Draft Document for Review February 7, 2025 4:44 pm SG24-8578-00

Redbooks
ibm.com/redbooks

IBM RACF Security Impact Forecasting using IBM zSecure

Bill White
Mike Riches
Elijah Swift
Jeroen Tiggelman
Scott Woolley
Tom Zeehandelaar

IBM Z

Security

IBM

Redbooks

IBM Redbooks

IBM RACF Security Impact Forecasting using IBM zSecure

February 2025

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (February 2025)

This edition applies to IBM z/OS Version 3, Release 1 (product number 5650-ZOS) and IBM zSecure Admin Version 3, Release 1, Modification 1 (product number 5655-ABB).

This document was created or updated on February 7, 2025.

Contents

Notices v
Trademarks vi

Preface vii
Authors vii
Now you can become a published author, too! viii
Comments welcome ix
Stay connected to IBM Redbooks ix

Chapter 1. Tighten access controls with confidence 1
1.1 IBM Z security challenges 2
1.2 Hardening practices 2
1.2.1 Zero trust 2
1.2.2 The principle of least privilege 3
1.2.3 Role-based access control 3
1.3 IBM Z security layers 3
1.3.1 External security managers on z/OS 3
1.3.2 RACF 4
1.4 A phased approach 4
1.4.1 Define new security rules to strengthen security 4
1.4.2 Remove unused security definitions 5
1.4.3 Establish role-based access control 5
1.4.4 Keep the defined privileges in the environment to a minimum 5
1.5 Lesser-known features in IBM zSecure Admin 5
1.5.1 zSecure Admin Access Monitor 5
1.5.2 zSecure Admin RACF-Offline 6
1.5.3 The Access Monitor function in zSecure Admin 6

Chapter 2. IBM zSecure capabilities for RACF administrators 7
2.1 IBM zSecure Admin for RACF administrators 8
2.2 zSecure Access Monitor 10
2.2.1 Possible uses for zSecure Access Monitor 10
2.2.2 Access Monitor environment 11
2.3 zSecure RACF-Offline 14
2.3.1 Testing RACF database cleanup with RACF-Offline 15
2.4 General preparation steps for the use cases 16
2.4.1 Creating a RACF-Offline database 16
2.4.2 Starting a RACF-Offline session 17
2.4.3 Logging on to RACF-Offline session 19
2.4.4 Resetting the RACF-Offline log data set 20
2.4.5 Starting and preparing your zSecure Admin session 20
2.5 Further access monitor capabilities 22

Chapter 3. Add a new set of profiles 27
3.1 Challenges with adding new resource profiles 28
3.2 Steering clear of default security settings 28
3.3 Temporarily allowing access for new resources or applications 28
3.4 Adding new profiles to administer more security checking 29
3.4.1 Analyzing statistics and discovering missing RACF resource profiles 29

3.4.2 Defining a new RACF resource profile in the RACF-Offline database 35
3.4.3 Conducting simulation of Access Monitor data against the offline RACF database . 39

Chapter 4. Remove unused security definitions 55
4.1 Challenges with unused RACF definitions 56
4.2 Keeping the RACF database orderly and well maintained 56
4.2.1 Address inactive users 56
4.2.2 Address unutilized groups. 57
4.2.3 Address users with outdated group connections 57
4.2.4 Remove unused resource profiles 58
4.3 Ensuring only the required authorization is in place 58
4.4 Removing unused resource profiles 59
4.4.1 Generating RACF commands to clean up resource profiles. 59
4.4.2 Running the generated RACF commands to clean up resource profiles 62
4.4.3 Recovering deleted resource profiles 64
4.4.4 Reporting potential security impact of deleted unused RACF profiles 67
4.4.5 Cleaning up unused profiles from the active primary RACF database 74
4.4.6 Cleaning up other unused security definitions from the active primary RACF database. 77

Chapter 5. Convert generic and specific access to group-based access 79
5.1 Challenges with generic and specific access permissions 80
5.2 Moving towards role-based access control 80
5.2.1 Minimize the use of universal and generic permissions 81
5.2.2 Minimize use of personalized permissions 81
5.2.3 Minimize the assignment of the OPERATIONS attribute 82
5.2.4 Minimize the use of the WARNING attribute. 82
5.2.5 Minimize the use of global access checking tables. 82
5.3 Establishing groups that accurately reflect user roles. 83
5.4 Converting to group-based access 83
5.4.1 Reporting successful access allowed through the UACC setting 84
5.4.2 Converting generic UACC access to group-based access 87
5.4.3 Reporting the effect of the UACC conversion commands. 95
5.4.4 Reporting successful access allowed through ID(*) 101
5.4.5 Converting generic ID(*) access to group-based access 105
5.4.6 Converting generic OPERATIONS access to group-based access 113

Chapter 6. Minimize access control privileges 121
6.1 Challenge with access control privileges 122
6.2 Avoiding the need to trust privileged users 122
6.2.1 Permitting higher access levels is bad practice 122
6.2.2 Minimize use of overly permissive RACF attributes 123
6.2.3 Minimize the use of the WARNING attribute. 123
6.2.4 Minimize the use of global access checking tables. 123
6.3 Revealing permitted access levels that exceed the need 124
6.4 Minimizing access control privileges 124
6.4.1 Reporting permitted DATASET access levels exceeding the actual used access levels. 124
6.4.2 Implementing least privilege for the DATASET class 128
6.4.3 Verifying the permit commands for the DATASET class. 134
6.4.4 Executing permit commands against the active primary RACF database 138
6.4.5 Reporting permitted general resource access levels exceeding the used access levels. 142

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at https://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

CICS®	IBM Z®	System z®
Db2®	RACF®	z/OS®
IBM®	Redbooks®	
IBM Security®	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

To guard against future cyberattacks efficiently, it is important to have a simple process to predict the impact of changes made to your security definitions. Security impact forecasting can provide a streamlined workflow in which historical data is captured automatically, and an intuitive interface allows for setting up security adaptations in a way that they can be quickly analyzed for effect and when deemed correct, applied automatically.

In the context of IBM® RACF®, changes to existing access control definitions can be done in a proactive way with confidence, using IBM zSecure Admin. IBM zSecure Admin capabilities can help assess and build stronger access controls against cyberthreats, rather than just reacting to them after they happen.

In this IBM Redbooks® publication, we look at the value of analytics for security impact forecasting in reference to IBM RACF and IBM zSecure Admin (RACF-Offline and the Access Monitor functions). Use cases, best practices, and step-by-step guidance with examples are also provided.

This publication is for IT Managers and Security Architects responsible for the technology that protects their assets, and the change management staff and security administrators responsible for the safeguarding applications and data from unauthorized access.

The reader is expected to have a basic understanding of IT security concepts and the principle of least privilege in a zero trust framework.

Authors

This book was produced by a team of specialists from around the world working with IBM Redbooks, Poughkeepsie Center.

Bill White is an IBM Redbooks Project Leader and Senior IT Infrastructure Specialist at IBM Poughkeepsie, New York.

Mike Riches is a Senior Technical Support Specialist in the United Kingdom. He has 13 years of in depth experience with mainframe security software, having worked with and supported mainframe networking software prior to that. Mike has worked at IBM for nearly 23 years. His areas of expertise include zSecure, IBM Z® Security and Compliance Center, and IBM Security® Key Lifecycle Manager for z/OS®, in conjunction with knowledge of RACF, ACF2 and Top Secret as external security managers. He has previously enjoyed being an author of Redbooks, for both IBM z/OS Communications Server and IBM Communication Controller for Linux on System z®.

Elijah Swift is a Backend Software Developer in the United States of America. He has 4 years of experience with mainframe security software, having previously worked elsewhere in both mainframe software and security engineering. Elijah has been with IBM for 5 years, supporting and working with a variety of mainframe software and products. He holds a Master’s degree in electrical and computer engineering from the State University of New York at Binghamton. Elijah’s area of expertise is RACF, having worked both as a technical support specialist for RACF and as a co-lead developer of the Python interface to the RACF Command interface—an open source python library that allows users to administer their

RACF environments in python. As a part of these efforts, he has written extensively on RACF and its use.

Jeroen Tiggelman is a Senior Software Engineer in the Netherlands. He has 29 years of experience in developing zSecure, and has been the zSecure development manager since 2001. Jeroen has been working for IBM since the company acquired Consul Risk Management in 2007. He holds two Master's degrees in discrete mathematics and theoretical computer science from the University of Leiden. His areas of expertise include zSecure, RACF, and security principles. Jeroen has supervised the creation of the first zSecure Alert User Reference Manual in 2003 and has written the introduction sections to the zSecure CARLa Command Language.

Scott Woolley is a z/OS software engineer at IBM Poughkeepsie in the United States. He was a member of the RACF development team for over 20 years, focusing on the areas of security auditing, cryptography, digital certificates, and identity mapping. Scott currently performs cybersecurity testing and analysis with the z/OS Secure Engineering team.

Tom Zeehandelaar is a zSecure Software Developer and Senior Technical Enablement Specialist. He joined IBM January 2007 as part of an acquisition of Consul Risk Management where he worked as a zSecure instructor and Security Consultant since December 2000. Currently, Tom works for the IBM Security zSecure development team that is based in Delft, the Netherlands. He is responsible for maintaining and building the compliance evaluation framework that the IBM Security zSecure Audit product supports. Other responsibilities include the development, maintenance, and delivery of IBM Security zSecure training courses. In addition, Tom is occasionally involved in pre- and post zSecure sales activities, z/OS security health checks, as well as second-level zSecure client support. Prior to joining Consul Risk Management, Tom spent 13 years working for KLM (Royal Dutch Airlines) in various IT functions, eventually the last 10 years as a Security Officer for the MVS-OS/390 platform. His expertise include information security, security administration, z/OS auditing, and business continuity planning processes.

Thanks to the following people for their contributions to this project:

Hans Schoone
Chief Architect, IBM zSecure

Michael Zagorski
Program Director, IBM Z Security

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on LinkedIn:

  https://www.linkedin.com/groups/2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/subscribe

- Stay current on recent Redbooks publications with RSS Feeds:

  https://www.redbooks.ibm.com/rss.html

1

Tighten access controls with confidence

Minimizing the risk of a data breach starts with controlling who can access specific resources. This chapter presents key concepts and standard security practices for granting access to the IBM Z platform. Some fundamental access control use cases are sketched and a brief discussion of lesser-known components of IBM zSecure Admin is provided.

The following topics are covered in this chapter:

- 1.1, “IBM Z security challenges” on page 2
- 1.2, “Hardening practices” on page 2
- 1.3, “IBM Z security layers” on page 3
- 1.4, “A phased approach” on page 4
- 1.5, “Lesser-known features in IBM zSecure Admin” on page 5

1.1 IBM Z security challenges

Today, a number of trends make IBM Z security an urgent topic of attention. A vast majority of the world's critical business data is managed and stored on IBM Z. Because of this, the mainframe is becoming the biggest target for bad actors, both external and internal to the organization. While IBM Z is the most securable platform in the industry, it is important to ensure that it is properly configured and that all security related capabilities are exploited.

Where in the past organizations would trust that the mainframe was safe, today we see more and more compliance regulations put into place to prove it. As a result, it is absolutely necessary to be able to explain clearly why particular people have access to resources.

Additionally, security skills needed to manage the environment are unique and finding the right skilled personnel is a challenge.

At a minimum the following is essential for restricting access to only those users and tasks that need it:

- Obtaining relevant data representing all the required accesses
- Summarizing and consolidating this data appropriately to keep analysis manageable
- Provide security impact forecasting on the proposed security definition changes

Refer to 2.2.2, “Access Monitor environment” on page 11 and 2.3.1, “Testing RACF database cleanup with RACF-Offline” on page 15 for more information.

1.2 Hardening practices

As you evaluate your security policy and consider changes that can further secure or harden your environment, it is important to keep in mind industry standard practices for managing security policies. The practices discussed in this document are underpinned by core security principles, namely the idea of “zero trust” and one of its core components “least privilege”, as well as “role-based access control”.

1.2.1 Zero trust

In the past, security management sometimes relied on securing a system's perimeter. In recent years, the consensus has become that the right assumption to make is that the perimeter might have been compromised, so any access done by any user or process to any resource must always be verified. This is known as a “Zero Trust Architecture” as described in the National Institute of Standards and Technology (NIST) special publication SP 800-207.

In simple terms, a zero trust environment is designed to trust nothing and no one until their identity and authorization are confirmed. There are several core components to a zero trust architecture that z/OS enforces well on its own. The ideas of “continuous” and “strict” authentication where the system regularly and for each request validates a user's access are managed by the system itself with the exception of application-based settings like user authorization caching that are beyond the scope of this document.

1.2.2 The principle of least privilege

Zero trust implies that no access must be allowed before a user or process has authenticated, and access controls must only allow access that is required for the user or process to do their job. This requirement is known as the principle of least privilege.

Risks can further be mitigated by encrypting sensitive data: even if someone could get at the data store, they would not be able to use the data unless they also obtained the key to decrypt the data.

A security administrator takes a more active role in enforcing the ideas of least privilege and “no implicit trust” as a part of a zero trust environment. No implicit trust means that, as laid out in the overarching term of zero trust, no user or agent (like a started task or job) shall be assumed to have access to the environment. You will see in later chapters of this publication how to avoid or minimize generic accesses and authorities that contradict this principle. Additionally, least privilege means that accesses and authorities granted to users and agents are strictly limited to the privileges that allow the user to perform their job. You will also see in later chapters some strategies to explicitly minimize privileges and eliminate authorities that go unused in order to maintain a least privilege environment.

1.2.3 Role-based access control

Ensuring strong authentication through approaches like multi-factor authentication and data encryption schemes are relatively easy to set up. Maintaining the principle of least privilege can be far more difficult. One reason making it difficult is determining why access was granted in the past.

Role-based access control (RBAC) is an approach where access is never granted directly to a user, but rather to a particular access control group that represents the job role a user is assigned. If the user moves to another job role, they can be removed from the role group assuring that no individual user permissions continue to exist for a task that is no longer required.

Of course, access monitoring is still required, as it is also possible that a particular task is no longer performed by a certain job role.

1.3 IBM Z security layers

The security capabilities incorporated in the IBM Z stack consists of a number of layers. For example, at the hardware level encryption, secure key management, and tamper-resistant technology are available. At the operating system level; many security measures are built in and the access decisions can be routed to an external security manager, such as IBM Resource Access Control Facility (RACF). Administration of security rules normally happens in a security database package as with RACF.

1.3.1 External security managers on z/OS

Security starts at the hardware level, where pages of storage can only be read or written by processes that have the right permissions for them. This is in principle controlled by the operation system. The term “trusted computing base” is used for the overall environment that ensures secure operations.

When a application that is part of the trusted computing base is asked to perform a certain operation, it must generally verify that the user or process requesting it has the right level of access granted for the request. In z/OS these verifications are directed to the System Authorization Facility (SAF) as: Does identity “x” have access type “y” on resource “z”?. The answer can be: “Yes”, “No”, or “Not known”, after which the application decides whether to perform the operation or refuse it.

The SAF interface usually passes the request to a security package that contains the access rules. This could be IBM RACF or another External Security Manager (ESM).

1.3.2 RACF

A central part of RACF is the security database—containing security rules that define the identities and access to resources. The RACF term for a logical record in the database is “profile”.

Profiles belong to one of four kinds of entities:

- USER profiles describe the users and tasks, attributes that they might have, and means of authentication.
- GROUP profiles group users together, so the accesses to the resources can be permitted to members of a group, so that simply adding a user to a group for a kind of access they need will take care of the right scope without the need for many updates.
- DATASET profiles describe the access rules for accessing data sets.
- General RESOURCE profiles control access to non-dataset resources. These exist in many different classes for the different kinds of resources.

A user or group can be permitted with a certain access level to a dataset or resource profile that controls the protection of a set of resources. In RACF, access types are hierarchical, so someone with the access level UPDATE also has the access level READ.

It is also possible to give access to everyone. In fact, this comes in two flavors. Access given to ID(*) requires that the identity of the requestor is known, while universal access (UACC) even allows access to non-authenticated tasks. Furthermore, there is also a global access table (GAT) that can specify that a certain level of access to a resource can be allowed without consulting the database.

The overall system is quite flexible, but for managing security with confidence it is often a good idea to establish groups for clear functional purposes, and minimize giving out both global and individual accesses.

1.4 A phased approach

Later chapters discuss best practices for RACF environments that apply to the following sample activities that administrators may perform as they manage their environments. Together they provide a phased approach for tightening your security with confidence.

1.4.1 Define new security rules to strengthen security

When you know access is needed for a particular user or role to a particular resource, that is currently serviced by a more generic definition, it might make sense to define a specific profile for this access, that would satisfy the need and not allow wider access at the same time.

You can do this as a first step, to remove the particular function from the more generic rule, which might then become redundant. As a next step, the access allowed to the more generic profile can then be reduced.

When a new application or function needs to be established, this should likewise be implemented in such a way that no more access is given than required.

See Chapter 3, “Add a new set of profiles” on page 27 for details.

1.4.2 Remove unused security definitions

When you have a full representative set of security access requests in use, you can use this to identify security definitions that are not in active use. These should be removed to strengthen security.

See Chapter 4, “Remove unused security definitions” on page 55 for details.

1.4.3 Establish role-based access control

If you can group individual users into role groups, you can replace permissions for individuals to permissions managed through groups representing user roles. This will make it easier to maintain least privilege when users move to a different role. It will also make auditing and proving compliance easier.

See Chapter 5, “Convert generic and specific access to group-based access” on page 79 for details.

1.4.4 Keep the defined privileges in the environment to a minimum

After implementing the previous steps, run periodic verifications that permitted accesses are still needed.

See Chapter 6, “Minimize access control privileges” on page 121 for details.

1.5 Lesser-known features in IBM zSecure Admin

IBM zSecure Admin can boost productivity for RACF administrators. Its best-known feature is a user interface where you can type over current permissions and RACF commands to make the changes are generated for you. However, zSecure Admin has other features very pertinent for hardening your security posture.

1.5.1 zSecure Admin Access Monitor

The Access Monitor component can observe all access requests that are posed to RACF, whether those are being logged or not. This is particularly important, since logging all access requests to the System Management Facilities (SMF) might be so voluminous as to make it prohibitive to do so. Access Monitor, by contrast, will consolidate all identical requests performed during a certain interval into a single record indicating the access request and the number of times it occurred.

This allows for capturing a full year's worth of access data and still being able to use the data for analysis, giving you good confidence that you have captured all accesses that normally

occur. (Of course, you need to be aware of emergency measures that might have to be in place but are not normally used.)

1.5.2 zSecure Admin RACF-Offline

The RACF-Offline component allows you to direct standard RACF commands to a database that is not the active security database. This allows you to build a RACF command stream to adapt security definitions and set up a database with the changes without these changes immediately impacting the active security definitions.

1.5.3 The Access Monitor function in zSecure Admin

The **AM** option in the zSecure Admin interface not only lets you work with ACCESS data in order to analyze it, but also allows you to evaluate the historic accesses against a different RACF database.

If you set up your adjusted security database using RACF-Offline, you can run an analysis to see what impact your proposed security changes have. This allows you to develop confidence in the command stream being built to change your security environment.

Note that the ACCESS and RACF_ACCESS report types used by the **AM** menu are part of the CKRCARLA program, also known as the CARLa Auditing and Reporting Language (CARLa) engine. It is also possible to do this analysis using a batch job.

See Chapter 2, “IBM zSecure capabilities for RACF administrators” on page 7 for more details on IBM zSecure Admin (Access Monitor and RACF-Offline) capabilities.

2

IBM zSecure capabilities for RACF administrators

Managing the security and risk of your IBM Z stack and external security managers can sometimes seem like a daunting task. However, the IBM zSecure portfolio is equipped to help overcome that challenge—automating security administrative tasks to help increase efficiency and reduce errors, detecting internal and external threats, issuing near-real-time alerts, and monitoring for compliance.

This chapter explains how specific IBM zSecure Admin capabilities can help build confidence in making access control decisions and policy changes to RACF. It also highlights the preparation steps needed to make use of those capabilities. The following topics are covered:

- 2.1, “IBM zSecure Admin for RACF administrators” on page 8
- 2.2, “zSecure Access Monitor ” on page 10
- 2.3, “zSecure RACF-Offline” on page 14
- 2.4, “General preparation steps for the use cases” on page 16
- 2.5, “Further access monitor capabilities” on page 22

2.1 IBM zSecure Admin for RACF administrators

As shown in Figure 2-1 there are several tools offered with the IBM zSecure portfolio. In this chapter, we concentrate only on a subset of the IBM zSecure portfolio that pertain to RACF administrator tasks, namely the “IBM zSecure Admin” capabilities.

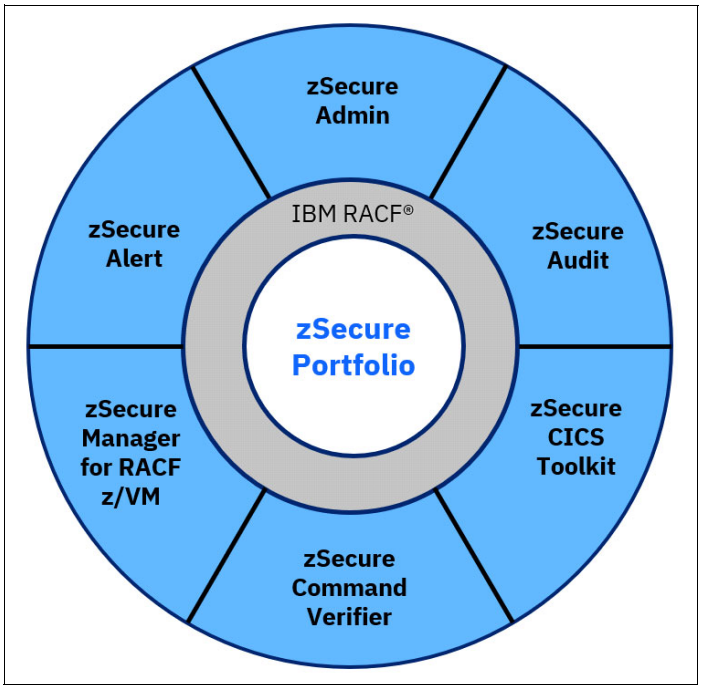


Figure 2-1 IBM zSecure portfolio overview

zSecure Admin provides a user-friendly layer on top of RACF and extends the functionality so users can enter and process administrative commands more quickly, generate custom reports, and clean up security databases. zSecure Admin enables RACF administrators to quickly identify and analyze problems in RACF to prevent mistakes before they become a threat—instilling confidence in RACF administrators. zSecure Admin also provides administrative authority in a more granular fashion so people only have the specific amount required for their job.

zSecure Admin provides many other features, for example, you can:

- Administer multiple systems with a single application interface.
- Monitor privileged users to help ensure old accounts are properly deleted.
- Compare profiles and report changes that have occurred over time, or report differences between profiles with the same name defined in different security databases.
- Copy or move users, groups, resources, applications, or entire databases between systems and rename IDs within the same database.
- Merge profiles from different databases, zSecure Admin performs extensive consistency checks and reports potential conflicts before generating commands, helping ease the burden of consolidation efforts.

In addition to increasing productivity and avoiding unintentional changes, zSecure Admin can also act as a training aid for security administrators unfamiliar with RACF, as they can learn about administration from the RACF commands that zSecure Admin generates.

zSecure Admin includes a multi-purpose program called CKRCARLA. It is a router program that calls the CARLa processing program or engine to request reports.[1]

The CKRCARLA program is used by many zSecure components to process data, including the zSecure Access Monitor task C2PACMON. C2PACMON is described in more detail in 2.2.2, “Access Monitor environment” on page 11.

zSecure Admin has a comprehensive ISPF user interface (UI) with a breadth of RACF administration functions (see Figure 2-2).

```
                          zSecure Admin - Main menu
Option===>_____________________________________________________________________
                                                                      More:     +
SE   Setup           Options and input data sets
RA   RACF            RACF Administration
  U    User            User information
  G    Group           Group information
  D    Data set        Data set profiles
  R    Resource        General resource profiles
  S    Settings        Setropts, RRSF, and class settings
  H    Helpdesk        One-panel helpdesk options
  Q    Quick admin     Quick User Administration
  1    Access          Access Check
  2    Queued          Display and action on profiles with QUEUED commands
  3    Reports         Reports with profiles and resources
  4    Mass update     Specify mass copy/recreate/delete actions
  5    RACDCERT        Work with certificates, key rings, filters and tokens
  C    Custom          Custom report
AU   Audit           Audit security and system resources
RE   Resource        Resource protection reports
AM   Access          RACF Access Monitor
CR   Command review  Review and run commands
CO   CARLa           Work with CARLa queries and libraries
IN   Information     Information and documentation
LO   Local           Locally defined options
```

Figure 2-2 zSecure Admin ISPF user interface Main menu

IBM zSecure Admin includes the following components that are the focus of this IBM Redbooks publication:

- Access Monitor, which can be used to collect and present information about actual usage of RACF resource profiles.
- RACF-Offline, which adds the ability to issue most RACF commands against an inactive RACF database to independently test changes.

[1] IBM CARLa, or CARLa Auditing and Reporting Language, is a programming language that allows users to generate audit reports and perform security administration tasks. Be aware that CARLa programming is not a skill that zSecure beginners are expected to perform. It is often seen as one of the more advanced skills that seasoned zSecure users pick up along the way when using zSecure functions and features. Alternatively, users can attend the 3-day IBM Security zSecure CARLa Audit and Reporting Language course to learn the fundaments of CARLa programming.

2.2 zSecure Access Monitor

zSecure Access Monitor was originally designed to provide the information required to allow effective RACF database cleanup. However, the data collected to provide this functionality also provides additional usage capabilities discussed in the next section.

To achieve the primary goal of effective RACF database cleanup, we need complete access information, meaning the data collected must represent all RACF access request events. SMF could be a possible data source, but it would be incomplete because RACF only writes SMF records where auditing events are specifically enabled. It is possible to enable auditing at the RACF class level, but then the SMF data would be too big. The data also needs to cover a suitable period of time; the more data collected, the higher level of confidence you can have in making the right decision. Of course, there will also be many duplicate access events over time, so the design of Access Monitor needed to cope with that, and this is achieved by the consolidation process. In summary, RACF access event data is collected by Access Monitor in addition to, and not instead of, SMF records.

More recently zSecure Access Monitor was enhanced to capture information about many z/OS UNIX file system events. The information is captured at the full path level, and not at the directory level. You can also use specific access events, the RACF VERIFY (RACINIT), for alerting through zSecure Alert, for example, alert 1122 when a site-defined sensitive user ID logs on.

These functions are not discussed here, however, further details can be found in the zSecure product documentation: https://www.ibm.com/docs/en/szs

2.2.1 Possible uses for zSecure Access Monitor

Some of the possible usage examples of zSecure Access Monitor, include:

- RACF database cleanup
- Access restructure
- Perform ad-hoc queries to evaluate profiles and access

RACF database cleanup

The access event information collected over time allows zSecure Admin to generate commands to clean up unused or obsolete definitions from your RACF database.

This includes:

- Access Control Lists – remove unused permits on access control lists (ACLs)
- Connects – remove unused group connects
- Profiles – remove unused profiles.
- RACF administrators and analysts can also use Access Monitor data to test resource profiles and access rights by running simulations against a candidate RACF database. The candidate database can be one prepared using zSecure Admin RACF-Offline, or a database on another z/OS system where you intend to host production processes.

Access restructure

The access event information collected over time allows you to restructure access definitions with the goal of tightening security by ensuring you follow the security principles of “least-privilege” access and “zero-trust”.

Examples of restructuring access include:

- Convert use of UACC to access via group
- Convert use of ID(*) to access via group
- Convert OPERATIONS access to access via ACL
- Split batch applications using a single user ID to use multiple user IDs based on job names.

Perform ad-hoc queries to evaluate profiles and access

Using the Access Monitor function, you can monitor access events and collect the data for reporting and analysis. From the reports, you can view and analyze the resource profile and access usage.

2.2.2 Access Monitor environment

The Access Monitor environment requires several building blocks to be in place to collect and record the access event data required to perform a RACF database cleanup and generate reports. Figure 2-3 depicts an overview of components that make up the Access Monitor environment.

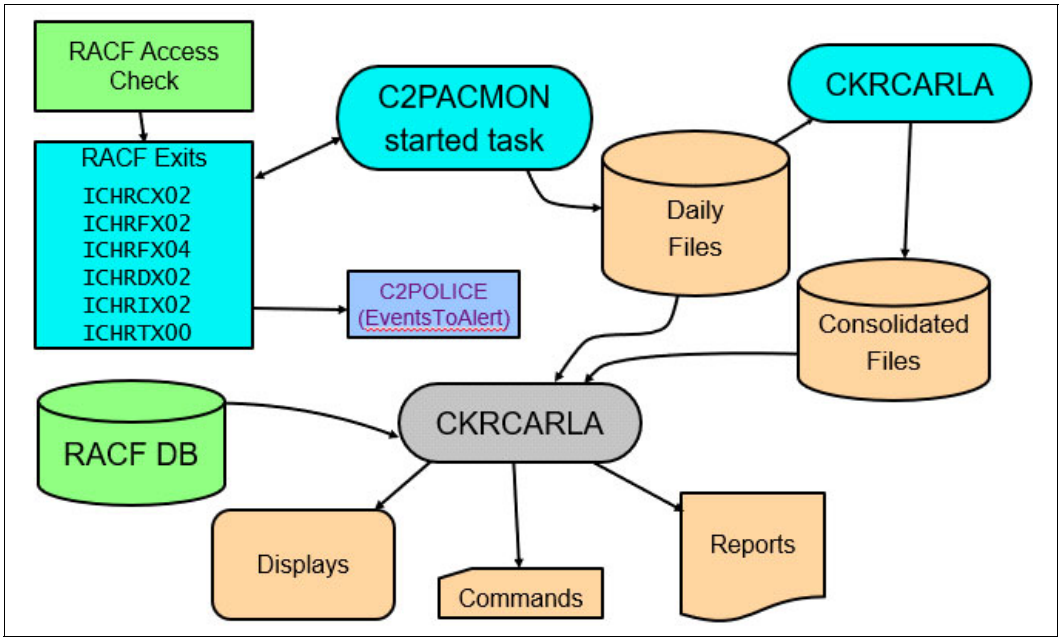


Figure 2-3 Schematic overview of Access Monitor environment

The access event data is collected by dynamically loaded RACF post-processing exits. The exit code is executed in the user address space where the RACF access requests occur. You generally do not need to concern yourself with the management of these exits unless your maintenance levels or release of zSecure changes.

The Access Monitor started task, which by default is called C2PACMON, must be started on each LPAR to record the RACF access event request data. C2PACMON dynamically loads the required RACF post-processing exits when it starts and removes them when it stops. The data collected by the RACF exits is written from in-memory buffer space by C2PACMON, using the CKRCARLA program, which combines the collected records and writes them to an LPAR specific daily collection data set. Each type of access event is saved in a corresponding

access record, representing the access from a user to a resource. Once per day the data from the collection data set, or data sets, is automatically consolidated into a single, LPAR specific, daily consolidation data set. The data set attributes for the daily collection and daily consolidation data sets are configured during the implementation of Access Monitor.

Batch jobs can be used to consolidate these daily consolidation data sets into weekly, monthly, quarterly, or yearly summaries. Any of these data sets can be used as input to zSecure Admin with an input data set type of ACCESS. In consolidated ACCESS data sets, only one record of information is stored per unique set of values. When the same user ID accesses the same resource in the same class with the same access level 500 times on the same system, these 500 access events are stored as one record in the ACCESS data set. This consolidated record then contains the time stamp of the last occurrence, system, user ID, resource, class name, access level, and a counter that indicates that this access occurred 500 times. This Access Monitor-specific consolidation process is the reason why the size of ACCESS data sets is manageable.

Users process the consolidated access data by running ad-hoc queries to evaluate the profile usage and access data. Processing can be set up and run interactively using the options available on the Access Monitor menu in the zSecure Admin UI. The access data is processed using CKRCARLA. Two CARLa report types are available for this purpose. The ACCESS report type uses the Access Monitor records to report about the collected events. The RACF_ACCESS report type shows profiles in the RACF database and annotates these profiles with usage data from the Access Monitor records.

zSecure Admin ISPF user interface

The zSecure Admin ISPF user interface (UI) provides many Access Monitor menu options to both report on access event data, and to automate the generation of cleanup commands. The AM – RACF Access Monitor menu options are shown in Figure 2-4.

```
                          zSecure Admin - Access
Option===>_____________________________________________________________________

1      Access          Access summary by user or profile
2      Compare         Compare monitored access against current RACF database
3      Permit usage    Permit usage information for current RACF database
4      Connect usage   Connect usage information for current RACF database
5      Profile usage   Profile usage information for current RACF database
6      Member usage    Member usage information for current RACF database
7      Global usage    Global usage summary for current RACF database
8      Remove          Remove unused profiles and authorizations
9      Cleanup         Remove permits, dataset and general resource profiles
V      ID verify       ID verification (logon/start/job-start) summary
I      ID usage        ID usage information for current RACF database
U      Unix events     Access summary for unix events by user, uid/gid, or path
```

Figure 2-4 Access Monitor menu options

Menu option **AM.1** allows selecting and reporting on historic access events from the allocated access monitor records, shown Figure 2-5 on page 13.

```
                        zSecure Admin - Access - Access
Command ===> ____________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . ________  (class or EGN mask)
SAF resource name . . . ________________________________________
RACF match on . . . . . ________________________________________

Advanced selection criteria
_  Further selection  _  Date selection   _  Current RACF DB selection

Output/run options
1  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  _  Show simulated fields   _  Timezone (U/L/H)
_  Print format             Customize title           Send as e-mail
      Background run         Full page form
```

Figure 2-5 AM.1 menu option

In the “Output/run options” section, you can specify how you prefer the layout and content to be produced. zSecure Admin supports four different summary layouts.

- Summary by userid – Produces an access overview of all historic access events and decisions that were collected in the allocated access monitor records summarized by the user ID involved in the access decision. If “Show simulated fields” is also selected, the detailed output will include a section titled “Current RACF database effect” showing the authority path simulated from the currently allocated RACF database.
- Summary by member class and profile - Produces an access overview of all historic access events and decisions that were collected in the allocated access monitor records summarized by resource class and profiles. If “Show simulated fields” is also selected, the detailed output will include a section titled “Current RACF database effect” showing the authority path simulated from the currently allocated RACF database.
- Summary by simulated authorization used – Produces an access overview of all historic access events and decisions that were collected in the allocated access monitor records summarized by the simulated authority that would allow or disallow that access based on the allocated RACF database source. The access event records do not contain all the possible authority paths because that information is not available from the RACF post-processing exits used by Access Monitor to collect the data. This simulation summary determines the authority path from the currently allocated RACF database and summarizes the output by the simulated authorization method.
- Summary by simulated groups used for access - Produces an access overview of all historic access events and decisions that were collected in the allocated access monitor records summarized by the group connection(s) that would allow or disallow that access based on the allocated RACF database source. The access event records do not contain the group connect information because that information is not available from the RACF post-processing exits used by Access Monitor to collect the data. This simulation summary determines the group connects from the currently allocated RACF database and summarizes the output by the simulated groups used for access.

The detailed examples of converting access shown in the subsequent chapters use different summary options best suited to the individual use case.

Figure 2-6 shows an example of reporting on the summarized access for user ID CRMBMJ2, using menu option **AM.1**.

```
               IBM Security zSecure ACCESS summary                Line 1 of 18
Command ===> ____________________________________________________ Scroll===> CSR_
Access monitor records for Userids like CRMBMJ2
   Occurrence Userid   Name                      First occurrenc Last occurrence
        41911 CRMBMJ2  FRED SMITH                27Oct2023 05:45 27Sep2024 10:39
   Occurrence Intent    Type   RetAll AccRC
         3476 READ      Auth              0
          491 READ      Auth              4
           48 READ      Auth              8
          198 READ      Fast
           28 READ      Fast              0
           13 UPDATE    Auth              0
           12 UPDATE    Auth              8
            7 DEFDELETE Define            0
            7 ALTER     Auth              0
            2 ALTER     Auth              8
        37629 ALTER     Auth   RetAll     8
```

Figure 2-6 Access Monitor Access Summary overview report

As previously mentioned, the Access Monitor menu options can be used to generate commands to perform a cleanup of unused accesses or profiles from the RACF database. This process is described in more detail in subsequent chapters.

Detailed information on how to implement zSecure Access Monitor can be found in the zSecure documentation manuals, Installation and Deployment Guide, and zSecure Admin and Audit for RACF User Reference Manual. The zSecure product documentation can be found here: https://www.ibm.com/docs/en/szs

2.3 zSecure RACF-Offline

RACF-Offline is a component of zSecure Admin that allows you to execute and test RACF commands on a RACF database that is not active in the system. Using this function allows you to test changes to RACF definitions without impacting any other software running on the system and without using a dedicated test system.

Using RACF-Offline, you can direct all RACF commands to an offline, or inactive, RACF database. RACF access verifications are not affected and are still performed against the system's active primary RACF database. Thus, the RACF-Offline environment applies only to RACF commands used to make changes to the Offline RACF database's security definitions.

The offline RACF database can be used as an input selection within the zSecure Admin UI and zSecure Admin can generate the RACF commands required to make the desired changes. Alternatively, RACF-Offline can be run in a batch job to process the RACF commands specified as input.

Detailed information on how to implement zSecure RACF-Offline can be found in the zSecure documentation manuals, Installation and Deployment Guide, and zSecure Admin and Audit

for RACF User Reference Manual. The zSecure manuals can be found here:
https://www.ibm.com/docs/en/szs

2.3.1 Testing RACF database cleanup with RACF-Offline

You can use Access Monitor, together with RACF-Offline, to generate commands to clean up your RACF database, and then execute those commands against an offline RACF database. You can then run a simulation of historic access events against the updated offline RACF database to test the results of the cleanup.

Figure 2-7 depicts a high-level overview of the components that make up the Access Monitor and RACF-Offline testing environment.

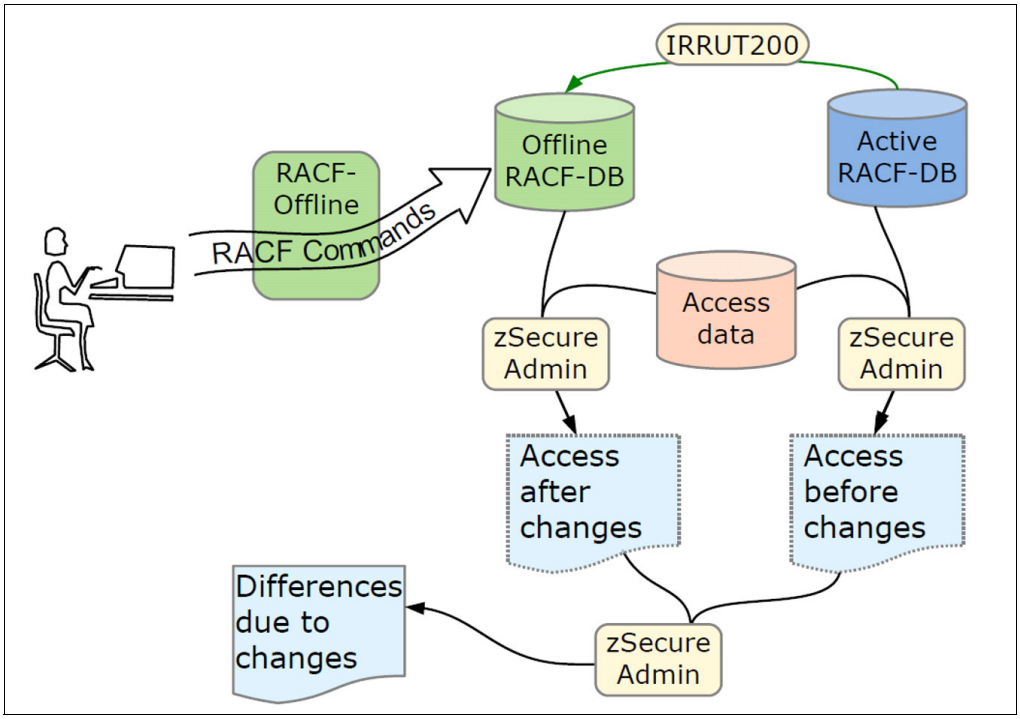


Figure 2-7 Schematic overview of Access Monitor & RACF-Offline testing environment

The following steps demonstrate how cleaning up unused definitions in the FACILITY class can be done:

1. A copy of the pertinent RACF database has been created and is ready to be used for a RACF-Offline session.
2. Start RACF-Offline with appropriate parameters.
3. Start zSecure Admin and select an input set that contains ACCESS, CKFREEZE, and RACF sources.
4. Analyze the profile and permit usage of the FACILITY class.
5. Generate and run commands to cleanup unused FACILITY class profiles and permits.
6. End the RACF-Offline session.
7. Start a zSecure Admin session.
8. Use the Access Monitor compare option to analyze the effects of cleanup commands.

9. When all is OK, you can decide to run the cleanup commands against the live RACF database.

This process will be described in more detail in the subsequent chapters.

2.4 General preparation steps for the use cases

The use cases presented in subsequent chapters will require the preparation steps outlined in this section. The steps are detailed here as a central reference, they include:

- Creating a RACF-Offline database
- Starting a RACF-Offline session
- Logging on to RACF-Offline session
- Resetting the RACF-Offline log data set
- Starting and preparing your zSecure Admin session

2.4.1 Creating a RACF-Offline database

When you intend to test running RACF cleanup commands against a RACF-Offline database, a good start is to copy the RACF primary or active backup database to a data set that you can use as offline database. RACF supports the IRRUT200 utility that can be used to create a backup of the primary RACF database. Figure 2-8 is sample batch job to create a RACF-Offline database that can be used for cleaning up unused security definitions.

```
//Add a valid job card for your environment here!
//*-------------------------------------------------------------------*/
//COPY    EXEC PGM=IRRUT200
//SYSRACF  DD DISP=SHR,DSN=<name of RACF primary/backup database>
//SYSUT1   DD DSN=<name of RACF offline database>,
//             DISP=SHR
//*            DCB=(LRECL=4096,RECFM=F),
//*            SPACE=(CYL,(300)),DISP=(NEW,CATLG),
//*            UNIT=3390
//SYSUT2    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
```

Figure 2-8 Sample batch job to create RACF-Offline database

You need to customize this job with the following definitions:

- Add a job card that is valid to run in your system environment.
- Supply the data set name of the RACF primary or active backup database that you want to copy to the RACF-Offline database.
- Specify the data set name of the RACF-Offline database.

If you need to allocate a new dataset for the RACF-Offline database, comment out DISP=SHR and uncomment the dataset attribute lines which are shown as comments. Adjust the SPACE specification if required. You can verify the space that the primary RACF database uses and specify the same space allocation for your RACF-Offline database.

When you run clean ups regularly, it might be a good idea to schedule a periodic batch job that creates a fresh copy of the primary RACF database in a time slot when the system is not heavily used.

2.4.2 Starting a RACF-Offline session

To run the cleanup commands against the offline RACF database that you just created or refreshed in 2.4.1, “Creating a RACF-Offline database” on page 16, you have the choice to use RACF-Offline in batch or interactively. In this IBM Redbooks publication, the interactive use of TSO commands with RACF-Offline is documented.

Important: To allow RACF-Offline to run as a TSO command, the TSO authorized commands list must be updated. If this has not been done you will see message IKJ56500I COMMAND B8RACF NOT FOUND. The optional steps to update the TSO authorized commands list is required for the examples shown in the subsequent chapters.

When RACF-Offline is installed, you can use the default options module named ‘B8ROPT’ to configure the default RACF-Offline database, log data sets, and SMF processing options. You use command **B8RACF** from the TSO ready prompt to activate RACF-Offline. Pressing Enter starts the RACF-Offline session using the options from module B8ROPT as configured (see Figure 2-9).

```
 READY
b8racf
 B8R100I B8RACF version 3.1.0
 B8R274I RACF DB to be used is CRMB.T.RACF.OFFLINE
 B8R268I LOG data set to be used is CRMB.T.RACF.OFFLINE.B8RLOG
 B8R304I New SMF-ID:$B8R
 B8R121I Completed processing B8ROPT options module
 B8R200A Enter RACF Command or "END"
```

Figure 2-9 RACF-Offline started with default configuration

The various B8RxxxI informational messages report the configured offline RACF database name, the name of the log data set, and SMF-ID that is to be used for the SMF records generated during your RACF-Offline session. The purpose of the LOG data set is to collect the RACF commands that you execute during your RACF-Offline session. Note that a different SMF-ID is configured to distinguish SMF records written during your RACF-Offline session from the SMF records of the ‘live’ system to avoid confusion between actual and RACF-Offline triggered SMF records.

When you are OK with the predefined configuration, you can continue with the logon command for RACF-Offline, see 2.4.3, “Logging on to RACF-Offline session” on page 19.

Steps to override default configuration settings for RACF-Offline

If you prefer to work with a different RACF-Offline configuration, you can override the default RACF-Offline settings that are configured by the systems programmer using either of the optional steps that follow:

- **Use prepared B8RPARM member**

  You can prepare a member that contains the `RACFDB`, `LOGDS`, and `SMF ID` parameters and the values that you prefer to use during your RACF-Offline session. Figure 2-10 on page 18 shows what that member might look like.

```
VIEW        CRMB.T.ZSECURE.TESTLIB(B8RPARM) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ****************************
001200 RACFDB 'CRMB.T.RACF.OFFLINE'
001210 LOGDS  'CRMB.T.RACF.OFFLINE.B8RLOG'
****** **************************** Bottom of Data ***************************
```

Figure 2-10 Sample B8RPARM member configuration

Use the command shown Figure 2-11 in to allocate your prepared RACF-Offline parameter member before you start a RACF-Offline session.

```
 READY
alloc fi(b8rparm) da('crmb.t.zsecure.testlib(b8rparm)')
 READY
b8racf
 B8R100I B8RACF version 3.1.0
 B8R274I RACF DB to be used is CRMB.T.RACF.OFFLINE
 B8R268I LOG data set to be used is CRMB.T.RACF.OFFLINE.B8RLOG
 B8R304I New SMF-ID:$B8R
 B8R121I Completed processing B8ROPT options module
 RACFDB 'CRMB.T.RACF.OFFLINE'
 B8R274I RACF DB to be used is CRMB.T.RACF.OFFLINE
 LOGDS  'CRMB.T.RACF.OFFLINE.B8RLOG'
 B8R268I LOG data set to be used is CRMB.T.RACF.OFFLINE.B8RLOG
 B8R143I Completed processing B8RPARM file
 B8R200A Enter RACF Command or "END"
```

Figure 2-11 Start RACF-Offline with parameters configured in B8RPARM member

- **Interactively and dynamically configure RACF-Offline configuration settings**

  If you prefer, you can also specify the `RACFDB`, `LOGDS`, and `SMF ID` configurations parameters interactively in your RACF-Offline session. In that case, you must allocate file B8RPARM dynamically to your session before you start RACF-Offline (see Figure 2-12 on page 19).

```
 READY
alloc fi(b8rparm) da(*)
 READY
b8racf
 B8R100I B8RACF version 3.1.0
 B8R274I RACF DB to be used is CRMB.T.RACF.OFFLINE
 B8R268I LOG data set to be used is CRMB.T.RACF.OFFLINE.B8RLOG
 B8R304I New SMF-ID:$B8R
 B8R121I Completed processing B8ROPT options module
racfdb 'crmb.t.racf.offline.other'
 RACFDB 'CRMB.T.RACF.OFFLINE.OTHER'
 B8R274I RACF DB to be used is CRMB.T.RACF.OFFLINE.OTHER
smf id($TST)
 SMF ID($TST)
 B8R304I New SMF-ID:$TST
end
 END
 B8R143I Completed processing B8RPARM file
 B8R200A Enter RACF Command or "END"
```

Figure 2-12 Specify RACF-Offline configuration parameters to use dynamically

After you start RACF-Offline, you can then specify the RACF-Offline parameters interactively provided that the RACFDB and LOGDS that you specify are predefined for this purpose and you have UPDATE access to them.

In general, it is probably most convenient that the RACF-Offline configuration that was prepared by the system programmer matches your requirements. When you are satisfied with the configuration of your RACF-Offline session, the next step is to logon to your RACF-Offline session.

2.4.3 Logging on to RACF-Offline session

When you want to use the same user ID for your RACF-Offline session as you use on the ‘live’ system, you can specify command `logon *` and press Enter to logon to RACF-Offline (see Figure 2-13).

```
READY
b8racf
 B8R100I B8RACF version 3.1.0
 B8R274I RACF DB to be used is CRMB.T.RACF.OFFLINE
 B8R268I LOG data set to be used is CRMB.T.RACF.OFFLINE.B8RLOG
 B8R304I New SMF-ID:$B8R
 B8R121I Completed processing B8ROPT options module
 B8R200A Enter RACF Command or "END"
logon *
 B8R251I CRMBTZ1 logged on
 B8R200A Enter RACF Command or "END"
```

Figure 2-13 Logon to RACF-Offline

That command logs you on with the same user ID that you already use. Optionally, if your user ID is not assigned the SPECIAL attribute, you can add the SPECIAL keyword to the

`logon *` command to assign the SPECIAL attribute to your user ID for the configured RACF-Offline database.

2.4.4 Resetting the RACF-Offline log data set

Before executing RACF commands in your RACF-Offline session, it is suggested to empty the associated log data set to prevent that RACF commands from previous RACF-Offline sessions are still stored in the log data set (see Figure 2-14).

```
b8racflg reset
 B8R277I RACF LOG file reset
 B8R200A Enter RACF Command or "END"
```

Figure 2-14 Clear the log data set that is configured in RACF-Offline

Next, you can issue command **ispf** and press Enter to start an ISPF session within your RACF-Offline session. That command results in the regular ISPF Primary Option Menu being displayed (see Figure 2-15).

```
                          ISPF Primary Option Menu
 Option ===> ____________________________________________________________________

 0  Settings      Terminal and user parameters           User ID . : CRMBTZ1
 1  View          Display source data or listings        Time. . . : 11:28
 2  Edit          Create or change source data           Terminal. : 3278
 3  Utilities     Perform utility functions              Screen. . : 1
 4  Foreground    Interactive language processing        Language. : ENGLISH
 5  Batch         Submit job for language processing     Appl ID . : ISR
 6  Command       Enter TSO or Workstation commands      TSO logon : TSOZSEC
 7  Dialog Test   Perform dialog testing                 TSO prefix: CRMBTZ1
 9  IBM Products  IBM program development products       System ID : NMPIPL87
 10 SCLM          SW Configuration Library Manager       MVS acct. : ACCT#
 11 Workplace     ISPF Object/Action Workplace           Release . : ISPF 8.1
 P  PRODUCTS      Dialogs for installed products
 G  GROUP         Dialogs used with your organization
 U  User          Your Own Dialogs
 S  SDSF          System Display and Search Facility

 C  Consul        Consul Risk Management applications
```

Figure 2-15 ISPF started within RACF-Offline session

2.4.5 Starting and preparing your zSecure Admin session

How to start zSecure Admin depends on how your organization has installed and configured it. Some customers configure a zSecure menu option for users to select whereas others prefer to use the supplied REXX exec, using, `tso ckr` from the command line to start zSecure. Start zSecure Admin using your regular method to start zSecure Admin in your system.

In all the examples in this Redbook we used the following confirmation options to ensure that commands are executed on the local system after confirmation. Select option **SE.4**, for Setup Confirm, to verify that you use the same options (see Figure 2-16 on page 21).

```
                        zSecure Admin - Setup - Confirm
Command ===> ____________________________________________________________

Action on command . . 2  1. Queue  2. Execute  3. Not allowed
                         Execute display commands (for option 1 only)
Confirmation  . . . . 5  1. None   2. Deletes  3. Passwords  4. All  5. Add
Command Routing . . . 2  1. Ask    2. Normal   3. Local only
```

Figure 2-16 zSecure Setup Confirm options

To generate RACF cleanup commands for security definitions that are not used in access decisions, you need to allocate a RACF input source (primary, active backup, unload, or RACF copy), CKFREEZE data set of the target system, and an ACCESS data set with preferably 1 year of historic consolidated access records.[2]

Select option **SE.1**, for Setup Input files, to define a new input set or select an already defined input set to use during your zSecure session. If you are not familiar with the naming convention of RACF, CKFREEZE, and ACCESS data sets on your system, you might need to consult a systems programmer to retrieve this information before you can set up an appropriate input set.

Figure 2-17 shows what the content of your zSecure input set might look like.

```
                        zSecure Admin - Setup - Input files        Row 1 of 3
Command ===> ____________________________________________________ Scroll ===> CSR

Description . . . . Recent RACF, CKFREEZE, ACCESS data sets_____
Complex . . . . . . TVT6003__  Version  . . . .  ____

Enter data set names and types.         Type END or press F3 when complete.
Enter dsname with .* to get a list      Type SAVE to save set, CANCEL to quit.
Valid line commands: E L I R D          Type REFRESH to submit unload  job.

   Data set name or DSNPREF=, or Unix file name           Type or ?   NJE node
_  'CRMB.T.RACF.OFFLINE'_____________________________     COPY.RACF__ ________
_  'CRMB.T.CKFREEZE'_________________________________     CKFREEZE___ ________
_  'C2PACMON.TVT6003.Y12MON'_________________________     ACCESS_____ ________
****************************** Bottom of data ********************************
```

Figure 2-17 zSecure input set for cleanup purposes

This sample input set contains an offline RACF database, a CKFREEZE, and a 12-month rolling ACCESS data set. The ACCESS data set contains consolidated access records of the last 12 months. For example, when today is October 25th, 2024, this ACCESS dataset contains the collected and consolidated access records from September 2023 to September 2024.

[2] When your RACF database is shared with more systems, you need to allocate CKFREEZE and ACCESS data set of each system that shares the RACF database that is targeted for the cleanup.

Important: When making decisions about cleaning up unused definitions in your RACF database, it is vital that all processing activities are considered. This includes those activities that occur only once a year, such as batch jobs that produce financial reports about last year's performance. Thus, collecting and using historical access event data of at least *one year* increases the confidence that your clean-up efforts will not disrupt access paths that are infrequently used.

Select this input set to use in your zSecure session with action command **S**. The selected input set must now be marked with label "**selected**" (see Figure 2-18 on page 22).

```
                          zSecure Admin - Setup - Input files     Row 1 from 18
Command ===> ____________________________________________________ Scroll ===> CSR_

(Un)select (U/S/C/M) set of input files or work with a set (B, E, R, I, D or F)

   Description                                                    Complex
_  Recent RACF, CKFREEZE, ACCESS data sets                        TVT6003  selected
_  Active primary RACF data base
_  Active primary RACF data base and live SMF data sets
_  Active backup RACF data base
_  Active backup RACF data base and live SMF data sets
_  Active backup ACF2 data base and live SMF data sets
_  Live settings
```

Figure 2-18 Select the input set to use for cleanup

2.5 Further access monitor capabilities

While not specifically described in further detail in this document, zSecure Access Monitor has many additional selection criteria and reporting functions that can help you in your journey towards the principles of "zero trust" and "least privilege".

For example, before performing intelligent cleanup guided by Access Monitor data, zSecure Admin option **AM.9** – Cleanup, option **1** will perform a scan of your RACF database and generate commands to remove redundant permits for user IDs which already have access via a group connect. Using option **2**, commands to remove redundant dataset profiles can be generated.

After the redundant definitions are removed using the static analysis, Access Monitor data can be used to perform dynamic analysis based on real access usage.

Another example relates to zSecure Admin option **AM.7** – Global usage and **AM.1** – Access with further selection option "Use of global access checking table". These reports allow you to check that access granted via the RACF Global Access Checking table is not excessive, and that only datasets and resources that are truly "public", and have a need to be accessed frequently, are defined to allow the performance gain from bypassing full access checking.

Figure 2-19 on page 23 shows the "Further selection" filtering options that are available when selected as "Advanced selection criteria" from menu option **AM.1** and **AM.2**. These allow you to refine your reports based on what the access event records show in relation to the historical access requests.

```
                  zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________
All access monitor records
Specify further selection criteria:
Jobname . . . . . _________  (jobname or EGN mask)
Port Of Entry class . . _________  (class or EGN mask)
Port Of Entry . . . . . ________________  (POE or EGN mask
Select access records(Y/N/blank)
_  Use of RACF commands                  _  Retrieval of access allowed
_  Use of global access checking table   _  Bypass JESSPOOL profiles
_  Use of discrete profiles              _  ID was undefined during event
_  System special authority used         _  User had special attribute
_  Operations authority used             _  User had operations attribute
_  Installation exit used                _  User had (ro)auditor attribute
_  User requesting access is owner

Resource action  Intended access    Result             Program status
_  Define        __  _  1. Read     _  Success         _  Defined program
_  Delete           _  2. Update   _  No profile      _  Controlled program
_  Addvol           _  3. Control  _  Not authorized  _  Specific program
_  Chgvol           _  4. Alter    _  Other           _  Controlled library
```

Figure 2-19 Further selection menu

Note that the “Specify further selection criteria” section, allowing additional filtering by Jobname and Port Of Entry, is only seen when “Show configured fields” is also selected from menu option **AM.1** and **AM.2**. Jobname and Port Of Entry information is not collected by default because it makes the consolidation of access records less effective, so configuring C2PACMON to collect this data must be done selectively for specific purposes.

Figure 2-20 on page 24 shows the “Current RACF DB selection” filtering options that are available when selected as “Advanced selection criteria” from menu option **AM.1** and **AM.2**. These allow you to select events based on the simulation of the event using the current RACF source.

```
                    zSecure Admin - Access - Current Database
Command ===> ____________________________________________________________

Specify simulated fields (Current DB effect) selection criteria:
Group(s) used for access  _____________________________  (group(s) or filter)
Essential group(s)  . . . _                              (Y/N/blank)
Not using group(s)  . . . _____________________________  (group(s) or filter)
Profile owned by  . . . . ________                       (id or filter)
RACF return code  . . . . __  ______  (operator: > >= < <= = <> ^=)

Authority used in current DB
_  APF          _  GRP_OPER     /  ID_USER      _  PRIVTRUS     _  SYS_OPER
_  CLAUTH       _  GRP_SPEC     _  INACTIVE     _  PROTFAIL     _  SYS_SPEC
_  CREATE       _  GRP_UACC     _  NO_CDT       _  QUALOWN      _  UACC
_  DFLTRC       _  ID_GROUP     _  NO_PROF      _  RESTRICTED   _  UNPROT
_  GLOBAL       _  ID_STAR      _  NOTHING      _  SPOOLRCVR    _  WARNING

User attributes in current DB (Y/N/blank)
_  ID present           _  ID Revoked
_  ID has special       _  ID has operations    _  ID has (ro)auditor
```

Figure 2-20 Current RACF DB simulation section menu

ID_USER is selected, which produces a report showing where access was granted because a user ID is permitted on the Access Control List (ACL) of a profile protecting a resource. This report could be used as a starting point to ascertain where in the journey to implement Role Based Access Control (RBAC) you are, where access should be granted by group based on the user's role.

zSecure Access Monitor can also help to identify user IDs which are not used for the period of time covered by the access event data allocated as input. Menu option **AM.V** – ID Verify reports on actual RACROUTE REQUEST=VERIFY (also known as RACINIT) events. Using the "Further selection" criteria shown in the following screen capture shows all RACF system special user IDs, which actually logged on (see Figure 2-21 on page 25).

```
                    zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________
All access monitor records
Specify further selection criteria:
Application name  . . . ________  (application name or EGN mask)




Select authentication method              Other flags(Y/N/blank)
_  Started        _  None                  _  Priv/Trust assigned
_  Password       _  MultiFactor           _  Password changed
_  Passphrase     _  IDToken               _  Passphrase changed
_  Passticket     _  Omitted               _  Passticket replay
                                           _  Undefined user
                                           _  Group-only verify

User attributes(Y/N/blank)
/  User has special attribute             _  User has operations attribute
_  User has (ro)auditor attribute
```

Figure 2-21 AM.V Further selection menu

The resulting report can then be checked against all user IDs defined with the system special attribute using menu option **RA.U**, selecting “Attributes” for the “Additional selection criteria” and selecting “Systemwide and group authorizations” of “Special”.

Following additional verification, any user IDs seen in the **RA.U** report which were not shown to be used by the **AM.V** report could potentially be removed.

These are just a few examples to demonstrate some of the additional functions that are available with zSecure Access Monitor. This document contains only a small number of use cases that are detailed in later chapters. But when you get more familiar with the value proposition of zSecure Admin, you will discover many more practical ways you could benefit from using other supported filters to fuel your potential future cleanup efforts. To learn more, review the zSecure documentation here: https://www.ibm.com/docs/en/szs

3

Add a new set of profiles

Often the role of a security administrator involves configuring their environment for new applications or expanding the security environment to better manage existing applications as features or security requirements change over time.

Additionally, security administrators may make changes to the environment's protections for data sets. Managing these changes and anticipating their impact on a security environment can be a challenging aspect to this process.

In this chapter, we discuss the following topics:

- 3.1, "Challenges with adding new resource profiles " on page 28
- 3.2, "Steering clear of default security settings" on page 28
- 3.3, "Temporarily allowing access for new resources or applications" on page 28
- 3.4, "Adding new profiles to administer more security checking" on page 29

3.1 Challenges with adding new resource profiles

Security administrators can be tasked with defining new resource profiles to RACF for various reasons. The definition of a new profile is often related to defining or controlling the use of a new application or new features for an application. Additionally, profiles are used to protect resources in RACF environments. When administrators are assigned such changes, they are challenged by the following factors:

- While targeting resource profiles that govern an application's new features or requirements for the application, the administrator faces uncertainty surrounding the existing use of the application and its resource profiles to minimize the security impact of these security definition changes on production workloads.
- When defining a set of resource profiles for a new application, it can be crucial to verify that these resource profiles do not have a security impact on existing applications or functions in the environment. When defining new dataset profiles, an administrator can struggle with evaluating these changes against the existing environment to ensure that required access is retained to key data and no additional access is granted. Leaving default or generic uses can allow for unintended use of key applications.

3.2 Steering clear of default security settings

RACF can only secure subsystems, applications, and resources that are configured to consult RACF for authentication and access verification. When consulted, RACF verifies the authentication and access verification requests by looking for RACF profile definitions to govern access to subsystems, application, or resources. Many applications and third party vendor products document how you can configure RACF or other security products to regulate access to the various features. It documents how you can enable certain features, disable certain features, or allow for any security checking and logging at all. Before installing or updating an application, it is a best practice to ensure that the environment is set up appropriately to secure this application.

When applications or their features are left unsecured by RACF or other products, they may hinge on defaults which allow for their use, but do not process security checking or logging. Such an implementation might allow for unintended and unsecured use of subsystems, applications, and resources through these default or generic permissions. Such a configuration is not acceptable in a "zero trust" environment, where no user should be trusted to access any resource or application feature by default. Additionally, any implementation of "least privilege" should not allow users whose job role does not require the use of an application or its features to allow this access.

3.3 Temporarily allowing access for new resources or applications

Most security administrators probably have familiarity with both the intended application profiles and their current security environment. This allows an administrator to establish new generic or backstop resource profiles with broad access control lists for new applications to function. These can be configured with auditing features to allow the administrator to later trim down the accesses granted to such profiles. Additionally, an administrator could leverage activation of the WARNING mode for a new profile to temporary permit ALTER access to non-permitted users to the resources that a new resource profile protects and cut audit

records that security administrators can use to grant more appropriate access later. Using WARNING mode allows all non-permitted users to access a given resource or application function with ALTER access, which still allows for unintended use of resources or applications.

With zSecure Access Monitor, you can review historic access events to determine resources or data sets that are currently not controlled by RACF profiles. Additionally, while Access Monitor can automatically generate commands to create access control lists for new profiles based on historic access to the resources. This is not the best approach because Access Monitor records bases such an access list on individual user access attempts. It would be more meaningful and secure for you to review the users in this report and define accesses to them through either existing or new groups that represent job roles. Lastly, with RACF-Offline through zSecure, you can validate that your new definitions and access control lists do not introduce overly permissive or unexpected access failures before applying them in your production environment.

3.4 Adding new profiles to administer more security checking

The details for the general use case preparation steps using automated (semi-) conversion and cleanup features in zSecure Admin are documented in 2.4, “General preparation steps for the use cases” on page 16, including these steps:

- Creating a RACF-Offline database
- Starting a RACF-Offline session
- Logging on to RACF-Offline session
- Resetting the RACF-Offline log data set
- Starting and preparing your zSecure Admin session

After completion of these steps, you can continue with the definition of new resource profiles, as follows:

- Analyzing statistics and discovering missing RACF resource profiles
- Defining a new RACF resource profile in the RACF-Offline database
- Conducting simulation of Access Monitor data against the offline RACF database

3.4.1 Analyzing statistics and discovering missing RACF resource profiles

In this section, you will learn how to analyze Access Monitor statistics and discover missing RACF resource profiles.

Suppose that your auditors demand that access to all TSOAUTH resources must be controlled with a RACF resource profile, and it is your job to implement appropriate protection of the TSOAUTH resources used in your company.

A good starting point for this activity is to first research if currently any TSOAUTH resources are accessed through the default return code (DFLTRC) that is configured for the TSOAUTH resource class in the class descriptor table (CDT). When you use option **RA.S**, for Setropts, RRSF, and class settings, it generates five standard audit reports. You can access the RACFCLAS report with action command **S**, to review the configuration and settings of all RACF resource classes in your system. You can issue `find tsoauth` in the command line and press Enter to find the class configuration settings of the TSOAUTH resource class (see Figure 3-1 on page 30). You can use action command **S** again to display the class configuration settings.

```
                    RACF class settings                          Line 1 of 38
Command ===> ____________________________________________________ Scroll===> CSR
                                          25 Oct 2024 12:31


  Class    Description
  TSOAUTH  TSO user authorities such as OPER and MOUNT

  Class SETROPTS settings                                  TVT6003  ZS34
  Protection active             Yes_____
  Command auditing active       Yes_____
  Logoptions                    Profile_
  GLOBAL (fast path) active     Yes_____
  Generics checked              Yes_____ Generics can be activated     Yes
  Generic commands allowed      Yes_____
  Profiles RACLISTed            Yes_____ Profiles can be RACLISTed     Yes
  Profiles GENLISTed            No______ Profiles can be GENLISTed     No
  Statistics collected          No______

  Profile syntax rules 1st     rest      Class properties
  Alphabetic allowed   Yes      Yes      Original order (class number) 259
  National allowed     Yes      Yes      Class identifier               48
  Numeric allowed      Yes      Yes      POSIT (options set id)        124
```

Figure 3-1 Display TSOAUTH class settings in the class descriptor table

From the “Yes” that is reported at option “Protection active”, you can derive that the protection of access to resources in the TSOAUTH class is active on this system. Press **F8** to scroll down to find the configured default return code for the TSOAUTH class (see Figure 3-2 on page 31).

```
                  RACF class settings                          Line 21 of 38
Command ===> _____________________________________________ Scroll===> CSR
                                          25 Oct 2024 12:31

  Maximum length                   8      Generic scan limit (quals)     0
  Maximum length with ENTITY       8      Installation-defined class    No
  Related grouping class                  Profile names case sensitive  No
  Related member class                    Default not-found RC           4

  Class activity options                  Profile residency options
  Profile definition forbidden  No        Profiles in dataspace         Yes
  OPERATIONS honored            No        RACLIST required              No
  Send ENF signal               No        RACLISTed by application only No

  Mandatory access control properties
  SECLABEL required             No
  Reverse MAC checking          No
  Equal MAC checking            No



****************************** Bottom of Data ******************************
```

Figure 3-2 Display TSOAUTH class default return code

The default return code for resource class TSOAUTH in the CDT is 4 as reported at option “Default not-found RC”. This means that when RACF does not find a matching resource profile for a TSOAUTH resource that a user tries to access, RACF provides return code 4. This return code means that RACF cannot make an access decision based on a resource profile and leaves the decision whether access is allowed or denied to the requesting resource manager (RM) such as z/OS, TSO, IMS, CICS®, Db2®, or other. Most RMs will probably allow access when the external security manager (ESM) RACF, ACF2, or TSS does not restrict access.

To verify whether any users can access TSOAUTH resources via this default return code, you can use the historic access decisions that Access Monitor collected.

Select option **AM**, for RACF Access Monitor, on the zSecure Admin Main menu and press Enter. Next, select option **1** from the zSecure Access Monitor menu (see Figure 3-3 on page 32).

```
                          zSecure Admin - Access
Option ===> 1_____________________________________________________________________

1      Access          Access summary by user or profile
2      Compare         Compare monitored access against current RACF database
3      Permit usage    Permit usage information for current RACF database
4      Connect usage   Connect usage information for current RACF database
5      Profile usage   Profile usage information for current RACF database
6      Member usage    Member usage information for current RACF database
7      Global usage    Global usage summary for current RACF database
8      Remove          Remove unused profiles and authorizations
9      Cleanup         Remove permits, dataset and general resource profiles
V      ID verify       ID verification (logon/start/job-start) summary
I      ID usage        ID usage information for current RACF database
U      Unix events     Access summary for unix events by user, uid/gid, or path
```

Figure 3-3 Select option Access from zSecure Admin Access menu

On the next menu, you specify to report all historical access events that are related to the TSOAUTH resource class. To use more granular selection filters for historical access events, you can select option **"Current RACF Database Selection"**. To report which users accessed TSOAUTH resources, you can select summary based on a member class and profile (see Figure 3-4).

```
                          zSecure Admin - Access - Access
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . tsoauth_  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
_  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
2_ 1. Summary by userid
__ 2. Summary by member class and profile
__ 3. Summary by simulated authorization used
__ 4. Summary by simulated groups used for access

_  Show configured fields  _  Show simulated fields   _  Timezone (U/L/H)
_  Print format             Customize title             Send as e-mail
      Background run          Full page form
```

Figure 3-4 Select access events for TSOAUTH resource class on AM.1 menu

In Figure 3-5 on page 33, you can specify which additional filtering criteria you want to use. In this scenario, you are interested in reporting on specific type of access events that used the default return code (DFLTRC). The DFLTRC setting reports historic access events that, when simulated against your allocated RACF source, returns the default return code for a resource access check for a class that is defined in RACF class descriptor table (CDT). The DFLTRC is

returned when no matching resource profile was found in the resource access check (RACHECK) call. The access events that are included in this report are the events that will be affected when you define a new RACF resource profile for the accessed resource.

```
                        zSecure Admin - Access - Current Database
Command ===> ____________________________________________________________________
Access monitor records for Classes like TSOAUTH
Specify simulated fields (Current DB effect) selection criteria:
Group(s) used for access  ____________________________  (group(s) or filter)
Essential group(s)  . . . _                            (Y/N/blank)
Not using group(s)  . . . ____________________________  (group(s) or filter)
Profile owned by  . . . . ________                      (id or filter)
RACF return code  . . . . __  ______  (operator: > >= < <= = <> ^=)

Authority used in current DB
_  APF          _  GRP_OPER     _  ID_USER      _  PRIVTRUS     _  SYS_OPER
_  CLAUTH       _  GRP_SPEC     _  INACTIVE     _  PROTFAIL     _  SYS_SPEC
_  CREATE       _  GRP_UACC     _  NO_CDT       _  QUALOWN      _  UACC
/  DFLTRC       _  ID_GROUP     _  NO_PROF      _  RESTRICTED   _  UNPROT
_  GLOBAL       _  ID_STAR      _  NOTHING      _  SPOOLRCVR    _  WARNING

User attributes in current DB (Y/N/blank)
_  ID present           _  ID Revoked
_  ID has special       _  ID has operations    _  ID has (ro)auditor
```

Figure 3-5 Select access events that use authority DFLTRC

Press Enter to produce the summary of access events where RACF returns the default return code for TSOAUTH class resources as is defined in the CDT (see Figure 3-6).

```
                IBM Security zSecure ACCESS summary    0 s elapsed, 0.1 s CPU
Command ===> ____________________________________________________ Scroll===> CSR
Access monitor records for Classes like TSOAUTH 16 Dec 2024 13:55
   Occurrence Class    First occurrenc Last occurrence
       751003 TSOAUTH   1Dec2023 22:31  2Nov2024 22:58
   Occurrence Profile key used
       751003
   Occurrence Intent   Type   RetAll AccRC
       751003 READ     Auth              4
   Occurrence Resource
__         525 CONOPER
S_      750478 MOUNT
******************************* Bottom of Data ********************************
```

Figure 3-6 Summary report of TSOAUTH access events that use DFLTRC

Exploring the summary statistics, you can identify that field “Profile key used” is missing in the report. Thus, this report indicates that two TSOAUTH resources; CONOPER and MOUNT are currently not protected by a profile in the allocated RACF database. You can use action command **S** preceding resource MOUNT and press Enter. This action provides you more detailed information about the users that accessed the TSOAUTH MOUNT resource during the last year (see Figure 3-7 on page 34).

```
                IBM Security zSecure ACCESS summary              Line 1 of 30
Command ===> _____________________________________________ Scroll===> CSR
Access monitor records for Classes like TSOAUTH 16 Dec 2024 13:55
   Occurrence Class    First occurrenc Last occurrence
       751003 TSOAUTH   1Dec2023 22:31  2Nov2024 22:58
   Occurrence Profile key used
       751003
   Occurrence Intent   Type   RetAll AccRC
       751003 READ     Auth              4
   Occurrence Resource
       750478 MOUNT
   Occurrence Userid   Name
__     749543 AXRUSER
__        333 CRMAUTO  ZTEAM AUTOTASKS
__          1 CRMBAB1  SYSPROG1 AB
__          1 CRMBAB2  SYSPROG2 AB
__         23 CRMBAH1  SYSPROG1 AH
__         18 CRMBAH2  SYSPROG2 AH
__          2 CRMBAH6  SYSPROG VK
__          2 CRMBEP1  SYSPROG1 EP
__          4 CRMBFN1  SYSPROG FN1
__          4 CRMBFN2  SYSPROG FN2
__          1 CRMBGUS  SYSPROG GUS
__         40 CRMBJK1  SYSPROG1 JK
```

Figure 3-7 Report detailed access statistics for TSOAUTH class resource MOUNT

After reviewing the outcome of the access simulation of historic access events, you decide to define a new resource profile named “MOUNT” in resource class TSOAUTH. Access to the MOUNT resource allows TSO/E users to issue dynamic allocation requests for datasets that result in the need for volume mounting. Next, you can simulate the security impact of adding profile MOUNT using your RACF-Offline database prior to defining the MOUNT profile in the TSOAUTH class of your active RACF database.

Analyzing the historic access events, you discovered that the majority of the reported users belong to either systems programmers or automation task users. You do not want to permit access to the TSOAUTH class MOUNT resource to individual users as your company is in the process of implementing role-based access control. You know that all systems programmers are connected to role-base group **SYSPROG** and automation task users are members of role-based group **SYS1**, therefore, you decide to permit READ access to these groups respectively. In a real-life scenario, it might be impossible to find an existing appropriate role-based group that represents a specific job role. In that case, you might want to consider introducing the new role-based group that allows efficient access management to this and potentially other sensitive z/OS resources.

Additionally, you might also discover that some of the users do not fall in the specific job roles or functions that you would expect to require access to TSOAUTH class MOUNT resource. In that case, the user access must not be provisioned in the TSOAUTH class MOUNT profile. Consequently, that user will no longer have this access after you introduce the TSOAUTH class MOUNT profile in the active RACF database.

3.4.2 Defining a new RACF resource profile in the RACF-Offline database

In this section you will learn how to define a new RACF resource profile in the RACF-Offline database.

To run the appropriate RACF commands to create TSOAUTH class resource profile MOUNT and define Access Control List (ACL) entries based on the results of your security assessment of historical access events for the TSOAUTH class MOUNT resource.

Select option **RA.R**, for RACF Administration, General resource profiles from the zSecure Admin Main menu (see Figure 3-8), and press Enter.

```
                zSecure Admin - RACF - Resource Selection
Command ===> ____________________________________________________   _ start panel

_   Add new general resource profile or segment
Show general profiles that fit all of the following criteria
Class name  . . . . tsoauth     (class or filter)
Resource profile  . ____________________________________________________________
____________________________________________________________  1 1 EGN mask
Owned by  . . . . . ________     (group or userid, or filter)      2 Exact
Installation data . ______________________________ (substring or *)   3 Match
                                                                     4 Any match
Additional selection criteria
_  Profile fields     _  Access list      _  Segment presence  _  Absence
_  Audit settings

Output/run options
_  Show segments      _  All              _  Enable full ACL   _  Specify scope
_  Show differences   _  Summarize by class
_  Print format          Customize title     Send as e-mail
      Background run     Full detail form    Sort differently     Narrow print
      Print ACL          Resolve to users    Incl operations      Print names
```

Figure 3-8 Select the profiles that are defined in the TSOAUTH class

Specify TSOAUTH at “Class name” and press Enter to see the profiles currently defined in the TSOAUTH class, shown in Figure 3-9 on page 36. Instead of reporting existing TSOAUTH profiles using this menu, as an alternative, you could use option “Add new general resource profile or segment” to build a new TSOAUTH class profile from scratch. In most cases, it is more efficient to copy an existing TSOAUTH class profile and adjust the generated commands to match your requirements rather than typing all required profile settings manually. Specify action command **C**, for copy, preceding profile PARMLIB and press Enter.

```
                                                          0 s elapsed, 0.0 s CPU
zSecure Admin General resource overview
Command ===> _____________________________________________________ Scroll===> CSR
Class TSOAUTH                                  16 Dec 2024 14:41
   Class    Profile key                              T UACC    Owner    S/F W
__ TSOAUTH  ACCT                                       READ    SYSAUTH  R R _
__ TSOAUTH  CONSOLE                                    NONE    SYSAUTH  R R _
__ TSOAUTH  JCL                                        READ    SYSAUTH  R R _
__ TSOAUTH  OPER                                       READ    SYSAUTH  R R _
C_ TSOAUTH  PARMLIB                                    NONE    SYSAUTH  R R _
__ TSOAUTH  RECOVER                                    READ    SYSAUTH  R R _
******************************* Bottom of Data ********************************
```

Figure 3-9 Copy profile PARMLIB in the TSOAUTH class

In the next panel (see Figure 3-10), specify that you want to copy TSOAUTH profile PARMLIB to TSOAUTH profile MOUNT. Use option **1**, for Create permanent profile, and sub-option **1**, for Copy segments and members, and press Enter.

```
                        zSecure Admin - RACF - Resource Copy
Command ===> ____________________________________________________________________

From class  . . . . . .  TSOAUTH
     profile  . . . . .  PARMLIB
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________
To class  . . . . . .  TSOAUTH_

   profile  . . . . . . MOUNT_____________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________

1  1. Create permanent profile
      1  1. Copy segments and members (commands are stored in CKRCMD)
         2. Use "copy from" (does not copy segments and members)
         _  RACF commands optimized for post-processing
   2. Create temporary profile
      Date of removal . . .  ___________  (ddmmmyyyy or yyyy-mm-dd or +days)
      Reason  . . . . . .  ____________________________________________
```

Figure 3-10 Specify TSOAUTH class profile name mount to be added

After pressing Enter, you return to the result of your original query as also shown in Figure 3-11 on page 37.

```
                                                                Queued in CKRCMD
zSecure Admin General resource overview
Command ===> ____________________________________________________ Scroll===> CSR_
Class TSOAUTH                                    16 Dec 2024 14:41
   Class    Profile key                                T UACC    Owner    S/F W
__ TSOAUTH  ACCT                                         READ    SYSAUTH  R R _
__ TSOAUTH  CONSOLE                                      NONE    SYSAUTH  R R _
__ TSOAUTH  JCL                                          READ    SYSAUTH  R R _
__ TSOAUTH  OPER                                         READ    SYSAUTH  R R _
__ TSOAUTH  PARMLIB                                      NONE    SYSAUTH  R R _
__ TSOAUTH  RECOVER                                      READ    SYSAUTH  R R _
****************************** Bottom of Data ********************************
```

Figure 3-11 Commands to copy TSOAUTH profile PARMLIB generated in CKRCMD work data set

The message “Queued in CKRCMD” in the top right corner of your display informs you that the RACF commands are generated in the CKRCMD work data set. Press **F3** to leave your display and access the work data set with the generated RACF commands (see Figure 3-12).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                   Columns 00001 00072
Command ===> ____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated 16 Dec 2024 14:41 */
000002  /* CKRCMD file CKRT1CMD complex TVT6003 NJE TVT6003 generated 16 Dec -
000003 2024 14:54 */
000004 rdefine  TSOAUTH  mount +
000005          owner(SYSAUTH) +
000006          uacc(NONE) +
000007          audit(all(READ))
000008 permit   mount +
000009          class(TSOAUTH) +
000010          id(SYSPROG) access(UPDATE)
000011 permit   mount +
000012          class(TSOAUTH) +
000013          id(CRMBGUS) access(UPDATE)
000014 permit   mount +
000015          class(TSOAUTH) +
000016          id(CRMBPH1) access(UPDATE)
```

Figure 3-12 RACF commands to copy TSOAUTH class PARMLIB profile to MOUNT generated

Adjust the generated commands to fit with your requirements for the to be added TSOAUTH class profile MOUNT (see Figure 3-13 on page 38).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                    Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR_
****** ***************************** Top of Data ******************************
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated 16 Dec 2024 14:41 */
000002  /* CKRCMD file CKRT1CMD complex TVT6003 NJE TVT6003 generated 16 Dec -
000003 2024 14:54 */
000004 rdefine  TSOAUTH  mount +
000005          owner(SYSAUTH) +
000006          uacc(NONE) +
000007          audit(failures(READ))
000008 permit   mount +
000009          class(TSOAUTH) +
000010          id(SYSPROG) access(read)
000011 permit   mount +
000012          class(TSOAUTH) +
000013          id(SYS1) access(read)
****** **************************** Bottom of Data ****************************
```

Figure 3-13 Commands to define TSOAUTH class profile MOUNT edited

When you are satisfied with the RACF commands to define TSOAUTH class profile MOUNT, either type `GO` to execute the generated commands, or press **F3** to return to the zSecure Admin Results menu. As indicated by the message in the top right corner of your panel enter **R** (see Figure 3-14), and press Enter to run the commands.

```
                         zSecure Admin - Results       Enter R to run commands
Command ===> ____________________________________________________________________

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 M E-mail report
 W Write file into seq. or partitioned data set

Enter a selection in front of a highlighted line below:
_  SYSPRINT  messages
_  REPORT    printable reports
_  CKRTSPRT  output from the last TSO command(s)
R  CKRCMD    queued TSO commands
_  CKR2PASS  queued commands for zSecure Admin
_  COMMANDS  zSecure Admin input commands from last query
_  SPFLIST   printable output from PRT primary command
_  OPTIONS   set print options

_  View files from recursive call (now on level 0)
```

Figure 3-14 Run commands to define TSOAUTH class profile MOUNT

You automatically transfer to the zSecure CKRTSPRT work data set that shows the results of the executed RACF commands. Because no “Command failed” message is shown in the top right corner of your display, that means that all RACF commands ran successfully (see Figure 3-15).

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000000 Col 001 080
 Command ===> ____________________________________________________ Scroll ===> CSR_
******************************** Top of Data ********************************
================================================================================
=== Multiple TSO command output file - scroll max down for overview ===
=== Input data set CRMBTZ1.DATA.C2R1DC6.CKRCMD                       ===
================================================================================
================================================================================
=== Commands for local node
================================================================================
/* CKRCMD file CKR1CMD complex TVT6003 generated 16 Dec 2024 14:41 */
/* CKRCMD file CKRT1CMD complex TVT6003 NJE TVT6003 generated 16 Dec 2024 14:54

============ 16Dec24 15:14:33.16377 start record 4 ============
rdefine  TSOAUTH  mount owner(SYSAUTH) uacc(NONE) audit(failures(READ))
ICH10006I RACLISTED PROFILES FOR TSOAUTH WILL NOT REFLECT THE ADDITION(S) UNTIL

============ 16Dec24 15:14:33.17609 start record 8 ============
permit   mount class(TSOAUTH) id(SYSPROG) access(read)
ICH06011I RACLISTED PROFILES FOR TSOAUTH  WILL NOT REFLECT THE UPDATE(S) UNTIL A

============ 16Dec24 15:14:33.18249 start record 11 ============
```

Figure 3-15 Commands to define TSOAUTH class profile MOUNT ran successfully

Optionally, you can press **F8** to scroll down to the comment box that reports the “All commands completed successfully” message.

The `TSOAUTH` class profile `MOUNT` is successfully added to your offline RACF database.

3.4.3 Conducting simulation of Access Monitor data against the offline RACF database

In this section, you will learn how to conduct a simulation of Access Monitor data against the offline RACF database.

After execution of the RACF commands to define TSOAUTH class MOUNT profile against your offline RACF database, you can verify the correct addition of this new resource profile.

Select option **RA.R**, for RACF Administration, General resource profiles from the zSecure Admin Main menu and press Enter. Specify class name TSOAUTH, resource profile MOUNT, and press Enter to list the profile (see Figure 3-16 on page 40).

```
                zSecure Admin - RACF - Resource Selection
Command ===> ____________________________________________________   _ start panel

_   Add new general resource profile or segment
Show general profiles that fit all of the following criteria
Class name  . . . . TSOAUTH_     (class or filter)
Resource profile  . mount_____________________________________________________
_____________________________________________________________  1 1 EGN mask
Owned by  . . . . . ________     (group or userid, or filter)      2 Exact
Installation data . ____________________________ (substring or *)   3 Match
                                                                    4 Any match
Additional selection criteria
_  Profile fields     _  Access list      _  Segment presence  _  Absence
_  Audit settings

Output/run options
_  Show segments      _  All              _  Enable full ACL   _  Specify scope
_  Show differences   _  Summarize by class
_  Print format          Customize title     Send as e-mail
      Background run     Full detail form    Sort differently     Narrow print
      Print ACL          Resolve to users    Incl operations      Print names
```

Figure 3-16 Use option RA.R to list details of resource profile MOUNT in the TSOAUTH class

After selecting RACF class TSOAUTH and profile MOUNT, press Enter to list the profile (see Figure 3-17).

```
zSecure Admin General resource overview
Command ===> ____________________________________________________ Scroll===> CSR_
Class TSOAUTH, like mount                        16 Dec 2024 15:30
   Class    Profile key                                T UACC    Owner    S/F W
S_ TSOAUTH  MOUNT                                       NONE___ SYSAUTH_ __R _
******************************* Bottom of Data ********************************
```

Figure 3-17 Access the detailed information of TSOAUTH class profile MOUNT

Use line command **S** preceding the TSOAUTH MOUNT profile and press Enter to review the correct definition of the resource profile and all its settings (see Figure 3-18 on page 41).

```
                                                                  Line 1 of 33
zSecure Admin General resource overview
Command ===> ____________________________________________________ Scroll===> CSR
Class TSOAUTH, like mount                      16 Dec 2024 15:37


_ Identification                                                    TVT6003
  Class                          TSOAUTH
  Profile name                   MOUNT
  Type
  Volume serial list
_ Owner                          SYSAUTH_                  AUTHORIZATION GRO
  Installation data              ________________________________________
  Application data               ________________________________________

  User     Access  ACL id   When              RI Name                DfltGrp
_ -group-  READ    SYSPROG_
_ -group-  READ    SYS1____



  Safeguards                           Other permissions
  User to notify of violation  ________ Allow all accesses    WARNING No_
  Audit access success/failures __R    Universal access authority    NONE__
```

Figure 3-18 Verify correct definition of TSOAUTH class profile MOUNT

As you can see, the profile is correctly added and shows the intended role-based groups SYSPROG and SYS1 with READ access on the access control list.

Next, you can simulate running the historic access events against the RACF-Offline database that contains the new TSOAUTH class resource profile MOUNT. For that purpose, you repeat running the **AM.1** report with the same selection parameter DFLTRC to verify whether the simulation still associates the outcome with a missing RACF resource profile. The report is expected return “No records found” as now TSOAUTH class profile MOUNT is defined (see Figure 3-19 on page 42).

```
                        zSecure Admin - Access - Access      No records found
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . TSOAUTH_  (class or EGN mask)
SAF resource name . . . mount___
RACF match on . . . . . ________________________________________

Advanced selection criteria
_  Further selection  _  Date selection   s  Current RACF DB selection

Output/run options
1  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  /  Show simulated fields   _  Timezone (U/L/H)
_  Print format               Customize title           Send as e-mail
      Background run          Full page form
```

Figure 3-19 Simulate access to TSOAUTH class resource MOUNT against RACF-Offline database

You can simulate access decisions that result from the new TSOAUTH class MOUNT profile using option **AM.2** from the zSecure Admin Access menu**.** You can use similar filtering criteria as you used in the previous step, except this time, you do not report authority used DFLTRC (see Figure 3-20).

```
                        zSecure Admin - Access - Compare
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . tsoauth_  (class or EGN mask)
SAF resource name . . . mount_____________________________________
RACF match on . . . . . ________________________________________
Comparison selection
Simulated results . . . /  Same      /  Less       /  More
Profile used  . . . . . 3  1. Same   2. Different  3. All

Advanced selection criteria
_  Further selection  _  Date selection   _  Current RACF DB selection

Output/run options
1  1. Summarize by userid  2. Summarize by member class and profile
_  Show configured fields  _  Timezone (U/L/H)
_  Print format               Customize title     Send as e-mail
      Background run          Full page form
```

Figure 3-20 Review access decisions for TSOAUTH class resource MOUNT via compare function

As shown in Figure 3-21, the access simulation results are divided into three separate reports:

- Report **SIMSAMED** shows access simulation results where the simulated access decisions are the same as the historic access decision.
- Report **SIMLESSD** shows access simulation results where the simulated access decisions allow less access than the historic access decision.
- Report **SIMMORED** shows access simulation results where the simulated access decisions allow more access than the historic access decision.

```
                 zSecure Admin Display Selection       0 s elapsed, 0.1 s CPU
Command ===> ____________________________________________________ Scroll===> CSR_

  Name     Summary Records Title
_ SIMSAMED       0       0 ACCESS summary simulated access is same
S SIMLESSD       9      13 ACCESS summary simulated access is less
S SIMMORED      21      37 ACCESS summary simulated access is more
****************************** Bottom of Data ********************************
```

Figure 3-21 Result of comparing simulated access against historic access

Report **SIMSAMED** is expected to report 0 records for the simulation to the TSOAUTH class resource MOUNT. Because you added the TSOAUTH class profile MOUNT, the simulated historic access events will no longer result in getting default return code of 4 now. This outcome proves that your profile is now successfully used to protect TSOAUTH class resource MOUNT.

When you permitted access to all users that historically accessed TSOAUTH class resource MOUNT, report **SIMLESSD** must also report 0 records. However, in this sample scenario, not all historic access through DFLTRC were honored with a permission. Thus, users that used TSOAUTH class resource MOUNT during the last year but are not connected to role-based groups SYSPROG and SYS1 now get a violation at their next attempt to use this resource. Use action command **S** preceding report **SIMLESSD** and press Enter to review which users no longer have access to TSOAUTH class resource MOUNT (see Figure 3-22).

```
                                                                0.0 s CPU, RC=4
ACCESS summary simulated access is less
Command ===> ____________________________________________________ Scroll===> CSR
Access monitor records for Classes like TSOAUTH 17 Dec 2024 09:40
   Occurrence Userid   Name                          First occurrenc Last occur
__            1 CRMBAB2  SYSPROG2 AB                   1Dec2023 22:31  1Dec2023
S             23 CRMBAH1  SYSPROG1 AH                   5Jun2024 13:53 12Sep2024
__            18 CRMBAH2  SYSPROG2 AH                  16Jan2024 14:07 12Mar2024
__             2 CRMBEP1  SYSPROG1 EP                  19Apr2024 13:21 19Apr2024
__             1 CRMBLU2  SYSPROG2 LU                   4Jan2024 12:12  4Jan2024
__             2 CRMBNAT  SYSPROG NAT                  25Mar2024 11:36 25Mar2024
__             1 CRMBSG1  SYSPROG1 SG                  29Jan2024 09:53 29Jan2024
__             1 CRMBSG2  SYSPROG2 SG                  29Jan2024 09:54 29Jan2024
__            14 HZSPROC                                2Nov2024 09:05  2Nov2024
****************************** Bottom of Data ********************************
```

Figure 3-22 Review users that no longer have access to TSOAUTH class resource MOUNT

You can use action command **S** once more to access the simulation details for user CRMBAH1 and press Enter (see Figure 3-23).

```
                                                                  Line 1 of 2
ACCESS summary simulated access is less
Command ===> _____________________________________________________ Scroll===> CSR
Access monitor records for Classes like TSOAUTH 17 Dec 2024 09:40
   Occurrence Userid   Name                          First occurrenc Last occur
           23 CRMBAH1  SYSPROG1 AH                    5Jun2024 13:53 12Sep2024
   Occurrence Intent   Type   RetAll AccRC SimRC
           23 READ     Auth              4     8
   Occurrence Class
           23 TSOAUTH
   Occurrence Resource                                          Profile key us
           23 MOUNT
   Occurrence Complex  Syst RGPJCAVP GUGSOPGXO SOA PCSL First occurrenc Last oc
           23 TVT6003  ZS34                    S A       5Jun2024 13:53 12Sep20
   Occurrence Timestamp
__          1  5Jun2024 13:53
__         22 12Sep2024 20:02
****************************** Bottom of Data ********************************
```

Figure 3-23 Simulated access decisions for user CRMBAH1 to TSOAUTH class resource MOUNT

The column “SimRC” reports that the simulated RC is 8, meaning that the user is not authorized by the currently allocated RACF database source, the offline RACF database. Column “AccRC” reports the historic access result of RC=4, meaning no matching profile found. Hence, adding the TSOAUTH class profile MOUNT successfully stops this user from accessing this resource when you define the same profile in the active RACF database. Potentially, you can decide that some of the users reported in report **SIMLESSD** do need to be authorized to use the TSOAUTH MOUNT resource after all. You can decide to add another permit to the access control list of TSOAUTH class profile MOUNT. When this user must not have this access in the future, adding the profile works as designed.

Report **SIMMORED** must contain records as shown in Figure 3-24 on page 45. The reason why report **SIMMORED** is expected to contain records where the simulated access results allow “more” or “higher” access than the historic access decision from the access records.is because you now permitted these users on the access list of the TSOAUTH class MOUNT profile.

```
                                                                  Line 1 of 21
ACCESS summary simulated access is more
Command ===> _____________________________________________ Scroll===> CSR_
Access monitor records for Classes like TSOAUTH 17 Dec 2024 09:40
   Occurrence Userid   Name                          First occurrenc Last occur
S_      749543 AXRUSER                               23Jun2024 11:51  2Nov2024
__         333 CRMAUTO  ZTEAM AUTOTASKS               22Jun2024 23:10  2Nov2024
__           1 CRMBAB1  SYSPROG1 AB                   1Dec2023 22:32  1Dec2023
__           2 CRMBAH6  SYSPROG VK                   12Sep2024 20:12 12Sep2024
__           4 CRMBFN1  SYSPROG FN1                   7May2024 14:09  7May2024
__           4 CRMBFN2  SYSPROG FN2                   7May2024 14:09  7May2024
__           1 CRMBGUS  SYSPROG GUS                   6Mar2024 09:21  6Mar2024
__          40 CRMBJK1  SYSPROG1 JK                  19Apr2024 07:25 31Oct2024
__           8 CRMBLU1  SYSPROG1 LU                  16May2024 13:12 16May2024
__           3 CRMBMC1  SYSPROG1 MC                  20May2024 06:40 20May2024
__           2 CRMBMJ1  SYSPROG1 MJ                   7Dec2023 09:35  7Dec2023
__          18 CRMBMK1  SYSPROG1 MK                  30Oct2024 16:07 30Oct2024
__           1 CRMBPH1  SYSPROG1 PH                   7Mar2024 10:22  7Mar2024
__           3 CRMBRL1  SYSPROG1 RL                  15Dec2023 13:54 11Sep2024
__           5 CRMBRS1  SYSPROG1 RS                  20Jun2024 11:06 11Sep2024
__           2 CRMBRT1  SYSPROG1 RT                  30Apr2024 09:56 30Apr2024
__          23 CRMBTZ1  SYSPROG1 TZ                   1Nov2024 16:02  1Nov2024
__          77 CRMBVK1  SYSPROG1 VK                  16Jan2024 15:55  1Nov2024
__           6 CRMBVK2  SYSPROG2 VK                   1Aug2024 12:36  1Aug2024
```

Figure 3-24 Review users that are permitted access to TSOAUTH class resource MOUNT

You can use action command **S** to review the access event details (see Figure 3-25).

```
                                                                  Line 1 of 2
ACCESS summary simulated access is more
Command ===> _____________________________________________ Scroll===> CSR_
Access monitor records for Classes like TSOAUTH 17 Dec 2024 09:40
   Occurrence Userid   Name                          First occurrenc Last occur
       749543 AXRUSER                               23Jun2024 11:51  2Nov2024
   Occurrence Intent   Type   RetAll AccRC SimRC
       749543 READ     Auth              4     0
   Occurrence Class
       749543 TSOAUTH
   Occurrence Resource                                        Profile key us
       749543 MOUNT
   Occurrence Complex  Syst RGPJCAVP GUGSOPGXO SOA PCSL First occurrenc Last oc
       749543 TVT6003  ZS34                            23Jun2024 11:51  2Nov20
   Occurrence Timestamp
S_        8824 23Jun2024 11:51
__      740719  2Nov2024 22:58
****************************** Bottom of Data ********************************
```

Figure 3-25 Review user AXRUSER that is permitted access to TSOAUTH class resource MOUNT

The column “SimRC” reports that the simulated RC is 0 from the currently allocated RACF database, the offline RACF database, meaning that the user is authorized to READ this resource. Column “AccRC” reports the historic access result of RC=4, hence adding the

TSOAUTH class profile MOUNT authorizes this user to access the TSOAUTH MOUNT resource with READ access when you define the same profile in the active RACF database.

To review the details of the access event record, issue action command **S** again and press Enter (see Figure 3-26).

```
                                                                  Line 1 of 47
ACCESS summary simulated access is more
Command ===> ____________________________________________________ Scroll===> CSR_
Access monitor records for Classes like TSOAUTH 17 Dec 2024 09:40


  Access summary
_ Security complex name         TVT6003
_ RACF userid                   AXRUSER
  Access intent                 READ
  Record type                   Auth
  System name                   ZS34
  SAF resource class            TSOAUTH
  SAF resource name             MOUNT
  Timestamp of last occurrence  23Jun2024 11:51
  Access count                  8824

  Request flags                           Define flags
  RACFIND                                 Use internal RACROUTE AUTH
  Limit to generics             No        Verify
  Private/CSA (no global)       No        Propagated
  Bypass JESSPOOL profiles      No
  Command                       No
  Retrieval of access allowed   No
```

Figure 3-26 Review access event details for AXRUSER to TSOAUTH class resource MOUNT

When you review the detailed access events statistics, you can find out which resource profile is selected as best matching resource profile during the access simulation. Press **F8** to scroll down (see Figure 3-27 on page 47).

```
                                                                Line 30 of 47
ACCESS summary simulated access is more
Command ===> _____________________________________________ Scroll===> CSR_
Access monitor records for Classes like TSOAUTH 17 Dec 2024 09:40

  Global access table used      No       Installation exit used        No
  Special authority used        No       Owner access                  No
  Operations authority used     No

  Access-time user attributes            Program status
  User systemwide SPECIAL       No       Defined program (any lib)
  User systemwide OPERATIONS    No       Controlled program
  User systemwide (RO)AUDITOR   No       Specific controlled program
                                         Controlled library (any pgm)

  Current RACF database effect
  RACF return code current DB   0
  Authority used in current DB  ID_GROUP
  RACF Profile type current DB  DISCRETE
_ RACF class and profile in DB  TSOAUTH  MOUNT
_ Profile owner                 SYSAUTH
  Installation data
  Creation/definition date      16 Dec 2024
****************************** Bottom of Data ********************************
```

Figure 3-27 Review which resource profile protects access to TSOAUTH class resource MOUNT

The simulation access event details are shown in section “Current RACF database effect” at the bottom of your display. From this section, you can conclude that user AXRUSER is allowed READ access to TSOAUTH class profile MOUNT that is permitted to a group, as indicated by ID_GROUP at field “Authority used in current DB”.

When you are satisfied with the outcome of your access simulation, the final step is to end your RACF-Offline session and define the TSOAUTH class profile MOUNT to the active RACF database.

Press **F3** several times until you reach the ISPF Primary Option Menu. Specify option X to leave ISPF and return to your RACF-Offline session (see Figure 3-28 on page 48).

```
                          ISPF Primary Option Menu
Option ===> X_____________________________________________________________
                                          More:     +
0  Settings      Terminal and user parameters          User ID . : CRMBTZ1
1  View          Display source data or listings       Time. . . : 14:12
2  Edit          Create or change source data          Terminal. : 3278
3  Utilities     Perform utility functions             Screen. . : 1
4  Foreground    Interactive language processing       Language. : ENGLISH
5  Batch         Submit job for language processing    Appl ID . : ISR
6  Command       Enter TSO or Workstation commands     TSO logon : TSOZSEC
7  Dialog Test   Perform dialog testing                TSO prefix: CRMBTZ1
9  IBM Products  IBM program development products      System ID : TVT6003
10 SCLM          SW Configuration Library Manager      MVS acct. : ACCT#
11 Workplace     ISPF Object/Action Workplace          Release . : ISPF 8.1
P  PRODUCTS      Dialogs for installed products
G  GROUP         Dialogs used with your organization
U  User          Your Own Dialogs
S  SDSF          System Display and Search Facility

C  Consul        Consul Risk Management applications

     Enter X to Terminate using log/list defaults
```

Figure 3-28 Exit ISPF

Specify what you want to do with your ISPF Log Data Set. Press Enter to return to the RACF-Offline session (see Figure 3-29).

```
                    Specify Disposition of Log Data Set
Command ===> ____________________________________________________________
                                                              More:     +
Log Data Set (CRMBTZ1.SPFLOG1.LIST) Disposition:
Process Option . . . . 2  1. Print data set and delete
                          2. Delete data set without printing
                          3. Keep data set - Same
                             (allocate same data set in next session)
Keep data set - New
                             (allocate new data set in next session)
Batch SYSOUT class . . _______________
Local printer ID or
writer-name  . . . . . _________________
Local SYSOUT class . . _______________

List Data Set Options not available

Press ENTER key to complete ISPF termination.
Enter END command to return to the primary option menu.

Job statement information:  (Required for system printer)
 ===> ____________________________________________________________________
 ===> ____________________________________________________________________
 ===> ____________________________________________________________________
```

Figure 3-29 Specify Log Data Set disposition option

You now return to the RACF-Offline interface. Specify command **end** (see Figure 3-30) and press Enter to exit your RACF-Offline session.

```
CRMBTZ1.SPFLOG1.LIST has been deleted.
 B8R200A Enter RACF Command or "END"
end
```

Figure 3-30 Exit your RACF-Offline session

You must now be back in TSO READY mode. Next, specify `ISPF` here once more and press Enter. You return to the ISPF Primary Option Menu again. But this time, you are no longer in a RACF-Offline session.

Start zSecure Admin using your regular method. You can continue to use the same zSecure Admin input set as you used during your RACF-Offline session noting that any RACF commands that you execute run against the production RACF database because you are no longer inside a RACF-Offline session (see Figure 3-31).

```
                        zSecure Admin - Setup - Input files        Row 1 of 3
Command ===> ____________________________________________________ Scroll ===> CSR

Description . . . . Recent RACF, CKFREEZE, ACCESS data sets_____________________
Complex . . . . . . TVT6003_   Version  . . . .  ____

Enter data set names and types.         Type END or press F3 when complete.
Enter dsname with .* to get a list      Type SAVE to save set, CANCEL to quit.
Valid line commands: E L I R D          Type REFRESH to submit unload  job.

   Data set name or DSNPREF=, or Unix file name           Type or ?   NJE node
_  'CRMB.T.RACF.OFFLINE'                                  COPY.RACF_  ________
_  'CRMB.T.CKFREEZE'                                      CKFREEZE__  ________
_  'C2PACMON.TVT6003.Y12MON'                              ACCESS____  ________
****************************** Bottom of data ********************************
```

Figure 3-31 Use the same input set for your zSecure Admin session

Because you still use your RACF-Offline database as input to this zSecure Admin session, you can easily define `TSOAUTH` class profile `MOUNT` by copying it from your offline RACF database.

Select option **RA.R**, for RACF Administration, General resource profiles from the zSecure Admin Main menu and press Enter (see Figure 3-32 on page 50).

```
                zSecure Admin - RACF - Resource Selection
Command ===> ____________________________________________________   _ start panel

_   Add new general resource profile or segment
Show general profiles that fit all of the following criteria
Class name  . . . . TSOAUTH_     (class or filter)
Resource profile  . mount_____________________________________________________
_____________________________________________________________  1 1 EGN mask
Owned by  . . . . . ________     (group or userid, or filter)      2 Exact
Installation data . ____________________________ (substring or *)   3 Match
                                                                    4 Any match
Additional selection criteria
_  Profile fields     _  Access list      _  Segment presence  _  Absence
_  Audit settings

Output/run options
_  Show segments      _  All              _  Enable full ACL   _  Specify scope
_  Show differences   _  Summarize by class
_  Print format          Customize title     Send as e-mail
      Background run     Full detail form    Sort differently     Narrow print
      Print ACL          Resolve to users    Incl operations      Print names
```

Figure 3-32 Use option RA.R to list details of resource profile MOUNT in the TSOAUTH class

Specify class name `TSOAUTH`, resource profile `MOUNT`, and press Enter to list the profile (see Figure 3-33).

```
zSecure Admin General resource overview
Command ===> ____________________________________________________ Scroll===> CSR_
Class TSOAUTH, like mount                        16 Dec 2024 15:30
   Class    Profile key                                T UACC    Owner    S/F W
C_ TSOAUTH  MOUNT                                        NONE___ SYSAUTH  __R _
****************************** Bottom of Data ********************************
```

Figure 3-33 Copy TSOAUTH class profile MOUNT

You can use action command **C**, for copy, to generate the appropriate RACF commands to copy `TSOAUTH` class profile `MOUNT` to the production RACF database (see Figure 3-34 on page 51).

```
                        zSecure Admin - RACF - Resource Copy
Command ===> ____________________________________________________________________

From class  . . . . . .  TSOAUTH
     profile  . . . . .  MOUNT
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________
To class  . . . . . .  TSOAUTH_
   profile  . . . . .  MOUNT_____________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________

1  1. Create permanent profile
       1  1. Copy segments and members (commands are stored in CKRCMD)
          2. Use "copy from" (does not copy segments and members)
       _  RACF commands optimized for post-processing
    2. Create temporary profile
       Date of removal . . .  ___________  (ddmmmyyyy or yyyy-mm-dd or +days)
       Reason  . . . . . .  ____________________________________________________
```

Figure 3-34 Copy TSOAUTH class profile MOUNT to TSOAUTH class MOUNT

Because the TSOAUTH class profile MOUNT does not yet exist in the active RACF database, you do not have to adjust the “To class” and “profile” specifications (see Figure 3-35). You can continue by pressing Enter to generate the commands.

```
                                                                Queued in CKRCMD
zSecure Admin General resource overview
Command ===> ____________________________________________________ Scroll===> CSR_
Class TSOAUTH, like mount                          17 Dec 2024 14:36
   Class    Profile key                                  T UACC    Owner    S/F W
__ TSOAUTH  MOUNT                                         NONE___ SYSAUTH  __R _
****************************** Bottom of Data ********************************
```

Figure 3-35 Commands to copy TSOAUTH profile MOUNT queued in CKRCMD work data set

The message “Queued in CKRCMD” in the top right corner of your display informs you that the RACF commands are successfully generated in the CKRCMD work data set. Press **F3** to leave your display and access the generated RACF commands in the CKRCMD work data set (see Figure 3-36 on page 52).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                          0.0 s CPU, RC=0
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated 17 Dec 2024 14:36 */
000002  /* CKRCMD file CKRT1CMD complex TVT6003 NJE TVT6003 generated 17 Dec -
000003 2024 14:43 */
000004 rdefine  TSOAUTH  MOUNT +
000005          owner(SYSAUTH) +
000006          uacc(NONE) +
000007          audit(failures(READ))
000008 permit   MOUNT +
000009          class(TSOAUTH) +
000010          id(SYSPROG) access(READ)
000011 permit   MOUNT +
000012          class(TSOAUTH) +
000013          id(SYS1) access(READ)
000014 setropts refresh raclist(TSOAUTH)
****** **************************** Bottom of Data ****************************
```

Figure 3-36 Commands to copy TSOAUTH class MOUNT profile to MOUNT generated

Either type `GO` to execute the generated commands, or press **F3** to return to the zSecure Admin Results menu. As indicated by the message in the top right corner of your panel you can enter action command **R** (see Figure 3-37), and press Enter to run the commands.

```
                        zSecure Admin - Results       Enter R to run commands
Command ===> ____________________________________________________________________

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 M E-mail report
 W Write file into seq. or partitioned data set

Enter a selection in front of a highlighted line below:
_  SYSPRINT  messages
_  REPORT    printable reports
_  CKRTSPRT  output from the last TSO command(s)
R  CKRCMD    queued TSO commands
_  CKR2PASS  queued commands for zSecure Admin
_  COMMANDS  zSecure Admin input commands from last query
_  SPFLIST   printable output from PRT primary command
_  OPTIONS   set print options

_  View files from recursive call (now on level 0)
```

Figure 3-37 Run commands to define TSOAUTH class profile MOUNT to production RACF database

You automatically transfer to the zSecure CKRTSPRT work data set that shows the results of the executed RACF commands. Because no “Command failed” message shows in the top

right corner of your display, this means that all RACF commands ran successfully. (see Figure 3-38).

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000000 Col 001 080
 Command ===> _____________________________________________ Scroll ===> CSR
******************************** Top of Data ********************************
================================================================================
=== Multiple TSO command output file - scroll max down for overview ===
=== Input data set CRMBTZ1.DATA.C2R1DC6.CKRCMD                        ===
================================================================================
================================================================================
=== Commands for local node
================================================================================
/* CKRCMD file CKR1CMD complex TVT6003 generated 17 Dec 2024 14:36 */
/* CKRCMD file CKRT1CMD complex TVT6003 NJE TVT6003 generated 17 Dec 2024 14:43

============ 17Dec24 14:53:23.513000 start record 4 ============
rdefine  TSOAUTH  MOUNT owner(SYSAUTH) uacc(NONE) audit(failures(READ))
ICH10006I RACLISTED PROFILES FOR TSOAUTH WILL NOT REFLECT THE ADDITION(S) UNTIL

============ 17Dec24 14:53:23.564400 start record 8 ============
permit   MOUNT class(TSOAUTH) id(SYSPROG) access(READ)
ICH06011I RACLISTED PROFILES FOR TSOAUTH  WILL NOT REFLECT THE UPDATE(S) UNTIL A

============ 17Dec24 14:53:23.571180 start record 11 ============
```

Figure 3-38 Output of RACF commands to define TSOAUTH class profile MOUNT

Optionally, you can press **F8** to scroll down to the comment box that reports the message “All commands completed successfully”.

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000009 Col 001 080
 Command ===> _____________________________________________ Scroll ===> CSR
/* CKRCMD file CKRT1CMD complex TVT6003 NJE TVT6003 generated 17 Dec 2024 14:43

============ 17Dec24 14:53:23.513000 start record 4 ============
rdefine  TSOAUTH  MOUNT owner(SYSAUTH) uacc(NONE) audit(failures(READ))
ICH10006I RACLISTED PROFILES FOR TSOAUTH WILL NOT REFLECT THE ADDITION(S) UNTIL

============ 17Dec24 14:53:23.564400 start record 8 ============
permit   MOUNT class(TSOAUTH) id(SYSPROG) access(READ)
ICH06011I RACLISTED PROFILES FOR TSOAUTH  WILL NOT REFLECT THE UPDATE(S) UNTIL A

============ 17Dec24 14:53:23.571180 start record 11 ============
permit   MOUNT class(TSOAUTH) id(SYS1) access(READ)
ICH06011I RACLISTED PROFILES FOR TSOAUTH  WILL NOT REFLECT THE UPDATE(S) UNTIL A

============ 17Dec24 14:53:23.577840 start record 14 ============
setropts refresh raclist(TSOAUTH)
================================================================================
=== All commands completed successfully                                   ===
================================================================================
******************************** Bottom of Data ********************************
```

Figure 3-39 RACF commands to define TSOAUTH class profile MOUNT ran successfully

The TSOAUTH class profile MOUNT is successfully added to your active RACF database. The access to TSOAUTH class resource MOUNT is now controlled as requested by your auditors.

This step concludes the walk through for adding TSOAUTH class profile MOUNT to control access to this z/OS sensitive resource.

Alternatively, you can use this same or a similar scenario to:

- Add another resource profile for TSOAUTH class resource CONOPER that was also reported as being accessed through authority DFLTRC, see Figure 3-6 on page 33.
- Report general resources in other general resource classes that were accessed through authority DFLTRC and define appropriate resource profiles for these resources as well.
- When your system has not set SETROPTS PROTECTALL(FAIL), you can report which data sets are accessed through authority UNPROT. Access simulation against the current RACF database returns UNPROT when no best matching dataset profile is found to control access to this data set. You can then use this same scenario to setup appropriate dataset profiles to protect these data sets.

4

Remove unused security definitions

One of the most difficult tasks for RACF security administrators is to keep the RACF database orderly, maintained, and compliant. Many RACF installations also suffer from the inability to identify security definitions that were created sometime in the past but are now no longer needed or used—causing security exposures that could potentially give an attacker unintended access to a system and sensitive data.

In this chapter, we discuss the following topics:

- 4.1, “Challenges with unused RACF definitions” on page 56
- 4.2, “Keeping the RACF database orderly and well maintained” on page 56
- 4.3, “Ensuring only the required authorization is in place” on page 58
- 4.4, “Removing unused resource profiles” on page 59

4.1 Challenges with unused RACF definitions

RACF databases tend to collect a lot of unused or outdated profile definitions which make them harder and more labor intensive to maintain. Also, passing audits is more challenging as auditors might pose questions about definitions that may not even be used by the company. Additionally, unused security definitions can pose security risks, as they can lead to further misconfiguration or unexpected permissions.

Some general reasons a RACF database can become unstructured and not meet specified standards, are as follows:

- RACF security administrators are often requested by business process owners to create certain RACF user, group, data set, or general resource profiles when required. However, when these profiles are no longer required, they are not informed to remove these security definitions from RACF.
- Resource names are changed to adhere to a new naming convention standard but thereafter the resource profiles that refer to the former resource names are not deleted.
- Security administrators are not informed when staff members leave the company causing their user IDs to remain defined in the RACF database.
- Business processes or applications that were used in the past are changed, stopped, or superseded by other processes, tooling, or applications. The required RACF security definitions to support the new or changed implementations are added to RACF, but the old definitions are often left behind.

4.2 Keeping the RACF database orderly and well maintained

As a best practice, RACF administrators should take actions to maintain the accuracy of their RACF database and keep the security definitions up to date. These actions help to minimize clutter that could cause problems for the administrators, auditors, compliance officers, users, or owners of the environment.

- Address inactive users
- Address unutilized groups
- Address users with outdated group connections
- Remove unused resource profiles

This document does not specify cadences for some of these checks because every environment is different and depending on your business requirements and the frequency that certain applications or workloads are run, you may wish to deviate from these practices to some degree.

4.2.1 Address inactive users

Removing unused users is an integral part of RACF database maintenance. Unused users can have logon credentials through multiple means, access to many types of applications and resources in the system. Without someone utilizing a user ID, this presents an unnecessary security exposure for an attacker to gain access to a system and all the privileges that this user is assigned. There is also potential for more deliberate misuse of these user IDs from former users of the system.

RACF includes several ways to minimize the risk associated with this problem. The *RACF Security Administrator's Guide* discusses "Automatically revoking unused user IDs

(INACTIVE option)” in more detail. However, this is a standard way of revoking users that have not had any activity in this environment for some time. Without the proper monitoring and configuration, this setting can be underutilized. Ensure the cadence specified in revoking user IDs meaningfully aligns with your security policies. Additionally, even with this automatic revocation, it is important to still actively take steps to delete unused user IDs when you are aware of a personnel change.

Leaving unutilized user IDs in your RACF environment is incompatible with a “least privilege” environment where only the user IDs which are necessary should exist and have appropriate privileges. As such, keeping defined user IDs current is an important aspect of maintaining such an environment.

4.2.2 Address unutilized groups

Removing unused groups is another aspect of RACF database maintenance. These groups can still have connections to active users as well as accesses to resource profiles and may persist past important updates to your security environment as they may not be documented or monitored once they are relegated to obscurity. Like inactive users, these unused group profiles can provide unforeseen or outdated accesses for an attacker or malicious user with a connection to this group.

Leaving outdated groups in your RACF environment is incompatible with a “least privilege” environment as they can lead to extraneous and outdated privileges for connected users. As such, keeping groups current is an important aspect of maintaining such an environment.

4.2.3 Address users with outdated group connections

While sometimes you have groups that are no longer being utilized at all, there might exist groups in your environment where the connected users have moved on and are no longer leveraging this group connection. In cases like these, it is vital to make sure that the security information in your environment is current and that these outdated connections are addressed. Users connected to groups that they no longer need can still use the permissions associated with these groups. This configuration allows for unforeseen or outdated accesses by an attacker or malicious user with a connection.

Another concern involving maintaining user and group profiles in a RACF environment is to manage users with multiple differing job roles or groups carefully. Unfortunately, users are not always removed from the role-based groups that no longer align with their current job role. To help monitor this, reviewing the groups and roles that a user uses is a vital tool to keeping these group connections and permissions current. Groups that represent different job roles can be “incompatible” or even “toxic” with one another job role. Users that are connected to incompatible or toxic groups can indicate a change in job role or function for this user occurred that was not adequately administered in RACF.

RACF provides a useful option in this case where an administrator may not want to completely disconnect the group connection right away. As RACF documents in their Command Language Reference, the CONNECT command contains a REVOKE option that allows an administrator to prevent a user from using a group connection to access resources while maintaining the existence of this connection. This REVOKE option for a connection allows for an administrator to restore these permissions in the event that this change is meant to be temporary, or to completely remove them after a job role change has been validated.

As discussed previously, leaving outdated groups in your RACF environment is incompatible with a “least privilege” environment as they can lead to extraneous and outdated privileges for

connected users. As such, keeping these groups, current is an important aspect of maintaining a “least privilege” environment.

4.2.4 Remove unused resource profiles

Removing unused resource profiles is another key aspect of RACF database maintenance. Just because a resource profile is no longer used, does not mean that it no longer offers privileges to the users and groups in its access lists. These privileges are another unnecessary security exposure for an attacker to gain access to whatever functions or resources that this profile protects. There is also potential for inadvertent or deliberate misuse of these functions, resources, or data sets by users that may lead to additional concerns in your environment.

Leaving unutilized resource profiles in your RACF environment is incompatible with a “least privilege” environment where only the minimum privileges required for functionality should be in place. As such, keeping resource profiles current is an important aspect of maintaining a “least privilege” environment.

4.3 Ensuring only the required authorization is in place

Typically, businesses and organizations state in their security policies that the RACF security definitions must only contain profiles that are used to protect sensitive resources. However, how does a RACF security administrator know that the deletion of security definitions could result in started tasks, jobs, applications, or subsystems no longer have the necessary authorization to work as expected?

A traditional method that is available for determining whether RACF definitions are used for access requests would be to log all successful access to all resources to SMF. But you can imagine that this solution is simply not feasible because of the sheer amount of SMF records that this implementation would produce, the required CPU time and storage.

Using zSecure Access Monitor, the actual use of profiles, group connections, and permissions can be analyzed and reported. Optionally, zSecure Admin can generate the appropriate RACF commands to delete profiles, connections, or permission that according to Access Monitor records are not used for a prolonged period, for example over a year. Additionally, with RACF-Offline, you can validate that removing any such profile, connection, or permission would not have a negative system impact based on the captured data.

Additionally, while this document highlights a number of best practices related to the idea of removing unused profiles, not all of these practices are discussed in detail in the following walkthrough.

4.4 Removing unused resource profiles

The details for the general use case preparation steps using automated (semi-) conversion and cleanup features in zSecure Admin are documented in 2.4, “General preparation steps for the use cases” on page 16, including these steps:

- Creating a RACF-Offline database
- Starting a RACF-Offline session
- Logging on to RACF-Offline session
- Resetting the RACF-Offline log data set
- Starting and preparing your zSecure Admin session

After completion of these steps, you can continue with the subsequent steps to remove unused resource profiles.

- Generating RACF commands to clean up resource profiles
- Running the generated RACF commands to clean up resource profiles
- Recovering deleted resource profiles
- Reporting potential security impact of deleted unused RACF profiles
- Cleaning up unused profiles from the active primary RACF database
- Cleaning up other unused security definitions from the active primary RACF database

4.4.1 Generating RACF commands to clean up resource profiles

In this section, you will learn how to generate RACF commands for cleaning up the resource profiles that are not used.

In zSecure Admin you can use option **AM.8** for Remove, to automatically generate RACF commands to cleanup profiles, permits, or connections that according to Access Monitor records were not used during last year.

To start with removing unused resource profiles, select option 1 and press Enter (see Figure 4-1).

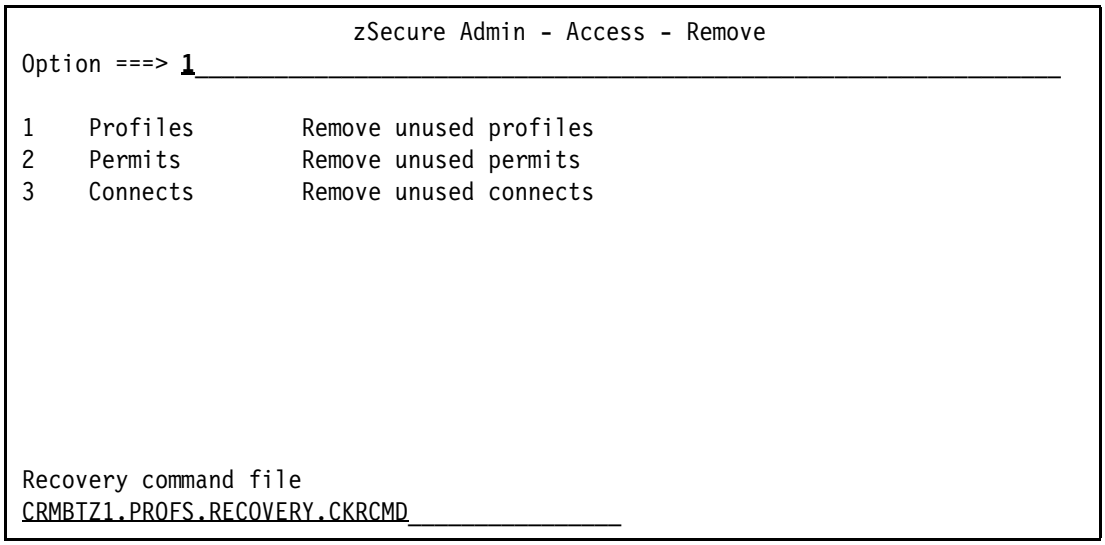


Figure 4-1 Select unused security definitions to be removed

Note: You can customize the name of the automatically generated a recovery command file. The commands in this recovery command file can be used to restore profiles that are cleaned up in case there is a requirement to rebuild the profile in its original state in the future.

In the next panel (see Figure 4-2), you can decide to remove unused profiles from all resource classes in the RACF database. In this example, the SURROGAT class is specified. Before pressing Enter, you must also select option “Additional profile filters” with an **S** or **/**. When you omit a class and RACF profile name on this panel, by default, zSecure Admin generates delete commands to remove the unused profiles of all resource classes.

```
                    zSecure Admin - Access - Profiles

Command ===> ____________________________________________________________

Remove unused profiles that fit all of the following criteria:
Class . . . . . . . . . surrogat  (class or EGN mask)
RACF profile name . . . ____________________________________________

Advanced selection criteria
/  Additional profile filters          _  Date selection

Output options

_  Background run
```

Figure 4-2 Optionally, specify a class name to restrict your cleanup to a resource class

After pressing Enter, the Further selection panel appears (see Figure 4-3). To prevent deleting recently added SURROGAT profiles, for example, during the last 2 weeks, you can specify `today-14` at option “Until” that verifies the profile creation date.

```
                 zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________
Access monitor records for classes like SURROGAT
Profile selection
Profile creation date   __________  Until   today-14__
Discrete profiles . . . _  (Y/N)
```

Figure 4-3 Specify additional profile filters on Further selection panel

Press Enter to produce the automatically generated cleanup commands for unused profiles in the SURROGAT class (see Figure 4-4 on page 61).

```
                          zSecure Admin – Results       Use END for other files

COMMAND ===> _____________________________________________________ Scroll ===> CSR_

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 W Write file into seq. or partitioned data set
 M E-mail report

CKRCMD for the specified environments:

   Complex   Njenode   Rrsfnode   System     zSecNode #Lines
S  TVT6003   <LOCAL>   ?          ZS34       ?        126
_  RECOVERY  <LOCAL>   ?          ZS34       ?        591
****************************** Bottom of data ********************************
```

Figure 4-4 Cleanup and recovery commands for SURROGAT class

Enter action command **S** to review the generated SURROGAT class cleanup commands and press Enter (see Figure 4-5).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                    Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR_
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated  1 Nov 2024 17:03 */
000002 rdelete  SURROGAT *.SUBMIT /* 18 Jun 2021 */
000003 rdelete  SURROGAT ALERT.SUBMIT /* 17 Mar 2014 */
000004 rdelete  SURROGAT BCSCGB*.SUBMIT /* 16 Feb 2010 */
000005 rdelete  SURROGAT BPX.SRV.ANONYMO /* 16 Feb 2010 */
000006 rdelete  SURROGAT BPX.SRV.CRMBRT1 /* 15 Apr 2014 */
000007 rdelete  SURROGAT BPX.SRV.CRMBVK2 /* 23 Sep 2020 */
000008 rdelete  SURROGAT BPX.SRV.CRMBVK3 /* 21 Jul 2021 */
000009 rdelete  SURROGAT BPX.SRV.GUEST /* 16 Feb 2010 */
000010 rdelete  SURROGAT BPX.SRV.INTERNAL /* 16 Feb 2010 */
```

Figure 4-5 Review the generated rdelete commands

zSecure Admin analyzes the allocated ACCESS data set to find SURROGAT class profiles for which no access decisions are found and then generates an RDELETE command for all SURROGAT profiles that were not used in any access decision during the past year.

Note: The profile creation date is reported as an eye catcher between CARLa comment signs (/*… */). When you execute the rdelete commands, the comment sections are ignored.

It is important to review the generated commands to make sure you do not delete profiles that must not be deleted despite their non-usage. There may be plausible reasons that (some of) these SURROGAT profiles are not used, such as:

- Some profiles are only used when your system is running from an alternate site.
- A security policy that requires all defined users and group profiles to have a top generic “hlq.**” dataset profile that must be defined independent of whether these profiles are used.
- Some profiles might only be used in emergency situations.
- Other reasons might be valid for your company.

zSecure Admin cannot take these company-specific security policies and implementations into account. Therefore, it is important that you add your company knowledge and experience to prevent the automatic deletion of profiles that must not be deleted.

4.4.2 Running the generated RACF commands to clean up resource profiles

In this section you will learn how to run the generated RACF commands for cleaning up the resource profiles that are not used.

As shown in the =NOTE= lines at the top of Figure 4-5, you can enter commands **GO** or **RUN** in the command line to execute the cleanup commands interactively. Alternatively, you can issue the command **SUB** or **SUBMIT** to submit a batch job to run the cleanup commands in the background. In that case, you access the zSecure Submit menu (see Figure 4-6).

```
                          zSecure Admin - Submit menu
Option ===> ____________________________________________________________________

1  View         View JCL
2  Edit         Edit JCL
3  Submit       Submit JCL for execution
4  Cancel       Do not submit the JCL
5  Select       Select an alternate set of input files


Batch input files   Active RACF backup, CKFREEZE, ACCESS data sets

Job statement information:  (Verify before proceeding)
> //CRMBTZ1A JOB ,'TOM ZEEHANDELAAR',TIME=(10),NOTIFY=&SYSUID,MSGCLASS=V
> ____________________________________________________________________________
> ____________________________________________________________________________
> ____________________________________________________________________________
```

Figure 4-6 zSecure Admin Submit menu

When you access the zSecure Admin Submit menu for the first time, the “Job statement information” section may not be populated yet. In that case, you must specify a job card that is valid on your installation. You can select the options **1**, for View JCL, or **2**, for Edit JCL, to access the JCL of the batch job prior to submitting the job with option **3**, for Submit JCL for execution.

Instead of executing or submitting the commands directly from within the CKRCMD work data set, you press **F3** to return to the Results panel (see Figure 4-7 on page 63). On this panel, depending on your preferences, you can run or submit the cleanup commands for SURROGAT class profiles from here.

```
                          zSecure Admin - Results       Enter R to run commands

COMMAND ===> _____________________________________________________ Scroll ===> CSR_

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 W Write file into seq. or partitioned data set
 M E-mail report

CKRCMD for the specified environments:

   Complex   Njenode   Rrsfnode   System     zSecNode #Lines
R  TVT6003   <LOCAL>   ?          ZS34       ?        126
_  RECOVERY  <LOCAL>   ?          ZS34       ?        591
****************************** Bottom of data ********************************
```

Figure 4-7 Run the cleanup commands in the foreground

After pressing Enter, the cleanup commands are executed against the offline RACF database. You automatically transfer to the CKRTSPRT work data set that reports the results of the executed RACF cleanup commands (see Figure 4-8).

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000000 Col 001 080
 Command ===> _____________________________________________ Scroll ===> CSR_
******************************** Top of Data ********************************
================================================================================
=== Multiple TSO command output file - scroll max down for overview ===
=== Input data set CRMBTZ1.DATA.C2R1DC6.CKRCMD                        ===
================================================================================
================================================================================
=== Commands for local node
================================================================================
/* CKRCMD file CKR1CMD complex TVT6003 generated  5 Nov 2024 11:17 */

============  5Nov24 11:18:10.40828 start record 2 ============
rdelete  SURROGAT *.SUBMIT /* 18 Jun 2021 */
ICH12002I RACLISTED PROFILES FOR SURROGAT WILL NOT REFLECT THE DELETION(S)
UNTIL

============  5Nov24 11:18:10.41639 start record 3 ============
rdelete  SURROGAT ALERT.SUBMIT /* 17 Mar 2014 */
ICH12002I RACLISTED PROFILES FOR SURROGAT WILL NOT REFLECT THE DELETION(S)
UNTIL

============  5Nov24 11:18:10.42267 start record 4 ============
rdelete  SURROGAT BCSCGB*.SUBMIT /* 16 Feb 2010 */
```

Figure 4-8 Review output of running cleanup commands

If one of the cleanup commands failed, a message “Command failed” shows up in the top right corner of your panel. In that case, a reference to the record number of the commands that failed is reported at the bottom of the CKRTSPRT work data set. Enter command **M** in the

command line, for maximum, and press **F8** to scroll to the bottom of the output (see Figure 4-9).

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000493 Col 001 080
 Command ===> ____________________________________________ Scroll ===> CSR

============  5Nov24 11:18:11.11977 start record 123 ============
rdelete  SURROGAT TWS.SUBMIT /* 26 Mar 2014 */
ICH12002I RACLISTED PROFILES FOR SURROGAT WILL NOT REFLECT THE DELETION(S)
UNTIL

============  5Nov24 11:18:11.12590 start record 124 ============
rdelete  SURROGAT U807018.SUBMIT /* 16 Feb 2010 */
ICH12002I RACLISTED PROFILES FOR SURROGAT WILL NOT REFLECT THE DELETION(S)
UNTIL

============  5Nov24 11:18:11.13163 start record 125 ============
rdelete  SURROGAT WZ903007.SUBMIT /* 16 Feb 2010 */
ICH12002I RACLISTED PROFILES FOR SURROGAT WILL NOT REFLECT THE DELETION(S)
UNTIL

============  5Nov24 11:18:11.13816 start record 126 ============
rdelete  SURROGAT Z714240.SUBMIT /* 16 Feb 2010 */
ICH12002I RACLISTED PROFILES FOR SURROGAT WILL NOT REFLECT THE DELETION(S)
UNTIL
=====================================================================
=== All commands completed successfully                          ===
=====================================================================
******************************* Bottom of Data ********************************
```

Figure 4-9 Confirm successful execution of cleanup commands

As expected, all cleanup commands are executed successfully. Pressing **F3** returns you to the results panel.

4.4.3 Recovering deleted resource profiles

In this section, you will learn how to review and save the RACF commands for recovering deleted resource profiles that are not used.

Note that the RECOVERY data set contains significantly more lines than the cleanup data set as reported in the `#Lines` column (see Figure 4-10 on page 65). This higher number of lines in the RECOVERY data set is because rebuilding a deleted profile requires multiple commands to redefine the resource profile and populate the ACL and CACL with their original permissions. You can use action commands **S**, **V**, **B**, or **E** to review or edit the RACF commands to recover the SURROGAT profiles that you just deleted from the offline RACF database. Please keep in mind that in the primary RACF database, these SURROGAT profiles still exist at this stage.

```
                          zSecure Admin – Results       Enter R to run commands
COMMAND ===> _____________________________________________ Scroll ===> CSR_

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 W Write file into seq. or partitioned data set
 M E-mail report

CKRCMD for the specified environments:

   Complex   Njenode   Rrsfnode   System     zSecNode #Lines
   TVT6003   <LOCAL>   ?          ZS34       ?        126
S  RECOVERY  <LOCAL>   ?          ZS34       ?        591
****************************** Bottom of data ********************************
```

Figure 4-10 Access recovery commands to rebuild deleted commands

The top of the RECOVERY work data set shows the **rdefine** commands to rebuild the SURROGAT profiles that you removed during your cleanup activity (see Figure 4-11). When you scroll down, you also encounter the required permit commands to restore the permitted IDs and access level that the original profile stored before you deleted it.

```
EDIT       CRMBTZ1.PROFS.RECOVERY.CKRCMD                 Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR_
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1RECV complex RECOVERY @#$@ generated  1 Nov 2024 -
000002 17:03 */
000003 rdefine SURROGAT *.SUBMIT owner(SYSAUTH) audit(failures(READ)) -
000004 uacc(NONE   )
000005 rdefine SURROGAT ALERT.SUBMIT owner(SYSAUTH) audit(all(READ)) -
000006 uacc(NONE   )
000007 rdefine SURROGAT BCSCGB*.SUBMIT owner(SYSAUTH) audit(all(READ)) -
000008 uacc(NONE   )
000009 rdefine SURROGAT BPX.SRV.ANONYMO owner(SYSAUTH) audit(all(READ)) -
000010 uacc(NONE   )
000011 rdefine SURROGAT BPX.SRV.CRMBRT1 owner(SYSAUTH) audit(all(READ)) -
000012 uacc(ALTER  )
000013 rdefine SURROGAT BPX.SRV.CRMBVK2 owner(SYSAUTH) audit(all(READ)) -
000014 uacc(NONE   )
000015 rdefine SURROGAT BPX.SRV.CRMBVK3 owner(SYSAUTH) audit(all(READ)) -
000016 uacc(NONE   )
000017 rdefine SURROGAT BPX.SRV.GUEST owner(SYSAUTH) audit(all(READ)) -
```

Figure 4-11 Review or edit recovery commands to rebuild the deleted SURROGAT profiles

The purpose of these RECOVERY commands is that they can be used to quickly restore profiles when the cleanup causes security issues. For that reason, it is suggested that when running cleanup commands against the primary RACF database, you save these RECOVERY commands to a permanent data set and put that data set in quarantine for some time, for example 3 - 12 months after the cleanup. That enables you to easily restore the

original SURROGAT profile should this be required in the (near) future. Press **F3** to return to the results panel.

Enter action command **W**, for Write, preceding the RECOVERY data set to save the commands in a sequential data set or partitioned data set of your preference (see Figure 4-12).

```
                          zSecure Admin – Results       Enter R to run commands
COMMAND ===> _____________________________________________ Scroll ===> CSR

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 W Write file into seq. or partitioned data set
 M E-mail report

CKRCMD for the specified environments:

   Complex   Njenode   Rrsfnode   System     zSecNode #Lines
_  TVT6003   <LOCAL>   ?          ZS34       ?        126
W  RECOVERY  <LOCAL>   ?          ZS34       ?        591
****************************** Bottom of data ********************************
```

Figure 4-12 Save the recovery commands for potential later use

Pressing Enter shows the panel that supports specifying in what data set or PDS, you want to save the recovery commands in this example, deleted SURROGAT class profiles (see Figure 4-13).

```
                      zSecure Admin - Results of last query
Command ===> ____________________________________________________________

Write the zSecure Admin+Audit for RACF message file to the following dataset:
Data set name . . . . . 'CRMB.T.RACF.RECOVERY.COMMANDS'_____________
Member  . . . . . SURROGAT
Disposition . . . . . 2  1. Append         2. Overwrite      3. Generate

Processing option after Write completed:
Go into Edit  . . . . . Y          (Y/N)
```

Figure 4-13 Specify location for saving your SURROGAT class recovery commands

Optionally, you can enter **Y** at option ‘Go into Edit’ to access the saved recovery commands in the specified location (see Figure 4-14 on page 67).

```
EDIT       CRMB.T.RACF.RECOVERY.COMMANDS(SURROGAT) - 01.00 Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001  /* CKRCMD file CKR1RECV complex RECOVERY @#$@ generated  1 Nov 2024 -
000002 17:03 */
000003 rdefine SURROGAT *.SUBMIT owner(SYSAUTH) audit(failures(READ)) -
000004 uacc(NONE   )
000005 rdefine SURROGAT ALERT.SUBMIT owner(SYSAUTH) audit(all(READ)) -
000006 uacc(NONE   )
000007 rdefine SURROGAT BCSCGB*.SUBMIT owner(SYSAUTH) audit(all(READ)) -
000008 uacc(NONE   )
000009 rdefine SURROGAT BPX.SRV.ANONYMO owner(SYSAUTH) audit(all(READ)) -
000010 uacc(NONE   )
000011 rdefine SURROGAT BPX.SRV.CRMBRT1 owner(SYSAUTH) audit(all(READ)) -
000012 uacc(ALTER  )
000013 rdefine SURROGAT BPX.SRV.CRMBVK2 owner(SYSAUTH) audit(all(READ)) -
000014 uacc(NONE   )
000015 rdefine SURROGAT BPX.SRV.CRMBVK3 owner(SYSAUTH) audit(all(READ)) -
000016 uacc(NONE   )
000017 rdefine SURROGAT BPX.SRV.GUEST owner(SYSAUTH) audit(all(READ)) -
```

Figure 4-14 Access the data set containing your SURROGAT class recovery commands

If you need to rebuild a SURROGAT profile in the future, you can execute only the relevant redefine and permit commands to restore the SURROGAT profile that you intend to restore. However, at this stage, you may not know whether this cleanup action could have a security impact. Press **F3** several times until you reach the zSecure Admin RACF Access Monitor panel.

4.4.4 Reporting potential security impact of deleted unused RACF profiles

In this section, you will learn how to report potential security impact of the deletion of the unused profiles.

The SURROGAT profiles that Access Monitor indicated were not used during the last year are now successfully deleted from the offline RACF database that you allocated as input. Next, you can use the Compare option that Access Monitor supports. This Compare function uses the identical access data set that you used for the cleanup as input, then simulates all recorded access events from last year against the allocated RACF input source. Your session uses the offline RACF database that no longer contains unused SURROGAT class profiles. The return code (SimRC) of each simulated access event is compared to the historic return code (AccRC) of the event.

To start the Compare function, specify option 2, for Compare and press Enter (see Figure 4-15 on page 68).

```
                              zSecure Admin - Access
Option ===> 2_____________________________________________________________________
                                                                        More:     +
1      Access          Access summary by user or profile
2      Compare         Compare monitored access against current RACF database
3      Permit usage    Permit usage information for current RACF database
4      Connect usage   Connect usage information for current RACF database
5      Profile usage   Profile usage information for current RACF database
6      Member usage    Member usage information for current RACF database
7      Global usage    Global usage summary for current RACF database
8      Remove          Remove unused profiles and authorizations
9      Cleanup         Remove permits, dataset and general resource profiles
V      ID verify       ID verification (logon/start/job-start) summary
I      ID usage        ID usage information for current RACF database
U      Unix events     Access summary for unix events by user, uid/gid, or path
```

Figure 4-15 Compare monitored access against current RACF database

On the Compare panel, you can specify that you want to restrict the Compare function to the SURROGAT class by specifying `surrogat` for option 'SAF resource class'.

Because you want to forecast the possible security impact of your cleanup commands, you can use the filters in the 'Comparison selection' section of this panel to omit occurrences where the historic AccRC is the same as the simulated SimRC. By removing the / in the Simulated results line preceding the option “Same”, your generated report only includes results where the historic and simulated return codes are not the same. That result might indicate that your cleanup commands will impact security because users have less or more access than before your cleanup of the unused SURROGAT profiles. Optionally, you can also specify 2 in the “Output/run options section” to summarize by resource rather than userid (see Figure 4-16).

```
                         zSecure Admin - Access - Compare
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . ________  (userid or EGN mask)
Complex . . . . . ________  (complex or EGN mask)
SAF resource class  . . surrogat  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________
Comparison selection
Simulated results . . . _  Same      /  Less       /  More
Profile used  . . . . . 3  1. Same   2. Different  3. All

Advanced selection criteria
_  Further selection  _  Date selection   _  Current RACF DB selection

Output/run options
2  1. Summarize by userid  2. Summarize by member class and profile
_  Show configured fields  _  Timezone (U/L/H)
_  Print format             Customize title     Send as e-mail
      Background run         Full page form
```

Figure 4-16 Report access events where historic and simulated access have a different return code

Press Enter to generate the simulation reports (see Figure 4-17).

```
                 zSecure Admin Display Selection        0 s elapsed, 0.1 s CPU
Command ===> ____________________________________________________ Scroll===> CSR_

  Name     Summary Records Title
_ SIMLESSD       0       0 ACCESS summary simulated access is less
S SIMMORED       1       1 ACCESS summary simulated access is more
******************************* Bottom of Data ********************************
```

Figure 4-17 Access simulation reports generated

The 0 records statistic that shows for report SIMLESSD indicates that the cleanup commands for the SURROGAT profiles does not impact security. All users that accessed SURROGAT resources during last year still get the same access after the cleanup of the unused SURROGAT profiles.

However, when you look at report SIMMORED, it shows 1 record where the simulated access decision allows more access as at the time of the event.

Specify action **S** to access the details of report SIMMORED (see Figure 4-18).

```
                                                                      Line 1 of 1
ACCESS summary simulated access is more
Command ===> ____________________________________________________ Scroll===> CSR_
Access monitor records for Classes like SURROGA 5 Nov 2024 11:34
   Occurrence Class    First occurrenc Last occurrence
            1 SURROGAT 12Mar2024 19:42 12Mar2024 19:42
   Occurrence Profile key used
            1 CRMBAH2.SUBMIT
   Occurrence Intent   Type   RetAll AccRC SimRC
            1 ALTER    Auth             8     4
   Occurrence Resource
            1 CRMBAH2.SUBMIT
   Occurrence Userid   Name
            1 CRMBAH2  SYSPROG2 AH
   Occurrence Complex  Syst RGPJCAVP GUGSOPGXO SOA PCSL First occurrenc Last oc
            1 TVT6003  ZS34     C                      12Mar2024 19:42 12Mar20
   Occurrence Timestamp
S           1 12Mar2024 19:42
******************************* Bottom of Data ********************************
```

Figure 4-18 Simulated access decision allows more access than historic access decision

This SIMMORED report shows 1 occurrence where the RC at the time of the event was 8 as reported in column “AccRC” whereas the simulated access RC, as shown in column “SimRC”, is 4. This SimRC 4 indicates that no profile is found in the allocated RACF database that protects SURROGAT class resource CRMBAH2.SUBMIT.

Specify command **S** in the last line and press Enter to access the details of the access record (see Figure 4-19 on page 70).

```
                                                                    Line 1 of 47
ACCESS summary simulated access is more
Command ===> _____________________________________________________ Scroll===> CSR_
Access monitor records for Classes like SURROGA 5 Nov 2024 11:34


  Access summary
_ Security complex name         TVT6003
_ RACF userid                   CRMBAH2  SYSPROG2 AH
  Access intent                 ALTER
  Record type                   Auth
  System name                   ZS34
  SAF resource class            SURROGAT
_ SAF resource name             CRMBAH2.SUBMIT
  Timestamp of last occurrence  12Mar2024 19:42
  Access count                  1

  Request flags                           Define flags
  RACFIND                                 Use internal RACROUTE AUTH
  Limit to generics             No        Verify
  Private/CSA (no global)       No        Propagated
  Bypass JESSPOOL profiles      No
  Command                       Yes
```

Figure 4-19 Review access record details

Notice that the Command flag reports “Yes”.

Position your cursor on field “Command” or “Yes” and press **F1** to access the field sensitive help information for this request flag (see Figure 4-20).

```
 CARLa field        : REQ_COMMAND
 Newlist type       : ACCCES
 Header default     : Cmd
 Field prefix header: Command

 This flag field (YES/NO) indicates whether the event occurred as the
 result of a RACF command. This can be because the command flag was set on
 the RACROUTE macro, or because the data collector found an active RACF
 command.

 The field is present in DEFINE and AUTH records.
```

Figure 4-20 Review help information about the meaning of the Command flag

The help information explains that the occurrence comes from a RACF command issued by the involved user CRMBAH2.

Press **F3** several times to add an extra filter to the simulation reports to suppress access events that are reported as the result of a RACF command, as this does not represent a real access event. Back on the Compare panel, activate the “Further selection” panel by tagging it with a /, and press Enter. Notice that the previous selection criteria that you used are saved in your ISPF session automatically (see Figure 4-21 on page 71).

```
                          zSecure Admin - Access - Compare      0.1 s CPU, RC=4
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . SURROGAT  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________
Comparison selection
Simulated results . . .    Same      /  Less       /  More
Profile used  . . . . . 3  1. Same   2. Different  3. All

Advanced selection criteria
/  Further selection  _  Date selection   _  Current RACF DB selection

Output/run options
1  1. Summarize by userid  2. Summarize by member class and profile
_  Show configured fields  _  Timezone (U/L/H)
_  Print format             Customize title     Send as e-mail
      Background run          Full page form
```

Figure 4-21 Rerun Compare function with additional selection filter

After pressing Enter, the Further selection panel appears. Specify an **N**, for No, to suppress access events from your reports that occurred as the result of a RACF command that was issued (see Figure 4-22).

```
                  zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________________
Access monitor records for Classes like SURROGAT



Select access records(Y/N/blank)
N  Use of RACF commands                  _  Retrieval of access allowed
_  Use of global access checking table   _  Bypass JESSPOOL profiles
_  Use of discrete profiles              _  ID was undefined during event
_  System special authority used         _  User had special attribute
_  Operations authority used             _  User had operations attribute
_  Installation exit used                _  User had (ro)auditor attribute
_  User requesting access is owner

Resource action  Intended access    Result             Program status
_  Define        __  _  1. Read     _  Success            Defined program
_  Delete              2. Update   _  No profile         Controlled program
_  Addvol              3. Control  _  Not authorized     Specific program
_  Chgvol              4. Alter    _  Other              Controlled library
```

Figure 4-22 Rerun Compare function with use of RACF commands suppressed

Press Enter to generate the compare reports again (see Figure 4-23 on page 72).

```
                  zSecure Admin Display Selection        0 s elapsed, 0.1 s CPU
Command ===> _____________________________________________ Scroll===> CSR_

  Name     Summary Records Title
_ SIMLESSD       0       0 ACCESS summary simulated access is less
_ SIMMORED       0       0 ACCESS summary simulated access is more
****************************** Bottom of Data ********************************
```

Figure 4-23 Access simulation reports regenerated

This result indicates that the cleanup of the SURROGAT profiles does not have a negative or positive security impact if the same access events that occurred last year for SURROGAT resources occur today against the cleaned RACF-Offline database.

This result does not mean with a 100% certainty that this cleanup will not have any security impact when you run it against the active primary RACF database. For example, historic SURROGAT access events occurring more than 1 year ago might still suffer from a violation when certain SURROGAT class profiles are no longer defined. But, at minimum, you can feel confident that your cleanup does not result in major incidents that cause important jobs to fail, started tasks crashing/running with reduced authorization, or subsystems, or applications that no longer function as expected.

Naturally, when you run the compare reports on your z/OS systems other issues might be reported. For example, these issues can be caused by changes that were made to the RACF database that you copied to define your RACF-Offline database but that are not covered in your 12 month rolling ACCESS data set. In these cases, you always need to investigate whether the issues reported in the SIMLESSD or SIMMORD reports are caused by the cleanup of the profiles, or they relate to a resource profile that you did not touch during your cleanup. If you run this same compare report against the active primary RACF database (that still contains the unused SURROGAT class profiles), the same occurrences must be reported. That indicates that an issue is not caused by your cleanup activities.

Sometimes, you might find occurrences in the SIMLESSD or SIMMORED reports that are related to profiles that you deleted. If for whatever reason, you cannot explain the issue, you can always decide not to delete the involved resource profile by removing the corresponding command(s) from the cleanup commands data set prior to execution.

When you are satisfied with the outcome of the compare reports, after analysis and resolving the reported issues, the final step is to end your RACF-Offline session and run the cleanup against the active primary RACF database.

Press **F3** several times until you reach the ISPF Primary Option Menu. Specify option **X** to leave ISPF and return to RACF-Offline (see Figure 4-24 on page 73).

```
                                                          ISPF Primary Option Menu
Option ===> X_____________________________________________________________________
                                          More:     +
0  Settings      Terminal and user parameters            User ID . : CRMBTZ1
1  View          Display source data or listings         Time. . . : 17:19
2  Edit          Create or change source data            Terminal. : 3278
3  Utilities     Perform utility functions               Screen. . : 1
4  Foreground    Interactive language processing         Language. : ENGLISH
5  Batch         Submit job for language processing      Appl ID . : ISR
6  Command       Enter TSO or Workstation commands       TSO logon : TSOZSEC
7  Dialog Test   Perform dialog testing                  TSO prefix: CRMBTZ1
9  IBM Products  IBM program development products        System ID : TVT6003
10 SCLM          SW Configuration Library Manager        MVS acct. : ACCT#
11 Workplace     ISPF Object/Action Workplace            Release . : ISPF 8.1
P  PRODUCTS      Dialogs for installed products
G  GROUP         Dialogs used with your organization
U  User          Your Own Dialogs
S  SDSF          System Display and Search Facility

C  Consul        Consul Risk Management applications

     Enter X to Terminate using log/list defaults
```

Figure 4-24 Exit ISPF

Specify what you want to do with your ISPF Log Data Set. Press Enter to reach the READY mode (see Figure 4-25).

```
                     Specify Disposition of Log Data Set
Command ===> ____________________________________________________________________
                                                                        More:     +
Log Data Set (CRMBTZ1.SPFLOG1.LIST) Disposition:
Process Option . . . . 2  1. Print data set and delete
                          2. Delete data set without printing
                          3. Keep data set - Same
                             (allocate same data set in next session)
                          4. Keep data set - New
                             (allocate new data set in next session)
Batch SYSOUT class . . _______________
Local printer ID or
writer-name  . . . . . _________________
Local SYSOUT class . . _______________

List Data Set Options not available

Press ENTER key to complete ISPF termination.
Enter END command to return to the primary option menu.

Job statement information:  (Required for system printer)
 ===> ____________________________________________________________________________
 ===> ____________________________________________________________________________
 ===> ____________________________________________________________________________
```

Figure 4-25 Specify Log Data Set disposition option

When you return to the RACF-Offline interface, specify command **end** and press Enter to exit your RACF-Offline session (see Figure 4-26).

```
CRMBTZ1.SPFLOG1.LIST has been deleted.
 B8R200A Enter RACF Command or "END"
end
```

Figure 4-26 Exit your RACF-Offline session

You must now be back in TSO READY mode. Next, specify command ISPF here once more and press Enter. You return to the **ISPF Primary Option Menu** again. But this time, you are no longer in a RACF-Offline session.

4.4.5 Cleaning up unused profiles from the active primary RACF database

In this section, you will learn how to cleanup unused profiles from the active primary RACF database.

Your next step is to run the cleanup of the unused SURROGAT profiles against the active primary RACF database. You can use the following two options to clean up the unused SURROGAT profiles:

- Rerun the logged RACF commands from your RACF-Offline session against the active RACF database
- Rerun option AM.8.1 for SURROGAT class and execute the commands against the active RACF database

Rerun the logged RACF commands against the active RACF database

You can decide to rerun the **RDELETE** commands from your RACF-Offline log data set CRMB.T.RACF.OFFLINE.B8RLOG. You might prefer to run the commands in the background using a batch job.

Alternatively, you can run the collected commands interactively by copying the RACF commands collected in data set CRMB.T.RACF.OFFLINE.B8RLOG to your CKRCMD work data set. Issue command result in the command line of any zSecure panel and press Enter to access the zSecure work data sets. Use action command **E**, for edit, preceding the CKRCMD data set (see Figure 4-27 on page 75) and press Enter.

```
                          zSecure Admin - Results

Command ===> ____________________________________________________________________

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 M E-mail report
 W Write file into seq. or partitioned data set

Enter a selection in front of a highlighted line below:
_  SYSPRINT  messages
_  REPORT    printable reports
_  CKRTSPRT  output from the last TSO command(s)
E  CKRCMD    queued TSO commands

_  CKR2PASS  queued commands for zSecure Admin+Audit for RACF
_  COMMANDS  zSecure Admin+Audit for RACF input commands from last query
_  SPFLIST   printable output from PRT primary command
_  OPTIONS   set print options
```

Figure 4-27 Access the CKRCMD work data set in edit mode

Specify command **COPY** 'crmb.t.racf.offline.b8rlog' in the command line and press Enter to copy the logged RACF-Offline commands to the CKRCMD work data set (see Figure 4-28).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                      Data set copied
Command ===> ____________________________________________ Scroll ===> CSR_
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001 CKGRACF show myaccess
000002 RDELETE  SURROGAT *.SUBMIT /* 18 Jun 2021 */
000003 RDELETE  SURROGAT ALERT.SUBMIT /* 17 Mar 2014 */
000004 RDELETE  SURROGAT BCSCGB*.SUBMIT /* 16 Feb 2010 */
000005 RDELETE  SURROGAT BPX.SRV.ANONYMO /* 16 Feb 2010 */
000006 RDELETE  SURROGAT BPX.SRV.CRMBRT1 /* 15 Apr 2014 */
000007 RDELETE  SURROGAT BPX.SRV.CRMBVK2 /* 23 Sep 2020 */
000008 RDELETE  SURROGAT BPX.SRV.CRMBVK3 /* 21 Jul 2021 */
000009 RDELETE  SURROGAT BPX.SRV.GUEST /* 16 Feb 2010 */
000010 RDELETE  SURROGAT BPX.SRV.INTERNAL /* 16 Feb 2010 */
000011 RDELETE  SURROGAT BPX.SRV.OMVSKERN /* 24 Jul 2019 */
000012 RDELETE  SURROGAT BPX.SRV.PRIVATE /* 16 Feb 2010 */
000013 RDELETE  SURROGAT BPX.SRV.PUBLIC /* 16 Feb 2010 */
000014 RDELETE  SURROGAT BPX.SRV.WEBADM /*  7 Jul 2003 */
000015 RDELETE  SURROGAT CICSUSER.DFHINSTL /* 16 Feb 2010 */
000016 RDELETE  SURROGAT CKNSERV%.SUBMIT /* 18 May 2010 */
000017 RDELETE  SURROGAT CRMA.DFHINSTL /* 16 Feb 2010 */
```

Figure 4-28 Copy cleanup commands from RACF-Offline log data set to CKRCMD work data set

You can execute the commands interactively with command **GO** or **RUN** in the command line or submit them in the background with command **SUB** or **SUBMIT**. That concludes the cleanup of unused SURROGAT profiles.

Rerun option AM.8.1 against the active RACF database

For this option, you must switch the input of your zSecure Admin session to use a different RACF input source (active primary RACF, active backup, recent UNLOAD, or a recent full RACF backup) than your offline RACF database with option **SE.1** (see Figure 4-29).

```
                          zSecure Admin – Setup – Input files        Row 1 of 3
Command ===> ____________________________________________________ Scroll ===> CSR_

Description . . . . Active RACF backup, CKFREEZE, ACCESS data sets________
Complex . . . . . . TVT6003_   Version  . . . .  ____

Enter data set names and types.         Type END or press F3 when complete.
Enter dsname with .* to get a list      Type SAVE to save set, CANCEL to quit.
Valid line commands: E L I R D          Type REFRESH to submit unload  job.

   Data set name or DSNPREF=, or Unix file name            Type or ?   NJE node
_  -DATA SET NAME DYNAMICALLY OBTAINED FROM COMMON STORAGE ACT.BACK    ________
_  'CRMB.T.CKFREEZE'                                       CKFREEZE    ________
_  'C2PACMON.TVT6003.Y12MON'                               ACCESS      ________
****************************** Bottom of data ********************************
```

Figure 4-29 Select input set with current active RACF database allocated

Next, you can rerun option **AM.8.1** again against the active RACF database that you allocated with option SE.1, that still contains the unused SURROGAT class profiles (see Figure 4-30).

```
                              zSecure  Admin  –  Results  Enter R to run commands
COMMAND ===> ____________________________________________ Scroll ===> CSR_

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 W Write file into seq. or partitioned data set
 M E-mail report

CKRCMD for the specified environments:

   Complex   Njenode   Rrsfnode   System     zSecNode #Lines
R  TVT6003   <LOCAL>   ?          ZS34       ?        126
_  RECOVERY  <LOCAL>   ?          ZS34       ?        591
****************************** Bottom of data ********************************
```

Figure 4-30 Rerun SURROGAT class cleanup commands against active RACF database

As expected the same 126 lines of clean up commands for the SURROGAT class are generated for the complex named TVT6003. When you issue action command **R**, for Run commands, and press Enter, the unused SURROGAT class profiles are then successfully removed from the active RACF database.

4.4.6 Cleaning up other unused security definitions from the active primary RACF database

In this section, you will learn how to cleanup other unused security definitions from active primary RACF database.

You can continue cleaning up unused resource profiles from other classes using this same methodology and following the same steps multiple times.

Similarly, you can also execute a similar scenario with option **AM.8.2** to remove unused permissions from the ACLs of resource profiles that according to the consolidated year of Access Monitor records are not used to access protected resources.

Option **AM.8.3** supports removing user to group connections where according to Access Monitor records the user did not use their group connection to access resources that are permitted to their connect group.

In addition, you can also use the options **AM.V**, for ID verify, or **AM.I**, for ID usage to analyze the usage of defined user IDs. These options can be used to report user IDs that represent good candidates for being deleted because these are not used during the last year. However, these options do not support the generation of automatically generated RACF commands to delete these unused user IDs including all their references. zSecure Admin supports a function to delete user IDs including the resource profiles that they own, RACF variables they are part of, master and user catalog entries, and even the data sets that start with that user ID. You can even code a scheduled CARLa script that automatically scans for user IDs whose last used date is more than 1 year old and generates automatic RACF commands to perform a clean sweep of all data sets, catalog pointer(s), dataset and general resource profile(s), and the concerning user profiles. But that script is outside the scope of this publication.

This concludes the walk through of use case: “Removing unused resource profiles”.

5

Convert generic and specific access to group-based access

Another key aspect of maintaining a RACF environment in the long term is to ensure that security policies align with a group or role-based access control model. It can be simpler and easier to establish either profiles that are generic enough to satisfy all users or simple enough to satisfy a specific user making a request, ensuring that this philosophy is followed is vital to the health of the environment and the hardening of your security posture.

In this chapter, we discuss the following topics:

- 5.1, “Challenges with generic and specific access permissions” on page 80
- 5.2, “Moving towards role-based access control” on page 80
- 5.3, “Establishing groups that accurately reflect user roles” on page 83
- 5.4, “Converting to group-based access” on page 83

5.1 Challenges with generic and specific access permissions

Specific access permissions, like personalized access permissions, can be secure, but make databases difficult to maintain and audit as they provide little context about what job role a particular user performs that requires access to a similar application. Additionally, users may change job roles, leading to a migrating away from the needs prescribed in the initial permission. Generic access permissions, like universal access to all users, do lead to more simplified database management, but lack the concept of individual accountability and can often lead to harmful security gaps.

Some reasons for the existence of generic access permissions are as follows:

- When new applications are configured for use, users may not have the full scope to understand which job roles have a valid requirement for what access level, leading to requests for RACF administrators to permit access based solely on specific user IDs.
- Occasionally, business imperatives to get an application released (or live) are given priority over securing it appropriately. That strategy often leads to permitting generic accesses that make certain to allow the application to function without security incidents, rather than using more job role-based accesses that are more secure.
- Security administrators often feel confident that types of protections and measures that they take are sufficiently secure. This feeling prevents the urgency for setting up for role-based controls to migrate away from using existing generic access paths. Internal and external auditors generally do not agree with the use of generic access paths to a company's sensitive resources and demand that more granular access rules are implemented instead of generic access paths.
- Existing security infrastructure, profile definitions, and access lists can predate modern security best practices. Often times administrators inherit environments that were set up a long time ago, when security policies may have had inaccurate, outdated, or incomplete information if they existed at all.

As a result, both generic and personalized access permissions pose maintenance challenges or even problems in passing security audits. Moving to role-based access control addresses many of these concerns.

5.2 Moving towards role-based access control

With RACF, access that users or applications have to resources can be based on multiple factors. Access permissions can be based on the user's identity (user ID), the group profile(s) a user is connected to, or the universal access level that is defined for a resource. As a best practice, RACF administrators must try to keep their environments aligned with the concept of role-based access control (RBAC). RBAC means that administrators must prioritize using groups that represent a job role in RACF to establish least privilege access to required resources. The following guidelines to manage your environment should be considered:

- Minimize the use of universal and generic permissions
- Minimize use of personalized permissions
- Minimize the assignment of the OPERATIONS attribute
- Minimize the use of the WARNING attribute
- Minimize the use of global access checking tables

Note that it is best when the use of generic and personalized accesses are minimized to those occurrences where it is still required and approved by management. There are probably valid reasons why certain resource profiles may need to have a universal access permission

defined. But, when speaking more broadly for generic and application-specific resources, the implementation of these best practices should be considered.

5.2.1 Minimize the use of universal and generic permissions

A common form of a generic access in RACF is the universal access (UACC) setting that all resource profiles store. The UACC setting designates the access level that applies to all users, applications, and processes in a RACF environment that request access for a resource that is protected by a particular profile where the requestor is not permitted directly or indirectly through their connect groups. Naturally, the use of UACC to provide access is not very secure and poses some obvious security risks. It allows for broad uncontrolled access to utilize a particular resource. Generally, from a security perspective, these types of uncontrolled accesses must be avoided as this approach lacks individual accountability. Even user IDs that are not defined in the local RACF database, that can access your system through means of NJE (Network Job Entry) or RJE (Remote Job Entry) are allowed to use resources when the UACC setting allows it. In the context of least privilege, RACF installations should have a security policy that states: By default, the UACC setting for all resource profiles must be NONE and deviations from this policy must approved.

Another common form of generic access in RACF is a permission that applies to all authenticated users. This is denoted in the form of the access being permitted to ID(*) which equivalates to all RACF-authenticated user IDs. While ID(*) is slightly more restrictive than UACC, it still grants authority to all authenticated users in the RACF environment, which also represents a lax security policy. It is preferred to explore which job roles require that access level to the protected resource and issue a permit command to the collection of corresponding role-based groups instead. Except for resources that are truly required for all RACF-authenticated users to be able to access in your environment.

Universal and generic access allow uncontrolled access to the resource(s) that a resource profile protects. The use of universal and generic access conflicts with both the concept of “zero trust” and “least privilege”. If anyone can access a resource, the system trusts them with this authority, thus this setting demonstrates a non-zero amount of trust in the environment. Likely, a subset of users or programs that do not require this access to function properly, which qualifies the universal access an unnecessary privilege permitted to these users.

As mentioned, some resource classes and profiles exist that have a valid business justification for setting the universal access level to a value that exceeds NONE. For example, the resource classes NODES, DIGTCERT, and XFACILIT have profiles that can require a universal access setting exceeding NONE. For more information on these classes and profiles and why they require this non-standard UACC access, refer to the *RACF Security Administrator’s Guide*.

5.2.2 Minimize use of personalized permissions

Another common form of access in RACF is to assign a permission to specific users. User-based access can often be done under the guise of increased security, as this is the most granular way to administer a RACF environment. Because this granular form of access is secure, prioritizing user-based access does not violate either of our security principles of “least privilege” or “zero trust”. Best practice is to avoid this because of the administrative complications of managing an environment in such a way. When accesses are based on users, one must rely on supplemental reasoning and documentation to support a need for this access. When managing an environment with hundreds of users requesting access to thousands of profiles, this implementation quickly becomes untenable. Auditing permissions by each individual user’s justified access becomes a monumental effort. Additionally, tracking

every user's job role and ensuring that accesses are updated for multiple profiles as organizational changes are made causes exponentially more work for security administrators than managing accesses through role-based groups. User-based access is coupled with other hurdles such as "what if the one person with access to this resource is unavailable". These administrative and logistical challenges make this sort of security management functionally impossible.

With role-based groups, these accesses can be consolidated, and often in ways that can be self-documenting. As an example, you would have to know why USER1 and USER2 needed access to a profile, but if both belong to group DEV1 and DEV1 holds this access, the access could be clearly rooted in the needs of the developer role. If there are changes made to the accesses required for the developer role, only the permissions of DEV1 need to change, not each individual developer user. Lastly, if a user leaves a role, they just need to be removed from one role-based group and possibly added to role-based group that represents the user's new job role. With a role-based group implementation there is no need remove or define explicit user permissions. Using role-based groups instead of users as the primary method of permitting access is the only way to manage a RACF environment at scale.

5.2.3 Minimize the assignment of the OPERATIONS attribute

User IDs that are assigned the OPERATIONS attribute are allowed ALTER access to resources in the dataset and the general resource classes that are configured to honor the OPERATIONS attribute. When an OPERATIONS user accesses such a resource in a class, ALTER access is allowed by default. If a user ID or one of their connect groups is permitted another access level, the access level permitted on the ACL or CACL is used instead. Auditors often report the use of the OPERATIONS attribute as non-controlled access. Use of OPERATIONS violates the idea of "least privilege" by assigning broad and permissive access. This also violates the idea of a "zero trust" environment by effectively assigning new default permissions for affected users. Replacing the assignment of the OPERATIONS attribute with appropriate permissions to the resources that an OPERATIONS user accesses in their job role improves the security posture of your system.

5.2.4 Minimize the use of the WARNING attribute

The activation of the WARNING attribute on a resource profile causes RACF to bypass access verification checks for this resource. The purpose of the WARNING attribute is to be used in the implementation phase when it is unclear which users require what access level to a resource. When a non-authorized user requests access to a resource that is protected by a profile in WARNING mode, ALTER access is allowed despite the fact that neither the user nor their connect groups are permitted access to the resource. The access is then logged to SMF allowing security administrators to populate the ACL and CACL with the appropriate permissions. Then, after a period of monitoring SMF records and adding permissions, security administrators can remove the WARNING attribute without causing a security impact. Obviously, the use of WARNING also violates the "least privilege" and "zero trust" environment by establishing broad and permissive access for all users.

5.2.5 Minimize the use of global access checking tables

Global access checking (GAC) tables can be implemented to boost the performance of access verification checks by quickly allowing a particular access level to certain resources in RACF resource classes without checking RACF profiles and without logging to SMF. Most companies probably have public resources that they publicly allow READ access to. For example, banks allow all users to READ their interest rates to attract more customers.

However, the GAC tables must not allow access to resources when access to the resource would not be allowed for certain users, or when the company needs to be able to report who accessed that resource. Therefore, the use of GAC tables and the list globally accessible resources needs the be carefully monitored. Improper use of these tables violates the principle of “zero trust” as it allows all users to access the specified resource.

5.3 Establishing groups that accurately reflect user roles

Businesses and organizations are steering their security policies towards role-based access control in RACF, but migrating to this state can pose a complex challenge for RACF administrator. As discussed previously, it is difficult for RACF security administrators to know or research whether deletion of security definitions results in a business impact because started tasks, jobs, applications, or information systems no longer work correctly due to a lack of authorization. When migrating from a more generic approach to a more specific one, these concerns are key. Additionally, when moving from user-based access to role-based access, it is a challenge to align the intended access with the user’s proper role to avoid opening up security holes when these security changes are made.

Traditional approaches to these migrations of RACF permissions are to log all successful access to all resources to SMF. Then, an administrator goes through the data for the generic profiles to identify accessing users, and then cross-reference these with common groups. This solution is simply not feasible because of the sheer amount of SMF records that this implementation would produce, the required CPU time, the storage associated with it, and the monumental manual work of associating users and groups involved.

With zSecure Access Monitor, you can generate reports based on captured access attempts that highlight user-based, ID(*)-based permissions or UACC-based access. You can use the “Cleanup” utility to generate appropriate commands to automatically convert ID(*) permissions or UACC access events to group-based permissions. Then, with RACF-Offline, you can test new group definitions and permissions against historic access attempts to resources to confirm that the conversion commands do not negatively impact production. This does mean that for conversions from user-based access to group-based access, a security administrator must manually review the access reports and determine what role-based groups to use or define, which users to add to those groups, and what access level to permit to which profiles for the role-based groups. This process must be done in consultation with organizational guidelines and security policies anyway to ensure that the new role-based groups accurately reflect appropriate roles for the involved users.

5.4 Converting to group-based access

The details for the general use case preparation steps using automated (semi-) conversion and cleanup features in zSecure Admin are documented in 2.4, “General preparation steps for the use cases” on page 16, including these steps:

- Creating a RACF-Offline database
- Starting a RACF-Offline session
- Logging on to RACF-Offline session
- Resetting the RACF-Offline log data set
- Starting and preparing your zSecure Admin session

After completion of these steps, you can continue with the subsequent steps to convert generic and specific access to group-based access:

- Reporting successful access allowed through the UACC setting
- Converting generic UACC access to group-based access
- Reporting the effect of the UACC conversion commands
- Reporting successful access allowed through ID(*)
- Converting generic ID(*) access to group-based access
- Converting generic OPERATIONS access to group-based access

5.4.1 Reporting successful access allowed through the UACC setting

In this section, you will learn how to report historic successful access that was allowed through the UACC setting of a resource profile.

The allocated access monitor records can be used to analyze and report which users successfully accessed resources because the UACC setting allowed the requested access during the last year. You can use option **AM.1**, for Access, to report which users accessed resources through the UACC setting of the protecting resource profile. By default, you can generate access overview reports for all users and resource classes. In this example, the report is restricted to the **OPERCMDS** class by specifying `opercmds` at option “SAF resource class”. To limit the report output to only successful access using UACC, you must also activate the options “Further selection” and “Current RACF DB selection” in the “Advanced selection criteria” section (see Figure 5-1).

To research which users get access to resources through the UACC setting, select option **3**, for “Summary by simulated authorization used”. This summary type groups access events that occur through the UACC setting of a resource profile together in a single display.

```
                        zSecure Admin - Access - Access
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . OPERCMDS  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
3  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  _  Show simulated fields   _  Timezone (U/L/H)
_  Print format             Customize title           Send as e-mail
      Background run         Full page form
```

Figure 5-1 Report historic access to resources in the OPERCMDS class

After you press Enter, the further selection panel appears. For this analysis report, you are only interested in reporting successful access events. Select option “Success” in the “Result” section with action command **/** or **S** (see Figure 5-2 on page 85) and press Enter.

```
                  zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________
Access monitor records for Classes like OPERCMDS




Select access records(Y/N/blank)
_  Use of RACF commands                  _  Retrieval of access allowed
_  Use of global access checking table   _  Bypass JESSPOOL profiles
_  Use of discrete profiles              _  ID was undefined during event
_  System special authority used         _  User had special attribute
_  Operations authority used             _  User had operations attribute
_  Installation exit used                _  User had (ro)auditor attribute
_  User requesting access is owner

Resource action  Intended access    Result             Program status
_  Define        __  _  1. Read     /  Success         _  Defined program
_  Delete              2. Update   _  No profile       _  Controlled program
_  Addvol              3. Control  _  Not authorized   _  Specific program
_  Chgvol              4. Alter    _  Other            _  Controlled library
```

Figure 5-2 Report only successful access events in the OPERCMDS class

Next, the activated “Current RACF DB selection” panel appears (see Figure 5-3). On this panel, you need to specify that you want your report to only show successful access through the UACC setting. Select option “UACC” in the “Authority used in current DB” section with action command **/** or **S** and press Enter.

```
                  zSecure Admin - Access - Current Database
Command ===> ____________________________________________________________
Access monitor records for Classes like OPERCMDS, Successes
Specify simulated fields (Current DB effect) selection criteria:
Group(s) used for access  _____________________________  (group(s) or filter)
Essential group(s)  . . . _                              (Y/N/blank)
Not using group(s)  . . . _____________________________  (group(s) or filter)
Profile owned by  . . . . ________                       (id or filter)
RACF return code  . . . . __  ______  (operator: > >= < <= = <> ^=)

Authority used in current DB
_  APF          _  GRP_OPER     _  ID_USER      _  PRIVTRUS     _  SYS_OPER
_  CLAUTH       _  GRP_SPEC     _  INACTIVE     _  PROTFAIL     _  SYS_SPEC
_  CREATE       _  GRP_UACC     _  NO_CDT       _  QUALOWN      /  UACC
_  DFLTRC       _  ID_GROUP     _  NO_PROF      _  RESTRICTED   _  UNPROT
_  GLOBAL       _  ID_STAR      _  NOTHING      _  SPOOLRCVR    _  WARNING

User attributes in current DB (Y/N/blank)
_  ID present           _  ID Revoked
_  ID has special       _  ID has operations    _  ID has (ro)auditor
```

Figure 5-3 Report only successful OPERCMDS access events that used UACC authority

If successful UACC access to OPERCMDS resources occurred during the last year, your report shows which users accessed OPERCMDS resources through UACC authority (see Figure 5-4).

```
                 IBM Security zSecure ACCESS summary    0 s elapsed, 0.1 s CPU
Command ===> ____________________________________________________ Scroll===> CSR_
Access monitor records for Classes like OPERCMD 6 Nov 2024 14:34
   Via        Occurrence First occurrenc Last occurrence
   UACC              124  2Nov2023 12:55 12Sep2024 23:36
   Occurrence Userid   Name                          First occurrenc Last occur
S_         101 CRMBAH1  SYSPROG1 AH                    4Jan2024 19:25 12Sep2024
__           1 CRMBAH2  SYSPROG2 AH                   15Jan2024 15:42 15Jan2024
__          22 CRMBEP1  SYSPROG1 EP                    2Nov2023 12:55 19Apr2024
****************************** Bottom of Data ******************************
```

Figure 5-4 Successful access through UACC to OPERCMDS class resources

The report reveals which users successfully accessed OPERCMDS resources through UACC authority. Specify action command **S**, and press Enter to review the access event details (see Figure 5-5).

```
                  IBM Security zSecure ACCESS summary              Line 1 of 8
Command ===> ____________________________________________________ Scroll===> CSR_
Access monitor records for Classes like OPERCMD 6 Nov 2024 14:41
   Via        Occurrence First occurrenc Last occurrence
   UACC              124  2Nov2023 12:55 12Sep2024 23:36
   Occurrence Userid   Name                          First occurrenc Last occur
          101 CRMBAH1  SYSPROG1 AH                     4Jan2024 19:25 12Sep2024
   Occurrence Intent    Type   RetAll AccRC SmRC
          101 READ      Auth              0    0
   Occurrence Class
          101 OPERCMDS
   Occurrence Resource                                          Profile key us
          101 MVS.MCSOPER.CRMBAH1                               MVS.MCSOPER.*
   Occurrence Complex  Syst RGPJCAVP GUGSOPGXO SOA PCSL First occurrenc Last oc
          101 TVT6003  ZS34          G                   4Jan2024 19:25 12Sep20
   Occurrence Timestamp
__          4  4Jan2024 19:25
__         23 16Jan2024 19:02
__          5 18Jan2024 14:45
__         35 12Mar2024 13:51
__          1 12Mar2024 21:07
__          5 11Sep2024 20:28
__          7 12Sep2024 15:27
__         21 12Sep2024 23:36
```

Figure 5-5 Successful access through UACC to OPERCMDS class resources details

From the “Profile key used” column that reports MVS.MCSOPER.*, you can conclude that the UACC setting of that OPERCMDS profile is at least READ. In this example, zooming in on the other reported users revealed that they all got READ access through the UACC of OPERCMDS profile MVS.MCSOPER.*. However, in practice, your RACF database might contain multiple OPERCMDS profiles with a lax UACC setting. In that case, you encounter other OPERCMDS profiles with a UACC setting that exceeds NONE in this report. In the next section, you learn how you can use zSecure Admin to convert access through UACC to group-based access permissions.

Important: Remember that the access monitor records only show the users that used the access path that the current UACC setting allowed. Keep in mind that this UACC implementation implies that all other RACF-authenticated, Network Job Entry (NJE), and Remote Job Entry (RJE) users are also allowed to successfully access these reported OPERCMDS resources with the same access level. Hopefully, in most cases the allowed access level through UACC does not exceed READ, but that might not always be the case. zSecure Admin supports features to (semi) automatically convert historic access that was allowed through a UACC setting, or an ID(*) permission, to group-based permissions and connections.

5.4.2 Converting generic UACC access to group-based access

In this section, you will learn how to generate and execute automatic RACF commands for converting generic UACC access to group-based access.

You can use option **AM.9**, for Cleanup, to automatically generate RACF commands to convert access that is currently allowed through the UACC setting. To start with converting access

through UACC to appropriate group permissions and connections, select option **4**, for UACC, and press Enter (see Figure 5-6).

```
                        zSecure Admin - Access - Cleanup
Option ===> 4_____________________________________________________________

1    User permit     Redundant permits to userids
2    Dataset         Redundant dataset profiles
3    Empty           Generic profiles without matching disk or tape datasets
4    UACC            Generate permits/connects to convert UACC access
5    ID(*)           Generate permits/connects to convert ID(*) access









Output options
_  Background run
```

Figure 5-6 Select option 4 UACC to convert UACC access to group-based permissions & connections

In this section, you learn how to convert the historic access through the UACC setting of OPERCMDS profile MVS.MCSOPER.*. If your system contains multiple resource profiles in the OPERCMDS or other resource classes with a UACC settings exceeding NONE, you can use this same strategy for converting UACC access to group-based permissions and connections using zSecure Admin.

Suppose that during your research into the historical use of UACC setting for OPERCMDS resources, you discovered that the users that accessed these resources were all z/OS systems programmers. The UACC conversion function allows you to define a new group to use for converting the historic UACC or use an already defined Role Based Access Control (RBAC) group, for systems programmers in your system. In the system that is used in this example, the group CRMB is defined as the RBAC group for z/OS systems programmers.

Specify OPERCMDS at option “SAF resource class” to restrict the UACC conversion to resources in the OPERCMDS class. Since you already established the RBAC group for systems programmers that exists, you decide to use group **CRMB** by specifying **CRMB** at option “Group for permit/connect” to convert to (see Figure 5-7). Press Enter to generate the UACC conversion commands.

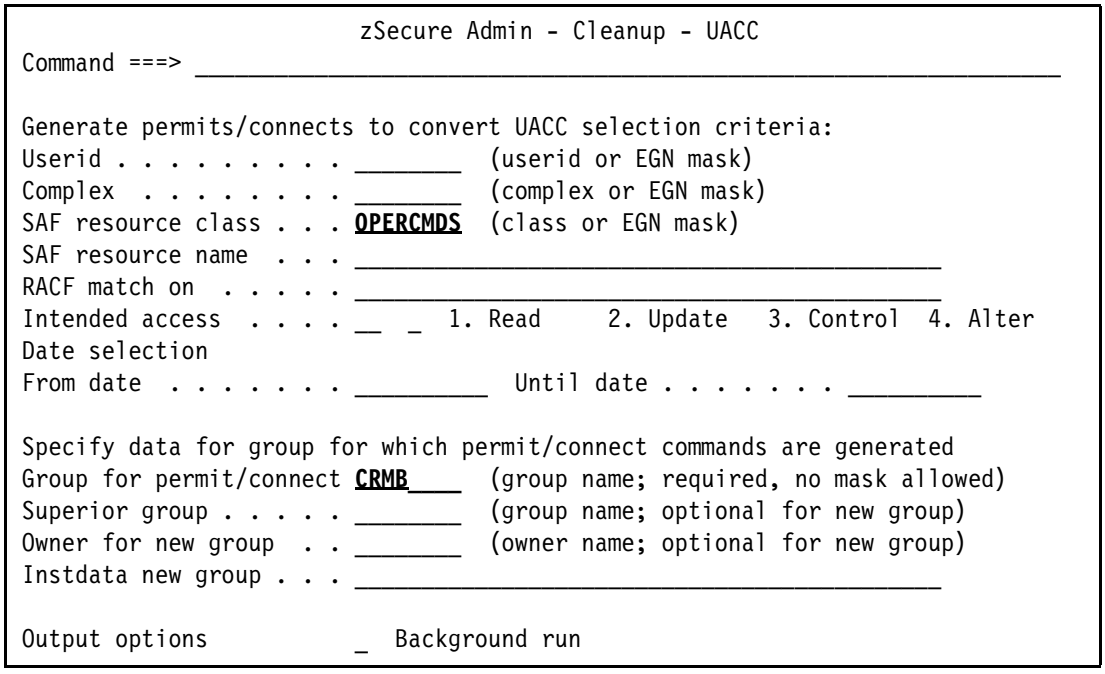


Figure 5-7 Convert historic access via UACC setting to existing role-based access group CRMB

The required conversion commands are successfully generated as shown in Figure 5-8 on page 90. As expected, connect commands to group **CRMB** are generated for the 3 users that you encountered in your access report earlier. When the users are already connected to group **CRMB**, you can remove the connect command prior to running the commands. However, leaving the connect commands in the CKRCMD work data set does not cause an issue when you run them anyway. The connect command runs successful but does not change the connection except when the original connection has a different group authority or connect privileges. Those connection settings will then be adjusted in accordance with generated connect command. If some of the users that historically used the OPERCMDS resources that profile MVS.MVSOPER.* protects are not systems programmers and they should not have accessed this resource, you can decide to delete the connect commands for these users. Naturally, these deleted connect commands can cause that a violation for these users occurs when they attempt to access this resource in the future after you reduce the UACC setting from READ to NONE.

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                          0.0 s CPU, RC=4
Command ===> go_____________________________________________ Scroll ===> CSR_
****** **************************** Top of Data ****************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated  6 Nov 2024 15:39 */
000002  /* No addgroup cmd created because group CRMB     already exists */
000003
000004  connect  CRMBAH1  group(CRMB) auth(use)
000005  connect  CRMBAH2  group(CRMB) auth(use)
000006  connect  CRMBEP1  group(CRMB) auth(use)
000007
000008  permit MVS.MCSOPER.* class(OPERCMDS) id(CRMB) access(READ)
000009
000010  setropts refresh raclist(OPERCMDS)
000011
****** **************************** Bottom of Data ****************************
```

Figure 5-8 UACC conversion commands for OPERCMDS resources successfully generated

When you are satisfied with your conversion commands, you can type **GO** or **RUN** in the command line and press Enter to execute the commands interactively. If you prefer to run these commands in the background with a batch job, you can use command `SUB` or `SUBMIT` to execute the commands in a batch job. In this example, the **GO** command is issued to run the commands interactively.

After pressing Enter, the UACC conversion commands are executed against the offline RACF database. You automatically transfer to the CKRTSPRT work data set that reports the results of the executed RACF conversion commands (see Figure 5-9 on page 90).

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000000 Col 001 080
 Command ===> _____________________________________________ Scroll ===> CSR_
******************************** Top of Data ********************************
================================================================================
=== Multiple TSO command output file - scroll max down for overview ===
=== Input data set CRMBTZ1.DATA.C2R1DC6.TEMP.CKRCMD                  ===
================================================================================




================================================================================
=== Commands for local node
================================================================================
/* CKRCMD file CKR1CMD complex TVT6003 generated  6 Nov 2024 15:39 */
/* No addgroup cmd created because group CRMB     already exists */

============  6Nov24 16:04:11.00019 start record 3 ============
connect  CRMBAH1  group(CRMB) auth(use)

============  6Nov24 16:04:11.02476 start record 5 ============
connect  CRMBAH2  group(CRMB) auth(use)
```

Figure 5-9 Review output of UACC setting running conversion commands

If one of the conversion commands failed, a message “Command failed” shows up in the top right corner of your panel. In that case, a reference to the record number of the commands that failed is reported at the bottom of the CKRTSPRT work data set. Enter command `M` in the command line, for maximum, and press **F8** to scroll to the bottom of the output (see Figure 5-10 on page 91).

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000015 Col 001 080
 Command ===> ____________________________________________ Scroll ===> CSR_
============  6Nov24 16:04:11.00019 start record 3 ============
connect  CRMBAH1  group(CRMB) auth(use)

============  6Nov24 16:04:11.02476 start record 5 ============
connect  CRMBAH2  group(CRMB) auth(use)

============  6Nov24 16:04:11.04182 start record 6 ============
connect  CRMBEP1  group(CRMB) auth(use)

============  6Nov24 16:04:11.05980 start record 7 ============
permit MVS.MCSOPER.* class(OPERCMDS) id(CRMB) access(READ)
ICH06011I RACLISTED PROFILES FOR OPERCMDS WILL NOT REFLECT THE UPDATE(S) UNTIL
A

============  6Nov24 16:04:11.06309 start record 9 ============
setropts refresh raclist(OPERCMDS)
B8R400I SETROPTS Command currently not supported
================================================================================
=== All commands completed successfully                                    ===
================================================================================
****************************** Bottom of Data ******************************
```

Figure 5-10 Confirm successful execution of UACC conversion commands

As expected, all UACC conversion commands executed successfully. Note, that command `setropts refresh raclist(OPERCMDS)` is reported as currently not supported.

Because you use RACF-Offline, a SETROPTS refresh command is not required. In fact, RACF-Offline does not even support the SETROPTS REFRESH command. Optionally, you can remove the command from your CKRCMD work data set prior to running the commands. Pressing **F3** returns you to the zSecure results panel (see Figure 5-11 on page 92).

```
                          zSecure Admin - Results       Enter R to run commands
=ra.r_____________________________________________________________________

The following selections are supported:
 B Browse file                S Default action (for each file)
 E Edit file                  R Run commands
 P Print file                 J Submit Job to execute commands
 V View file                  M E-mail report
 W Write file into seq. or partitioned data set

Enter a selection in front of a highlighted line below:
_  SYSPRINT  messages
_  REPORT    printable reports
_  CKRTSPRT  output from the last TSO command(s)
_  CKRCMD    queued TSO commands
_  CKR2PASS  queued commands for zSecure Admin for RACF
_  COMMANDS  zSecure Admin for RACF input commands from last query
_  SPFLIST   printable output from PRT primary command
_  OPTIONS   set print options
```

Figure 5-11 Jump to RACF administration panel for general resource profiles

Tip: If the timespan of the allocated access monitor records covers last month only, you might postpone reducing of the UACC setting at this point. When you run the same analysis report next month, the users that are now permitted via the **CRMB** group no longer get access through the UACC setting. Perhaps other users might be reported that accessed MVS.MCSOPER.* resources via the UACC setting. Hypothetically, you can repeat the monthly monitoring report until no other users are reported to get access through the UACC(READ) setting anymore. At that time, you can decide to change the UACC setting from READ to NONE.

Your next step is to reduce the UACC setting for OPERCMDS profile MVS.MCSOPER.* from READ to NONE. To increase efficiency, you could have added a manually coded RACF command to change the UACC setting of involved profile(s) to NONE to your CKRCMD work data set prior to running the UACC conversion commands.

However, this document illustrates an alternative UI supported function to accomplish this task. Specify command `=ra.r` in the command line to instruct zSecure Admin to jump to the **RA.R** panel and press Enter.

Specify `OPERCMDS` at option “Class name”, `MVS.MVSOPER.*` at option “Resource profile”, and specify option **2**, for exact (see Figure 5-12 on page 93), and press Enter.

```
                  zSecure Admin - RACF - Resource Selection
Command ===> ____________________________________________________   _ start panel

_   Add new general resource profile or segment
Show general profiles that fit all of the following criteria
Class name  . . . . opercmds     (class or filter)
Resource profile  . mvs.mcsoper.*_____________________________________________
_____________________________________________________________  2 1 EGN mask
Owned by  . . . . . ________     (group or userid, or filter)      2 Exact
Installation data . ____________________________ (substring or *)   3 Match
                                                                   4 Any match
Additional selection criteria
_  Profile fields     _  Access list      _  Segment presence  _  Absence
_  Audit settings

Output/run options
_  Show segments      _  All             /  Enable full ACL   _  Specify scope
_  Show differences   /  Summarize by class
_  Print format          Customize title     Send as e-mail
      Background run     Full detail form    Sort differently     Narrow print
      Print ACL          Resolve to users    Incl operations      Print names
```

*Figure 5-12 List OPERCMDS class profile MVS.MCSOPER.**

That query shows a display with the current settings for OPERCMDS profile MVS.MCSOPER.*. As expected, the current UACC is set to READ. Move your cursor to the UACC column and overtype READ with the desired value NONE (see Figure 5-13).

```
                                                      0 s elapsed, 0.0 s CPU
               zSecure Admin General resource overview
Command ===> ____________________________________________________ Scroll===> CSR_
Class OPERCMDS, key mvs.mcsoper.*              6 Nov 2024 16:25
   Class    Profile key                              T UACC    Owner    S/F W
__ OPERCMDS MVS.MCSOPER.*                           G NONE    SYSAUTH  R R _
******************************** Bottom of Data ********************************
```

Figure 5-13 Overtype the UACC of OPERCMDS profile MVS.MCSOPER. with desired value*

When you press Enter, the appropriate RACF command to adjust the UACC setting is generated. Depending on your zSecure Admin command confirmation setting, the command is immediately executed or, as in this example, your zSecure session is configured to confirm generated commands prior to execution. The command confirmation option, and whether to queue or execute immediately after confirmation, for your zSecure session is configured with option **SE.4**, for Setup Confirm.

Important: Confirm the generated commands prior to execution to prevent mistakes as well as to enable learning the RACF command syntax while using zSecure Admin.

On the command confirmation panel, select option **1**, for EXECUTE RACF command (see Figure 5-14 on page 94), and press Enter to run the command.

```
                        zSecure Admin - Confirm command
Command ===> ____________________________________________________________




Confirm or edit the following command
ralter OPERCMDS MVS.MCSOPER.* uacc(NONE)
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________



Command execution . 1  1. EXECUTE RACF command
                       2. EXECUTE CKGRACF command (allows use of Reason)
                       3. ASK administrator to execute CKGRACF command
                       4. REQUEST CKGRACF command for later execution
                       5. WITHDRAW CKGRACF command




Reason . . . . . . . ____________________________________________________
Press ENTER to continue or END to cancel the command
```

Figure 5-14 Confirm and execute the generated command to change the UACC setting to NONE

That action returns you to your original display listing OPERCMDS profile MVS.MCSOPER.*. It shows that your command successfully modified the UACC that now reports NONE (see Figure 5-15).

```
                                                          Successfully modified
               zSecure Admin General resource overview
Command ===> ____________________________________________________ Scroll===> CSR
Class OPERCMDS, key mvs.mcsoper.*              6 Nov 2024 16:41
   Class    Profile key                              T UACC    Owner    S/F W
__ OPERCMDS MVS.MCSOPER.*                           G NONE    SYSAUTH  R R _
****************************** Bottom of Data ********************************
```

Figure 5-15 Change of UACC setting to NONE successfully executed

Depending on your Setup Confirm configuration, when you press F3, you might see the panel shown in Figure 5-16 on page 95.

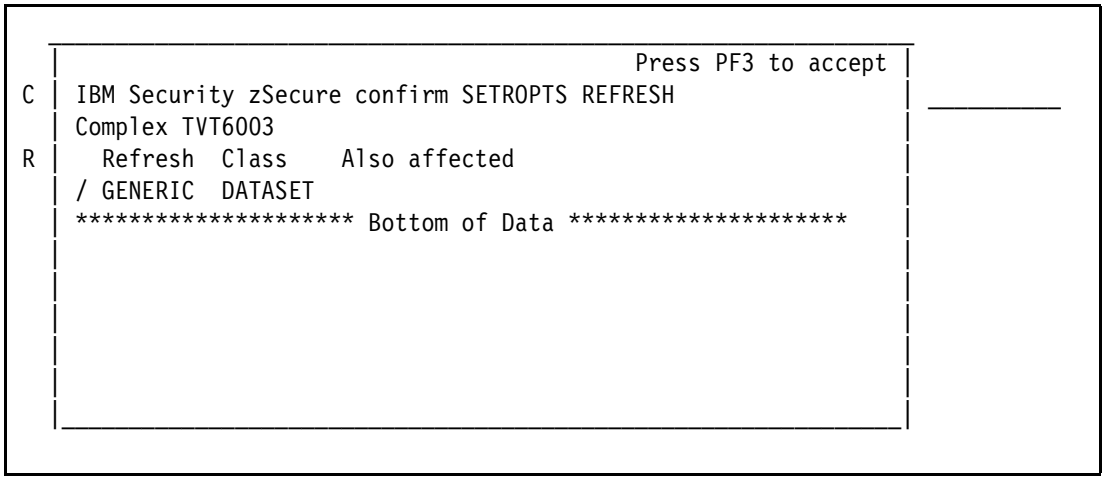


Figure 5-16 Required SETROPTS refresh command generated

RACF-Offline does not support the SETROPTS REFRESH command. Remove the **/** before you press **F3** to indicate that you do **not** want to run the SETROPTS REFRESH command at this stage.

5.4.3 Reporting the effect of the UACC conversion commands

In this section, you will learn how to report the effect of the UACC conversion commands.

Now that you reduced the UACC setting from READ to NONE, users must no longer be allowed universal access to OPERCMDS class resources that are protected by OPERCMDS class profile MVS.MCSOPER.*. You can use option AM.1, for Access, again to run the same report that you used earlier to analyze which users can successfully access OPERCMDS resources through the UACC setting. You can use the selections as shown in Figure 5-17.

```
                        zSecure Admin - Access – Access
Command ===> ____________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . ________  (userid or EGN mask)
Complex . . . . . ________  (complex or EGN mask)
SAF resource class  . . OPERCMDS  (class or EGN mask)
SAF resource name . . . mvs.mcsoper.*_____________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
3  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  /  Show simulated fields   _  Timezone (U/L/H)
_  Print format               Customize title           Send as e-mail
      Background run          Full page form
```

Figure 5-17 Report historic successful access to OPERCMDS resources with prefix MVS.MCSOPER

After you press Enter, the further selection panel appears. For this analysis report, you are only interested to report successful access events. Select option “Success” in the “Result” section with action command **/** or **S** (see Figure 5-18), and press Enter.

```
                    zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________
Access monitor records for Classes like OPERCMDS




Select access records(Y/N/blank)
_  Use of RACF commands                 _  Retrieval of access allowed
_  Use of global access checking table  _  Bypass JESSPOOL profiles
_  Use of discrete profiles             _  ID was undefined during event
_  System special authority used        _  User had special attribute
_  Operations authority used            _  User had operations attribute
_  Installation exit used               _  User had (ro)auditor attribute
_  User requesting access is owner

Resource action  Intended access    Result             Program status
_  Define        __  _  1. Read     /  Success         _  Defined program
_  Delete              2. Update   _  No profile       _  Controlled program
_  Addvol              3. Control  _  Not authorized   _  Specific program
_  Chgvol              4. Alter    _  Other            _  Controlled library
```

Figure 5-18 Report only successful access events to OPERCMDS resources

Next, the activated “Current RACF DB selection” panel appears. On this panel, you need to specify that you want your report to only show successful access through UACC. Select

option “UACC” in the “Authority used in current DB” section with action command **/** or **S** (see Figure 5-19), and press Enter.

```
                  zSecure Admin - Access - Current Database
Command ===> ____________________________________________________________
Access monitor records for Classes like OPERCMDS, Successes
Specify simulated fields (Current DB effect) selection criteria:
Group(s) used for access  _____________________________  (group(s) or filter)
Essential group(s)  . . . _                              (Y/N/blank)
Not using group(s)  . . . _____________________________  (group(s) or filter)
Profile owned by  . . . . ________                       (id or filter)
RACF return code  . . . . __  ______  (operator: > >= < <= = <> ^=)

Authority used in current DB
_  APF          _  GRP_OPER     _  ID_USER      _  PRIVTRUS     _  SYS_OPER
_  CLAUTH       _  GRP_SPEC     _  INACTIVE     _  PROTFAIL     _  SYS_SPEC
_  CREATE       _  GRP_UACC     _  NO_CDT       _  QUALOWN      /  UACC
_  DFLTRC       _  ID_GROUP     _  NO_PROF      _  RESTRICTED   _  UNPROT
_  GLOBAL       _  ID_STAR      _  NOTHING      _  SPOOLRCVR    _  WARNING

User attributes in current DB (Y/N/blank)
_  ID present           _  ID Revoked
_  ID has special       _  ID has operations    _  ID has (ro)auditor
```

Figure 5-19 Report only successful OPERCMDS access events that used UACC authority

If successful access to OPERCMDS resources occurred during the last year, your report shows which users accessed what OPERCMDS resources through UACC authority (see Figure 5-20).

```
                           zSecure Admin - Access - Access      No records found
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . OPERCMDS_ (class or EGN mask)
SAF resource name . . . mvs.mcsoper.*_____________________________
RACF match on . . . . . _____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
3  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  /  Show simulated fields   _  Timezone (U/L/H)
_  Print format               Customize title           Send as e-mail
      Background run          Full page form
```

Figure 5-20 No successful access events are found that used UACC authority

As expected, no simulated access to OPERCMDS resources starting with prefix MVS.MCSOPER using UACC is reported. The users CRMBAH1, CRMBAH2, and CRMBEP1 no longer use the UACC setting to get access but their connection to group **CRMB** to access OPERCMDS resources starting with prefix MVS.MCSOPER.

You can change your selection to verify this conclusion. Remove the **/** at option “Current RACF DB selection” (see Figure 5-21 on page 98), and press Enter.

```
                        zSecure Admin - Access - Access
Command ===> ____________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . ________  (userid or EGN mask)
Complex . . . . . ________  (complex or EGN mask)
SAF resource class  . . OPERCMDS_ (class or EGN mask)
SAF resource name . . . mvs.mcsoper.*_____________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   _  Current RACF DB selection

Output/run options
3  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  /  Show simulated fields   _  Timezone (U/L/H)
_  Print format               Customize title           Send as e-mail
      Background run          Full page form
```

Figure 5-21 Remove the option Current RACF DB selection

Now some access to OPERCMDS resources is reported through user and group permissions. Because, you already know that the involved users get access through group **CRMB**, select the report named ID_GROUP with an **/** or **S** action command (see Figure 5-22), and press Enter.

```
                  IBM Security zSecure ACCESS summary    0 s elapsed, 0.1 s CPU
Command ===> ____________________________________________ Scroll===> CSR_
Access monitor records for Classes like OPERCMD 7 Nov 2024 16:52
   Via        Occurrence First occurrenc Last occurrence
__ ID_USER        170232  3Apr2024 07:09 31Oct2024 22:45
S_ ID_GROUP          338  2Nov2023 12:55 31Oct2024 07:56
****************************** Bottom of Data ********************************
```

Figure 5-22 Review access events for OPERCMDS resources through group permissions

That shows the details of the users that accessed OPERCMDS resources through a permission to one of their connect groups. If all is well, the users that you connected to group **CRMB** are included in the report. Select one of the involved users with action command **/** or **S** (see Figure 5-23 on page 99), and press Enter.

```
                IBM Security zSecure ACCESS summary             Line 1 of 10
Command ===> _____________________________________________ Scroll===> CSR_
Access monitor records for Classes like OPERCMD 7 Nov 2024 16:52
   Via        Occurrence First occurrenc Last occurrence
   ID_GROUP          338  2Nov2023 12:55 31Oct2024 07:56
   Occurrence Userid   Name                          First occurrenc Last occur
S_        101 CRMBAH1  SYSPROG1 AH                    4Jan2024 19:25 12Sep2024
__          1 CRMBAH2  SYSPROG2 AH                   15Jan2024 15:42 15Jan2024
__         22 CRMBEP1  SYSPROG1 EP                    2Nov2023 12:55 19Apr2024
__         42 CRMBJK1  SYSPROG1 JK                    9Nov2023 12:23 31Oct2024
__         22 CRMBLU1  SYSPROG1 LU                   27Nov2023 07:56  5Jan2024
__          7 CRMBMK1  SYSPROG1 MK                   25Oct2024 10:23 30Oct2024
__          1 CRMBPH1  SYSPROG1 PH                    7Mar2024 10:21  7Mar2024
__          1 CRMBRL1  SYSPROG1 RL                   23Feb2024 13:35 23Feb2024
__        133 CRMBVK1  SYSPROG1 VK                    6Dec2023 13:45 25Oct2024
__          8 CRMBVK2  SYSPROG2 VK                   17Nov2023 14:24 12Apr2024
****************************** Bottom of Data ********************************
```

Figure 5-23 Review access event details for OPERCMDS resources

On the next panel, issue action command **/** or **S** again until you reach the “Access summary” detail panel (see Figure 5-24). Use command `M`, for maximum, and press **F8** to scroll to the bottom of the details.

```
                IBM Security zSecure ACCESS summary             Line 40 of 48
Command ===> _____________________________________________ Scroll===> CSR_
Access monitor records for Classes like OPERCMD 7 Nov 2024 16:52

  Current RACF database effect
  RACF return code current DB   0
  Authority used in current DB  ID_GROUP
_ Grp permit used in current DB CRMB
  RACF Profile type current DB  GENERIC
_ RACF class and profile in DB  OPERCMDS MVS.MCSOPER.*
_ Profile owner                 SYSAUTH
  Installation data
  Creation/definition date      16 Feb 2010
****************************** Bottom of Data ********************************
```

Figure 5-24 Access to OPERCMDS class MVS.MCSOPER. is now allowed through group CRMB*

That report proves that your conversion worked successfully, and the users CRMBAH1, CRMBAH2, and CRMBEP1 are now allowed to access the resource without using the UACC setting.

Note: Your access summary report might also show other user IDs that were not involved in the conversion. These users were already connected to another group than CRMB that was already permitted access to the MVS.MCSOPER.* resources.

When you are satisfied with the outcome of your conversion commands, the final step is to end your RACF-Offline session and apply the same UACC conversion commands to your active primary RACF database.

Press **F3** several times until you reach the ISPF Primary Option Menu. Specify option X to leave ISPF and return to your RACF-Offline session (see Figure 5-25).

```
                          ISPF Primary Option Menu
Option ===> X_____________________________________________________________________
                                          More:     +
0  Settings      Terminal and user parameters            User ID . : CRMBTZ1
1  View          Display source data or listings         Time. . . : 17:19
2  Edit          Create or change source data            Terminal. : 3278
3  Utilities     Perform utility functions               Screen. . : 1
4  Foreground    Interactive language processing         Language. : ENGLISH
5  Batch         Submit job for language processing      Appl ID . : ISR
6  Command       Enter TSO or Workstation commands       TSO logon : TSOZSEC
7  Dialog Test   Perform dialog testing                  TSO prefix: CRMBTZ1
9  IBM Products  IBM program development products        System ID : TVT6003
10 SCLM          SW Configuration Library Manager        MVS acct. : ACCT#
11 Workplace     ISPF Object/Action Workplace            Release . : ISPF 8.1
P  PRODUCTS      Dialogs for installed products
G  GROUP         Dialogs used with your organization
U  User          Your Own Dialogs
S  SDSF          System Display and Search Facility

C  Consul        Consul Risk Management applications

     Enter X to Terminate using log/list defaults
```

Figure 5-25 Exit ISPF

Next, you must specify what you want to do with your ISPF Log Data Set. Select the option of your preference (see Figure 5-26), and press Enter to reach the READY mode.

```
 Specify Disposition of Log Data Set
Command ===> ____________________________________________________________
                                                                More:     +
Log Data Set (CRMBTZ1.SPFLOG1.LIST) Disposition:
Process Option . . . . 2  1. Print data set and delete
                          2. Delete data set without printing
                          3. Keep data set - Same
                             (allocate same data set in next session)
                          4. Keep data set - New
                             (allocate new data set in next session)
Batch SYSOUT class . . _______________
Local printer ID or
writer-name  . . . . . _________________
Local SYSOUT class . . _______________

List Data Set Options not available

Press ENTER key to complete ISPF termination.
Enter END command to return to the primary option menu.

Job statement information:  (Required for system printer)
 ===> ____________________________________________________________________
 ===> ____________________________________________________________________
 ===> ____________________________________________________________________
```

Figure 5-26 Specify Log Data Set disposition

You return to the RACF-Offline interface. Specify command `end` and press Enter to exit your RACF-Offline session (see Figure 5-27).

```
 CRMBTZ1.SPFLOG1.LIST has been deleted.
 B8R200A Enter RACF Command or "END"
end
```

Figure 5-27 Exit your RACF-Offline session

You must now be back in TSO READY mode. Specify command `ISPF` here once more and press Enter to return to the ISPF Primary Option Menu again. But this time, you are no longer in a RACF-Offline session.

You can repeat the same steps to generate and execute the UACC conversion commands against the active primary RACF database.

Tip: You can replay the same scenario as often as is needed to also convert UACC settings from other OPERCMDS profiles or resource profiles from different resource classes that have a UACC setting that exceeds NONE.

5.4.4 Reporting successful access allowed through ID(*)

In this section, you will learn how to report historic successful access that was allowed through a permission to ID(*).

When, you want to convert access permitted to ID(*), you can also use the same approach as you used for converting successful access through UACC. However, this time you must use different filters to report about successful access through a permission to ID(*).

Once again, you can use option **AM.1**, for Access, and activate similar options as you used for reporting access through UACC setting. By default, you can generate access overview reports for all resource classes that contain resource profiles containing a permission to ID(*). When your report includes all resource classes, activate options “Further selection” and “Current RACF DB selection” in the “Advanced selection criteria” section. Because your goal is the report simulated access through the permission to ID(*), you can use option 2, for “Summary by member class and profile” (see Figure 5-28). This summary generates a display per resource class and profile that contain a permission to ID(*).

```
                        zSecure Admin - Access - Access
Command ===> ____________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . ________  (userid or EGN mask)
Complex . . . . . ________  (complex or EGN mask)
SAF resource class  . . ________  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
2  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  _  Show simulated fields   _  Timezone (U/L/H)
_  Print format             Customize title           Send as e-mail
      Background run         Full page form
```

Figure 5-28 Report historic access to resources summarized by class and profile

After you press Enter, the further selection panel appears. For this analysis report, you are only interested to report successful access events. Select option “Success” in the “Result” section with action command **/** or **S** (see Figure 5-29 on page 103), and press Enter.

```
                  zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________________
All access monitor records




Select access records(Y/N/blank)
_  Use of RACF commands                  _  Retrieval of access allowed
_  Use of global access checking table   _  Bypass JESSPOOL profiles
_  Use of discrete profiles              _  ID was undefined during event
_  System special authority used         _  User had special attribute
_  Operations authority used             _  User had operations attribute
_  Installation exit used                _  User had (ro)auditor attribute
_  User requesting access is owner

Resource action  Intended access    Result             Program status
_  Define        __  _  1. Read     /  Success         _  Defined program
_  Delete           _  2. Update    _  No profile      _  Controlled program
_  Addvol           _  3. Control   _  Not authorized  _  Specific program
_  Chgvol           _  4. Alter     _  Other           _  Controlled library
```

Figure 5-29 Report only successful access events

Next, the activated “Current RACF DB selection” panel appears. On this panel, you need to specify that you want your report to only report successful access through a permission to ID(*). Select option “ID_STAR” in the “Authority used in current DB” section with action command **/** or **S** (see Figure 5-30), and press Enter.

```
                  zSecure Admin - Access - Current Database
Command ===> ____________________________________________________________________
All access monitor records, Successes
Specify simulated fields (Current DB effect) selection criteria:
Group(s) used for access  _____________________________  (group(s) or filter)
Essential group(s)  . . . _                              (Y/N/blank)
Not using group(s)  . . . _____________________________  (group(s) or filter)
Profile owned by  . . . . ________                        (id or filter)
RACF return code  . . . . __  ______  (operator: > >= < <= = <> ^=)

Authority used in current DB
_  APF          _  GRP_OPER     _  ID_USER      _  PRIVTRUS     _  SYS_OPER
_  CLAUTH       _  GRP_SPEC     _  INACTIVE     _  PROTFAIL     _  SYS_SPEC
_  CREATE       _  GRP_UACC     _  NO_CDT       _  QUALOWN      _  UACC
_  DFLTRC       _  ID_GROUP     _  NO_PROF      _  RESTRICTED   _  UNPROT
_  GLOBAL       /  ID_STAR      _  NOTHING      _  SPOOLRCVR    _  WARNING

User attributes in current DB (Y/N/blank)
_  ID present           _  ID Revoked
_  ID has special       _  ID has operations    _  ID has (ro)auditor
```

Figure 5-30 Report only the successful access events that used ID(*) authority

The generated report shows an overview for all resource classes that contain a profile that includes a permission to ID(*) on the ACL that was used by users to access the protected

resources. When you want to zoom in to the DATASET class, specify action command / or S to review the involved profiles in the DATASET class (see Figure 5-31), and press Enter.

```
                IBM Security zSecure ACCESS summary    0 s elapsed, 0.2 s CPU
Command ===> _____________________________________________ Scroll===> CSR_
All access monitor records, Successes, Current  8 Nov 2024 10:41
   Occurrence Class    First occurrenc Last occurrence
/_      26772 DATASET   1Nov2023 04:30 31Oct2024 21:05
__       2012 SDSF     17Nov2023 14:24 31Oct2024 22:49
__      45846 SERVAUTH 17Nov2023 14:25 31Oct2024 22:49
****************************** Bottom of Data ********************************
```

Figure 5-31 Review DATASET profiles with permission to ID(*)

The overview shows the, in this case, three DATASET class profiles that contain a permit to ID(*). You can zoom in to the details of the DATASET profile USER.*.** with action command **/** (see Figure 5-32), and press Enter.

```
                     IBM Security zSecure ACCESS summary
Command ===> _____________________________________________ Scroll===> CSR_
All access monitor records, Successes, Current  8 Nov 2024 10:37
   Occurrence Class    First occurrenc Last occurrence
        26772 DATASET   1Nov2023 04:30 31Oct2024 21:05
   Occurrence Profile key used
__      13011 CATALOG.**
__      10226 CRMA.X.**
/_       3535 USER.*.**
****************************** Bottom of Data ********************************
```

Figure 5-32 Zoom in to the details of DATASET profile USER..***

When you use action command **/** or **S** a couple of times more until you reach the detail panel of the access record and when you scroll to the bottom, you encounter the following information in the “Current RACF database effect” section (see Figure 5-33).

```
                IBM Security zSecure ACCESS summary            Line 39 of 48
Command ===> _____________________________________________ Scroll===> CSR_
All access monitor records, Successes, Current  8 Nov 2024 10:41


  Current RACF database effect
  RACF return code current DB   0
  Authority used in current DB  ID_STAR
  Grp permit used in current DB
  RACF Profile type current DB  GENERIC
_ RACF class and profile in DB  DATASET  USER.*.**
_ Profile owner                 CRMA
  Installation data
  Creation/definition date      23 Apr 2010
****************************** Bottom of Data ********************************
```

Figure 5-33 Current RACF database effect reveals that ID(*) is used for the access

However, when you scroll up one page with **F7**, you see the “Access time effect” section. That section shows the involved RACF class and profile. Specify action command **P**, for Show profile (see Figure 5-34), and press Enter.

```
                IBM Security zSecure ACCESS summary            Line 21 of 48
Command ===> ____________________________________________________ Scroll===> CSR_
All access monitor records, Successes, Current  8 Nov 2024 10:41

  Access time effect
P RACF class and profile        DATASET  USER.*.**
  RACF Profile type used        GENERIC
  Access allowed                READ
  RACF return code              0
```

Figure 5-34 Access the involved DATASET profile from access record details panel

That action runs a recursive call from your report to list the settings of DATASET class profile USER.*.** (see Figure 5-35).

```
                zSecure Admin DATASET Overview         0 s elapsed, 0.0 s CPU
Command ===> ____________________________________________________ Scroll===> CSR_
key USER.*.**                                  8 Nov 2024 11:02


_ Identification                                                    TVT6003
  Profile name                 USER.*.**
  Type                         GENERIC
  Volume serial list
_ Effective first qualifier    USER                        DATASET OWNER?
_ Owner                        SYSAUTH                     AUTHORIZATION GRO
  Installation data            ____________________________________________

  User     Access  ACL id   When               RI Name                DfltGrp
_ - any -  READ    *
_ -group-  ALTER   SUPERUSR
_ -group-  ALTER   SUPPORT
_ -group-  ALTER   SYSPROG
_ -group-  ALTER   SYS1
```

Figure 5-35 Show details of DATASET profile USER..** from the allocated RACF input source*

As expected, indeed ID(*) is permitted READ access on the ACL of this profile. When you would inspect the other reported DATASET class (or the SDSF and SERVAUTH) profiles, they will all show a permission to ID(*) on the ACL or CACL (Conditional ACL).

5.4.5 Converting generic ID(*) access to group-based access

This section explains the necessary steps for generating and executing automatic RACF commands for converting generic ID(*) access to group-based access.

Important: If you logged off from your RACF-Offline session earlier, now is the time to start a RACF-Offline session again before generating the ID(*) conversion commands.

Suppose that your auditors insist that you convert the ID(*) permissions in DATASET class profiles to a new group named IDSTAR with owner and superior group CRMB.

Use option **AM.9.5**, for Cleanup – ID(*), to generate the appropriate RACF conversion commands (see Figure 5-36).

```
                        zSecure Admin - Cleanup - ID(*)
Command ===> ____________________________________________________________

Generate permits/connects to convert ID(*) selection criteria:
Userid . . . . . . . . . ________  (userid or EGN mask)
Complex  . . . . . . . . ________  (complex or EGN mask)
SAF resource class . . . dataset_  (class or EGN mask)
SAF resource name  . . . ____________________________________________
RACF match on  . . . . . ____________________________________________
Intended access  . . . . __  _  1. Read     2. Update   3. Control  4. Alter
Date selection
From date  . . . . . . . _________  Until date . . . . . . . _________

Specify data for group for which permit/connect commands are generated
Group for permit/connect idstar__  (group name; required, no mask allowed)
Superior group . . . . . crmb____  (group name; optional for new group)
Owner for new group  . . crmb____  (owner name; optional for new group)
Instdata new group . . . Conversion of ID(*) dataset permissions_____

Output options           _  Background run
```

Figure 5-36 Generate ID(*) conversion commands for new group named IDSTAR

When you press Enter, the ID(*) conversion commands are successfully generated in the CKRCMD work data set (see Figure 5-37).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                    Columns 00001 00072
Command ===> ____________________________________________ Scroll ===> CSR_
****** ***************************** Top of Data ********************************
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated  8 Nov 2024 11:21 */
000002  addgroup IDSTAR sup(CRMB) owner(CRMB) data('CONVERSION OF ID(*) -
000003 DATASET PERMISSIONS')
000004
000005  connect  CKNSERVE  group(IDSTAR) auth(use)
000006  connect  CKNSERV1  group(IDSTAR) auth(use)
000007  connect  CRMBAB2  group(IDSTAR) auth(use)
000008  connect  CRMBAH1  group(IDSTAR) auth(use)
000009  connect  CRMBAH2  group(IDSTAR) auth(use)
000010  connect  CRMBEP1  group(IDSTAR) auth(use)
000011  connect  CRMBNAT  group(IDSTAR) auth(use)
000012  connect  CRMBSG2  group(IDSTAR) auth(use)
```

Figure 5-37 ID(*) conversion commands for new group IDSTAR successfully generated

At the top of the CKRCMD data set is the ADDGROUP command and the appropriate CONNECT commands for users that historically used the permission to ID(*) for DATASET profiles during the last year (see Figure 5-38 on page 107). When you scroll to the bottom of

the CKRCMD work data set, you encounter the three permit commands to group IDSTAR that permit the same access level to group IDSTAR as is permitted to ID(*).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                    Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR_
000019  connect  PAGENT  group(IDSTAR) auth(use)
000020  connect  SSHDAEM  group(IDSTAR) auth(use)
000021
000022  permit  'CATALOG.**' GEN id(IDSTAR) access(UPDATE)
000023  permit  'CRMA.X.**' GEN id(IDSTAR) access(READ)
000024  permit  'USER.*.**' GEN id(IDSTAR) access(READ)
000025
000026  setropts refresh generic(DATASET)
****** *************************** Bottom of Data ***************************
```

Figure 5-38 ID(*) conversion commands for new group IDSTAR successfully generated (continued)

Before running the commands, you must make sure that all users that are about to be connected to group IDSTAR indeed require this access level to the involved data sets.

You can run the commands interactively with command **GO** or **RUN** and pressing Enter. You can also press **F3** and use action command R, for run, from the “Results” panel. You are automatically transferred to the CKRTSPRT work data set (see Figure 5-39). Scroll to the bottom to confirm that all command ran successful.

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000060 Col 001 080
 Command ===> _____________________________________________ Scroll ===> CSR_
connect  PAGENT  group(IDSTAR) auth(use)

============  8Nov24 11:32:53.22015 start record 20 ============
connect  SSHDAEM  group(IDSTAR) auth(use)

============  8Nov24 11:32:53.26403 start record 21 ============
permit  'CATALOG.**' GEN id(IDSTAR) access(UPDATE)

============  8Nov24 11:32:53.29307 start record 23 ============
permit  'CRMA.X.**' GEN id(IDSTAR) access(READ)

============  8Nov24 11:32:53.32110 start record 24 ============
permit  'USER.*.**' GEN id(IDSTAR) access(READ)

============  8Nov24 11:32:53.36686 start record 25 ============
setropts refresh generic(DATASET)
=========================================================================
=== All commands completed successfully                              ===
=========================================================================
****************************** Bottom of Data ******************************
```

Figure 5-39 Review ID(*) conversion successful command execution

In practice, you can monitor the use of ID(*) access to access data sets for a prolonged period and connect more users that use the permission to ID(*) to access the protected data sets. When after some period, no conversion commands are generated anymore, the next step is

to remove the permissions to ID(*) from the involved DATASET class profiles from the Offline RACF database.

Use option **RA.D**, for RACF Administration - Data set, to report DATASET profiles with ID(*) on the ACL. Select option “Access list” in the “Additional selection criteria” section (see Figure 5-40), and press Enter.

```
                zSecure Admin - RACF - Data set Selection
Command ===> ____________________________________________________   start panel

    Add new DATASET profile or segment

Show dataset profiles that fit all of the following criteria
Dataset profile . . ____________________________________________  1 1 EGN mask
Owned by  . . . . . ________     (group or userid, or filter)       2 Exact
High level qual . . ________     (qualifier or filter)              3 Match
Installation data . ________________________ (substring or *)     4 Any match

Additional selection criteria
_  Profile fields     /  Access list      _  Segment presence  _  Absence
   Audit settings
Output/run options
_  Show segments      _  All  _  Together _  Enable full ACL   _  Specify scope
_  Show differences
_  Print format          Customize title     Send as e-mail
      Background run     Full detail form    Sort differently     Narrow print
      Print ACL          Resolve to users    Incl operations      Print names
```

Figure 5-40 Select DATASET profiles based on Access list details

Specify `'*'` at option “Id on access list”. Notice that you must enclose the `*` with single quotation marks to prevent zSecure Admin interpreting the `*` as a wildcard character (see Figure 5-41).

```
                zSecure Admin - RACF - Data set Selection
Command ===> ____________________________________________________________________
All profiles
  Specify additional selection criteria:
  Find a combination of the following in the access list
  # permits . . . . . __   ______            (operator: < <= > >= = <> ^= )
  # conditional permits __   ______          (operator: < <= > >= = <> ^= )
_ Id on access list . . '*'_____             (*, group or userid, or filter)
  When resource . . . . _______________       (resource name or filter)
  Access level  . . . . __      1. None       When class  . .    1. PROGRAM
                                2. Execute                       2. CONSOLE
                                3. Read                          3. APPCPORT
                                4. Update                        4. TERMINAL
                                5. Control                       5. JESINPUT
                                6. Alter                         6. SERVAUTH
                                7. Ignore                        7. Present
                                                                 8. Ignore
  Access list filtering
  _  Only show matching ACL entries
```

Figure 5-41 Select DATASET profiles with permission to ID(*)

After pressing Enter, the following display is generated (see Figure 5-42).

```
                zSecure Admin DATASET Overview                  Line 1 of 14
Command ===> ____________________________________________________ Scroll===> CSR
All profiles with * on ACL                      8 Nov 2024 12:11
   Profile key                                  Type    UACC    Owner    S/F W
__ AA1.DDIRS%.AASRV1.*                          GENERIC READ    AA1      U U Y
__ AA1.DDIRTA.AASRV1.APPLID.ANYTRAN             GENERIC NONE    AA1       R _
__ AA1.DDIRTA.AASRV1.APPLID.TRANID              GENERIC NONE    AA1       R _
__ AA1.DDIRT*.AASRV1.APPLID.ANYTRAN             GENERIC NONE    AA1       R _
__ AA1.DDIRT*.AASRV1.APPLID.TRANID              GENERIC NONE    AA1       R _
__ AA1.SERVER.REXXAPI.AASRV1                    GENERIC NONE    AA1       R _
__ AA1.SEVER.IMPORT.**                          GENERIC NONE    AA1       R _
__ CATALOG.**                                   GENERIC READ    SYSAUTH   R _
__ CRMA.X.**                                    GENERIC NONE    CRMA     U U _
__ CRMBJU1.LOADLIB                              GENERIC READ    CRMBMB1   R _
__ CRMBRS3.*.**                                 GENERIC NONE    CRMBRS3   R _
__ SYSP.*.**                                    GENERIC NONE    SYSAUTH  U R _
__ SYS1.*.**                                    GENERIC NONE    SYSAUTH  R R _
__ USER.*.**                                    GENERIC READ    SYSAUTH  R R _
****************************** Bottom of Data ********************************
```

Figure 5-42 DATASET profiles with permission to ID(*) selected

All DATASET profiles with a permission to ID(*) are reported. The fact that you find more DATASET profiles than your access monitor report revealed earlier, means that the permissions to ID(*) were not used last year to access the protected data sets. You can decide based on that conclusion, that these DATASET profiles do not require the permission to ID(*) and clean them up right now. Your company might have a procedure that demands that you open a change management ticket and needs management approval before you can remove the permissions to ID(*).

Pressing **F11**, to scroll right, shows a column that is named ID(*) and shows the permitted access level to ID(*) in the ACL or CACL of the involved DATASET profile (see Figure 5-43 on page 110).

```
                zSecure Admin DATASET Overview                  Line 1 of 14
Command ===> ____________________________________________________ Scroll===> CSR
All profiles with * on ACL                      8 Nov 2024 12:11
   Profile key                                  E SgF ID(*)    Complex  Notify
__ AA1.DDIRS%.AASRV1.*                          _ ___ UPDATE   TVT6003  _______
__ AA1.DDIRTA.AASRV1.APPLID.ANYTRAN             _ ___ READ     TVT6003  _______
__ AA1.DDIRTA.AASRV1.APPLID.TRANID              _ ___ UPDATE   TVT6003  _______
__ AA1.DDIRT*.AASRV1.APPLID.ANYTRAN             _ ___ READ     TVT6003  _______
__ AA1.DDIRT*.AASRV1.APPLID.TRANID              _ ___ UPDATE   TVT6003  _______
__ AA1.SERVER.REXXAPI.AASRV1                    _ ___ NONE     TVT6003  _______
__ AA1.SEVER.IMPORT.**                          _ ___ READ     TVT6003  _______
__ CATALOG.**                                   _ ___ UPDATE   TVT6003  _______
__ CRMA.X.**                                    _ ___ READ     TVT6003  _______
__ CRMBJU1.LOADLIB                              _ ___ READ     TVT6003  _______
__ CRMBRS3.*.**                                 _ ___ UPDATE   TVT6003  _______
__ SYSP.*.**                                    _ ___ READ     TVT6003  _______
__ SYS1.*.**                                    _ ___ EXECUTE  TVT6003  _______
S_ USER.*.**                                    _ ___ READ     TVT6003  _______
****************************** Bottom of Data ********************************
```

Figure 5-43 DATASET profiles with access level of permission to ID(*) reported

You can use action command **S** to access the details for DATASET profile USER.*.**. That action shows the current settings of this profile. Specify action command **D**, for Delete, preceding the ACL entry for ID(*) (see Figure 5-44), and press Enter to delete the permission.

```
                zSecure Admin DATASET Overview                  Line 1 of 51
Command ===> ____________________________________________________ Scroll===> CSR_
All profiles with * on ACL                      8 Nov 2024 12:19


_ Identification                                                    TVT6003
  Profile name                  USER.*.**
  Type                          GENERIC
  Volume serial list
_ Effective first qualifier     USER                          DATASET OWNER?
_ Owner                         SYSAUTH                       AUTHORIZATION GRO
  Installation data             ____________________________________________

  User     Access  ACL id   When              RI Name                 DfltGrp
D - any -  READ    *
_ -group-  ALTER   SUPERUSR
_ -group-  ALTER   SUPPORT
_ -group-  ALTER   SYSPROG
_ -group-  ALTER   SYS1
_ AXRSTC   READ    AXRSTC                       AXR USER2               AXRGRP
_ AXRUSER  READ    AXRUSER                                              AXRGRP
_ CRMBEP1  UPDATE  CRMBEP1                      SYSPROG1 EP             CRMB
_ CRMBGUS  UPDATE  CRMBGUS                      SYSPROG GUS             CRMA
```

Figure 5-44 Request the deletion of the permission to ID(*)

Specify option **1** and press Enter to execute the RACF command (see Figure 5-45).

```
                        zSecure Admin - Confirm command
Command ===> ____________________________________________________________________




Confirm the following delete command
permit'USER.*.**'id(*)delete_____________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________



Command execution . 1  1. EXECUTE RACF command
                       2. EXECUTE CKGRACF command (allows use of Reason)
                       3. ASK administrator to execute CKGRACF command
                       4. REQUEST CKGRACF command for later execution
                       5. WITHDRAW CKGRACF command

Specify date for command to be executed
Start date . . . . . __________  (ddmmmyyyy, yyyy-mm-dd or TODAY)
Until/for  . . . . . __________  (ddmmmyyyy, yyyy-mm-dd or number of days)
Reason . . . . . . . _____________________________________________________
Press ENTER to continue or END to cancel the command
```

Figure 5-45 Run the deletion of the permission to ID(*)

Repeat the same steps to delete the ID(*) permissions from DATASET profiles CATALOG.** and CRMA.X.**.

Exit, your RACF-Offline session and start `ISPF` to verify the effect of this conversion against the offline RACF database.

You can now verify that your conversion commands had the desired effect. Simulating the historic access events against your offline RACF database for the DATASET class must show that access through ID(*) is not occurring anymore. Use option **AM.1** again to report the access by ID(*) to DATASET class profiles (see Figure 5-46 on page 112).

```
                        zSecure Admin – Access - Access
Command ===> ____________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . DATASET_  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
2  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  /  Show simulated fields   _  Timezone (U/L/H)
_  Print format              Customize title           Send as e-mail
      Background run         Full page form
```

Figure 5-46 Verify whether DATASET access through a permission to ID(*) still occurs

On the next panels, not displayed in this document, select only the successful access events, and select option “ID_STAR” in the “Authority used in current DB” section (see Figure 5-47), and press Enter to generate the report.

```
                        zSecure Admin - Access - Access      No records found
Command ===> ____________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . DATASET_  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
2  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  /  Show simulated fields   _  Timezone (U/L/H)
_  Print format              Customize title           Send as e-mail
      Background run         Full page form
```

Figure 5-47 Results of query whether permission to ID(*) to access DATASET resources

The “No records found” message in the top right corner of your display confirms that after your conversion, no data sets are accessed through a permission to ID(*). Your conversion of

ID(*) permissions to group IDSTAR is successful. You can now repeat the same steps to run the same ID(*) conversion commands against the active primary RACF database.

You can replay similar steps to also migrate the permissions to ID(*) in the other reported classes SDSF and SERVAUTH. However, you can also report all profiles in your RACF database that contain a permission to ID(*). The classes and profiles that are reported, but were not shown in your access overview report are all profiles where the permission to ID(*) was not used during the last year! These are good candidates for an even more tidy cleanup.

5.4.6 Converting generic OPERATIONS access to group-based access

This section explains the necessary steps for generating and executing automatic RACF commands for converting generic OPERATIONS access to group-based access

Naturally, the way to convert OPERATIONS access to group-based permission uses the same approach as UACC an ID(*) conversion. However, the zSecure Admin user interface does not support the automatic generation of OPERATIONS access to group-based access. But, with some manual tweaking of the CARLa code that zSecure Admin uses ‘under the covers’ that conversion can be performed semi-automatically.

You can use option **AM.1** to report the successful access that occurred through OPERATIONS to data sets (see Figure 5-48).

```
                        zSecure Admin - Access - Access
Command ===> ____________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . ________  (userid or EGN mask)
Complex . . . . . ________  (complex or EGN mask)
SAF resource class  . . DATASET_  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
1_ 1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  _  Show simulated fields   _  Timezone (U/L/H)
_  Print format             Customize title           Send as e-mail
      Background run         Full page form
```

Figure 5-48 Report historic access to resources summarized by userid

After you press Enter, the further selection panel appears. For this analysis report, you are only interested to report successful access events. Select option “Succes” in the “Result” section with action command **/** or **S** (see Figure 5-49 on page 114), and press Enter.

```
                  zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________
Access monitor records for Profiles like **, Classes like DATASET



Select access records(Y/N/blank)
_  Use of RACF commands                 _  Retrieval of access allowed
_  Use of global access checking table  _  Bypass JESSPOOL profiles
_  Use of discrete profiles             _  ID was undefined during event
_  System special authority used        _  User had special attribute
_  Operations authority used            _  User had operations attribute
_  Installation exit used               _  User had (ro)auditor attribute
_  User requesting access is owner

Resource action  Intended access    Result             Program status
_  Define        __  _  1. Read     /  Success         _  Defined program
_  Delete              2. Update   _  No profile      _  Controlled program
_  Addvol              3. Control  _  Not authorized  _  Specific program
_  Chgvol              4. Alter    _  Other           _  Controlled library
```

Figure 5-49 Report only successful access events

Next, the activated “Current RACF DB selection” panel appears. On this panel, you need to specify that you want your report to only show successful access through the system-OPERATIONS attribute. Select option “SYS_OPER” in the “Authority used in current DB” section with action command **/** or **S** (see Figure 5-50), and press Enter.

```
                  zSecure Admin - Access - Current Database
Command ===> ____________________________________________________________
Access monitor records for Profiles like **, Classes like DATASET, Successes
Specify simulated fields (Current DB effect) selection criteria:
Group(s) used for access  _____________________________  (group(s) or filter)
Essential group(s)  . . . _                              (Y/N/blank)
Not using group(s)  . . . _____________________________  (group(s) or filter)
Profile owned by  . . . . ________                        (id or filter)
RACF return code  . . . . __  ______  (operator: > >= < <= = <> ^=)

Authority used in current DB
_  APF          _  GRP_OPER    _  ID_USER     _  PRIVTRUS    /  SYS_OPER
_  CLAUTH       _  GRP_SPEC    _  INACTIVE    _  PROTFAIL    _  SYS_SPEC
_  CREATE       _  GRP_UACC    _  NO_CDT      _  QUALOWN     _  UACC
_  DFLTRC       _  ID_GROUP    _  NO_PROF     _  RESTRICTED  _  UNPROT
_  GLOBAL       _  ID_STAR     _  NOTHING     _  SPOOLRCVR   _  WARNING

User attributes in current DB (Y/N/blank)
_  ID present           _  ID Revoked
_  ID has special       _  ID has operations   _  ID has (ro)auditor
```

Figure 5-50 Report the successful access events that used system-OPERATIONS

In Figure 5-51, the resulting report shows that two users are assigned the OPERATIONS attribute used it to successfully access data sets.

```
                IBM Security zSecure ACCESS summary    1 s elapsed, 0.3 s CPU
Command ===> ____________________________________________________ Scroll===> CSR_
Access monitor records for Profiles like **, Cl 8 Nov 2024 17:16
   Occurrence Userid   Name                          First occurrenc Last occur
__       10693 IBMUSER  TO BE REVOKED-LATER           31Oct2023 23:00 31Oct2024
__         930 IWST     IWS TRACKER                    3Nov2023 22:46 31Oct2024
****************************** Bottom of Data ********************************
```

Figure 5-51 Successful DATASET access events that used system-OPERATIONS

Because option **AM.9** does not support an option to automatically convert OPERATIONS usage, you must use a different more manual approach. This method does require more advanced knowledge of, and experience with, the zSecure-specific CARLa programming language.

Press **F3** to return to the **AM.1** report. Next, specify command `RESULT` in the command and press Enter to access the zSecure Results panel that shows the zSecure work data sets (see Figure 5-52).

```
                            zSecure Admin - Results
Command ===> ____________________________________________________________________

The following selections are supported:
 B Browse file                S Default action (for each file)
 E Edit file                  R Run commands
 P Print file                 J Submit Job to execute commands
 V View file                  M E-mail report
 W Write file into seq. or partitioned data set

Enter a selection in front of a highlighted line below:
_  SYSPRINT  messages
_  REPORT    printable reports
_  CKRTSPRT  output from the last TSO command(s)
_  CKRCMD    queued TSO commands
_  CKR2PASS  queued commands for zSecure Admin for RACF
S  COMMANDS  zSecure Admin for RACF input commands from last query
_  SPFLIST   printable output from PRT primary command
_  OPTIONS   set print options
```

Figure 5-52 Results panel of successful DATASET access events that used system-OPERATIONS

The CARLa script that was composed by the selection filters that you specified is stored in the COMMANDS work data set. Use action command **S** to access the CARLa code. Press Enter to access COMMANDS work data set containing the CARLa code that produced your report (see Figure 5-53 on page 116).

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR_
****** **************************** Top of Data ******************************
000001 /* generated by CKRP3AMZ */
000002 newlist type=access nodetailinherit required,
000003  st=",
000004 Access monitor records for Classes like DATASET, Resources like **,,
000005  Successes, Current RACF DB=, SYS_OPER",
000006  sumhelppanel=ckrt3ami,
000007  helppanel=ckrt3ami,
000008  detailhelppanel=ckrt3ami
000009  define tot_count("Occurrence",10,udec$abbr) sum(access_count_big)
000010  define avg_reclen avg(record_length)
000011  define first_tod_sum(,"First occurrence") min(last_tod)
000012  define last_tod_sum(,"Last occurrence") max(last_tod)
000013 select ,class=DATASET,access_profile=**, sim_via=(,,
000014  SYS_OPER),,access_result=("00"x) rectype=(auth,fast,def)
000015  display last_tod(nd) access_count("Occurrence") last_tod,
000016   / "Access summary"(d,ch),
000017   / complex(d,p),
000018   / userid(d,p) userid:name(d),
```

Figure 5-53 COMMANDS work data set shows CARLa used to produce OPERATIONS usage report

The resulting SELECT statement in **bold** shows the result of the panel selections that you entered to produce the system-OPERATIONS access report. You can copy this select statement and use it to build your own CARLa program to generate system-OPERATIONS conversion commands instead of a report.

After copying the select statement press **F3** and next issue command `CARLA` on the command line and press Enter. That command takes you to the CARLa editor. If you have used the CARLa editor before, your previous CARLa code is saved there (see Figure 5-54). When you use this editor for the first time, the editor will be empty.

```
EDIT       CRMBTZ1.CARLA.SAMPLES(#CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR_
****** **************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= END or SAVE to save in ISPF profile
000001
''''''
''''''
''''''
```

Figure 5-54 CARLa editor at first usage

Next, copy the following CARLa code to your CARLa editor (see Figure 5-55 on page 117).

```
EDIT       CRMBTZ1.CARLA.SAMPLES(#CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 newlist type=racf dd=ckrcmd nopage outlim=1
000002  sortlist "addgroup <group name> sup(<superior group>) owner(<owner>)",
000003           "data('<Desrciption of system-OPERATIONS access group>')"
000004
000005 newlist type=access dd=ckrcmd nopage nodup
000006  select class=DATASET sim_via=(SYS_OPER) access_result=("00"x),
000007         rectype=(auth,fast)
000008  sortlist "pe '" | access_profile(0) | "' generic id(<group name>)",
000009           "access(" | intent(0) | ")"
000010  sortlist "co" userid(0) "group(<group name>) authority(use) uacc(none)"
****** **************************** Bottom of Data ****************************
```

Figure 5-55 CARLa code to generate automatic system-OPERATIONS conversion commands

Additional notes and explanations about the copied CARLa code:

- When you use an existing group to convert system-OPERATIONS access to, then you can remove the first 4 lines that define a new group profile as this is not necessary.
- All `NEWLIST` statements in this CARLa program redirect their output to the CKRCMD work data set using the dd=ckrcmd specification.
- The `NOPAGE` keyword suppresses all page layout characteristics, used because this CARLa generates RACF commands instead of a report.
- The `OUTLIM=1` specification in line 1 causes the newlist statement to only produce 1 output record.
- The `SORTLIST` statement in lines 2 & 3 produces the appropriate ADDGROUP command to define a new RACF group.
- The values shown with “<....>” serve as variables for you to replace with the values that you want to define for this new RACF group.
- The `NODUP`, for no duplicates, specification in the newlist statement in line 5, suppresses generation of duplicate permit commands in case the user accessed the same data set many times.
- The `SELECT` statement in lines 6 & 7 is copied from your earlier report. It is sanitized by removing superfluous commas and parentheses to improve the readability of the CARLa code. Also, the rectype=def filter is removed to prevent that the CARLa script produces conversion commands for DEFINE requests that report pseudo access levels DEFCREATE, DEFDELETE, DEFCHGVOL and DEFADDVOL.
- The `SORTLIST` statement in lines 8 & 9 generates the appropriate permit commands for the system-OPERATIONS users to the target group with the access level that was requested according to the access monitor record. The GENERIC keyword is generated to cater for occurrences where the system-OPERATIONS user accessed a fully qualified generic (FQG) dataset profile. The vertical bars (|) are concatenation characters used to suppress otherwise automatically inserted blanks between columns.
- The `SORTLIST` statement in line 10 generates the connect commands to the target group that you use as target for this conversion.

Running the CARLa with command **`GO`** or **`RUN`** might produce something similar to Figure 5-56 on page 118.

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                    Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR_
****** **************************** Top of Data ******************************
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated  8 Nov 2024 18:18 */
000002 addgroup grpoper sup(crmb) owner(crmb) data('Conversion of OPERATIONS -
000003 dataset access group')
000004 pe 'CRMBEP1.*.**' generic id(grpoper) access(READ)
000005 pe 'CRMBEP1.*.**' generic id(grpoper) access(ALTER)
000006 pe 'CRMBMK1.*.**' generic id(grpoper) access(READ)
000007 pe 'CRMBPH1.*.**' generic id(grpoper) access(READ)
000008 pe 'CRMBPH1.*.**' generic id(grpoper) access(ALTER)
000009 pe 'CRMBRL1.*.**' generic id(grpoper) access(ALTER)
000010 pe 'CRMBRS1.*.**' generic id(grpoper) access(READ)
000011 pe 'CRMBRS1.*.**' generic id(grpoper) access(ALTER)
000012 pe 'CRMBTZ1.*.**' generic id(grpoper) access(ALTER)
000013 pe 'CRMBVK1.*.**' generic id(grpoper) access(ALTER)
000014 pe 'CRMBVK2.*.**' generic id(grpoper) access(ALTER)
000015 pe 'C2PACMON.*.**' generic id(grpoper) access(READ)
000016 pe 'SYSAPPL.**' generic id(grpoper) access(READ)
000017 co IBMUSER group(grpoper) authority(use) uacc(none)
000018 co IWST group(grpoper) authority(use) uacc(none)
**************************** Bottom of Data ********************************
```

Figure 5-56 Automatically generated system-OPERATIONS conversion commands

Before running the commands, you must review the generated PERMIT commands to remove duplicate commands to the same DATASET profile with different access levels. You must keep the PERMIT command with the highest access level. In this example, you must delete the commands in lines 4, 7, and 10 before running the commands.

Running the commands transfers you to the CKRTSPRT work data set as usual (see Figure 5-57). Scroll to the bottom to confirm that all commands ran successful.

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000032 Col 001 080
 Command ===> ____________________________________________________ Scroll ===> CSR_
pe 'CRMBVK1.*.**' generic id(grpoper) access(ALTER)

============  8Nov24 18:20:28.23371 start record 11 ============
pe 'CRMBVK2.*.**' generic id(grpoper) access(ALTER)

============  8Nov24 18:20:28.23724 start record 12 ============
pe 'C2PACMON.*.**' generic id(grpoper) access(READ)

============  8Nov24 18:20:28.24029 start record 13 ============
pe 'SYSAPPL.**' generic id(grpoper) access(READ)

============  8Nov24 18:20:28.24356 start record 14 ============
co IBMUSER group(grpoper) authority(use) uacc(none)

============  8Nov24 18:20:28.25080 start record 15 ============
co IWST group(grpoper) authority(use) uacc(none)
=============================================================================
=== All commands completed successfully                                  ===
=============================================================================
******************************** Bottom of Data ********************************
```

Figure 5-57 Confirm successful execution of system-OPERATIONS conversion commands

When the involved system-OPERATIONS users did not use OPERATIONS access to other resources in classes that honor the OPERATIONS attribute (which you have not verified), you can now safely remove the OPERATIONS attribute from the users IBMUSER and IWST.

Use the same approach as outlined in Figure 5-25 on page 100 through Figure 5-27 on page 101 to exit your RACF-Offline session and then verify that now no system-OPERATIONS access for DATASET class is reported (see Figure 5-58 on page 120).

```
                          zSecure Admin - Access - Access      No records found
Command ===> ____________________________________________________________________

Show records that fit all of the following criteria:
Userid  . . . . . . . . ________  (userid or EGN mask)
Complex . . . . . . . . ________  (complex or EGN mask)
SAF resource class  . . DATASET_  (class or EGN mask)
SAF resource name . . . ____________________________________________
RACF match on . . . . . ____________________________________________

Advanced selection criteria
/  Further selection  _  Date selection   /  Current RACF DB selection

Output/run options
1  1. Summary by userid
   2. Summary by member class and profile
   3. Summary by simulated authorization used
   4. Summary by simulated groups used for access
_  Show configured fields  /  Show simulated fields   _  Timezone (U/L/H)
_  Print format               Customize title           Send as e-mail
      Background run          Full page form
```

Figure 5-58 Confirm that no system-OPERATIONS data set access occurs

When everything worked according to plan, no system-OPERATIONS access is reported for the DATASET class. You can use the same approach and steps to convert system-OPERATIONS use in other resource classes.

You can now repeat the same steps to run the same OPERATIONS attribute conversion commands against the active primary RACF database.

This concludes the walk through of use case: “Converting to group-based access”.

6

Minimize access control privileges

Minimizing access control privilege, also known as the “principle of least privilege,” is crucial to reducing the potential of security breaches. Ensuring users only have the minimum level of access needed to perform their job limits damage that can be done if their credentials are compromised or misused, whether from malicious intent or negligence.

In this chapter, we discuss the following topics:

- 6.1, “Challenge with access control privileges” on page 122
- 6.2, “Avoiding the need to trust privileged users” on page 122
- 6.3, “Revealing permitted access levels that exceed the need” on page 124
- 6.4, “Minimizing access control privileges” on page 124

6.1 Challenge with access control privileges

Another challenge that RACF administrators face is minimizing the access control privileges in their environment to protect sensitive data and systems from unauthorized access and manipulation. In many RACF environments, permissions exist that are no longer needed, or have more access than is required.

Some reasons for RACF databases to end up in a state were allowed access exceeds required access are as follows.

- RACF security administrators are often requested by business process owners to define RACF permissions when required. However, when accesses are no longer required, they are not informed to remove or redefine these privileges in RACF.
- A requestor of a security definition change may not fully understand the needs of the application or use-case of the resource that they request access for. They might request the RACF administrator to permit them a higher access level or privilege than they require for their job role.
- Over time, the security requirements of certain job roles or uses of an application can change leaving a RACF database with permissions that are out of sync with the current organizational needs.
- Overly permissive resource access definitions can present problems for an organization or information system owner. Permissions that are unnecessary or overly permissive can cause a company to fail internal or external audits. Additionally, these badly defined permissions can leave security exposures in place as features of applications and resources are accessible to users who do not require access to them.

6.2 Avoiding the need to trust privileged users

The concept of “least privilege” is very prominent in modern and healthy security environments. Minimizing the privileges in RACF environments to embody this principle is vital to the maintenance of your RACF environment. The following practices are important aspects of this concept:

- Permitting higher access levels is bad practice
- Minimize use of overly permissive RACF attributes
- Minimize the use of the WARNING attribute
- Minimize the use of global access checking tables

6.2.1 Permitting higher access levels is bad practice

Permitting higher access levels than necessary to a resource is bad practice. Permissions in RACF are discussed as a series of “access levels” that offer the privileges of preceding levels. The general idea of this practice is that if a user only requires a specific permission the use of an application, dataset, or feature in their job role. The permitted access level that they are permitted to the profile that governs this access must not exceed their required access level. As an example, a user who requires READ access to a specific resource that is protected by a profile, must not be permitted UPDATE, CONTROL, or ALTER access level instead. While this permission does not cause RACF authorization errors, it does leave environments open to potential security exposures.

This is an illustration of the idea of “least privilege” that is discussed elsewhere. If a user only requires READ access to a resource and they are permitted ALTER access instead, this

access does not represent the least privilege access that they require to successfully perform their job role. Overly permissive access definitions can lead to security exposures that potentially can be leveraged by compromised or malicious users to perform actions beyond the required access for a given job role.

6.2.2 Minimize use of overly permissive RACF attributes

In RACF, there are other attributes and methods for providing access that must be carefully monitored to ensure that the least privilege principle is adhered to. Some of these are attributes that users can be assigned like AUDITOR, OPERATIONS, SPECIAL, TRUSTED, or PRIVILEGED. The assignment of these privileges must be carefully monitored and documented to ensure that the authorities associated with them are not assigned to users who do not require them.

User IDs that are assigned the OPERATIONS attribute are allowed ALTER access to resources in the dataset and the general resource classes that are configured to honor the OPERATIONS attribute. When an OPERATIONS user accesses such a resource in a class, ALTER access is allowed by default. If a user ID or one of their connect groups is permitted another access level, the access level permitted on the ACL or CACL is used instead. Auditors often report the use of the OPERATIONS attribute as non-controlled access. Use of OPERATIONS violates the idea of “least privilege” by assigning broad and permissive access. This also violates the idea of a “zero trust” environment by effectively assigning new default permissions for affected users. Replacing the assignment of the OPERATIONS attribute with appropriate permissions to the resources that an OPERATIONS user accesses in their job role improves the security posture of your system.

6.2.3 Minimize the use of the WARNING attribute

The activation of the WARNING attribute on a resource profile causes RACF to bypass access verification checks for this resource. The purpose of the WARNING attribute is to be used in the implementation phase when it is unclear which users require what access level to a resource. When a non-authorized user requests access to a resource that is protected by a profile in WARNING mode, ALTER access is allowed despite the fact that neither the user nor their connect groups are permitted access to the resource. The access is then logged to SMF allowing security administrators to populate the ACL and CACL with the appropriate permissions. Then, after a period of monitoring SMF records and adding permissions, security administrators can remove the WARNING attribute without causing a security impact. Obviously, the use of WARNING also violates the “least privilege” and “zero trust” environment by establishing broad and permissive access for all users.

6.2.4 Minimize the use of global access checking tables

Global access checking (GAC) tables can be implemented to boost the performance of access verification checks by quickly allowing a particular access level to certain resources in RACF resource classes without checking RACF profiles and without logging to SMF. Most companies probably have public resources that they publicly allow READ access to. For example, banks allow all users to READ their interest rates to attract more customers. However, the GAC tables must not allow access to resources when access to the resource would not be allowed for certain users, or when the company needs to be able to report who accessed that resource. Therefore, the use of GAC tables and the list globally accessible resources needs the be carefully monitored. Improper use of these tables violates the principle of “zero trust” as it allows all users to access the specified resource.

6.3 Revealing permitted access levels that exceed the need

Reporting overly permissive access is a particularly difficult challenge to tackle in a native RACF environment. There are SMF records that can indicate when profile permissions are used or permitted as well as the level of privilege the user has to the resource. But running bulk analysis of SMF records would be a large task and requires auditing successful access to all profiles within the scope of this effort. Using zSecure Access Monitor, you can generate reports about ACL entries that reveal that the permitted access level exceeds the access level that the user requested. Additionally, you can code CARLa scripts that support the automatic generation of RACF commands to transition from this higher authority to a “least privilege” environment. In addition, with zSecure, you can first test the effect of running these security changes against an offline RACF database with RACF-Offline prior to running them in your active RACF database.

While several practices related to managing permissive options for users or profiles like RACF OPERATIONS, WARNING, or GAC are discussed, these practices also relate to privilege minimization. The steps for identifying and tuning such attributes are covered in detail in Chapter 5, “Convert generic and specific access to group-based access” on page 79.

6.4 Minimizing access control privileges

The details for the general use case preparation steps using automated (semi-) conversion and cleanup features in zSecure Admin are documented in 2.4, “General preparation steps for the use cases” on page 16, including these steps:

- Creating a RACF-Offline database
- Starting a RACF-Offline session
- Logging on to RACF-Offline session
- Resetting the RACF-Offline log data set
- Starting and preparing your zSecure Admin session

After completion of these steps, you can continue with the subsequent steps to minimize access control privileges.

- Reporting permitted DATASET access levels exceeding the actual used access levels
- Implementing least privilege for the DATASET class
- Verifying the permit commands for the DATASET class
- Executing permit commands against the active primary RACF database
- Reporting permitted general resource access levels exceeding the used access levels

6.4.1 Reporting permitted DATASET access levels exceeding the actual used access levels

In this section, you will learn how to report historic DATAESET access where the allowed access level exceeds the used access level.

The allocated access monitor records can be used to analyze and report which users successfully accessed resources where the allowed access level exceeds their used access level. You can use option **AM.3**, for Permit usage, to report which users accessed resources with a requested access level that is less than their permitted access level. By default, you can generate permit usage overview reports for all users and resource classes in a single report.

In Figure 6-1 the scope of the permit usage report is restricted to the DATASET class by specifying `dataset` at option “Class”. Because for this query, you are only interested in permissions that are used, you must also activate the options “Non-zero counts” and “Further selection” in the “Advanced selection criteria” section, and option 2, for “Simulate access in database to find current profile”.

```
                    zSecure Admin - Access - Permit usage
Command ===> ____________________________________________________________________

Show permits that fit all of the following criteria:
Permit id . . . . . ________  (permit id or EGN mask on access list)
Class . . . . . dataset_  (class or EGN mask)
RACF profile name . . . _____________________________________________
Complex . . . . . ________  (complex or EGN mask)
Show accesses . . . . . /  Non-zero counts     _  Zero counts

Profile to use  2  1. Use historic profile name in access summary if present
                   2. Simulate access in database to find current profile

Advanced selection criteria
/  Further selection                    _  Date selection

Output/run options
_  Print format          Customize title     Send as e-mail
      Background run     Full page form
```

Figure 6-1 Report historic access to resources in the DATASET class

After pressing Enter, the further selection panel appears. On this panel, you must select option “Highest access used less than access allowed” to find occurrences where the “least privilege” principle appears not to be correctly implemented (see Figure 6-2). Press Enter.

```
                  zSecure Admin - Access - Further selection
Command ===> ____________________________________________________________________
Access monitor records for Classes like DATASET, non-zero counts
Access selection
#Accesses allowed . . . __  __________
#Accesses prevented . . __  _____
#Accesses unexplained    __  _____

Permit selection
Creation date from  . . __________  Until   __________

/  Highest access used less than access allowed

Access allowed        Highest access used   Lowest violation
>=  _  1. Execute         <=  _  1. Read        >=  _  1. Read
       2. Read                  2. Update            2. Update
       3. Update                3. Control           3. Control
       4. Control               4. Alter             4. Alter
       5. Alter
```

Figure 6-2 Report historic DATASET access where highest access used is less than access allowed

That selection generates a permit usage overview report of DATASET profiles that users used to access protected data sets, where the permitted access level exceeds the highest access that permitted users used. You can use action command **S** (see Figure 6-3) and press Enter to access the DATASET profile details to verify which ACL permissions are potential candidates to be tightened.

```
                                                                  Line 1 of 32
Unconditional permits and UACC, by class complex/profile
Command ===> _____________________________________________________ Scroll===> CSR
Access monitor records for Classes like DATASET 12 Nov 2024 14:30
   Allowed Deny  Unexp LastUse Class    Complex
   4108400   264     0 31Oct24 DATASET  TVT6003
   Allowed Deny  Unexp LastUse Type     Profile
__   13011     0     0 31Oct24 GENERIC  CATALOG.**
S_   20442     0     0 31Oct24 GENERIC  CKNSERVE.**
__       7     0     0 23Oct24 GENERIC  CRMA.T.**
__   20710     0     0 31Oct24 GENERIC  CRMA.X.**
__    6378     0     0 31Oct24 GENERIC  CRMAUTO.**
__       1     0     0 12Sep24 GENERIC  CRMBAH1.TEST12.**
__     187     0     0  5Dec23 GENERIC  CRMBEP1.*.**
__      96     0     0 31Oct24 GENERIC  CRMBJK1.*.**
__      65     0     0 17Sep24 GENERIC  CRMBJU1.**
__       2     0     0 17Sep24 GENERIC  CRMBJU1.LOADLIB
__    4888     0     0 30Oct24 GENERIC  CRMBMK1.*.**
__       2     0     0 13Sep24 GENERIC  CRMBMK1.TE%T
__      30     0     0 22Mar24 GENERIC  CRMBPH1.*.**
```

Figure 6-3 Report DATASET profiles where allowed access level exceeds used access level

In the top right, message “Line 1 of 32”, indicating that 32 DATASET profiles exist where permitted access exceeds used access. You can conclude that 32 DATASET profiles need to be tightened to implement the least privilege principle.

The next panel reports the permissions for which the allowed access level exceeds the used access level (see Figure 6-4).

```
Unconditional permits and UACC, by class complex/profile
Command ===> _____________________________________________________ Scroll===> CSR
Access monitor records for Classes like DATASET 12 Nov 2024 14:30
   Allowed Deny  Unexp LastUse Class    Complex
   4108400   264     0 31Oct24 DATASET  TVT6003
   Allowed Deny  Unexp LastUse Type     Profile
     20442     0     0 31Oct24 GENERIC  CKNSERVE.**
   Allowed Deny  Unexp LastUse Id       Access    Used    Failed  Red RdM Name
__       1     0     0 25Oct24 CKNSERVE QUALOWN   READ            No      MULTI
S_   20441     0     0 31Oct24 C2RSERVG UPDATE    READ            No
******************************* Bottom of Data ********************************
```

*Figure 6-4 Profile CKNSERVE.** permissions where allowed access exceeds used access*

Note: By default, Access Monitor includes administrative authority as indicated with access level “QUALOWN”. That QUALOWN access level is not the result of a permission, it indicates that the user ID that matches the high-level-qualifier (HLQ) of the data sets is the implicit owner of that data set and therefore is always allowed ALTER access to the, in this example, CKNSERVE data sets.

The report reveals that user C2RSERVG successfully used the UPDATE permission 20441 times to access data sets with HLQ CKNSERVE. However, during all these historic access events by user C2RSERVG, the used access level was always READ whereas the permitted access level for this user is UPDATE.

When you specify action command **S** and press Enter, you can review the access record details. In Figure 6-5, optionally, you can issue action command **P**, for Show profile, at option “RACF class and profile”. When you press Enter, a recursive call is performed from your report to list the profile definitions of, in this example, DATASET profile CKNSERVE.**.

```
                                                                Line 1 of 28
Unconditional permits and UACC, by class complex/profile
Command ===> _____________________________________________ Scroll===> CSR
Access monitor records for Classes like DATASET 12 Nov 2024 16:03


  Profile in current database
_ Security complex name         TVT6003
P RACF class and profile        DATASET  CKNSERVE.**
  Profile type                  GENERIC
  Volume serial
  Resource location
  Installation data

  Unconditional permit
_ Permit or connect id          C2RSERVG
  Access or authority           UPDATE
  Highest access used           READ
  Lowest access prevented
  Permit reduces access         No
  Merged permit reduces access
  User name
  Installation data
```

*Figure 6-5 Profile CKNSERVE.** permission to C2RSERVG*

After pressing Enter, the security definitions of DATASET profile named CKNSERVE.** are displayed (see Figure 6-6 on page 128). As expected, the ACL contains a permission to group ID C2RSERVG with UPDATE access level. You can tighten security manually by overtyping access level UPDATE with **READ**, pressing Enter, and executing the generated command. However, there were more permissions to other DATASET profiles where according to access monitor records the allowed access level exceeds the used access level.

```
                                                          0 s elapsed, 0.0 s CPU
                        zSecure Admin DATASET Overview
Command ===> _____________________________________________________ Scroll===> CSR
key CKNSERVE.**                                  12 Nov 2024 16:07


_ Identification                                                        TVT6003
  Profile name                 CKNSERVE.**
  Type                         GENERIC
  Volume serial list
_ Effective first qualifier    CKNSERVE MULTI-SYS SERVER     ZSECURE 1.12+ MUL
_ Owner                        SYSPROG                       SYSTEM PROGRAMMIN
  Installation data            ____________________________________________

  User     Access  ACL id   When                RI Name                 DfltGrp
_ -group-  READ___ CRMBZDEV
_ -group-  READ___ CRMCXGRP
_ -group-  READ___ CRMQA___
_ -group-  UPDATE_ C2RSERVG
_ -group-  ALTER__ SYSPROG_
_ CKNSERVE ALTER__ CKNSERVE                    MULTI-SYS SERVER     C2RSERV
_ CRMBGUS  UPDATE_ CRMBGUS_                    SYSPROG GUS          CRMA
_ CRMBPH1  UPDATE_ CRMBPH1_                    SYSPROG1 PH          CRMA
```

*Figure 6-6 DATASET profile CKNSERVE.** security settings reported*

Instead of manually changing the permitted access level on this and the other 32 involved DATASET profiles, you can also use your CARLa programming skills. Using CARLa, you can automatically generate the appropriate permit commands to reduce the permitted access level, to the highest used access level for permitted IDs. Press **F3** a couple of times until you are back on the **AM.3** panel.

6.4.2 Implementing least privilege for the DATASET class

In this section, you will learn how to generate and execute automatic permit commands to implement the least privilege principle for the DATASET class.

Your recursive call to list the security definitions of DATASET profile CKNSERVE.** overwrote the CARLa code that was composed to generate your report of DATASET profiles where the used access level is less than the permitted access level. To rebuild the CARLa code that produced your initial report, run the permit usage overview report again. This time do not display the settings of DATASET profile CKNSERVE.** with the **P** action command. Instead, press **F3** once to return to the **AM.3** panel.

Next, specify command **`result`** on the command line and press Enter to access the zSecure work data sets on the result panel (see Figure 6-7 on page 129).

```
                          zSecure Admin – Access - Per          0.2 s CPU, RC=4
Command ===> result_____________________________________________________________

Show permits that fit all of the following criteria:
Permit id . . . . . _______  (permit id or EGN mask on access list)
Class . . . . . dataset  (class or EGN mask)
RACF profile name . . . _____________________________________
Complex . . . . . _______  (complex or EGN mask)
Show accesses . . . . . /  Non-zero counts     _  Zero counts

 Profile to use  2  1. Use historic profile name in access summary if present
                    2. Simulate access in database to find current profile

Advanced selection criteria
/  Further selection                    _  Date selection

Output/run options
_  Print format          Customize title     Send as e-mail
      Background run     Full page form
```

Figure 6-7 Use result command to access zSecure work data sets

After pressing Enter, the zSecure Results panel appears. The CARLa code that is used to produce your report showing which DATASET profiles store permissions that allow a higher access level than the permitted IDs use, is stored in the COMMANDS work data set. Use action command **S** or **E** to access the COMMANDS work data set (see Figure 6-8).

```
                             zSecure Admin - Results
Command ===> ____________________________________________________________________

The following selections are supported:
 B Browse file                S Default action (for each file)
 E Edit file                  R Run commands
 P Print file                 J Submit Job to execute commands
 V View file                  M E-mail report
 W Write file into seq. or partitioned data set

Enter a selection in front of a highlighted line below:
_  SYSPRINT  messages
_  REPORT    printable reports
_  CKRTSPRT  output from the last TSO command(s)
_  CKRCMD    queued TSO commands
_  CKR2PASS  queued commands for zSecure Admin
S  COMMANDS  zSecure Admin input commands from last query
_  SPFLIST   printable output from PRT primary command
_  OPTIONS   set print options
```

Figure 6-8 Use action command S to access COMMANDS work data set from Results panel

Pressing Enter reveals the content of the COMMANDS work data set. The top part of the CARLa script contains CARLa statements that build the report layout as is shown once the report is generated (see Figure 6-3 on page 126). The define statements are used to define various statistical fields and counters that are part of the generated standard permit usage report. That part of this CARLa program is not the focus of your interest. You want to discover

the composed select statement that filters the access monitor records to only show the access events where the permitted access level exceeds the used access level. Press **F8** and press Enter to scroll down to find the select statement that is composed using the UI-based selection filters that you entered to produce the permit usage report (see Figure 6-9).

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= CREATE or REPLACE to save these commands in your own dataset
000001 simulate racf_access
000002 n type=RACF_ACCESS name=CTBYPRFD nodetailinherit required,
000003 st=",
000004 Access monitor records for Classes like DATASET, non-zero counts,,
000005  highest access lower than allowed" I=CTBYPROF,
000006 t="Unconditional permits and UACC, by class complex/profile"
000007  /* generated by CKRP3AMX */
000008  define count_suc(7,"Allowed",udec$abbr,bw,noprop)
000009                   sum(access_count_suc)
000010  define count_vio(5,"Deny",udec$abbr,bw,noprop)
000011                   sum(access_count_vio)
000012  define count_unk(5,"Unexp",udec$abbr,bw,noprop)
000013                   sum(access_count_unk)
000014  define lastuse(7,"LastUse",noprop) max(access_lastuse)
000015  define firstuse(7,"Firstuse",noprop) min(access_lastuse)
000016  define helppanel=C2R3Z241,
```

Figure 6-9 Top part of the COMMANDS work data set

On the next page of the CARLa program (see Figure 6-10 on page 131), you encounter the select statement that produced the overview of the 32 DATASET class profiles where at least one of their permissions allows a higher access level then the permitted IDs requires.

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
000019  define helppanel=C2R3Z243,
000020         Miss#(7,"Missing",dec,noprop)  count where
000021                    (proftype="missing"C)
000022  define Pres#(8,"Permits",dec,noprop) count where
000023               (proftype<>"missing"C)
000024  define helppanel=C2R3Z242,
000025         Perm#(8,"Permits and UACC",dec,noprop) count
000026 select raclist_merge=no class<>group,
000027  proftype<>GLOBAL,class=DATASET,(access_count_suc>0 or,
000028  access_count_vio>0 or access_count_unk>0),(,,
000029  access_intent_max_suc<<access)
000030 display id(nd),
000031    access_count_suc(7,key),
000032    access_count_vio(5,"Deny",key),
000033    access_count_unk(5,key),
000034    access_lastuse(7,key,"LastUse"),
000035    id(pas,key) access access_intent_max_suc("Used"),
000036    access_intent_min_vio access_reduced merged_access_reduced,
000037    id:name id:revoke(1) | id:revoke_inactive(1) " ",
```

Figure 6-10 Composed select statement generated in the COMMANDS work data

Your next challenge is to adjust this CARLa script to automatically produce RACF permit commands instead of a permit usage report. As a starting point, you can use this same select statement for your customized CARLa script to automatically generate permit commands that tighten the permissions for the applicable DATASET profiles.

The customized CARLa script might look like this example in Figure 6-11.

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 simulate racf_access
000002 n type=RACF_ACCESS nopage dd=ckrcmd
000003 select raclist_merge=no class=dataset access<>(QUALOWN,ALTER-O),
000004  access_intent_max_suc<>(DEFCREATE,DEFDELETE,DEFCHGVOL,DEFADDVOL),
000005  proftype<>GLOBAL (access_count_suc>0 or access_count_vio>0 or,
000006  access_count_unk>0) access_intent_max_suc<<access
000007  sortlist "pe '" | profile(0) | "' class(dataset) generic id(" |,
000008           id(0) | ") ACCESS(" | access_intent_max_suc(0) | ")" / ,
000009           "     /* Access level was" access(0) "*/"
****** **************************** Bottom of Data ****************************
```

Figure 6-11 Customized CARLa script to produce RACF permit commands instead of the report

Notes and explanations regarding the adjustments performed to the original CARLa program in the COMMANDS work data set:

- The simulate statement in line 1 was not changed.
- The newlist statement in line 2, that is abbreviated to n, that was defined in the lines 2 to 6 is replaced with a single line now.

  - The `NOPAGE` keyword suppresses generation of page layout characteristics (titles, column headers, page numbers, etc) as your goal is to generate RACF permit commands instead of a permit usage report.
  - The `DD=CKRCMD` specification redirects the output of your CARLa script to the CKRCMD work data set which is the designated zSecure work data set to run commands from.
- All original DEFINE statements in lines 8 to 25 that produced counters and statistics used in the report are not required for the RACF permit command generation. Therefore, all DEFINE statements are removed from the customized CARLa script.
- The original select statement is slightly sanitized and improved:
  - The superfluous commas and parentheses that support all supported filters in the UI but are not required in this CARLa script are removed.
  - Extra filter `ACCESS<>(QUALOWN,ALTER-O)` is added to the select statement to prevent generating permit commands for these pseudo access levels QUALOWN and ALTER-O. These pseudo access levels indicate access by qualifier owner or access through the OPERATIONS attribute.
  - Extra filter `access_intent_max_suc<>(DEFCREATE,DEFDELETE,DEFCHGVOL,DEFADDVOL)` is added to prevent that permit commands for pseudo access levels reported for define-type access events are generated.
  - The other filters are identical to those that were used to produce the original permit usage report.
- The DISPLAY statement is replaced with a `SORTLIST` statement that produces the automatically generated appropriate permit commands:
  - The specification `"pe '"` generates the string `pe '` in the to be produced RACF command in the CKRCMD work data set.
  - `profile(0)` generates the name of the concerning DATASET profile following the pe ' string. The length 0 causes the trailing blanks of a DATASET profile name to be truncated.
  - A vertical bar (`|`) suppresses a blank that is otherwise inserted by default between columns that are specified on the SORTLIST statement.
  - The `"' class(dataset) generic id("` specification generates the literal string `' class(dataset) generic id(` in the permit command following the DATASET profile name.
  - The specification `ID(0)` generates the name of the permitted ID in the permit command.
  - The `) "ACCESS("` specification generates the literal string `) ACCESS(` in the permit command.
  - The variable `access_intent_max_suc(0)` generates the maximum access level that the involved permitted ID used according to Access Monitor in the permit command.
  - The `)` specification generates the closing `)` and the slash `(/)` instructs CARLa to continue the output on the next line of the CKRCMD work data set.
  - Finally, the string `"/* Access level was" access(0) "*/"` generates a CARLa comment statement `/* Access level was" access(0) */` that reports the original permitted access level that the ID was permitted with the purpose to serve as an eye catcher for the reviewer of the generated commands.

Tip: When you intend to use this CARLa script more often in the future, you might want to save the CARLa script in your CARLa samples library.

Running the CARLa script with the command **GO** or **RUN** transfers you automatically to the CKRCMD work data set that stores the generated permit commands to implement the least privilege principle for DATASET profiles on your system (see Figure 6-12).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                      Columns 00001 00072
Command ===>  _____________________________________________ Scroll ===> CSR
****** **************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated 20 Nov 2024 12:29 */
000002 pe 'CATALOG.**' class(dataset) generic id(IDSTAR) ACCESS(READ)
000003      /* Access level was UPDATE */
000004 pe 'CKNSERVE.**' class(dataset) generic id(C2RSERVG) ACCESS(READ)
000005      /* Access level was UPDATE */
000006 pe 'CRMA.X.**' class(dataset) generic id(SYSPROG) ACCESS(READ)
000007      /* Access level was ALTER */
000008 pe 'CRMAUTO.**' class(dataset) generic id(SYSPROG) ACCESS(UPDATE)
000009      /* Access level was ALTER */
000010 pe 'CRMBJK1.*.**' class(dataset) generic id(CRMBVK1) ACCESS(READ)
000011      /* Access level was ALTER */
000012 pe 'CRMBJU1.**' class(dataset) generic id(CRMB) ACCESS(READ)
000013      /* Access level was UPDATE */
000014 pe 'CRMBJU1.LOADLIB' class(dataset) generic id(CRMB) ACCESS(READ)
000015      /* Access level was UPDATE */
000016 pe 'CSF.SCSFMOD0' class(dataset) generic id(SYSPROG) ACCESS(READ)
000017      /* Access level was UPDATE */
```

Figure 6-12 Automated RACF permit commands to implement least privilege principle generated

Important: Before running these automatically generated commands, it is strongly suggested that you carefully review the generated commands before executing them.

Verify that the generated permit access level is not a pseudo access level from a define-type access event that does not RACF does not support for data set. The only valid values that you can encounter are UPDATE, CONTROL, or ALTER. When you encounter other values, you can adjust the `access_intent_max_suc<>` filters in your CARLa script and rerun it. Alternatively, you can also just delete the generated permit command from this CKRCMD work data set before executing the commands.

Double check that eye catcher that reports the to be replaced access level does not contain any values that are not a real access level that RACF supports, such as QUALOWN, ALTER-O, or other. The only valid values that you can encounter are UPDATE, CONTROL, or ALTER. When you encounter other values, you can adjust the ACCESS <> filters in your CARLa script and rerun it. Alternatively, you can also just delete the generated permit command from this CKRCMD work data set before executing the commands.

Furthermore, you must consider for DATASET profiles that some job role must be the custodian of these data sets. That custodian needs the authority to create, update, and delete the protected data sets. So even when that custodian job role did not use ALTER access last year, it is probably not a good idea to reduce the ALTER access. Therefore, you must diagnose whether the ALTER permission is the only ID with ALTER access on the ACL. In that case, it is probably better to leave the ALTER permission unchanged. However, if the HLQ of the DATASET profile matches a user ID, it is probably OK to reduce the ALTER

permissions. Using personal user data sets in a production environment is not a good implementation by any means.

In the example, the diagnostics review revealed that the ALTER permissions to group SYSPROG must not be reduced as the z/OS systems programmers are the custodians for these data sets that do require ALTER access in certain circumstances.

After removing the generated permit commands to ID SYSPROG from your CKRCMD work data set, you can run the commands against your offline RACF database with command **GO** or **RUN** (see Figure 6-13), and pressing Enter.

```
BROWSE    CRMBTZ1.DATA.C2R1DC6.CKRTSPRT           Line 0000000000 Col 001 080
 Command ===> ____________________________________________ Scroll ===> CSR
******************************** Top of Data ********************************
================================================================================
=== Multiple TSO command output file - scroll max down for overview ===
=== Input data set CRMBTZ1.DATA.C2R1DC6.TEMP.CKRCMD                  ===
================================================================================
================================================================================
=== Commands for local node
================================================================================
/* CKRCMD file CKR1CMD complex TVT6003 generated 13 Nov 2024 09:21 */

============ 13Nov24 09:42:57.04579 start record 2 ============
pe 'CATALOG.**' class(dataset) generic id(IDSTAR) ACCESS(READ)
/* Access level was UPDATE */

============ 13Nov24 09:42:57.05440 start record 4 ============
pe 'CKNSERVE.**' class(dataset) generic id(C2RSERVG) ACCESS(READ)
/* Access level was UPDATE */

============ 13Nov24 09:42:57.06144 start record 6 ============
```

Figure 6-13 CKRTSPRT permit commands to implement least privilege principle executed

As usual, you are automatically transferred to the CKRTSPRT work data set that shows the results of the executed permit commands. All commands processed successful. You can scroll down to the bottom of the CKRTSPRT work data set that reports the message that all commands processed successfully.

6.4.3 Verifying the permit commands for the DATASET class

In this section, you will learn how to verify that the executed permit commands to improve the least privilege principle implementation for DATASET class worked as anticipated.

Press **F3** a couple of times to leave ISPF, and then terminate your RACF-Offline session with the **END** command. Next, issue command **ISPF** in the command line and press Enter, to start a regular TSO session that uses the primary RACF database for access verification checks.

The next step is to verify that your commands improved the implementation of the least privilege principle in your environment. Start zSecure Admin and ensure that your allocated input set contains your offline RACF database that you ran the permit commands against.

Next, use option **AM.3** again to report the occurrences where DATASET profiles permit a higher access level to some IDs than the permitted IDs use (see Figure 6-14).

```
                    zSecure Admin - Access - Permit usage
Command ===> ____________________________________________________________

Show permits that fit all of the following criteria:
Permit id . . . . . _______  (permit id or EGN mask on access list)
Class . . . . . dataset_  (class or EGN mask)
RACF profile name . . . _____________________________________________
Complex . . . . . _______  (complex or EGN mask)
Show accesses . . . . . /  Non-zero counts     _  Zero counts

   Profile to use  2  1. Use historic profile name in access summary if present
                      2. Simulate access in database to find current profile

Advanced selection criteria
/  Further selection                   _  Date selection

Output/run options
_  Print format          Customize title     Send as e-mail
      Background run     Full page form
```

Figure 6-14 Report historic access to DATASET class resources against offline RACF database

After you press Enter, the further selection panel appears. On this panel, you must select option “Highest access used less than access allowed” to find occurrences where the “least privilege” principle appears not to be correctly implemented (see Figure 6-15 on page 135). Press Enter.

```
                    zSecure Admin - Access - Further selec
Command ===> ____________________________________________________________
Access monitor records for Classes like DATASET, non-zero counts
Access selection
#Accesses allowed . . . __  __________
#Accesses prevented . . __  _____
#Accesses unexplained    __  _____

Permit selection
Creation date from  . . __________  Until   __________

/  Highest access used less than access allowed

Access allowed        Highest access used   Lowest violation
>=  _  1. Execute        <=  _  1. Read       >=  _  1. Read
       2. Read                  2. Update           2. Update
       3. Update                3. Control          3. Control
       4. Control               4. Alter            4. Alter
       5. Alter
```

Figure 6-15 Report historic DATASET access where highest access used is less than access allowed

That selection generates a permits usage overview report of DATASET profiles that are used to access protected data sets, where the permitted access level exceeds the highest access used by permitted IDs (see Figure 6-16). You can use action command **S** and press Enter to access the DATASET class profile details to verify which permissions are potential candidates to be tightened.

```
                                                                    Line 1 of 26
Unconditional permits and UACC, by class complex/profile
Command ===> _____________________________________________ Scroll===> CSR
Access monitor records for Classes like DATASET 13 Nov 2024 12:17
   Allowed Deny  Unexp LastUse Class    Complex
     56232   248     0 31Oct24 DATASET  TVT6003
   Allowed Deny  Unexp LastUse Type     Profile
__       1     0     0 25Oct24 GENERIC  CKNSERVE.**
__       7     0     0 23Oct24 GENERIC  CRMA.T.**
__   20710     0     0 31Oct24 GENERIC  CRMA.X.**
__    6378     0     0 31Oct24 GENERIC  CRMAUTO.**
__       1     0     0 12Sep24 GENERIC  CRMBAH1.TEST12.**
__     187     0     0  5Dec23 GENERIC  CRMBEP1.*.**
__      87     0     0 31Oct24 GENERIC  CRMBJK1.*.**
__    4888     0     0 30Oct24 GENERIC  CRMBMK1.*.**
__       2     0     0 13Sep24 GENERIC  CRMBMK1.TE%T
__      30     0     0 22Mar24 GENERIC  CRMBPH1.*.**
__      10     0     0 16Oct24 GENERIC  CRMBRL1.*.**
__     162     0     0 16Sep24 GENERIC  CRMBRS1.*.**
__    3344     0     0 31Oct24 GENERIC  CRMBTZ1.*.**
__    1472     0     0 31Oct24 GENERIC  CRMBVK1.*.**
__      58     0     0 16Oct24 GENERIC  CRMBVK2.*.**
__      13   248     0 31Oct24 GENERIC  CSF.SCSFMOD0
__   11597     0     0 31Oct24 GENERIC  C2PACMON.*.**
```

Figure 6-16 Report DATASET profiles where permitted access exceeds used access

As illustrated, the permit usage report still reports 26 DATASET profiles with permissions that permits an access level that exceeds the used access level. However, when you zoom in to the profile details, you notice that the pseudo access levels such as QUALOWN and ALTER-O are still included in this report.

Press **F3**, issue the **RESULT** command on the command line and press Enter to access the Results panel. Next, use action command **S** or **E** to access the COMMANDS work data set and press Enter. Scroll down with **F8** and add filter `access<>(QUALOWN,ALTER-O)` to the select statement and issue command **GO** or **RUN** to regenerate the permit usage report.

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> GO                                              Scroll ===> CSR
000018  define auth#(5,"Auth",dec,bw,noprop)  count
000019  define helppanel=C2R3Z243,
000020         Miss#(7,"Missing",dec,noprop)  count where
000021                    (proftype="missing"C)
000022  define Pres#(8,"Permits",dec,noprop) count where
000023               (proftype<>"missing"C)
000024  define helppanel=C2R3Z242,
000025         Perm#(8,"Permits and UACC",dec,noprop) count
000026 select raclist_merge=no class<>group access<>(QUALOWN,ALTER-0),
000027  proftype<>GLOBAL,class=DATASET,(access_count_suc>0 or,
000028  access_count_vio>0 or access_count_unk>0),(,,
000029  access_intent_max_suc<<access)
000030 display id(nd),
000031    access_count_suc(7,key),
000032    access_count_vio(5,"Deny",key),
000033    access_count_unk(5,key),
000034    access_lastuse(7,key,"LastUse"),
000035    id(pas,key) access access_intent_max_suc("Used"),
000036    access_intent_min_vio access_reduced merged_access_reduced,
000037    id:name id:revoke(1) | id:revoke_inactive(1) " ",
```

Figure 6-17 Adjust the CARLa script that the UI generated under the covers and rerun the report

When you press Enter, your adjusted CARLa script is executed and produces a customized version of the original permit usage report about DATASET class profiles with permissions that have a higher access level than permitted IDs use (see Figure 6-18 on page 137).

```
                                                                Line 1 of 11
Unconditional permits and UACC, by class complex/profile
 Command ===> _____________________________________________ Scroll===> CSR
,Access monitor records for Classes like DATASE 13 Nov 2024 12:29
   Allowed Deny  Unexp LastUse Class    Complex
     30548   248     0 31Oct24 DATASET  TVT6003
   Allowed Deny  Unexp LastUse Type     Profile
__       7     0     0 23Oct24 GENERIC  CRMA.T.**
__   20710     0     0 31Oct24 GENERIC  CRMA.X.**
__    6378     0     0 31Oct24 GENERIC  CRMAUTO.**
__      13   248     0 31Oct24 GENERIC  CSF.SCSFMOD0
__      79     0     0 27Nov23 GENERIC  C2RSRV#P.**
__       4     0     0 16Sep24 GENERIC  SYS1.BRODCAST
__       1     0     0  2Nov23 GENERIC  SYS1.DSDBCTRL
__       1     0     0  2Nov23 GENERIC  SYS1.DSDB1
__       1     0     0  2Nov23 GENERIC  SYS1.DSDB2
__       3     0     0 12Apr24 GENERIC  SYS1.MAN*
__    3351     0     0 31Oct24 GENERIC  SYS1.TVT6003.**
****************************** Bottom of Data ******************************
```

Figure 6-18 Report remaining DATASET profiles where permitted access exceeds used access

As illustrated by the fact that your report now only includes 11 DATASET profiles. That number dropped from 32 in Figure 6-3 on page 126 and from 26 in Figure 6-16 on page 136.

Why does the report still show 11 DATASET profiles with permitted access levels that exceed the used access level? Remember that your research revealed that for some of the originally reported DATASET profiles the SYSPROG group is the custodian job role for these protected data sets. You then decided to remove the generated permit commands to SYSPROG from the CKRCMD work data set prior to running the commands.

Thus, if all worked as expected, the remaining 11 DATASET profiles still report a permission with ALTER access to ID SYSPROG that need to be able to add or delete these data sets in specific circumstances (see Figure 6-19). You can confirm that theory using action command **S** to access the profile details.

```
                                                                  Line 1 of 1
Unconditional permits and UACC, by class complex/profile
Command ===> _____________________________________________ Scroll===> CSR
,Access monitor records for Classes like DATASE 13 Nov 2024 12:29
   Allowed Deny  Unexp LastUse Class    Complex
     30548   248     0 31Oct24 DATASET  TVT6003
   Allowed Deny  Unexp LastUse Type     Profile
     20710     0     0 31Oct24 GENERIC  CRMA.X.**
   Allowed Deny  Unexp LastUse Id       Access    Used    Failed  Red RdM Name
__   20710     0     0 31Oct24 SYSPROG  ALTER     READ            No
****************************** Bottom of Data ********************************
```

Figure 6-19 Report DATASET profiles where allowed access intentionally exceeds used access

If you are happy with this outcome, the next step is to execute the same permit commands against the primary RACF database.

6.4.4 Executing permit commands against the active primary RACF database

In this section, you will learn how to execute the permit commands to implement the least privilege principle for DATASET class against active primary RACF database.

When you saved the CARLa script that is shown in Figure 6-11 on page 131, you can reuse that same CARLa script. In fact, you can even improve the script by adding filter `ID<>SYSPROG` to the select statement to prevent that permits for group SYSPROG are generated (see Figure 6-20).

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 simulate racf_access
000002 n type=RACF_ACCESS nopage dd=ckrcmd
000003 select raclist_merge=no class=dataset access<>(QUALOWN,ALTER-0),
000004  proftype<>GLOBAL (access_count_suc>0 or access_count_vio>0 or,
000005  access_count_unk>0) access_intent_max_suc<<access ID<>SYSPROG
000006  sortlist "pe '" | profile(0) | "' class(dataset) generic id(" |,
000007           id(0) | ") ACCESS(" | access_intent_max_suc(0) | ")" / ,
000008           "     /* Access level was" access(0) "*/"
****** **************************** Bottom of Data ****************************
```

Figure 6-20 Improved CARLa script to generate appropriate permit commands

Before running the CARLa script, you must switch your selected input set in zSecure. Use option **SE.1** to select an input set that contains a different RACF input source than your offline RACF database, and your access monitor consolidation data set.

Running this version of the script produces the following commands in CKRCMD (see Figure 6-21 on page 139).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                    Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** **************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated 13 Nov 2024 13:43 */
000002 pe 'CATALOG.**' class(dataset) generic id(IDSTAR) ACCESS(READ)
000003      /* Access level was UPDATE */
000004 pe 'CKNSERVE.**' class(dataset) generic id(C2RSERVG) ACCESS(READ)
000005      /* Access level was UPDATE */
000006 pe 'CRMBJK1.*.**' class(dataset) generic id(CRMBVK1) ACCESS(READ)
000007      /* Access level was ALTER */
000008 pe 'CRMBJU1.**' class(dataset) generic id(CRMB) ACCESS(READ)
000009      /* Access level was UPDATE */
000010 pe 'CRMBJU1.LOADLIB' class(dataset) generic id(CRMB) ACCESS(READ)
000011      /* Access level was UPDATE */
000012 pe 'SYSAPPL.**' class(dataset) generic id(CRMAUTO) ACCESS(UPDATE)
000013      /* Access level was ALTER */
000014 pe 'SYS1.*.**' class(dataset) generic id(CRMBVK9) ACCESS(READ)
000015      /* Access level was UPDATE */
000016 pe 'SYS1.*.**' class(dataset) generic id(SYS1) ACCESS(READ)
```

Figure 6-21 Automated RACF permit commands to implement least privilege principle regenerated

You can issue command **GO** or **RUN** to execute them against the primary RACF database.

Another way to rerun the permit commands against the active primary RACF database is using your RACF-Offline log data set. The log data set stores the RACF commands that you executed during your last RACF-Offline session.

After switching to an zSecure Admin input set that contains a different RACF source than your offline RACF database, issue command **RESULT** in the command line and press Enter (see Figure 6-22 on page 140).

```
                          zSecure Admin - Results       Enter R to run commands
Command ===> ____________________________________________________________________

The following selections are supported:
 B Browse file               S Default action (for each file)
 E Edit file                 R Run commands
 P Print file                J Submit Job to execute commands
 V View file                 M E-mail report
 W Write file into seq. or partitioned data set

Enter a selection in front of a highlighted line below:
_  SYSPRINT  messages
_  REPORT    printable reports
_  CKRTSPRT  output from the last TSO command(s)
E  CKRCMD    queued TSO commands
_  CKR2PASS  queued commands for zSecure Admin
_  COMMANDS  zSecure Admin input commands from last query
_  SPFLIST   printable output from PRT primary command
_  OPTIONS   set print options
```

Figure 6-22 Edit the CKRCMD work data set

Use action command **E**, for Edit, to access the CKRCMD data set and press Enter (see Figure 6-23).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                   Columns 00001 00072
Command ===> copy 'CRMB.T.RACF.OFFLINE.B8RLOG'           Scroll ===> CSR
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** Bottom of Data ****************************
```

Figure 6-23 Copy RACF commands from last RACF-Offline session to the CKRCMD work data set

After pressing Enter, you receive warning message (see Figure 6-24).

```
                              EDIT - Confirm Copy
Command ===> ____________________________________________________________________

Data set attributes are inconsistent. Truncation may result in the right-most
portions of some "from" records if copy is performed.

  "From" data set attributes:
     Data set name . : CRMB.T.RACF.OFFLINE.B8RLOG
     Record format . : VARIABLE
     Record length . : 32,752

  "Current" data set attributes:
     Data set name . : CRMBTZ1.DATA.C2R1DC6.CKRCMD
     Record format . : FIXED
     Record length . : 80

Press ENTER key to copy with truncation.
Enter END command to cancel copy.
```

Figure 6-24 Truncation warning for copying RACF-Offline commands to the CKRCMD work data set

Because you know that none of the generated permit commands are longer than 80 characters, you can press Enter to copy with truncation. However, instead of using the zSecure CKRCMD work data set, you may prefer to use a batch job to run these commands from data set “CRMB.T.RACF.OFFLINE.B8RLOG” in the background.

After you pressed Enter, the logged commands are successfully copied from your RACF-Offline log data set “CRMB.T.RACF.OFFLINE.B8RLOG” to your CKRCMD work data set (see Figure 6-25).

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                   Columns 00001 00072
Command ===> ____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001 CKGRACF show myaccess
000002 PE 'CATALOG.**' class(dataset) generic id(IDSTAR) ACCESS(READ)
000003 PE 'CKNSERVE.**' class(dataset) generic id(C2RSERVG) ACCESS(READ)
000004 PE 'CRMBJK1.*.**' class(dataset) generic id(CRMBVK1) ACCESS(READ)
000005 PE 'CRMBJU1.**' class(dataset) generic id(CRMB) ACCESS(READ)
000006 PE 'CRMBJU1.LOADLIB' class(dataset) generic id(CRMB) ACCESS(READ)
000007 PE 'SYSAPPL.**' class(dataset) generic id(CRMAUTO) ACCESS(UPDATE)
000008 PE 'SYS1.*.**' class(dataset) generic id(CRMBVK9) ACCESS(READ)
000009 PE 'SYS1.*.**' class(dataset) generic id(SYS1) ACCESS(READ)
000010 PE 'SYS1.LOGREC' class(dataset) generic id(SYS1) ACCESS(UPDATE)
000011 PE 'SYS1.MAN*' class(dataset) generic id(SYS1) ACCESS(CONTROL)
000012 PE 'SYS1.TVT6003.**' class(dataset) generic id(OMVSGRP) ACCESS(READ)
000013 PE 'USER.*.**' class(dataset) generic id(CRMBEP1) ACCESS(READ)
000014 PE 'USER.*.**' class(dataset) generic id(CRMBGUS) ACCESS(READ)
000015 PE 'USER.*.**' class(dataset) generic id(CRMBMC1) ACCESS(READ)
000016 PE 'USER.*.**' class(dataset) generic id(CRMBPH1) ACCESS(READ)
```

Figure 6-25 RACF commands from last RACF-Offline session copied to CKRCMD work data set

Note that the command "`CKGRACF show myaccess`" in line 1 is executed when you start zSecure Admin in an ISPF environment. zSecure supports customizing menu options and action commands that a zSecure user is authorized to use. The CKGRACF show myacces command is executed to verify which zSecure menu options and action commands you are authorized to use. Access checks to the resources starting with prefix CKR.ACTION and CKR.ACTION in, by default, the XFACILIT class control which menu options and action commands a zSecure user can use. However, most zSecure customers have not defined CKR.ACTION.** and CKR.ACTION.** profiles in the XFACILIT class. It does not matter whether you delete the CKGRACF show myacces command or leave it in when you run the commands.

With command **`GO`** or **`RUN`**, you can run these commands against the active RACF database.

This step concludes the walk through for the implementation of least privilege for DATASET class. However, because the permit command syntax for general resource classes differs from DATASET class, you need a slightly different CARLa script for general resources.

6.4.5 Reporting permitted general resource access levels exceeding the used access levels

In this section, you will learn how to report historic successful general resource access where the allowed access level exceeds the used access level.

Use the same option **AM.3**, for Permit usage, but this time specify a different class name than DATASET, for example FACILITY, and use the same filters as you used to report the DATASET class access.

That selection generates a permit usage report along the lines of Figure 6-26.

```
                                                                  Line 1 of 9
Unconditional permits and UACC, by class complex/profile
Command ===> _____________________________________________ Scroll===> CSR
Access monitor records for Classes like FACILIT 13 Nov 2024 14:54
   Allowed Deny  Unexp LastUse Class    Complex
    121151     0     0 31Oct24 FACILITY TVT6003
   Allowed Deny  Unexp LastUse Type     Profile
__  118983     0     0 25Oct24 GENERIC  CSVAPF.**
__     245     0     0 12Sep24 GENERIC  CSVDYNEX.**
__       8     0     0 25Oct24 GENERIC  CSVDYNL.**
__     655     0     0 25Oct24 GENERIC  CSVLLA.**
__     330     0     0 31Oct24 DISCRETE IRR.DIGTCERT.GENCERT
__     329     0     0 31Oct24 DISCRETE IRR.DIGTCERT.LISTRING
__     396     0     0 29Oct24 GENERIC  STGADMIN.ADR.*
__     198     0     0 29Oct24 GENERIC  STGADMIN.ADR.STGADMIN.*
__       7     0     0 16Sep24 GENERIC  STGADMIN.IGG.*
****************************** Bottom of Data ********************************
```

Figure 6-26 Report FACILITY profiles where allowed access exceeds used access

Zooming into a FACILITY profile level with action command **S** reveals which permissions allow a higher access level as the permitted IDs used (see Figure 6-27).

```
                                                                  Line 1 of 2
Unconditional permits and UACC, by class complex/profile
Command ===> _____________________________________________ Scroll===> CSR
Access monitor records for Classes like FACILIT 13 Nov 2024 14:54
   Allowed Deny  Unexp LastUse Class    Complex
    121151     0     0 31Oct24 FACILITY TVT6003
   Allowed Deny  Unexp LastUse Type     Profile
       245     0     0 12Sep24 GENERIC  CSVDYNEX.**
   Allowed Deny  Unexp LastUse Id       Access    Used    Failed  Red RdM Name
__     222     0     0 12Sep24 CRMBZDEV ALTER     UPDATE          No
__      23     0     0 30May24 SYSPROG  ALTER     UPDATE          No
****************************** Bottom of Data ********************************
```

Figure 6-27 Report FACILITY profiles where permitted access exceeds used access

In this example, two permissions are reported to allow ALTER access to IDs CRMBZDEV and SYSPROG, but according to access monitor their highest access level used last year is UPDATE. These permits are good candidates to be reduced to improve least privilege implementation.

Note that in contradiction to the DATASET class, general resource classes contain profiles that do not protect a physical resource such as a program, terminal, console, transaction, or other resource type. Some general resource profiles use a resource access check to control the use of certain functions and/or features of an application or a sub-system but do not protect a physical resource. For example:

- FACILITY class profile BPX.SUPERUSER controls which users are authorized in UNIX System Services (USS) to switch to UNIX SUPERUSER mode UID(0).
- UNIXPRIV class profile SHARED.IDS controls which RACF administrators are authorized to assign a shared UID value to multiple users.
- Many other such profiles in different general resource classes might exist.

For resource profiles that do not protect a physical resource, the role of custodian for that resource does not apply. In other words, for that type of resource profiles it is acceptable that the ACL does not contain any permissions with ALTER access level.

Press **F3** twice to leave the report that shows the reported FACILITY class profiles, issue command `result` in the command line, and press Enter to access the zSecure Results panel.

You want to create a CARLa script that automatically produces permit commands to improve the least privilege principle implementation for all general resource classes on your system.

Scroll down with **F8** to review the composed select statement (see Figure 6-28 on page 144).

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
000020         Miss#(7,"Missing",dec,noprop)  count where
000021                    (proftype="missing"C)
000022  define Pres#(8,"Permits",dec,noprop) count where
000023               (proftype<>"missing"C)
000024  define helppanel=C2R3Z242,
000025         Perm#(8,"Permits and UACC",dec,noprop) count
000026 select raclist_merge=no class<>group,
000027  proftype<>GLOBAL,class=FACILITY,(access_count_suc>0 or,
000028  access_count_vio>0 or access_count_unk>0),(,,
000029  access_intent_max_suc<<access)
000030 display id(nd),
000031    access_count_suc(7,key),
000032    access_count_vio(5,"Deny",key),
000033    access_count_unk(5,key),
000034    access_lastuse(7,key,"LastUse"),
000035    id(pas,key) access access_intent_max_suc("Used"),
000036    access_intent_min_vio access_reduced merged_access_reduced,
000037    id:name id:revoke(1) | id:revoke_inactive(1) " ",
000038    id:dfltgrp id:instdata,
000039  / "Profile in current database"(d,ch),
```

Figure 6-28 Composed select statement generated in the COMMANDS work data set

Like the DATASET class, you can use this composed select statement for your customized CARLa script that produces permit commands that implement the least privilege principle for the general resource class profiles in your system.

Using the same logic as is used for the DATASET class, you can adjust the CARLa script to generate permit commands for all general resource class profiles where permitted access level exceeds the used access level (see Figure 6-29).

```
EDIT       CRMBTZ1.CARLA.SAMPLES(@CRMBTZ1) - 01.00         Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 simulate racf_access
000002 n type=RACF_ACCESS nopage dd=ckrcmd
000003 select raclist_merge=no class<>(dataset,group) access<>ALTER-0,
000004  proftype<>GLOBAL (access_count_suc>0 or access_count_vio>0 or,
000005  access_count_unk>0) (access_intent_max_suc<<access)
000006  sortlist 'pe' profile(0) 'class(' | class(0) | ') id(' | id(0) | ')',
000007           'ACCESS(' | access_intent_max_suc(0) ')' / ,
000008           '     /* Access level was' access(0) '*/'
****** **************************** Bottom of Data ****************************
```

Figure 6-29 Customized CARLa script to produce permit commands for general resource classes

Notes regarding the CARLa code adjustments:

- Lines 1 and 2 are the same as in the CARLa script for DATASET class.
- In the select statement in lines 3 to 5 the following changes are performed.
  - The `class<>group` specification is changed to `class<>(dataset,group)` to exclude profiles from DATASET class as well.

- Filter `access<>ALTER-0` is added to prevent permit commands being generated to replace the pseudo access level in general resource class that honour the OPERATIONS attribute and allow ALTER access.
- The `class=FACILITY` filter is removed to generate permit commands for resource profiles of all general resource classes, if applicable.

You can run your customized CARLa script with command **GO** or **RUN** (see Figure 6-30), and press Enter.

```
EDIT       CRMBTZ1.DATA.C2R1DC6.CKRCMD                          0.2 s CPU, RC=4
Command ===> _____________________________________________ Scroll ===> CSR
****** **************************** Top of Data ******************************
=NOTE= Enter GO or RUN to execute commands, SUB or SUBMIT to generate batch job
=NOTE= You can also press PF3, enter R at the cursor location, and press ENTER.
000001  /* CKRCMD file CKR1CMD complex TVT6003 generated 13 Nov 2024 17:33 */
000002 pe ** class(TSOPROC) id(SYSPROG) ACCESS(READ )
000003      /* Access level was ALTER */
000004 pe B8R.RACF.OFFLINE class(XFACILIT) id(CRMBQA) ACCESS(READ )
000005      /* Access level was UPDATE */
000006 pe B8R.RACFDB.*.** class(XFACILIT) id(CRMBQA) ACCESS(UPDATE )
000007      /* Access level was CONTROL */
000008 pe B8R.RACFDB.*.** class(XFACILIT) id(CRMBZDEV) ACCESS(UPDATE )
000009      /* Access level was CONTROL */
000010 pe B8R.RACFDB.*.** class(XFACILIT) id(SYSPROG) ACCESS(UPDATE )
000011      /* Access level was CONTROL */
000012 pe CKG.CMD.FIELD.* class(XFACILIT) id(CRMBLU1) ACCESS(READ )
000013      /* Access level was UPDATE */
000014 pe CKG.CMD.FIELD.* class(XFACILIT) id(SYSPROG) ACCESS(READ )
000015      /* Access level was UPDATE */
000016 pe CKG.CMD.FIELD.PWDX class(XFACILIT) id(CRMBLU1) ACCESS(READ )
```

Figure 6-30 Automated RACF permit commands to implement least privilege principle generated

The appropriate RACF permit commands to reduce permitted access levels based on evidence from the access monitor records are successfully generated in the CKRCMD work data set.

Important: Make sure you review all generated commands and perform inquiries when any generated commands are unexpected. For general resource profiles that protect a physical resource, you need to ensure not to the remove the ALTER access for the custodian even when ALTER access has not been used in the last year.

When you are finished with your review and you removed the commands that must not be executed, you can run the commands against the offline RACF database. Use commands **GO** or **RUN** to execute them in the foreground or use command **SUB** or **SUBMIT** to run them in the background as a batch job instead.

The next steps to take, include:

- Verify that the executed permit commands to improve the least privilege principle implementation for general resource classes worked using the same steps as documented for DATASET class in the previous section.
- Execute the same permit commands to implement the least privilege principle for general resource classes against active primary RACF database.

This concludes the walk through of use case: “Minimizing access control privileges”.

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5” spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

Redbooks

IBM RACF Security Impact Forecasting

SG24-8578-00

ISBN

IBM

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

Redbooks

IBM RACF Security Impact Forecasting using

SG24-8578-00

ISBN

IBM

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Redbooks

IBM RACF Security Impact Forecasting using IBM zSecure

SG24-8578-00

ISBN

IBM

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Redbooks

IBM RACF Security Impact Forecasting using IBM zSecure

(0.2”spine)
0.17”<->0.473”
90<->249 pages

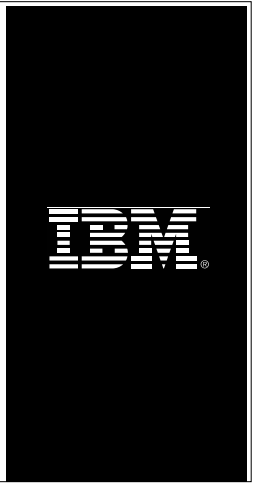
(0.1”spine)
0.1”<->0.169”
53<->89 pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5” spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

IBM RACF Security Impact

SG24-8578-00

ISBN

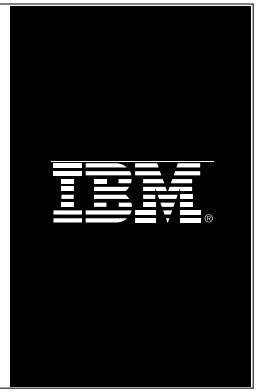


(2.5” spine)
2.5”<->nnn.n”
1315<-> nnnn pages

IBM RACF Security Impact Forecasting using IBM zSecure

SG24-8578-00

ISBN



(2.0” spine)
2.0” <-> 2.498”
1052 <-> 1314 pages

SG24-8578-00

ISBN

Printed in U.S.A.