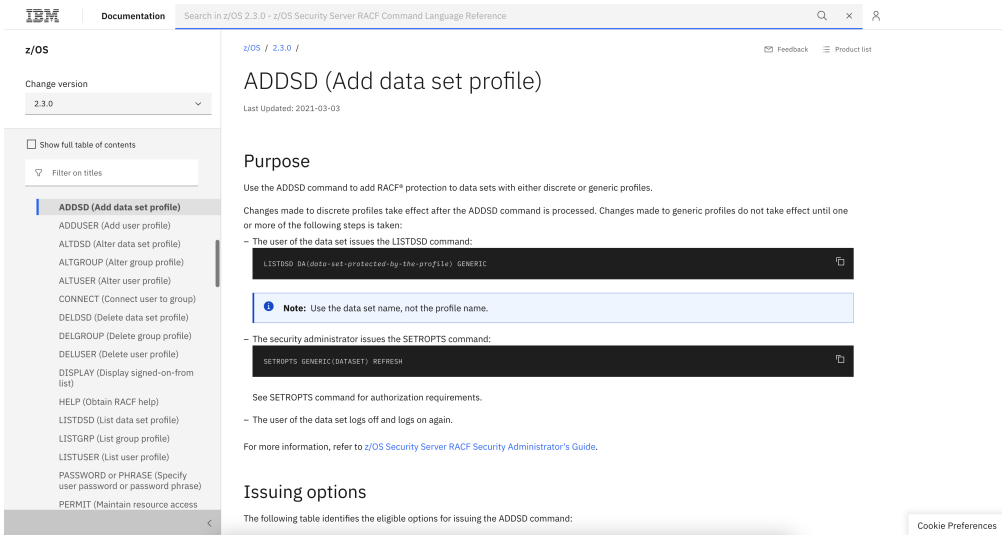
Terminology	Description
Resource Classes	Designs the installation’s custom fields
Resource Profiles	Describes security characteristics of a user or group of users within a resource class
Group Profiles	Describes a group of users and information on the group
User Profiles	These are the subjects, the way to permit access to objects. **Example: Your IBM zSystems ID**
System Options	Global behavior of the environment

Along with the structure and profiles, there are commands used in order to add, change or delete permissions given to a user. Users, like yourself, have limited RACF authority. As a general user, you can restrict or permit other users to access objects that you own. So, you can add, change, or delete permissions to any object that YOU own to any user, such as a data set.

We will see how this works in the following steps.

3 DO YOUR RESEARCH

When learning something new on an IBM zSystem or any new skill, you won't always have all of the answers in front of you. Part of learning is using documentation on the technology and figuring things out on your own.



Use your research skills to figure out how a user can create, add, update and delete RACF permissions. Try googling "racf command syntax".

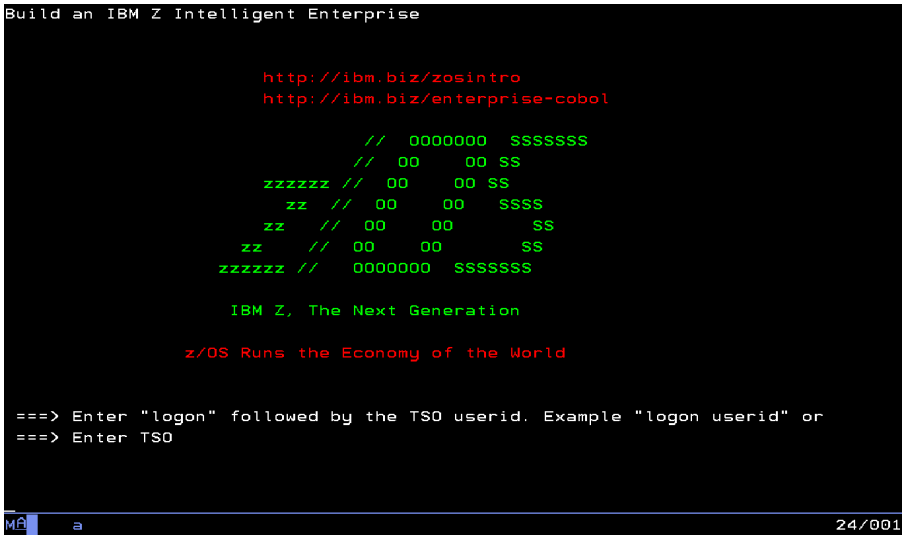
Make sure you look at the command, command operands, and operand values.

Hint: One command we are using is the 'ADDSD' command, the UACC operand, and the 'read' operand value. Try starting there and see how they fit together!

These permissions are important. This is how you have control over resources that *YOU* own. They allow you to give authority to others to read/update/delete/etc. We will use permissions next in this challenge.

Let's jump over to our 3270 terminal and get familiar with RACF.

4 LOG INTO YOUR EMULATOR



You should already be familiar on how to set up and log into your emulator.

Remember: Your emulator will depend on the device you are using.

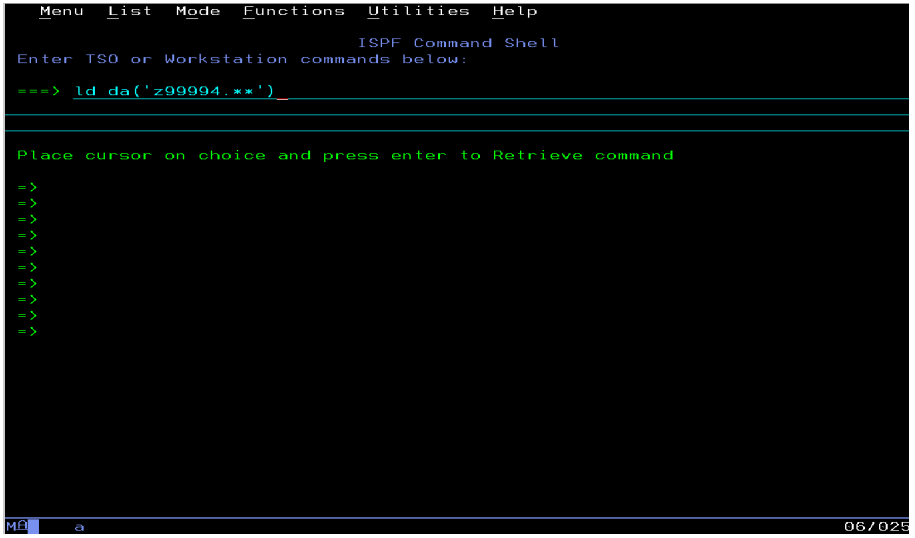
Log in using your ID and Password and navigate to ISPF Command Shell (*Option 6*).

5 UNDERSTANDING COMMANDS USING ISPF COMMAND SHELL

Let’s look at how you list the permissions for data sets under your own ID.

On the command line, type `listdsd da('YOURZID.**')`.

Tip: The short version for listdsd is ld



What do you see?

Look at the ‘Owner’, ‘Universal Access’ and ‘Your Access’ columns.

- What does *alter* mean?
- What does *none* under Universal Access mean?

This is where researching documents comes in handy.

Note: The asterisks being used by RACF are to show all of the security/protection a user has. Above, you are asking to list the security permissions you have for any data set under your ownership. Later, we will specify specific datasets we want to use.

6 ADDING RACF PERMISSIONS

Now that you understand how to view your own profile, let's look at how you add, change and delete permissions to a dataset.

The dataset we will be working with is *NOT* created yet. In RACF2, you will add in the dataset. Right now, we are simply adding permissions to our future dataset.

6.1 ADD PERMISSIONS

First let's add a dataset. On the command shell line, type: `addsd 'YOURZID.secret' generic uacc(read)`.

Press enter.

The way you write out permissions varies depending on the command you want to use and what permissions you want to give. To understand the syntax for each command, you can either look up the documentation online or type: `tso help [insert command you need help with] syntax`

In this example, we are putting:

`[the command we want]` + `[resource profile]` + `[generic]` + `[permission]`

so…

`addsd` + `yourzid.dataset` + `generic` + `uacc (read)`

6.1.1 HELPFUL HINT: Generic Syntax

Generic typically is used when you have * or ** to indicate that the permissions apply to all datasets that qualify.

For example, `addsd 'ZXP.PUBLIC.**' generic uacc(none)` sets the default access for all ZXP.PUBLIC datasets to none.

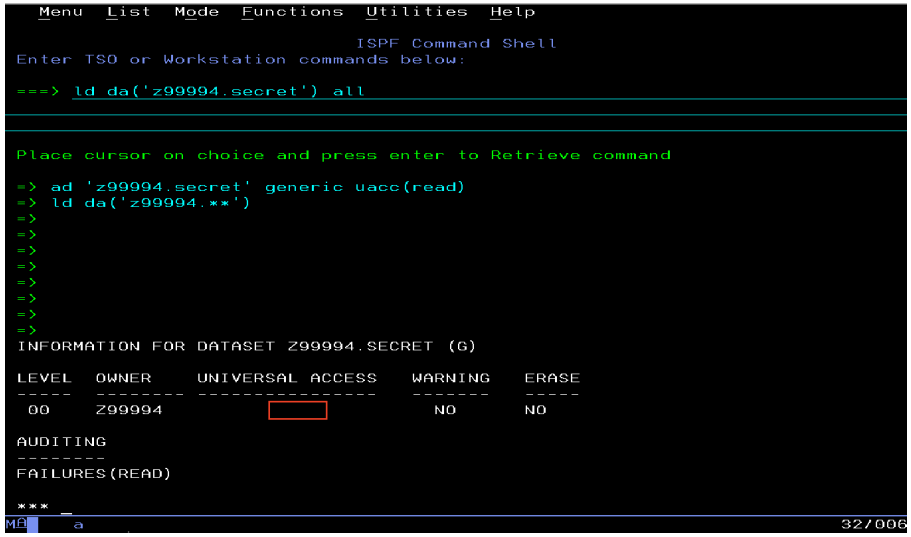
Generic can also be used when you have not yet created the dataset you are adding permissions to. Since we have not yet created the “secret” dataset, we are using generic. We are letting RACF know that if/when it is created, these are the permissions.

Now that we have added permissions to a dataset, let’s see how our permissions have changed since our first list dataset command (ld).

Remember how we listed our user profile permissions? Go back a page if you need a reminder. Instead of using **, this time we will specifically list the dataset we want to see permissions for, in this case, ‘secret’.

Type `ld da('YOURZID.secret') all`.

Press enter.



You should see something different under the Universal Access column, what does it say?

7 SEE HOW IT WORKS

You've now added read permissions to a non-existent dataset. So, how do you know if permissions work? Let's see what it looks like with datasets that are already created.

7.1 PERMISSION VS. RESTRICTION

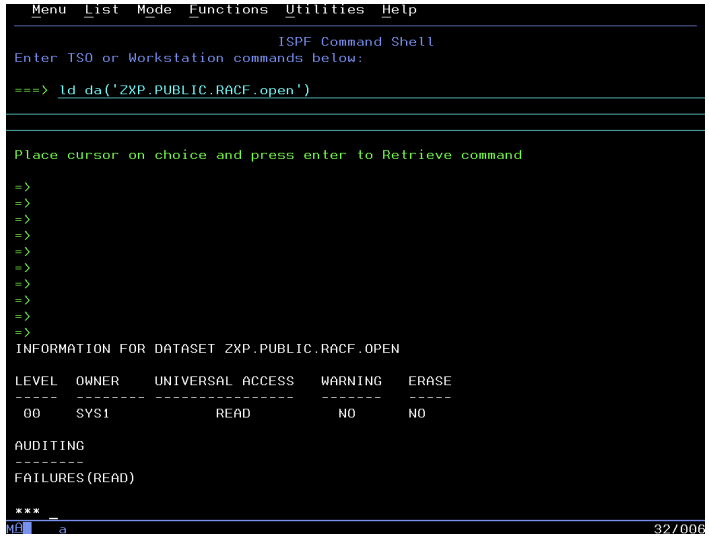
We have created two different datasets in ZXP.PUBLIC:

- ZXP.PUBLIC.RACF.OPEN
- ZXP.PUBLIC.RACF.CLOSED

Let's look at how *read* permission affects a users access to a dataset.

On the Command Shell, try using the list dataset command (listdsd or ld) to read the two datasets.

Hint: For ZXP.PUBLIC.RACF.CLOSED, you will need to add generic at the end of your command.



Are you able to read both datasets? To double check, head over to =3.4 and view the two datasets. Can you view them?

8 ALTERING AND DELETING RACF PERMISSIONS

Let’s look at some of the other access authorities available.

8.1 COMMON COMMANDS

Command	Short-version
LISTDSD	LD
LISTUSER	LU
ADDSD	AD
ADDUSER	AU
ALTDSD	ALD
DELDSD	DD
DELUSER	DU
PERMIT	PE

Let’s put some of these commands to use.

8.2 ALTER PERMISSIONS

We've now added read permissions to our *secret* dataset using **read**. What about if you want to alter a data set? How would you change the universal access(access-authority) to another access? Head back over to ISPF Command Shell.

Change the access from **read** to **update**.

Tip: Update means to "write", giving the user editing access.

Let's make sure the change was implemented correctly. Repeat the list dataset command to see how UACC has changed. If it hasn't, go through the steps again.

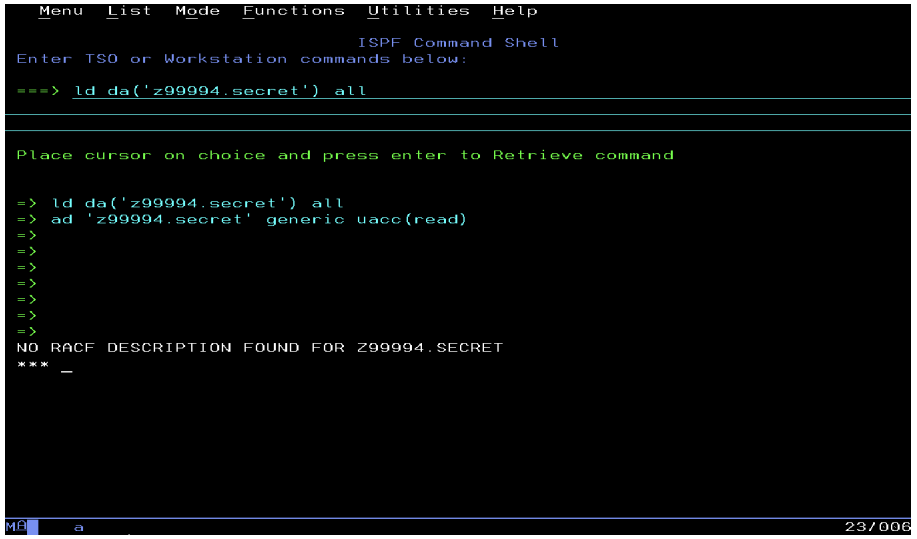
8.3 DELETE PERMISSIONS

Let's delete permissions.

Deleting a dataset profile is similar, but a bit different. Can you figure out what the command syntax is?

Hint: You don't need to assign uacc to a dataset profile you are deleting permissions to.

Check to make sure you deleted the dataset profile correctly. Use the list dataset command again, this time, you should see that the description is not found.



9 PUT IT ALL TOGETHER

Create a sequential (PS) dataset called ‘YOURZID.magic’.

Ex: ‘z99994.magic’.

If you don’t remember how to create a dataset, go back to the TSO challenges.

Go back to the ISPF Command Shell:

1. Create a DSD profile that matches the dataset (addsd) with UACC (NONE)
2. Permit user IBM0001 to have write (UPDATE) access to your new dataset.

Hint #1: You will be using the permit command to do this. If you aren’t sure of syntax, look up the documentation.

Hint #2: Since the dataset is already created, generic is not needed for these commands.

If you are having trouble, review the other steps in this challenge and/or review the RACF documentation online. There are a lot of resources to help you.

After you have successfully done this, find ZXP.PUBLIC.JCL and submit the member CHKRACF1.

Nice job - let’s recap	**Next up …**
You have been introduced to RACF and how to create, change, display and delete a data set profile. These are only a few of the many commands you can use. Check out https://www.ibm.com/docs/en/zos/2.3.0?topic=reference-racf-command-syntax to see all of the other RACF commands and their syntax.	Deepen your RACF skills in RACF2. You will be taken through a mini scavenger hunt to create a dataset and RACF permissions for that dataset.