



IBM z/OS Mainframe Security and Audit Management Using the IBM Security zSecure Suite

Increase the efficiency of your RACF security management

Address mainframe audit and compliance

Understand all zSecure components



Axel Buecker
Michael Cairns
Monique Conway
Mark S. Hahn
Deborah McLemore
Jamie Pease
Lili Xie

ibm.com/redbooks

Redbooks



International Technical Support Organization

**IBM z/OS Mainframe Security and Audit
Management Using the IBM Security zSecure Suite**

August 2011

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

Second Edition (August 2011)

This edition applies to version 1, release 12 of IBM Security zSecure products and to all subsequent releases and modifications until otherwise indicated.

© Copyright International Business Machines Corporation 2008, 2011. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xi
Trademarks	xii
Preface	xiii
The team who wrote this book	xiii
Now you can become a published author, too!	xvi
Comments welcome	xvii
Stay connected to IBM Redbooks	xvii
Summary of changes	xix
August 2011, Second Edition	xix
Part 1. Architecture and design	1
Chapter 1. Business context	3
1.1 Today's challenges	4
1.2 Risk management, IT governance, and compliance	5
1.2.1 Risk management	5
1.2.2 IT governance	6
1.2.3 Regulatory compliance	6
1.3 Business enablement	7
1.3.1 IT Service Management and IT Infrastructure Library	7
1.3.2 System z and IBM Security	8
1.4 Conclusion	8
Chapter 2. IBM Security zSecure component structure	11
2.1 zSecure at a glance	12
2.1.1 Operating systems supported	13
2.1.2 Security systems supported	14
2.2 zSecure Admin	14
2.3 zSecure Visual	17
2.4 zSecure CICS Toolkit	18
2.4.1 Command interface	19
2.4.2 Application programming interface	20
2.5 zSecure Audit	20
2.6 zSecure Alert	22
2.7 zSecure Command Verifier	24
2.8 zSecure Manager for RACF z/VM	25
2.9 IBM Security zSecure Compliance Insight Manager Enabler for z/OS ...	26

2.10 Conclusion	27
Chapter 3. IBM Security zSecure Admin	29
3.1 An easy to use RACF administration interface	31
3.1.1 Initial setup	31
3.1.2 An easy to use display of RACF profiles	32
3.1.3 Adding a new general resource profile	37
3.2 Automating and simplifying routine administration tasks	42
3.2.1 Mass changes to RACF and block command support	42
3.2.2 RACF Offline	49
3.2.3 Timed actions	52
3.2.4 Single action to perform an access check	57
3.2.5 Complete access report	57
3.2.6 Automated verification and cleanup	59
3.2.7 Access Monitor for additional cleanup	60
3.2.8 Automated reporting using CARLa	61
3.2.9 Recovering from administrator errors	61
3.3 Delegating RACF administration tasks	63
3.4 Preventing and identifying problems to minimize threats	64
3.4.1 Using reports provided by zSecure Admin	65
3.4.2 Customizing your own report display	67
3.4.3 Integration with zSecure Audit	68
3.5 Other enhancements for RACF administration	68
3.5.1 Storing user data and installation data in RACF	69
3.5.2 Access list display modes	70
3.5.3 RACF database merge processing	71
3.6 Conclusion	71
Chapter 4. IBM Security zSecure Alert	73
4.1 Product positioning and features	74
4.2 zSecure Alert architecture and processing	75
4.2.1 zSecure Alert data flow	75
4.2.2 zSecure Alert components	76
4.2.3 Address space and data collection mechanism	77
4.3 Implementation suggestions	78
4.3.1 Initial setup	78
4.3.2 Selecting alerts ready for use	79
4.3.3 Sending alerts to their destination	81
4.3.4 Adding your own alerts	83
4.4 Integration guidelines	85
4.5 Conclusion	87
Chapter 5. IBM Security zSecure Audit	89
5.1 zSecure Audit architecture	90

5.2	Initial setup of zSecure Audit	92
5.2.1	Input data used by zSecure Audit	92
5.2.2	Security controls for zSecure Audit	96
5.3	System environment reporting	97
5.3.1	Audit priorities and policies	98
5.3.2	Trust, profile audit concerns, and sensitive data trustees	99
5.3.3	Automated vulnerability analysis	100
5.3.4	Data sets used by System Status	101
5.4	Database verification and cleanup	101
5.4.1	Requirements for using verify reports	102
5.5	Event reporting	103
5.5.1	HTTP reporting	105
5.5.2	Requirements for using event and HTTP reports	105
5.6	Change tracking	106
5.6.1	Data sets used by change tracking	107
5.6.2	Security controls for change tracking use	108
5.7	Library and sequential data set audit	108
5.7.1	Data sets used by data set audit	109
5.7.2	Security controls for using library analysis	110
5.8	Conclusion	110
Chapter 6.	IBM Security zSecure Visual	111
6.1	zSecure Visual architecture and implementation	112
6.2	Usage scenarios	113
6.2.1	zSecure Visual default roles	113
6.2.2	RACF setup to support zSecure Visual	114
6.2.3	CKGRACF ID level scoping	115
6.2.4	CKGRACF USER and GROUP level scoping	116
6.2.5	Using RACF group special	119
6.2.6	Access level on scoping profiles	120
6.3	RACF scoping rules and examples	120
6.3.1	Local password administrator	121
6.3.2	Branch wide group connections administrator	122
6.3.3	Staff wide user administrator	123
6.3.4	Applications data administrator	125
6.3.5	Resource access list administrator	126
6.3.6	Multiple system support	128
6.4	Conclusion	129
Chapter 7.	IBM Security zSecure Command Verifier	131
7.1	zSecure Command Verifier architecture	132
7.2	Controlling RACF commands	133
7.2.1	Example policy profiles to control RACF changes	135

7.3	Replacing user exits	136
7.3.1	Profile locking	137
7.3.2	Temporary system special	138
7.4	Command audit trail feature	138
7.4.1	Activating command audit trail for profiles	139
7.4.2	Auditing changes failed by zSecure Command Verifier	140
7.4.3	Reviewing profile changes	140
7.4.4	Maintaining the command audit trail information.	141
7.5	Alerting and action capabilities	141
7.6	Conclusion.	143
Chapter 8. IBM z/OS compliance enablers		145
8.1	What enablers are	146
8.2	Currently available Enablers for z/OS	147
8.3	Why you would want to use Enablers for z/OS	147
8.4	How Enablers for z/OS work	148
8.4.1	Enabler	149
8.4.2	Agent	149
8.4.3	Actuator	149
8.4.4	The data	149
8.4.5	The process	150
8.4.6	At a glance	151
8.5	Sample screen captures of Enablers for z/OS in action	151
8.6	Conclusion.	158
Chapter 9. IBM Security zSecure CICS Toolkit		161
9.1	zSecure CICS Toolkit architecture	162
9.1.1	Command interface architecture	162
9.1.2	Application programming interface architecture	164
9.2	Command interface usage	165
9.2.1	Customized panels	166
9.3	Application programming interface usage	166
9.3.1	Using the API for resource authorization checking	167
9.3.2	Using the API for RACF administration	168
9.4	Conclusion.	172
Chapter 10. Planning for deployment		173
10.1	Services engagement preparation	174
10.1.1	Implementation skills	174
10.1.2	Available resources	175
10.2	Solution descriptions	176
10.2.1	Audit and compliance solution	177
10.2.2	Administration solution	178
10.2.3	Monitoring solution	178

10.2.4 Reporting solution	179
10.3 Services engagement overview	179
10.3.1 Executive assessment	180
10.3.2 Demonstrating the system setup.	180
10.3.3 Analyzing solution tasks	181
10.3.4 Creating a contract	181
10.3.5 Defining solution tasks	182
10.3.6 Deployment tasks	185
10.4 Conclusion.	185
Part 2. Customer scenario	187
Chapter 11. Delft Transport Authority	189
11.1 Delft Transport company profile	190
11.2 Delft Transport IT security architecture	190
11.3 Corporate business vision and objectives.	191
11.4 Acquisition project	193
11.5 Conclusion.	193
Chapter 12. Project requirements and design	195
12.1 Business requirements	196
12.2 Functional requirements	197
12.3 Design approach	198
12.4 Implementation approach	199
12.5 Conclusion.	199
Chapter 13. Implementation phase I	201
13.1 Post systems programmer installation setup.	202
13.2 CKFREEZE, Signature, and UNLOAD generation data groups	203
13.3 RACF security for IBM zSecure	208
13.3.1 Program Access to Datasets.	212
13.3.2 Conclusion for RACF security.	212
13.4 Running initial analysis reports	212
13.4.1 Status audit reports.	213
13.4.2 Reviewing the current RACF group tree	229
13.5 Implementing initial improvements in system security posture	231
13.5.1 Implementing SETROPTS improvements.	232
13.5.2 Cleaning up badly defined data set profiles	235
13.5.3 Implementing an improved RACF group tree structure.	241
13.5.4 Planning for PROTECTALL implementation	249
13.6 Post implementation verification reports	254
13.6.1 Reducing trust levels.	257
13.6.2 RACF group structure	262
13.7 Conclusion.	263

Chapter 14. Implementation phase II	265
14.1 Audit reporting	266
14.1.1 Using supplied reports	266
14.1.2 Sensitive data set analysis	273
14.1.3 XML format audit reporting using CARLa	275
14.2 Ongoing monitoring	282
14.2.1 Using the change tracking feature	283
14.2.2 Delta reporting: Comparing profiles and databases	294
14.2.3 SYSLOG trapping in zSecure Alert	297
14.2.4 Sending SNMP data	298
14.2.5 Monitoring specific resources	301
14.2.6 Monitoring for critical system events	311
14.2.7 Monitoring RACF OPERATIONS attribute use	316
14.3 Conclusion	319
 Chapter 15. Implementation phase III	 321
15.1 Delegated RACF administration	322
15.1.1 Implementing zSecure Admin scoping	327
15.2 Ensuring system integrity	348
15.2.1 Enforcing standards	349
15.2.2 Preventing unwanted SETROPTS changes	354
15.2.3 No profiles in WARNING mode	354
15.2.4 No high UACC	355
15.2.5 Preventing or allowing elevation of authority	357
15.2.6 Lockdown profiles for segregation of responsibilities	358
15.2.7 Additional controls required for group special users	358
15.2.8 Assigning mandatory values	361
15.3 Processes for managing authorization	361
15.3.1 Timed (queued) commands	361
15.3.2 Temporary (queued) commands	371
15.3.3 Workflow for RACF commands	376
15.3.4 Access re-validation reporting	383
15.4 Reporting processes	385
15.4.1 Advanced use of CARLa for email bundle reporting	385
15.5 Joiners, leavers, and movers processing	389
15.5.1 Flagging users for revocation, revoking them, and changing ownership of those users	389
15.5.2 Reporting on deleted user IDs	392
15.5.3 Leavers processing	393
15.5.4 Joiners processing	395
15.5.5 Movers processing	396
15.6 Segregation of duties	401
15.6.1 Separating administrators by specialized function	401

15.6.2 Conflict detection in permits/roles	403
15.6.3 Mutually exclusive access reporting	404
15.7 Conclusion	407
Part 3. Appendices	409
Appendix A. Troubleshooting	411
Installation challenges	412
Ordering the software	412
Obtaining the licensed documentation	412
Receiving multiple zSecure products through SMP/E	415
Receiving zSecure Command Verifier	417
CKRINST	418
How to get help from zSecure Support	418
How to contact zSecure Support	421
What information to send to zSecure Support	422
Researching zSecure product maintenance	424
Finding zSecure technical documentation	425
Conclusion	426
Appendix B. An introduction to CARLa	427
About CARLa	428
Data sources	429
Writing a CARLa program	431
Basic CARLa to get you started	432
Where to write and run your CARLa program	432
Examples of basic CARLa programs for RACF reporting	436
Examples of basic CARLa programs for SMF reporting	444
Where to store your CARLa programs for reuse	450
Useful primary commands	451
Additional examples of CARLa	452
Conclusion	454
Appendix C. User roles for IBM Security zSecure Visual	455
RACF commands to generate zSecure Visual user roles	456
RACF commands sample	456
Appendix D. A look at the Consul to IBM Tivoli transformation	461
Information for users migrating from previous releases	462
Program name changes	462
RACF resource class changes	462

Consul and zSecure history 463

A personal note from Jamie Pease 464

A personal note from Mike Cairns 465

Related publications 467

IBM Redbooks 467

Other publications 467

How to get Redbooks 468

Help from IBM 468

Notices

This information was developed for products and services offered in the U.S.A.

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 give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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.

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 the names and addresses used by an actual business enterprise 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	NetView®	Tivoli Enterprise Console®
CICS®	RACF®	Tivoli®
DB2®	Redbooks®	VTAM®
IBM®	Redbooks (logo)  ®	z/OS®
Informix®	System i®	z/VM®
MVS™	System z®	

The following terms are trademarks of other companies:

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Novell, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

CA, CA ACF-2, and CA-Top Secret are trademarks of CA, Inc. in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

Every organization has a core set of mission-critical data that must be protected. Security lapses and failures are not simply disruptions—they can be catastrophic events, and the consequences can be felt across the entire organization. As a result, security administrators face serious challenges in protecting the company's sensitive data. IT staff are challenged to provide detailed audit and controls documentation at a time when they are already facing increasing demands on their time, due to events such as mergers, reorganizations, and other changes. Many organizations do not have enough experienced mainframe security administrators to meet these objectives, and expanding employee skillsets with low-level mainframe security technologies can be time-consuming.

The IBM® Security zSecure suite consists of multiple components designed to help you administer your mainframe security server, monitor for threats, audit usage and configurations, and enforce policy compliance. Administration, provisioning, and management components can significantly reduce administration, contributing to improved productivity, faster response time, and reduced training time needed for new administrators.

This IBM Redbooks® publication is a valuable resource for security officers, administrators, and architects who wish to better understand their mainframe security solutions.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.



Axel Buecker is a Certified Consulting Software IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on areas of Software Security Architecture and Network Computing Technologies. He holds a degree in computer science from the University of Bremen, Germany. He has 25 years of experience in a variety of areas related to Workstation and Systems Management, Network Computing, and e-business Solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture.



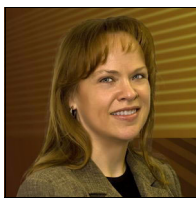
Michael Cairns is a Technical Sales Specialist with IBM Tivoli® ANZ. He has worked directly for a wide variety of IBM mainframe customers since 1986 in Australia, New Zealand, and the UK, both large and small. He joined IBM in 2007 with the acquisition of the zSecure suite of mainframe security management products. He specializes in IBM z/OS® security, particularly the IBM RACF® Security Server and associated products. His background includes Application Development, Systems Programming, Capacity and Performance Management, and Security Architecture. He teaches and mentors in mainframe security throughout the Asia Pacific region and is a Technical Editor at IBM Systems Magazine, regularly writing about z/OS management and security issues.



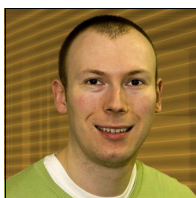
Monique Conway is the Level 2 Technical Support Team Lead for the zSecure suite of products. She has worked as a mainframe Systems Programmer for 23 years, concentrating on Security and zSecure for the past three years. Monique graduated from the three year Business Administration program at Algonquin College in Ottawa, Canada, with a concentration in Information Technology. Over the course of her mainframe career, Monique has been responsible for hardware and software support, computer room environments, and security. She has been an active volunteer with SHARE for the past 10 years and in her current role is Project Manager for the Security and Audit Project.



Mark S. Hahn is a Level 2 Technical Support member for the zSecure suite of products. He has worked in computer security for over 30 years and holds a degree in Computer Science from California Polytechnic State University, San Luis Obispo. Mark was a part of Global Services previously within IBM. Mark writes for mainframe security publications as well as speaking at multiple conferences and user groups regarding mainframe security in addition to being an active ribbon-wearer in the Security and Audit project in SHARE.



Deborah McLemore has been with IBM for 26 years and is currently working at IBM Austin, Texas. Over the course of her career she has held various positions across the Product Development, Sales, Support, and Services organizations as a Software Engineer. Deborah has designed and developed both mainframe applications and distributed applications ranging across z/OS, i5OS, IBM AIX®, Windows, and OS/2 Communications Manager, with primary programming roles on the Personal Communications 3270 and 5250 Emulation product suite. Deborah has also had engineering responsibilities for High Availability solutions and Security technologies that are embedded with IBM Branded software products. Deborah has published white papers and articles for various IBM Magazines and for Industry Publications related to 3270 and 5250 emulation. Deborah is currently specializing in IBM Education Delivery for the Americas geography for the Tivoli Brand.



Jamie Pease is an IT Specialist, currently working in a pre-sales technical role covering the IBM Security zSecure Suite within the United Kingdom, Ireland, and South Africa. Most of his career has been spent working on IBM System z®, specifically in security. Prior to joining IBM, Jamie spent 13 years at Aviva (Norwich Union), working in the Systems Security team as both a consultant and team manager, focusing on security development and support for System z security. During his time at Aviva, Jamie worked extensively with the zSecure software, including implementation, customization, support, and general usage of the software. After leaving Aviva, he joined HSBC Bank, where he worked within the Internal Audit department as an IT Auditor.



Lili Xie is an Advisory IT specialist with the technical sales support team supporting mainframe Tivoli products to customers in China. She has a Master's degree in Computer Science from Dong Hua University in Shanghai, China. She has seven years of experience in the mainframe systems management area, and her major expertise includes System Automation, performance monitoring, IBM Security zSecure solution, and batch workload scheduling on z/OS. She is currently engaged in mainframe security projects in China.

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

Wade Wallace

International Technical Support Organization, Austin Center

Special thanks go to Rob van Hoboken (IBM Netherlands) and Reese Meadows (IBM US) who spent the first week with the team and provided a lot of additional input to the overall effort over the duration of this project.

Another set of special thanks go to Tom Zeehandelaar (IBM Security zSecure Course Developer, IBM Netherlands) and Jeroen Tiggelman (IBM Security zSecure Development Manager, IBM Netherlands) for their extensive technical review and heaps of productive comments that we were able to incorporate into this publication.

Joel Tilton, Bob Haimowitz, Rich Conway, Andreina Monsalve, Julie Bergh,
Dimple Ahluwalia

IBM US

Richard P. Orr

**IT Security Architect, Technology Risk and Security
National Australia Bank**

Doc Farmer

Senior Security Specialist at InfoSec, Inc. (US)

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 ITSO 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, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7633-01
for IBM z/OS Mainframe Security and Audit Management Using the IBM Security zSecure Suite
as created or updated on August 18, 2011.

August 2011, Second Edition

This revision reflects the addition of new information described below.

New information

- ▶ RACF Offline (this capability has been added with IBM Security zSecure Admin V1.9.1)
The zSecure Admin RACF Offline function provides the possibility to issue most RACF commands against an inactive RACF database.
- ▶ Access Monitor (this capability has been added with zSecure Admin V1.11)
Access Monitor provides an automated means of collecting access data for the systematic and programmatic clean up of your RACF database, specifically to remove unused and obsolete entries in the PERMIT list.
- ▶ Multi-system support (this capability has been added with zSecure Admin V1.12)
This support allows direct access to multiple (networked) system security information repositories (RACF databases and CKFREEZE configuration data).
- ▶ New destinations have been added to which IBM Security zSecure Alert can route information.



Part 1

Architecture and design

In this part, we discuss the general business context for and introduce the components of IBM Security zSecure V1.12 to show what it has to offer in the mainframe security and compliance management area of the overall security architecture. After talking about architectures and design, Part 2, “Customer scenario” on page 187 provides a more solution oriented, scenario based approach.



Business context

In this chapter, we discuss enterprise *mainframe security and audit management*. Across the wide variety of companies that exploit mainframe technology, a range of security architectures are used, from little or no added security through to sophisticated architectures designed to allow for growth and extension as business requirements evolve. All of these architectures rely on the underlying security facilities provided by the z/OS mainframe operating system. This function is often performed by an IBM software product called the *Resource Access Control Facility (RACF)*.

In the following sections, we discuss the challenges faced by organizations in managing RACF and z/OS security. We briefly outline how each product of the IBM Security zSecure Suite helps facilitate better security management on the mainframe. We also describe how these tools can help your organization to meet the compliance and regulatory issues that all IT departments are facing.

1.1 Today's challenges

In today's business environment, the challenge of protecting sensitive corporate data can seem both technically overwhelming and prohibitively costly. A balance must be found between unreasonably rigid access controls and too generous access to data by your business users. Additionally, you must monitor and audit your privileged user community, the systems administrators with the highest levels of access to your company data.

Mainframe based systems implement security using an *External Security Manager* (ESM). Examples of ESMs are RACF, CA-ACF2, and CA-Top Secret. We will use the term ESM interchangeably with the term RACF to refer to whatever security mechanism is used on your system. You may have deployed more than one of the available types of ESM across your various systems.

Day to day management of your chosen ESMs can be complex. It requires rare and potentially expensive technical skills to perform this job properly. Recent changes in the regulatory landscape have made it harder to meet audit requirements, and increased the frequency with which your organization faces these audits. The activity to answer the audit requirements usually takes time away from your specialist mainframe security managers and therefore less time is available to actually manage system security. The auditing burden actually increases your costs, without necessarily gaining any real improvement in security.

Managing security using the facilities provided natively with most ESMs is a highly technical and demanding task. The complexity of these security systems and the operating systems they support, combined with the lack of time available to most system administrators, serve to inhibit your security management and any improvements you might like to see made.

What is needed are automated tools that remove the drudgery and provide an intelligent analysis of the real risks present in your system. Given this, both your auditors and security managers are better suited to perform the job we really want them to, that is, improving and monitoring the protection and controls that guard your corporate data.

The IBM Security zSecure Suite delivers an integrated solution that addresses these security challenges facing your organization today.

In the next section, we introduce some risk management, IT governance, and compliance principles and objectives. We then discuss how IBM Security zSecure can help you meet these objectives in the administration and audit of your System z environment.

1.2 Risk management, IT governance, and compliance

Before discussing the capabilities of IBM Security zSecure, we want to introduce the following topics:

- ▶ Risk management
- ▶ IT governance
- ▶ Regulatory compliance

1.2.1 Risk management

Risk management in an IT sense is the act of measuring the business risks introduced by your IT systems, and mitigating these to meet your organization's acceptable business risk. This sounds simple enough, right?

How then do you actually approach this mystical "Risk Management"?

IBM Security zSecure provides the capabilities necessary to administer, audit, and monitor the enterprise security controls configured on your mainframe systems. It does this using sophisticated, built-in program logic that understands your security system and the relationship between your security policies and the parameters and settings required to enforce these policies within the system. Your IT security policies are derived from your risk management analysis.

By removing the necessity for understanding detailed RACF command syntax, the administration of the security system becomes easier for your security administrators. Thus, security administrators can now concentrate their attention on the implementation of IT security policy rather than the detailed commands and configurations necessary to do it.

IBM Security zSecure enables you to assess your risks, plan your security policies and standards, and then implement a security solution that helps you monitor, maintain, and strengthen your enterprise mainframe security.

The capabilities built into IBM Security zSecure remove the burden from your IT staff of writing batch jobs, scripts, health checkers, databases, and other processes just to manage security implementation details. IBM Security zSecure contains a knowledge base of industry leading audit standards, which it uses to automatically analyze your system for vulnerabilities and configuration errors.

Components within the IBM Security zSecure suite provide real-time, periodic, or on demand threat monitoring and audit reporting. These can be integrated with your enterprise data center monitoring solution to become a part of your infrastructure wide IT monitoring solution.

With your IT security administrators now focused on administering security policy in alignment with your acceptable business risk, we shall discuss IT governance and compliance.

1.2.2 IT governance

If you allow only the right people the right access to the right information and never needed to change this, then security would be easy. However, there are a few questions to answer in this context:

- ▶ How do you define the “right people”?
- ▶ How do you define the “right access”?
- ▶ How do you define the “right information”?
- ▶ When do you make changes?
- ▶ How do you ensure the changes are accurate?

The tasks of collecting and analyzing this information, making policy decisions, creating and maintaining security policy, creating business procedures and IT controls to implement the policies, and then testing all of this, are covered in the process of *IT governance*.

Following a process of disciplined IT governance for your security administration is how your company achieves regulatory compliance.

1.2.3 Regulatory compliance

Regulatory compliance is the ability to demonstrate that the IT security controls implied by your security policies are actually in place and effective. Often this is demonstrated using the process of both internal and external audit.

Compliance itself expresses that whatever intentions you declared toward securing your systems are actually in place. In other words, your IT infrastructures actual implementation is in compliance with your declared security policies.

Of course in certain industries, particularly the financial sector, the policies you must adhere to are often set by outside organizations. Sometimes these are purely industry associations, other times this compliance is mandated by law. This is what is called regulatory compliance.

IBM Security zSecure Audit assists your business by automating the audit and reporting processes. Using this automation and eliminating home grown auditing processes and scripts, the measurement of your compliance can be more accurate and costs can be reduced.

IBM Security zSecure knows your ESM and how it works. zSecure Audit uses a built-in knowledge base of security vulnerabilities and an expert system to systematically compare your configuration against the knowledge base. After you have mitigated the findings of this thorough internal audit, you can then use the built-in change tracking facility to ensure that your system does not subsequently deviate from best practice without your knowledge.

In the next section, we discuss how IBM Security zSecure fits into an overall IT Service Management and IT Infrastructure Library strategy.

1.3 Business enablement

Adding to the immediate benefits of IBM Security zSecure and its ability to automate and analyze your mainframe security, IBM Security zSecure is also a business enabler.

It assists you in planning for and delivering an IT Service Management architecture solution that positions your enterprise to meet industry best practices, specifically, a security management architecture aligned with the associated IT Infrastructure Library (ITIL) guidelines.

1.3.1 IT Service Management and IT Infrastructure Library

Establishing best practices for security compliance management enables your business to streamline operations, minimize costs, and proactively stay current with new initiatives. The IT Service Management (ITSM) component of the IT Infrastructure Library (ITIL) can assist you in defining and delivering these benefits.

IBM Security zSecure provides a framework for standardizing your IT security processes for mainframe using its various components:

- ▶ IBM Security zSecure Admin
- ▶ IBM Security zSecure Audit
- ▶ IBM Security zSecure Alert
- ▶ IBM Security zSecure Command Verifier
- ▶ IBM Security zSecure CICS® Toolkit

- ▶ IBM Security zSecure Visual
- ▶ IBM Tivoli zSecure Manager for RACF z/VM®

Note: IBM Tivoli zSecure Manager for RACF z/VM still carries the Tivoli brand in its name because it is the only component that is still in Version 1.11.

These components of IBM Security zSecure allow you to deploy solutions to the highest priority issues first, and then to confidently add components that can integrate with these solutions as your needs evolve.

The level of customization and automation you choose to exploit from the available product features can influence the savings achievable over your existing solutions. Fully integrating all zSecure features into your existing security workflow can provide significant savings in service and maintenance costs, not to mention the reduction in overall risk, knowing that your mainframe security is meeting best practice standards.

1.3.2 System z and IBM Security

IBM Security products address the complete life cycle management of your security requirements. Integrated security features in the z/OS operating system, reliability, availability, scalability, data and communication encryption, comprehensive administration and audit capabilities, and identity and access management across the enterprise are key business differentiators in long term strategies for IT governance and compliance management.

IBM Security zSecure is just one piece of the overall IBM Security architecture that can deliver a complete identity and access management solution for your organization. Establishing System z as the enterprise security hub provides the framework for a unified security, risk, and compliance management strategy that leverages your mainframes inherent security strengths and the IBM Security portfolio.

1.4 Conclusion

The following business goals can be achieved by using IBM Security zSecure to manage mainframe security:

- ▶ Reduce the cost of security management by eliminating repetitive tasks.
- ▶ Protect information from unauthorized individuals, external threats, and mitigate the risks of access by privileged users.

- ▶ Provide automated audit reports to both internal and external auditors.
- ▶ Reduce the introduction of errors and misconfigurations by using automation and command verification.
- ▶ Automate the manual clean-up necessary for ongoing maintenance.

The following security architecture goals can be achieved using IBM Security zSecure to design and manage your security solution:

- ▶ Improve security posture by identifying and correcting known exposures and identified inconsistencies.
- ▶ Authorize users to the minimum level of access required.
- ▶ Authorize privileged users to the minimum security privileges required.
- ▶ Early identification and correction of misconfigurations to minimize duration of exposure to risk.
- ▶ Automatic periodic baseline compliance comparison reporting.
- ▶ Complete security life cycle management process (security policy, change management, and compliance reporting).

Security management of your ESM (be it RACF, CA-ACF2, or CA-Top Secret) is a complex and sensitive endeavor. IBM Security zSecure provides the audit and administrative tools that can help you apply an IT governance model to your mainframe security.

This allows you to manage your IT security processes with reduced cost and to meet ongoing regulatory requirements. Compliance, when well applied, can be less of a cost to the business, and more of an enabler around the acceptance of reasonable business risk. Effective deployment of IBM Security zSecure on your mainframe can help you turn an IT cost into a business profit.



IBM Security zSecure component structure

In this chapter, we introduce all the components that are part of the IBM zSecure Suite. We describe each component at a high level to give you an understanding of their purpose and how they can provide a comprehensive and integrated security management solution for the mainframe.

This chapter covers the following topics:

- ▶ zSecure at a glance
- ▶ Operating systems supported
- ▶ Security systems supported
- ▶ IBM zSecure Admin
- ▶ IBM zSecure Visual
- ▶ IBM zSecure CICS Toolkit
- ▶ IBM zSecure Audit
- ▶ IBM zSecure Alert
- ▶ IBM zSecure Command Verifier

- ▶ “zSecure Manager for RACF z/VM” on page 25
- ▶ “IBM Security zSecure Compliance Insight Manager Enabler for z/OS” on page 26

2.1 zSecure at a glance

The IBM zSecure Suite consists of multiple modular components and tools, which are designed to help you quickly and efficiently manage your RACF database. zSecure can help you monitor for threats, conduct status audits, help manage control self-assessments, and assist with the enforcement of policy compliance.

The use of these tools can help you improve your security posture, comply with industry regulations, identify and aid in the reduction of audit findings, aid in compliance initiatives, and help you improve overall efficiency.

Figure 2-1 shows all of the components available in the IBM zSecure suite and these are categorized into two domains:

- Security audit and compliance
- Administration management

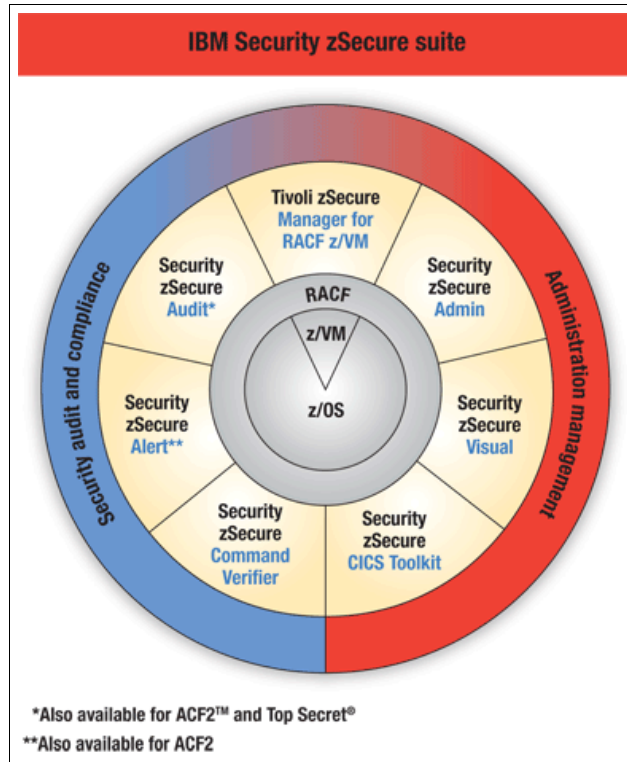


Figure 2-1 The IBM zSecure Suite

2.1.1 Operating systems supported

IBM zSecure supports both z/OS and z/VM operating systems. In the z/VM environment, only the zSecure Manager for RACF z/VM component is supported.

2.1.2 Security systems supported

The External Security Managers (ESMs) supported by IBM zSecure are:

- ▶ IBM RACF
- ▶ CA ACF2 (supported in zSecure Audit and zSecure Alert)
- ▶ CA Top Secret (supported in zSecure Audit)

2.2 zSecure Admin

zSecure Admin provides a comprehensive, easy to use workbench for RACF administration, catering from beginner to advanced security administrators. The product can also be used by local security administrators, auditors, system programmers, and business users who have requirements to produce RACF reports, analyze security settings, and where applicable, execute RACF commands.

Figure 2-2 shows the main ISPF menu for zSecure Admin.

Menu Options Info Commands Setup			
zSecure Admin+Audit for RACF - Main menu			
Option ==> _____			
SE	Setup	Options and input data sets	More: +
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 and class settings	
H	Helpdesk	One-panel helpdesk options	
Q	Quick admin	Quick User Administration	
W	Windows	zSecure Visual 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	DIGTCERT	Work with digital certificates	
C	Custom	Custom report	
AU	Audit	Audit security and system resources	

Figure 2-2 zSecure Admin menu

In the absence of IBM zSecure, a RACF administrator uses native RACF commands, either from a TSO command prompt or from ISPF panels provided with RACF. The organization may also use home grown (in-house written) utilities to perform RACF administration, including a selection of batch JCL containing predefined RACF commands. The administrator may often find these methods difficult, time consuming, inflexible, and hard to maintain.

Another approach is to use an unloaded RACF database, which may be read into IBM DB2® tables (or another database of their choice). The approach then would be to use SQL (or another query language) to generate reports and maybe some commands from the data as it existed at the time of unload. This approach can lead to administrators working with outdated information and hence cause unexpected or inappropriate actions.

RACF administration needs to be a structured and repeatable process, using appropriate tools that are easy to use and flexible to meet the ever changing needs of an organization. zSecure Admin provides an easy to use RACF administration interface that shows profiles in a user friendly display (as shown in Figure 2-3). Profiles can be displayed easily with clear selection and search criteria. zSecure Admin provides administrators with the powerful capability to actually overtype the display fields on the panel to allow them to quickly make changes to the RACF profiles (this generates standard RACF commands). The ISPF interface of zSecure honors your sessions default CUAATTR settings. If your settings underscore input fields, then you will see something similar to Figure 2-3. The underscored fields are where you can overtype and generate a RACF command for verification or modification prior to running the command(s).

zSecure Admin+Audit for RACF USER overview									
Command ==> _____ 0 s elapsed, 0.0 s CPU									
Users like D* with special OR operations OR aud 31 Mar 2008 20:07 Scroll==> <u>PAGE</u>									
User	Complex	Name	DfltGrp	Owner	RIRP	SOA	gC	LCX	Grp
— DATAMOVE	SC76		SYS1	IBMUSER		SO		X	1
— DAUBMAN	SC76	ALEX L	SYS1	HAIMO		SO		X	2
— DAVISRD	SC76	RICARD DAVIS	SYS1	HAIMO		SO			1
— DB	SC76	DB	SYS1	WELLIE3		SO		X	1
— DEBMC	SC76	DEBORAH MCLEMORE	ZSEC001	ZSEC001		A			3
— DETRO	SC76	GARY DETRO	SYS1	HAIMO		SO			1
— DIRMAINT	SC76		SYS1	IBMUSER		SO			1
— DRIEVER	SC76	PATTY DRIEVER	SYS1	HAIMO		SO		X	1
***** Bottom of Data *****									

Figure 2-3 zSecure Admin RACF user overview

zSecure Admin works with the active RACF database so that an administrator can work with the actual profiles in the RACF database. The software allows you to work with an unloaded RACF database if you want. The unloaded RACF database can be used to do historical analysis, such as recovering a deleted profile or analyzing access permissions at a point in time. zSecure Admin also has the ability to work in an offline environment where a RACF database can be built and used to model changes and analyze the outcome of such changes without affecting production RACF databases.

zSecure Admin provides in-depth drill down reporting capability for RACF profiles to enable administrators to get a clear view of security parameters and settings. The product can also help identify misconfigurations and audit concerns.

zSecure Admin is designed for both the beginner and advanced RACF administrator. The software provides the administrator with an excellent view of the RACF database, coupled with powerful automation capabilities to help speed up RACF administration tasks. The product quickly generates the required syntax for RACF commands (based on the input from the window), and then allows the administrator to review the command(s) prior to their actual execution. This enables administrators to research the generated parameters prior to execution, to help assess impact and increase RACF knowledge and skills, while taking the headache out of remembering the various parameters. In all cases, generating RACF commands automatically helps reduce errors that could potentially lead to security exposures or system downtime.

Another powerful function of zSecure Admin is the ability to delegate RACF administration tasks. For example, this function allows you to delegate password resets to an administrator without having to grant group SPECIAL or access to the FACILITY profile IRR.PASSWORD.RESET. zSecure Admin provides a mechanism for enforcing tighter controls over these administrators to ensure they can only manage user IDs within their scope in addition to issuing commands that are appropriate for their job function. This function is described in further detail in 15.1, “Delegated RACF administration” on page 322.

For further information about zSecure Admin, refer to Chapter 3, “IBM Security zSecure Admin” on page 29.

2.3 zSecure Visual

zSecure Visual is a Microsoft Windows based graphical user interface (GUI). This product allows RACF administration tasks to be delegated to junior security administrators. This helps minimize the security exposure of having junior administrators needing higher than necessary privileges in the RACF database. zSecure Visual communicates with a server running under z/OS UNIX to perform the native RACF commands. This insulates the zSecure Visual administrator from the complexities of native RACF and TSO/ISPF.

Figure 2-4 shows the GUI interface for zSecure Visual.

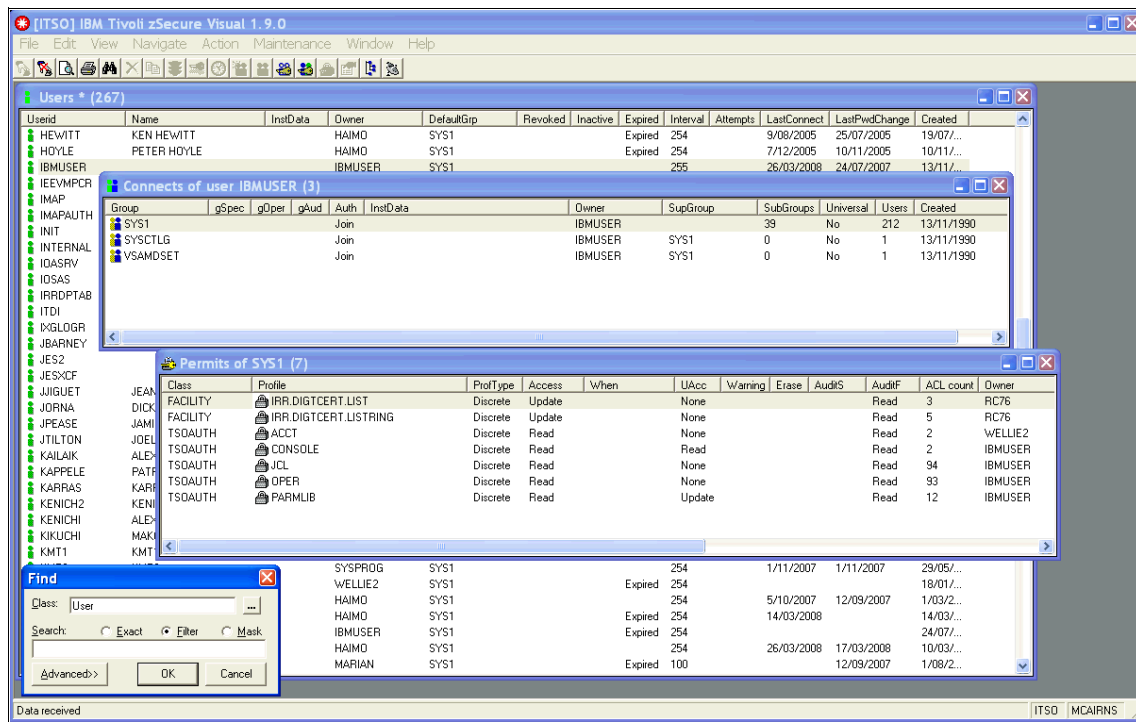


Figure 2-4 zSecure Visual GUI

zSecure Visual is suitable for central security administrators, local security administrators, and help desks. Typically, customers use zSecure Visual as a method of relieving central security administrators from tasks such as password resets and user provisioning. These tasks can be delegated to specialized teams such as the help desk.

Many organizations prefer to bring security administration closer to the business by allowing local security administrators within the business units to perform a subset of tasks, using zSecure Visual. These administrators work close to their users and usually have a good understanding of roles within their department and can grant access that is appropriate for a job function. They can also respond quickly to joiners, movers, and leavers, removing the dependency on a central security team to process such requests.

zSecure Visual is an ideal tool for allowing controlled and fully audited RACF administration to take place, without investing large amounts of money in training employees in mainframe and RACF administration. Thus, user administration can be delegated to non-technical users, freeing up precious time within the central security administration team.

zSecure Visual shows RACF information as directories of information, and the RACF group tree as an actual explorer tree. The administrator can use the mouse to navigate through the windows and click profiles to request actions. The list of actions is controlled by the authority of the administrator in RACF, so zSecure Visual administrators can only see actions that their user ID is permitted to use.

zSecure Visual administrators do not see any RACF commands, so there is no need for them to remember RACF command syntax. If authorized, the administrator can work with user, group, data set, and general resource profiles without needing to know a single RACF command.

For further information about zSecure Visual, refer to Chapter 6, “IBM Security zSecure Visual” on page 111.

2.4 zSecure CICS Toolkit

zSecure CICS Toolkit fulfills two major functions. The first is a command interface, which provides the ability to issue RACF commands from CICS. The second function is an application programming interface (API) that can be used by application designers and programmers to bring external security to CICS and established CICS applications. Thus, access controls and auditing can be managed by RACF rather than relying on embedded security mechanisms within an application.

2.4.1 Command interface

The RACF administration command interface for zSecure CICS Toolkit is commonly used by security administrators who work with CICS based applications and have little or no knowledge of TSO. An advantage to using this product is the high performance and efficiency you get from CICS and this is particularly important for organizations that have a large network of security administrators who need to use a common tool to perform RACF administration tasks.

zSecure CICS Toolkit allows both central and local security administrators to perform RACF administration tasks. This can be carried out through the IBM supplied command interface (as shown in Figure 2-5), or through customized screens.

Termid = OS13	IBM Tivoli zSecure CICS Toolkit	Date = 2008/130
Userid = ALEX01	MAIN MENU	Time = 11:12:30
Name = REDBOOK USER		

PF01 ADGRP	PF02 ADUSER	PF03 ALTGRP	PF04 ALUSER	PF05 CONNCT	PF06 DELDSD
PF07 DELGRP	PF08 DELUSR	PF09 LDSO	PF10 LGRP	PF11 LUSER	PF12 PERMIT
PF13 RALTER	PF14 RACLNK	PF15 RDEFNE	PF16 RDELTE	PF17 REMOVE	PF18 RLIST
PF19 USRDAT					

Number ==> __

Licensed Materials - Property of IBM
5655-T05 Copyright IBM Corp. 1988, 2007. All Rights Reserved.

Use PF key or enter NUMBER for desired command. Press CLEAR to exit

Figure 2-5 zSecure CICS Toolkit main menu

2.4.2 Application programming interface

The application programming interface (API) is a powerful feature that allows you to interface with RACF from a CICS application. You can perform RACF administration activities or resource authorization checks to remove the dependency on internal security. An example of internal security is where a DB2 table or VSAM file is checked to determine whether a user is authorized to perform an action. This is not a best practice and normally requires separate maintenance of security tables and files, which can lead to user IDs having excessive access to applications.

Using an external security manager (ESM), such as RACF, is the preferred method for centralized access control and auditing. A major advantage is that you are placing security administration into the hands of security administrators rather than database administrators or application programmers.

zSecure CICS Toolkit allows you to externalize resource authorization checking so that RACF becomes the security manager, thus making use of the powerful security features already built into z/OS.

Another advantage of the zSecure CICS Toolkit API is the ability to issue RACF commands from CICS. Some organizations use the API as part of their in-house written user provisioning system to provision users to RACF, such as user ID creation, group connects, and password resets.

For further information about zSecure CICS Toolkit, refer to Chapter 9, “IBM Security zSecure CICS Toolkit” on page 161.

2.5 zSecure Audit

zSecure Audit is a comprehensive mainframe compliance and audit solution. It enables you to analyze and report on mainframe events, and automatically detect security exposures and misconfigurations. It does this through extensive status auditing and automated analysis using a built-in knowledge base. zSecure Audit has extensive change tracking facilities, which enables you to establish a security baseline and automatically track changes to it.

Trusted users sometimes keep their RACF authorities and privileges even after a job change, so typically you will see users with unexpected, and non-compliant, authorities in your RACF database. zSecure Audit offers a *Trust analysis* that identifies privileges and resources in z/OS that have to be protected, and assigns them a priority. It then lists the users that have access to these authorities as *Trusted users*.

Figure 2-6 shows a screen capture of some privileges of a user along with an audit concern that documents why this is an issue. These have been identified by zSecure Audit during a status audit of z/OS and RACF. Note that the audit concerns have been prioritized (shown in the “Pri” column) to help you decide the urgency of audit concerns. Prioritization assists in both security planning and auditing, such as deciding where to allocate project resources to close audit concerns or when conducting risk assessments.

```
Trusted userids (may bypass security)                               Line 1 of 3
Command ===> _____ Scroll==> CSR
                                     2 Apr 2008 19:46

Pri Complex Trusted userids
49 SC76                               108
Pri Reasons Userid Name RIP DfltGrp InstData
10      3 ALEX01  ALEX RESIDENT 1      SYS1
Pri Cnt Audit concern
__ 10    1 Systemwide authority to change/define security
__ 9     1 Systemwide authority to process data sets
__ 8     1 Superuser authority, can do anything in USS
***** Bottom of Data *****
```

Figure 2-6 zSecure Audit - Sample list of audit concerns

zSecure Audit is generally used by security personnel and IT auditors (internal and external) to help meet compliance and audit requirements. Security personnel use the software to:

- ▶ Interrogate audit logs, such as SMF, to analyze mainframe events, detect security breaches, perform trend analysis, and fix problems.
- ▶ Conduct regular security reviews (control self assessments) to assess the current state of system security.
- ▶ Set up continuous automated auditing to track changes and highlight exposures.
- ▶ Use validation utilities to help maintain a secure and clean RACF database.

IT auditors often refer to zSecure Audit as a Computer Assisted Audit Technique (CAAT) and may use the software as part of an audit of z/OS or business application(s) to help them assess controls. An IT auditor may follow a set of documented audit procedures during an audit to help them in the assessment process. zSecure Audit can help automate this process through running either predefined or customized reports. The vast majority of documented z/OS audit procedures in use by IT auditors today are already built into predefined reports within zSecure Audit. This also includes the IBM supplied DSMON reports, which are used, among other manual efforts, by auditors in the absence of zSecure Audit.

We often find that system programmers benefit from zSecure Audit. As an example, the system programming team is about to or have been subject to an audit of z/OS. zSecure Audit helps them assess the status of z/OS, such as supervisor calls (SVCs), exits, and program property table (PPT) and authorized program facility (APF) settings to determine where audit concerns exist.

zSecure Audit can help you reduce your reporting and analysis costs through in-depth, ready for use reporting and automation capabilities. zSecure Audit can correlate information from several different sources and systems, helping you consolidate reporting and significantly strengthen controls. Reporting is available in various formats, including extensible markup language (XML).

For further information about zSecure Audit, refer to Chapter 5, “IBM Security zSecure Audit” on page 89.

2.6 zSecure Alert

zSecure Alert is a real-time threat monitoring solution that allows you to monitor for security breaches and changes to your mainframe security configuration in addition to inappropriate settings. zSecure Alert can take automated action upon detecting an event, such as revoking a user with excessive violations, or triggering automation to complete a security request, for example, adding a user to IBM NetView®.

Traditionally, organizations scan audit logs on a daily basis, usually overnight to produce a number of security reports detailing events that the security team may be concerned about. In some cases, the reporting takes place 24 hours after the event occurred, which means potentially a major exposure is resident on the system that could be exploited or even cause severe disruption to the operating system or applications. zSecure Alert helps a security team to quickly respond and verify whether the action that triggered the event was appropriate. For example, zSecure Alert can send an alert when:

- ▶ An administrator has assigned a privileged attribute.
- ▶ Read or update activity occurs for sensitive data.
- ▶ Changes occur to RACF system-wide settings.
- ▶ Loss of SMF data (audit trail) starts.

zSecure Alert comes with a large number of predefined alerts, which are simple to activate. Some of these predefined alerts can be customized so you can add your own installation definitions, such as a list of sensitive RACF groups that you need to monitor for connect activity. You can also add your own alerts, giving you the flexibility to provide real-time alerting for other events you need to monitor. Alerts can be sent as:

- ▶ Email (to one or more mailboxes).
- ▶ Text message to a cell phone.
- ▶ SNMP trap to enable you to consolidate events into your enterprise security monitoring software, such as IBM Tivoli Security Operations Manager.
- ▶ Write to operator (WTO).

A sample of alerts is shown in Figure 2-7.

zSecure Admin+Audit for RACF - Setu Row 1 to 13 of 15					
Command ==> _____ Scroll ==> <u>CSR</u>					
User alerts					
Select the alert you want to work with.					
The following line commands are available: A(Preview), C(opy), D(elete),					
E(dit), I(nsert), W(Who/Where), S(elect), U(nselect), B(rowse)					

	Alert	Id	Sel	gECSW	C
=	Logon by unknown user	1101	Yes	gE	W
—	Logon with emergency userid	1102	Yes	gE	W Y
—	Logon of a userid with UID(0) (Unix superuser)	1103	Yes	gE	W
—	Highly authorized user revoked for pwd violations	1104	Yes	gE	W
—	System authority granted	1105	Yes	gE	W
—	System authority removed	1106	Yes	gE	W
—	Group authority granted	1107	Yes	gE	W
—	Group authority removed	1108	Yes	gE	W
—	SPECIAL authority used by non-SPECIAL user	1109	Yes	gE	W
—	non-OPERATIONS user accessed data set with OPERAT	1110	Yes	gE	W
—	Invalid password attempts exceed limit	1111	Yes	gE	W
—	Password history flushed	1112	Yes	gE	W
—	Suspect password changes	1113	Yes	gE	W

Figure 2-7 zSecure Alert - Sample list of user alerts

zSecure Alert uses the CARLa reporting language to generate alerts. This language is also used to generate reports in zSecure Admin and zSecure Audit, which gives you the advantage of having the same reporting language across the products.

For further information about zSecure Alert, refer to Chapter 4, “IBM Security zSecure Alert” on page 73.

2.7 zSecure Command Verifier

zSecure Command Verifier is an automated policy enforcement solution that adds granular controls for keywords and parameters in RACF commands. It helps enforce mainframe compliance to company and regulatory policies by preventing non-compliant RACF commands, a sample of which is shown in Figure 2-8.

```
READY
CD JPEASE GROUP(SYSPROG)
  You may not connect yourself to group SYSPROG, command terminated
READY
PERMIT 'SYS1.**' GENERIC ID(ALEX01) AC(R)
  Management of locked profiles not allowed, command terminated
READY
ALTDSD 'CKR.**' GENERIC UACC(READ)
  UACC READ setting not allowed, command terminated
READY
```

Figure 2-8 zSecure Command Verifier - Output from non-compliant RACF commands

The product is a *preventative* and *corrective control*. For example, you can automatically prevent security exposures from occurring, such as the specification of high universal access, default passwords, and deactivation of critical security settings. The product can also correct RACF commands as they are being issued. For example, if a security administrator is creating a user ID and incorrect naming standards have been used within the RACF command, zSecure Command Verifier can automatically deny or correct the command to enforce standards and reduce clean-up efforts.

zSecure Command Verifier can also help enforce *segregation of duties*. This may be applicable where you have a large team of security administrators with RACF system SPECIAL or even a large network of local administrators with RACF group SPECIAL. You can establish policies to ensure that administrators can only perform tasks according to their job role. The product can also help prevent system outages due to incorrect settings on RACF profiles or deletion of critical profiles.

Organizations that need to achieve some level of policy enforcement for RACF commands would make use of the RACF command exit, available in the RACF Security Server. However, this may require extensive coding. Organizations may find this method extremely inflexible as security policy changes. The use of zSecure Command Verifier enables you to enforce policy dynamically without any coding. Policies are defined using RACF profiles that are referenced by zSecure Command Verifier. All you require is a RACF administrator to define and manage the policies for you.

After the product is configured and policies have been defined (policies are defined as RACF profiles), zSecure Command Verifier sits in the background. If a security administrator tries to issue a command that does not comply with policy, the command is either automatically corrected or prevented from being executed.

Another powerful feature of zSecure Command Verifier is the RACF command audit trail. This enables security administrators to instantly view a change history for each RACF profile through using normal LIST commands. This can save time by reducing the need to examine historical SMF records for change activity.

In conclusion, zSecure Command Verifier helps to maintain a good security posture, reduce potential audit concerns, and reduce clean-up efforts.

For more information about zSecure Command Verifier, refer to Chapter 7, “IBM Security zSecure Command Verifier” on page 131.

2.8 zSecure Manager for RACF z/VM

zSecure Manager for RACF z/VM adds a user friendly layer onto the mainframe that enables effective and efficient administration, coupled with audit capabilities for the z/VM RACF feature. The product enhances user management and provisioning for the VM environment through automating complex, time consuming z/VM security management tasks with simple, one-step actions that can be performed without detailed knowledge of RACF command syntax. zSecure Manager for RACF z/VM can quickly identify and prevent problems in RACF before they become a threat to security and compliance.

The software also provides audit capabilities by reading the RACF database, analyzing SMF records generated by RACF z/VM, and providing user privileges from both RACF and the VM directory.

zSecure Manager for RACF z/VM allows you to generate and view customized audit reports with flexible schedule and event selections. The software can help you to significantly increase your security posture, thus helping you to pass audits more easily.

A book has been produced to cover security on z/VM in addition to zSecure Manager for RACF z/VM. We recommend you read the IBM Redbooks publication *Security on z/VM*, SG24-7471 should you want to explore this area further.

2.9 IBM Security zSecure Compliance Insight Manager Enabler for z/OS

Using IBM Security zSecure Compliance Insight Manager Enabler for z/OS, mainframe data can be fed into the enterprise security and compliance dashboard of IBM Tivoli Security Information and Event Manager. This product provides automated user activity, monitoring across heterogeneous systems, with dashboard and reporting to help measure your security posture and respond to auditors requests.

A little confusion: The integration component that is needed between zSecure and Tivoli Security Information and Event Manager is still called IBM Tivoli Compliance Insight Manager Enabler for z/OS. Be aware that IBM Tivoli Compliance Insight Manager was renamed to IBM Tivoli Security Information and Event Manager, but the zSecure Enablers still bear the old name. This change, however, does not impact functionality.

The Event Sources provided by the Enabler for z/OS can provide various data for analysis using Tivoli Security Information and Event Manager depending on your licensed z/OS Event Sources:

- ▶ z/OS events and configuration information
- ▶ RACF events and user information including group membership
- ▶ CA-ACF2 events and user information
- ▶ CA-Top Secret (TSS) events
- ▶ z/OS UNIX events
- ▶ DB2 events

For more information about the IBM Security zSecure Compliance Insight Manager Enabler for z/OS, refer to Chapter 8, “IBM z/OS compliance enablers” on page 145.

2.10 Conclusion

IBM zSecure is the next generation of mainframe security. The software allows an organization to take mainframe security to a new level. An organization can significantly strengthen their security posture, using structured and repeatable processes that will help you comply with company and regulatory policies.

Today, we face an increased number of regulations compared with the past. This means that corporate auditors will be actively reviewing and assessing controls for their effectiveness. The technology built into IBM zSecure can help you manage your audit requirements using industry proven, proactive methodology, coupled with powerful efficiency techniques. This can help you to improve accuracy and reduce efforts in security administration so that overall you see a reduction in costs.



IBM Security zSecure Admin

In this chapter, we introduce IBM Security zSecure Admin, which provides a user-friendly ISPF interface to RACF and extends RACF functionality. zSecure Admin allows you to enter and process administrative commands faster and more accurately. It also provides a wide range of custom reports that can be used to clean up RACF databases. Additionally, it provides administration authority in a more granular fashion, so RACF administrators only receive the specific amount of authority they need to perform their job - a basic principle of IT Security. In this fashion, it makes RACF administration easier, faster, and less error-prone. The object-action metaphor employed in the 8000+ ISPF panels makes navigation intuitive. For example, you can copy a selected user ID using a single line command instead of hundreds of individual RACF commands.

IBM Security zSecure Admin can:

- ▶ Automate routine tasks to simplify administration
- ▶ Identify and analyze problems to minimize threats
- ▶ Merge databases quickly and efficiently
- ▶ Display data from active (live) RACF databases, both the local system copy and from zSecure network connected systems
- ▶ Create blocks of commands and easily update large numbers of profiles with similar commands
- ▶ Assist in the clean up of obsolete permissions within the RACF database

- ▶ Integrate smoothly with IBM Security zSecure Audit
- ▶ Store non-RACF data to reduce organizational costs
- ▶ Validate RACF commands against a copy of the RACF database without affecting the production database

To provide you with a good understanding of this product and efficiently manage your mainframe security administration by using it, we show detailed examples of how to use zSecure Admin for daily activities and how it can help you automate recurring tasks, reduce errors, and demonstrate compliance with regulatory or audit requirements.

This chapter covers the following topics:

- ▶ An easy to use RACF administration interface
- ▶ Automating and simplifying routine administration tasks
- ▶ Delegating RACF administration tasks
- ▶ Preventing and identifying problems to minimize threats
- ▶ Other enhancements for RACF administration

3.1 An easy to use RACF administration interface

zSecure Admin makes RACF administration faster and easier. It provides quick access to frequently used functions and extensive reporting capabilities, enriched with context-aware actions that help you to easily organize your RACF database. It provides an easy to use ISPF interface (the primary menu is shown in Figure 3-1). To do this, it uses ISPF dialog environment functions invoking the IBM zSecure application programs as needed.

zSecure Admin+Audit for RACF - 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 and class settings	
H	Helpdesk	One-panel helpdesk options	
Q	Quick admin	Quick User Administration	
W	Windows	zSecure Visual 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	DIGTCERT	Work with digital certificates	
C	Custom	Custom report	

Figure 3-1 Main menu for zSecure Admin

In this section, we discuss how to exploit zSecure Admin easily with examples of using the ISPF interface to perform RACF administration more efficiently. For more detailed information about how to use zSecure Admin, refer to the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12, LC27-2773*.

3.1.1 Initial setup

zSecure Admin can use several different types of data as input files, including the primary/backup RACF database or copies of these, unloaded RACF databases, and a file known as the CKFREEZE file that contains z/OS control block information, DASD contents data, and other system configuration information.

Before you start to use zSecure Admin for the first time, you may need to set up and select a set of input files using the zSecure Admin panel option SE.1, as shown in Figure 3-2.

```

zSecure Admin+Audit for RACF - Setup - I Row 1 from 6
Command ==> _____ Scroll ==> CSR

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

Description                                Complex
- test daily ckfreeze gdg input file        SC76      selected
- Active backup RACF data base and live SMF data sets  SC76      selected
- Active primary RACF data base              SC76
- test daily signature gdg input file        SC76
- test daily unload gdg input file          SC76
- Active backup RACF data base              SC76
***** Bottom of data *****

```

Figure 3-2 Selecting your input file

3.1.2 An easy to use display of RACF profiles

zSecure Admin displays RACF profiles in a user friendly manner, with comprehensive selection and search criteria available to select the profiles you need to see. For example, as shown in Figure 3-3, the option RA.U displays RACF user IDs. By using any of the *additional search criteria*, you can choose from a wide variety of search criteria not available in native RACF.

```

zSecure Admin+Audit for RACF - RACF - User Selection
Command ==> _____ _ start panel

_ Add new user or segment

Show userids that fit all of the following criteria
Userid . . . . . ITSOTS* (user profile key or filter)
Name . . . . . _____ (name/part of name, no filter)
Installation data . _____ (data scan, no filter except *)
Owned by . . . . . _____ (group or userid, or filter)
Default group . . . _____ (group or filter)
Connect group . . . _____ (group or filter)

Additional selection criteria
/_ Other fields    /_ Attributes    _ Segment presence  _ Absence

Output/run options
- Show segments    - All            - Specify scope
- Print format      - Customize title - Send as e-mail
- Background run    Full page form  Sort differently    Narrow print

```

Figure 3-3 zSecure Admin search for user IDs ISPF interface

In this example, we tag the *Other fields* and *Attributes*. After pressing Enter, we see the next two panels to specify more selection criteria, as shown in Figure 3-4 and Figure 3-5. The first panel provides many selection conditions related to dates, such as last logon date, password change date, and so on. In this example, we use Last logon/connect. <= 2007-04-08 as one criteria to find users that last logged on over one year ago.

```

zSecure Admin+Audit for RACF - RACF - User Selection
Command ==> _____
Users like ITSOTS*
Specify additional selection criteria:
Selection by date
Last logon/connect. <= 2007-04-08 (operator: < <= > >= = <> ? )
Last logon/update . _ (date: yyyy-mm-dd, ddMMMyyyy
Password changed . _ NEVER, DUMPDATE, DUMPDATE-nnn,
Pass phrase changed _ DUMPDATE-INACTIVE, TODAY,
Creation date . . . _ TODAY-nnn, TODAY-INACTIVE)
Revoke date . . . _

Logdays selection
_ Sun _ Mon _ Tue _ Wed _ Thu _ Fri _ Sat

Miscellaneous fields
Password interval . _ (operator+number or Y/N)
Schedule name . . _ (schedule name or filter)
Complex . . . . . _ (complex name or filter)

```

Figure 3-4 More criteria for zSecure Admin search - 1

The next panel provides the selection conditions related to system-wide and group authorizations, logon status, user properties, and CKGRACF features. We tag *User audited* as another criteria, and then press Enter.

```

zSecure Admin+Audit for RACF - RACF - User Attributes
Command ==> _____
Users like ITSOTS* with last logon <= 2007-04-08
Specify groups of criteria that the userids must meet:
Systemwide and group authorizations
OR _ Special _ Operations _ Auditor _ Class auth
   _ Group-special _ Group-oper _ Group-audit

Logon status
OR _ Revoked _ Inactive _ Protected _ Passw expired
   _ Revoked group _ Certificate _ Pass phrase _ Phrase expired
   _ When day/time

User properties
OR _ Has RACLINK _ Restricted _ / User audited _ Mixed case pwd

CKGRACF features
OR _ Queued cmds _ Schedules _ Userdata _ MultiAuthority

```

Figure 3-5 More criteria for zSecure Admin search - 2

So you can then see the search result, as shown in Figure 3-6.

The columns in this display show commonly used fields, for example, the name of the users, their associated groups, and a number of flags. A non-blank value is shown for a flag when the flag value is exceptional. For ten of the eleven users displayed, the letter X is shown, indicating that the users' passwords have expired, as you would expect for an ID that has been inactive for more than one year.

If you tag any field of *output/run options* in the previous panel shown in Figure 3-3 on page 32, you can also specify a wide variety of output options, such as email, various panel formats, expanded information, or batch submission of the report.

zSecure Admin+Audit for RACF USER overview										Line 1 of 11
Command ==>										Scroll==> CSR
Users like ITSOTS* with last logon <= 2007-04-0 16 Apr 2008 17:20										
User	Complex	Name	DfltGrp	Owner	RIRP	SOA	gC	LCX	Grp	
ITSOTSA	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTSB	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTSC	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS2	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR					1	
ITSOTS3	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS4	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS5	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS6	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS7	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS8	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS9	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
***** Bottom of Data *****										

Figure 3-6 zSecure Admin user friendly ISPF interface - User overview

zSecure Admin allows security administrators to overtyping fields for quick changes to RACF profiles instead of having to remember and type complex RACF commands.

For example, if a staff has the user ID ITSOTSA in the current system and his job role changes from a trainee to an operator in the security team, then the security administrator needs to change the RACF profiles because different job roles mean different authorities granted.

As shown in Figure 3-7, we overwrite the DfltGrp field for the user ITSOTSA, changing it from @TSTACC to @ZSEC002, and also overwrite the Owner field for the same user, changing it from #TSTUSR to #SECADM.

zSecure Admin+Audit for RACF USER overview										Line 1 of 11
Command ==>										Scroll==> CSR
Users like ITSOTS* with last logon <= 2007-04-0 16 Apr 2008 18:06										
User	Complex	Name	DfltGrp	Owner	RIRP	SOA	gC	LCX	Grp	
ITSOTSA	SC76	TEST FOR ITS0	@ZSEC002	#SECADM				X	1	
ITSOTSB	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTSC	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS2	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR					1	
ITSOTS3	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS4	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS5	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS6	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS7	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS8	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
ITSOTS9	SC76	TEST FOR ITS0	@TSTACC	#TSTUSR				X	1	
***** Bottom of Data *****										

Figure 3-7 An example of making quick changes to RACF profiles - Step 1

Depending on the specific command and your SE.4 default processing options, these changes do not take effect when you first press Enter; you are presented with the subsequent panel shown in Figure 3-8. Here you can see the relevant RACF commands have been generated and command execution options are provided. Because the user ITSOTSA was not initially connected to group @ZSEC002, there are two commands automatically generated: The first to connect the user, the second to change the default group (this is a RACF quirk; you cannot simply change a user ID's default group unless they already have a connection to the intended group). Similar concepts can be found throughout the zSecure Admin ISPF interface, the aim being to save the administrator from having to remember the many RACF quirks such as this, and assisting a new administrator in learning these fundamental RACF processing rules.

zSecure Admin+Audit for RACF - Confirm commands	
Command ==>	
Confirm or edit the following commands	
<u>connect ITSOTSA group(@ZSEC002)</u>	
<u>altuser ITSOTSA dfltgrp(@ZSEC002) owner(#SECADM)</u>	
Command execution . 2	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 commands	

Figure 3-8 An example of making quick changes to RACF - Step 2

The administrator can now choose from various options for execution of the command. The command may be edited directly, executed as is, or CKGRACF may be invoked to provide sophisticated options, such as command scheduling, or workflow to a manager for approval prior to execution. All this can be done, from an intuitive interface, with no requirement for the steep learning curve of RACF command syntax. Thus, one of the main causes of error and resulting security misconfiguration and data exposures is alleviated.

Note: The functions provided by the CKGRACF command are used by zSecure Admin and zSecure Visual. CKGRACF allows you to provide local administrators with specific RACF privileges and can avoid the use of group special. Group special often gives wider authority to local security administrators than they require. Using CKGRACF instead, you can have fine grained distinctions between authority levels in RACF.

3.1.3 Adding a new general resource profile

Using the zSecure Admin ISPF interface, you can easily add new profiles or segments to profiles. For example, if you have zSecure Command Verifier, you may want to add a new general resource C4R.DATASET.ACL.=RACUID.** in class XFACILIT. The intention of this profile is to prevent security administrators from performing a RACF permit against their own user IDs for any resource in the class DATASET. You could use a profile such as C4R.*.ACL.=RACUID.** to cover multiple classes. You can do this using the zSecure Admin ISPF panel RA.R, by selecting / in the Add new general resource profile or segment option. See Figure 3-9.

```

zSecure Admin+Audit for RACF - 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 . . . . XFACILIT (class or filter)
Resource profile . C4R.DATASET.ACL.=RACUID.**

Owned by . . . . . (group or userid, or filter)
Installation data . (substring or *)

Additional selection criteria
_ Profile fields _ Access list _ Segment presence _ Absence

Output/run options
_ Show segments _ All _ Enable full ACL _ Specify scope
_ 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 3-9 Adding a new general resource profile

```

zSecure Admin+Audit for RACF General resource overvie    0 s elapsed, 0.0 s CPU
Command ==> _____ Scroll==> PAGE
Class XFACILIT, like C4R.**                               1 Apr 2008 19:48
  Class Profile key                                         T UACC Owner      S/F W
  _ XFACILIT C4R.CONNECT.ID.*.=RACUID.**                   G NONE SYS1      R _
  _ XFACILIT C4R.DATASET.=NOCHANGE.SYS1.**                 G NONE SYS1      R _
  _ XFACILIT C4R.DATASET.ACL.=RACGPID.**                   G NONE SYS1      R _
  _ XFACILIT C4R.FACILITY.=CMDAUD.=ACL.**                  G NONE SYS1      R _
  _ XFACILIT C4R.FACILITY.=CMDAUD.=ATTR.**                 G NONE SYS1      R _
  _ XFACILIT C4R.FACILITY.=CMDAUD.=MAINT.**                 G NONE SYS1      R _
  _ XFACILIT C4R.FACILITY.=CMDAUD.=SEGMENT.**              G NONE SYS1      R _
  _ XFACILIT C4R.USER.DFLTGRP.**                           G NONE SYS1      R _
***** Bottom of Data *****

```

You can see that a similar profile already exists, C4R.DATASET.ACL.=RACGPID.**, which can be used as a model from which to copy. If you enter a / on the line command for this profile and press Enter, you are presented with a comprehensive list of functions for your selection. Select C to copy the general resource profile, as shown in Figure 3-11. You can select C directly without using the / line command first; all commands presented by the / line command may be selected by their shortcut one or two character prefix.

[illegible]

38 IBM z/OS Mainframe Security and Audit Management Using the IBM Security zSecure Suite

After pressing Enter, you will be presented with the next panel, Figure 3-12. Here you specify the new profile name and class, if different from the original, as well as options for precisely how the copy is to be performed. If you are exploiting CKGRACF features, you can also use option 2 to create a temporary profile and specify a removal date for the profile. Using this option, the profile will be created immediately.

zSecure Admin+Audit for RACF - RACF - Resource Copy

Command ==> _____

From class XFACILIT
profile C4R.DATASET.ACL.=RACGPID.**

To class XFACILIT
profile C4R.DATASET.ACL.=RACUID.**

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)

2. Create temporary profile
Date of removal _____ (ddmmgyyyy or yyyy-mm-dd or +days)
Reason _____

Figure 3-12 Adding a new general resource profile - Step 3

After pressing Enter again, you are returned to the initial profile search results panel where you can now perform additional RACF work. If the profile class you were working with requires a RACLIST REFRESH before your changes become active, you will also see the Refresh registered message in the top right corner. This reminder is shown in Figure 3-13. Once again we see a productivity feature; not only does zSecure Admin remind you to refresh the class, it can perform the refresh for you automatically if your Setup Options are configured to allow it. Again, this removes the opportunity for a common administrative error to occur, that is, forgetting to refresh after profile changes, and the resulting confusion as to why a change has not been effective.

zSecure Admin+Audit for RACF General resource overview					Refresh registered	
Command ==> _____					Scroll==> PAGE	
Class XFACILIT, like C4R.**					1 Apr 2008 19:48	
Class	Profile key	T	UACC	Owner	S/F W	
— XFACILIT	C4R.CONNECT.ID.*.=RACUID.**	G	NONE	SYS1	R —	
— XFACILIT	C4R.DATASET.=NOCHANGE.SYS1.**	G	NONE	SYS1	R —	
— XFACILIT	C4R.DATASET.ACL.=RACGPID.**	G	NONE	SYS1	R —	
— XFACILIT	C4R.FACILITY.=CMDAUD.=ACL.**	G	NONE	SYS1	R —	
— XFACILIT	C4R.FACILITY.=CMDAUD.=ATTR.**	G	NONE	SYS1	R —	
— XFACILIT	C4R.FACILITY.=CMDAUD.=MAINT.**	G	NONE	SYS1	R —	
— XFACILIT	C4R.FACILITY.=CMDAUD.=SEGMENT.**	G	NONE	SYS1	R —	
— XFACILIT	C4R.USER.DFLTGRP.**	G	NONE	SYS1	R —	
***** Bottom of Data *****						

Figure 3-13 Adding a new general resource profile - Prompt information

After pressing F3, you are presented with a listing of RACF commands generated from your current session so far. This may contain many commands, created from various actions on multiple profiles from your current search selection criteria. This is another of the many efficiency improvements available using zSecure Admin, that is, bulk changes to RACF profiles no longer need to be typed into batch jobs or navigated individually through many ISPF panels.

A recommendation is to use the search and filter options to display lists of profiles, rather than just one profile at a time. This way, not only can you make bulk changes more readily, but it is also clearer when one or more profiles in the list have different settings than others, and may be in error. For example, incorrect audit settings, warn mode, or universal access value are all displayed clearly in a column aligned format where any anomalies readily stand out.

The RACF commands from our profile copy process have been generated automatically and are shown in Figure 3-14.

```

EDIT          LILIXIE.C2R1B9A.CKRCMD                      Columns 00001 00072
Command ==> _____ Scroll ==> PAGE
***** ***** 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 CKR1CMD complex SC76 NJE WTSC76 generated  1 Apr
000002          /* CKRCMD file CKR1CMD complex SC76 NJE WTSC76 generated  1 Apr
000003          rdefine XFACILIT C4R.DATASET.ACL.=RACUID.** owner(SYS1) audit(fa
000004          permit C4R.DATASET.ACL.=RACUID.** class(XFACILIT) GENERIC id(SYS
000005          SETROPTS REFRESH RACLIST(XFACILIT) /* POSIT 8 also affects GXFAC
***** ***** Bottom of Data *****

                                     ␣

Press PF3, enter R at the cursor location, press ENTER to run these commands

```

Figure 3-14 Adding a new general resource profile - Step 4

As noted in the prompt information on this panel, you can press F3 to leave the panel, followed by entering R at the default cursor location on the next panel and pressing Enter to run these commands. When you are comfortable using this sequence of panels, it becomes a second nature to follow these steps. If your installation can be done through menu option SE.4 (Setup Confirm), you can edit the commands prior to execution, or include other commands from a data set using standard ISPF copy facilities.

After executing the commands, depending on your SE.4 settings mentioned above, you may be presented with a command output file. This lists the commands and any resulting RACF messages. Usually you would scroll to the bottom of this ISPF browse panel, and check for the message All commands completed successfully. If a command failed, the record number of the command or commands is listed, and you can easily search or scroll to this command to identify the error. Commands may be corrected and re-run by just selecting E for edit against the commands file CKRCMD, correct the commands, press F3, and then enter R again to run the corrected commands.

The commands execution detailed output in the ISPF browse is shown in Figure 3-15.

```

BROWSE      SYS08092.T185438.RA000.LILIXIE.R0128973      Line 00000000 Col 001 080
Command ==> _____ Scroll ==> PAGE
***** Top of Data *****
=====
=== Multiple TSO command output file - scroll max down for overview ===
=== Input data set LILIXIE.C2R1B9A.CKRCMD ===
=====
/* CKRCMD file CKR1CMD complex SC76 NJE WTSC76 generated  1 Apr 2008 20:48 */
/* CKRCMD file CKR1CMD complex SC76 NJE WTSC76 generated  1 Apr 2008 20:56 */

===== 1Apr08 20:57:37.76748 start record 3 =====
rdefine XFACILIT C4R.DATASET.ACL.=RACUID.** owner(SYS1) audit(failures(READ)) ua
RACLISTED PROFILES FOR XFACILIT WILL NOT REFLECT THE ADDITION(S) UNTIL A SETROPT

===== 1Apr08 20:57:37.80024 start record 4 =====
permit C4R.DATASET.ACL.=RACUID.** class(XFACILIT) GENERIC id(SYS1) access(UPDATE
RACLISTED PROFILES FOR XFACILIT WILL NOT REFLECT THE UPDATE(S) UNTIL A SETROPTS

===== 1Apr08 20:57:37.81627 start record 5 =====
SETROPTS REFRESH RACLIST(XFACILIT) /* POSIT 8 also affects GXFACILI FACILITY */
=====
=== All commands completed successfully ===

```

Figure 3-15 Adding a new general resource profile - Result

For detailed information about RACF profiles prefixed with the string C4R. in class XFACILIT, refer to Chapter 7, “IBM Security zSecure Command Verifier” on page 131.

3.2 Automating and simplifying routine administration tasks

zSecure Admin is designed to help manage the security administration of RACF with less system resources and administrator time than conventional RACF administration requires. To provide some of these benefits, zSecure Admin enables you to automate recurring and time consuming security tasks, such as:

- ▶ Adding or deleting user IDs and groups
- ▶ Granting access to users and user groups
- ▶ Setting passwords and resetting revoked user IDs
- ▶ Determining access of user IDs or groups
- ▶ Providing both periodic and one-off reports

In this section, we introduce some practical uses of zSecure Admin and provide examples and suggestions to help you automate routine tasks.

3.2.1 Mass changes to RACF and block command support

The *mass update* feature of zSecure Admin provides an easy way to change, add, or delete multiple profiles from one panel. It can greatly reduce the time spent doing bulk RACF administration. With this feature, zSecure Admin enables the security administrators to perform bulk administration easily on the types of RACF profiles shown in Figure 3-16.

zSecure Admin+Audit for RACF - RACF - Mass update		
Option ==> _____		
0	Copy user	Copy existing user(s) to new user(s)
1	Copy group	Copy existing group(s) to new group(s)
2	Copy dataset	Copy dataset profile(s) to another high level qualifier
3	Copy resource	Copy general resource profile(s) to another class
4	Delete user	Delete user(s)
5	Delete group	Delete group(s)
6	Recreate user	Recreate user(s)
7	Recreate grp	Recreate group(s)
8	Recreate ds	Recreate data set profile(s)
9	Recreate res	Recreate general resource profile(s)
C	Copy CICS	Copy CICS prefixed profile(s) or member(s)

Figure 3-16 Options of the mass update feature

If you need to add a number of new users with similar characteristics copied from an existing user, option RA.4.0 of zSecure Admin is the best approach. Figure 3-17 shows an example.

zSecure Admin+Audit for RACF - RACF - User Multiple copy

Command ==>

Create new user(s) like existing user(s):

Model user	New user	Password	Name	Owner	Dfltgrp	Data
CSV001	csv005	start				
=	csv006	=				
=	csv007	=				
=	csv008	=				
csv002	csv009	=		#B002		
=	csv010	=		=		
=	csv011	=		=		
=	csv012	=		=		

Enter = to copy value from preceding line, leave blank to copy from model.
Press ENTER to specify optional parameters.

Figure 3-17 Add new users using the mass update feature

For adding up to ten new user IDs based on existing models, just enter new user IDs and their models in the panel shown in Figure 3-17. You can enter the model user ID, then an equals sign (=) for all fields that you want copied from the previous line, leave the field blank to copy from the model user, or fill in the fields that you want to contain unique data rather than the model user information. This makes it easy to add a small number of similar user IDs using one or more user IDs as models.

For this example, though, we demonstrate a technique that can easily be used to add dozens of user IDs relatively simply. For this task, we have already prepared some RACF groups using the RA.G options for copying groups.

We need to add four new user IDs in each of five default groups, making a total of 20 new user IDs. We use a combination of mass update functions and the ISPF editor features to make this task simple.

Using mass update, as shown in Figure 3-17 on page 43, we enter the first new user ID, leaving the Name, Owner, Dfltgrp, and Data fields blank so that they can inherit their field information from the model user ID. After pressing Enter, you are presented with additional options for this new user ID, as shown in Figure 3-18.

If the user ID is to be a TSO user, you would likely have data set profiles and catalog aliases automatically. Aliases require a CKFREEZE file to be allocated as one of your current input files through SE.1 and invokes z/OS DFSMS IDCAMS to create the alias. You can also specify extra groups or groups that you do not want to be copied to the target user.

```
zSecure Admin+Audit for RACF - RACF - User Multiple copy
Command ==> _____

Optional parameters
Do not connect new user(s) to following group(s):
_____
_____
Also connect new user(s) to following group(s):
_____
_____

_ Generate RACF commands even when the target user exists
_ Copy USERDATA

Specify options for new userid
_ Revoke new userid
/ Copy catalog aliases (only if CKFREEZE is present)
_ Issue ADDSD/RDEF for dataset and resource profiles related to the user
  _ Copy RACFVARS profiles/members too

Press ENTER to generate TSO and RACF commands.
```

Figure 3-18 Add new users using the mass update feature - Optional parameters

After this panel you can see the commands generated for adding the first user ID, as shown in Figure 3-19. As before, you can again choose to modify these commands should your default settings allow it.

```

EDIT          LILIXIE.C2R1B9A.CKRCMD                      Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
your edit profile using the command RECOVERY ON.
000001      /* CKRCMD file CKR1CMD complex SC76 generated 16 Apr 2008 20:15
000002      /* Commands generated by COPY USER/GROUP */
000003      adduser CSV005  password(start) +
000004          name('CUST SERV REP 01  ') +
000005          data('S/N 000001 - JUNIOR CUST SERV REP') +
000006          owner(#B001) +
000007          dfltgrp(@CSV002)
000008      adduser CSV006  password(start) +
000009          name('CUST SERV REP 01  ') +
000010          data('S/N 000001 - JUNIOR CUST SERV REP') +
000011          owner(#B001) +
000012          dfltgrp(@CSV002)
000013      adduser CSV007  password(start) +

Press PF3, enter R at the cursor location, press ENTER to run these commands

000017          dfltgrp(@CSV002)

```

Figure 3-19 Add new users using the mass update feature - Generated commands

When you press F3 to leave the above panel, you could choose to type R at the cursor location and then press Enter to execute the queued TSO commands. However, for this example, we still have additional user IDs to add, so instead we select the COMMANDS output file zSecure Admin+Audit for RACF input commands from last query and press Enter, as shown in Figure 3-20.

```

                                zSecure Admin+Audit for RACF  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                  W Write file into seq. or partitioned data set
M E-mail report

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+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 3-20 Add new users using the mass update feature - Select commands

You can now see the CARLa script that generated the TSO commands for adding the eight new user IDs, as shown in Figure 3-21.

```

EDIT          CKR.SCKRCARL(@LILIXIE) - 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 suppress addsd
000002 suppress copyusrdata
000003 copy user=CSV001 touser=CSV005 newpassword(start)
000004 copy user=CSV001 touser=CSV006 newpassword(start)
000005 copy user=CSV001 touser=CSV007 newpassword(start)
000006 copy user=CSV001 touser=CSV008 newpassword(start)
000007 copy user=CSV002 touser=CSV009 newpassword(start),
000008     newowner=#B002
000009 copy user=CSV002 touser=CSV010 newpassword(start),
000010     newowner=#B002
000011 copy user=CSV002 touser=CSV011 newpassword(start),
000012     newowner=#B002
000013 copy user=CSV002 touser=CSV012 newpassword(start),
000014     newowner=#B002
000015
***** ***** Bottom of Data *****

```

Figure 3-21 Add new users using the mass update feature - CARLa script

Now we edit this file to add the additional user IDs that we require, as shown in Figure 3-22. We use the ISPF block editor feature to generate another 12 new user IDs, some with different owner attributes.

```

EDIT          CKR.SCKRCARL(@LILIXIE) - 01.00                      Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000021 copy user=CSV003 touser=CSV016 newpassword(start),
000022     newowner=#B003
000023
000024 copy user=CSV004 touser=CSV017 newpassword(start),
000025     newowner=#B004
000026 copy user=CSV004 touser=CSV018 newpassword(start),
000027     newowner=#B004
000028 copy user=CSV004 touser=CSV019 newpassword(start),
000029     newowner=#B004
000030 copy user=CSV004 touser=CSV020 newpassword(start),
000031     newowner=#B004
000032
000033 copy user=CSV002 touser=CSV021 newpassword(start),
000034     newowner=#B005
000035 copy user=CSV002 touser=CSV022 newpassword(start),
000036     newowner=#B005
000037 copy user=CSV002 touser=CSV023 newpassword(start),
000038     newowner=#B005
000039 copy user=CSV002 touser=CSV024 newpassword(start),
000040     newowner=#B005

```

Figure 3-22 Add new users using the mass update feature - Add more user IDs

After editing this file of CARLa commands to your satisfaction, enter G0 in the command line and you will be presented with a similar panel to that shown in Figure 3-19 on page 45. However, you will note that the entire set of TSO commands required to perform the user ID definitions has been generated automatically for each user ID. In this example, we have used the power of the CARLa engine to generate many RACF commands using only a few simple looking CARLa commands that were much easier to block edit than their TSO equivalents. Again, the zSecure Admin tools have accelerated our ability to provide the required work in less time and with greater accuracy.

Now we can press F3 to leave the commands output display, enter R at the cursor location, and then press Enter to execute the queued TSO commands. The results are shown in Figure 3-23.

```

BROWSE      SYS08108.T121556.RA000.LILIXIE.R0103529      Line 00000075 Col 001 080
Command ==> _____ Scroll ==> PAGE
connect CSV020  group(@CSV001) owner(@CSV001)  auth(USE) uacc(NONE)

===== 17Apr08 12:15:57.68299 start record 74 =====
connect CSV020  group(@CSV002) owner(@CSV002)  auth(USE) uacc(NONE)

===== 17Apr08 12:15:57.74240 start record 75 =====
connect CSV021  group(@CSV001) owner(@CSV001)  auth(USE) uacc(NONE)

===== 17Apr08 12:15:57.76526 start record 76 =====
connect CSV022  group(@CSV001) owner(@CSV001)  auth(USE) uacc(NONE)

===== 17Apr08 12:15:57.78839 start record 77 =====
connect CSV023  group(@CSV001) owner(@CSV001)  auth(USE) uacc(NONE)

===== 17Apr08 12:15:57.81075 start record 78 =====
connect CSV024  group(@CSV001) owner(@CSV001)  auth(USE) uacc(NONE)
=====
=== All commands completed successfully ===
=====
***** Bottom of Data *****

```

Figure 3-23 Add new users using the mass update feature - Commands output

Adding the new users, as in Figure 3-23, is just one good use of the mass update feature. After an administrator is comfortable and experienced in the use of the feature, they can use it to do many advanced security administration functions. Tasks that could take between a day and a week depending on your current administration processes (such as cloning a CICS regions definitions) may now be performed in literally minutes.

Another common use of mass update is to copy profiles definitions between different RACF databases. If you maintain multiple RACF databases for your enterprise that must have selected definitions occasionally copied between databases, mass update can reduce the effort required to a trivial task.

Working with multiple RACF databases: If you want to compare, display, or copy profiles from multiple RACF databases, you must have these databases a copy of them, or unload them through SE.1. You may only have one RACF database allocated to any zSecure input set, so to allocate multiple RACF databases for analysis, you must have at least two zSecure input sets selected for your current zSecure session.

New to zSecure V1.12 is the block command option. When reviewing ISPF displays, it is now possible to *edit* the listing using standard ISPF block line format commands. For example, by placing DD on the first and last line of a range of lines, all the entries in that block will have delete commands built for them. In a similar fashion, using RR in a block of entries will issue a RECREATE command for each line.

You can also issue multiple D or R commands and have them executed together.

There is also a FORALL command, which can be used with the Z or ZZ (to select) or the X or XX (to exclude) line commands. FORALL builds commands for all the profiles selected (or not excluded) using a prototype command (with substitution variables). The command is passed as entered, except it is scanned for exclamation marks(!) to replace the substitution variables.

Figure 3-24 shows an example that selects users C##CY30, C##CY31, C##CY32, and C##CY34.

zSecure Audit for RACF USER overview									
Command ==> forall									
Line 325 of 1776									
Scroll==> CSR									
All users									
8 Oct 2010 00:07									
	User	Complex	Name	DfltGrp	Owner	RIRP	SOA	gC	LCX Grp
zz	C##CY30	DINO	TEST USER	C##C	C##C	R			X 2
__	C##CY31	DINO	TEST USER	C##C	C##C	R			X 2
zz	C##CY32	DINO	TEST USER	C##C	C##C	R			X 2
__	C##CY33	DINO	TEST USER	C##C	C##C	R			X 2
z_	C##CY34	DINO	TEST USER	C##C	C##C	R			X 2
__	C##CY35	DINO	TEST USER	C##C	C##C	R			X 2

Figure 3-24 Using ISPF block line commands

Most RACF profile displays support FORALL with the following substitution variables.

- ▶ !KEY (profile key)
- ▶ !CLASS (profile class)

- ▶ !TYPE (profile type)
- ▶ !VOLSER (volume serial for discrete profiles)
- ▶ !GENERIC (GENERIC for fully qualified generic profiles)
- ▶ !KEY_MODIFIERS - (a phrase to make a profile key unique, for example, volser(volume))

Figure 3-25 shows how the FORALL and block selection commands within zSecure Audit and Admin.

```

zSecure Audit for RACF - FORALL Command Shell
Enter FORALL command below:

====>altuser_!key_owner(c##arob)_____
_____
_____

Produces commands like:
000001      /* CKRCMD file CKR1CMD complex DINO generated  8 Oct 2010 21:49
000002      altuser C##CX30 owner(c##arob)
000003      altuser C##CX31 owner(c##arob)
000004      altuser C##CX32 owner(c##arob)
000005      altuser C##CX34 owner(c##arob)

```

Figure 3-25 The FORALL command prototype and its output (in queued commands)

3.2.2 RACF Offline

The RACF Offline capability facilitates administration and auditability of changes by creating a mirrored copy of the RACF database, which allows the RACF administrator or analyst to check and verify their configuration changes offline without affecting the production database. The ability to test changes and review the results before implementing them provides a reduced risk of introducing errors into the production environment.

RACF Offline can be used for a variety of circumstances, such as when combining workloads from newly acquired companies, providing a simulated RACF environment to aid in the testing of changes to the RACF profiles that are used in the z/OS system, without impacting normal production workload or running the risk of unexpected security results. RACF Offline allows the execution of most RACF commands against an inactive RACF environment, and allows easy switching between different environments. The function also provides a replay facility to test the same scenario on multiple RACF environments.

For example, RACF administrators or analysts can use RACF Offline to implement a new structure and show the benefit of using the implementation built through RACF Offline, without impacting any production business processes. Customers can use RACF Offline to test major profile and policy changes or additions, before implementing them into production, providing the benefits of lower risk of implementations for new policies, reduced human error, and reduced chance of possible outages related to implementing security changes.

In the standard RACF environment, all RACF verifications (like logon and access to data sets) are verified using the information in the primary RACF database. All updates are normally performed against both the primary and the secondary (backup) RACF database. Thus, if a RACF administrator issues a command to add a profile, the new profile is added to both databases. Using zSecure Admin RACF Offline, you can direct all RACF commands to a third, alternate RACF database. RACF access verifications are not affected, and are still performed against the System RACF database. Thus, the RACF Offline environment applies only to the RACF commands. (It must be explicitly activated before it is available.) After terminating the RACF Offline environment, the regular System RACF environment is re-established. Currently, you cannot use the RACF Offline environment for logon and resource access verification. These functions are always performed against the System RACF environment.

To illustrate the difference between the handling of commands and verifications, consider the situation when a user issues the ISPF command while in the RACF Offline environment. All standard ISPF functions remain available. Depending on the type of actions being performed while in ISPF, either the System RACF database or the RACF Offline database is used. For example, when the user uses Option-1 to BROWSE a data set, the System RACF database is used to verify READ access to the data set. If the user uses Option-6 or the RACF panels to issue a RACF command, the RACF Offline database is used. The main thing to remember is that only the RACF commands are affected by zSecure Admin RACF Offline.

The zSecure Admin RACF Offline environment, as set up using the B8RACF command, can be compared with an RRSF environment. Whenever you are in B8RACF, it likely and automatically adds the AT keyword to each and every RACF command. It appears that the command is always sent to another system, which uses another RACF database.

The zSecure Admin RACF Offline environment can be entered interactively under TSO, but it is also possible to run the B8RACF command in a batch job. All zSecure Admin RACF Offline functions are available both in TSO and in batch. On one side, we have the regular System RACF environment that is used for logon and access verification. On the other side, we have the RACF Offline environment that is used for all RACF commands while in B8RACF. As soon as the user exits B8RACF, the RACF commands use the System RACF database again. The B8RACF command facilitates quick switching between different RACF databases.

Figure 3-26 illustrates the use of B8RACF to establish a RACF Offline environment. Several commands are issued, and the flow of commands follows the solid arrows. The dotted arrows show the usage of a particular RACF database.

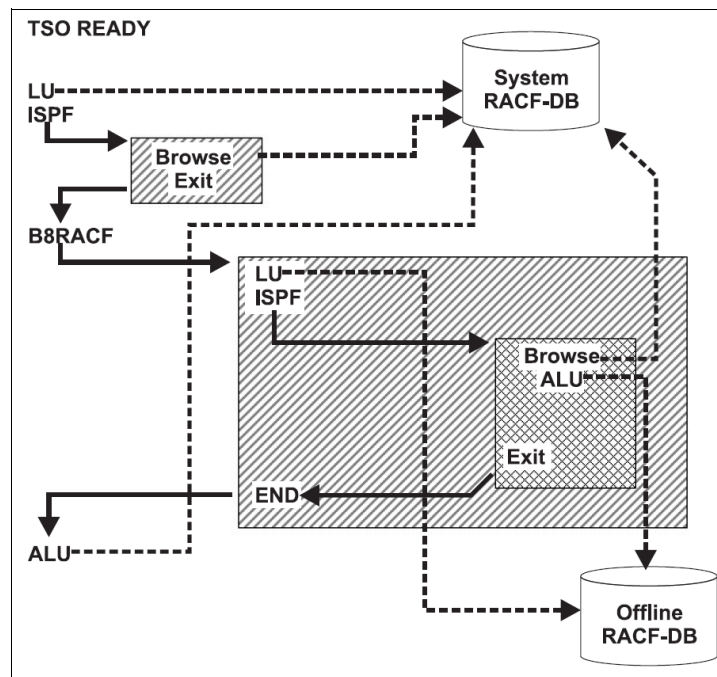


Figure 3-26 Commands and access verification in RACF Offline database

The first LU command is issued in TSO READY mode (line mode) and accesses the System RACF database (dotted arrow). Inside the first ISPF session, the System RACF database is used as well. After issuing the B8RACF command, the LU command accesses the RACF Offline database. The BROWSE function of ISPF still involves the System RACF database, and the ALU uses the RACF Offline database. In summary, access verifications always use the System RACF database, and RACF commands go to the System RACF database when outside B8RACF, and to the RACF Offline database when inside B8RACF.

3.2.3 Timed actions

zSecure Admin allows you to queue RACF commands for later execution, a feature that is useful for granting and removing temporary permissions or managing user ID revocation, for example, during a temporary absence. If, for example, you want to temporarily permit the user SP001 to have UPDATE access on SYS1.PARMLIB from April 18th 2008 until April 19th, zSecure Admin can do this easily. You can see the SYS1.PARMLIB data set profile shown in Figure 3-27.

zSecure Admin+Audit for RACF DATASET Overview						Line 1 of 34
Command ==>						Scroll==> CSR
like SYS1.PARMLIB						17 Apr 2008 12:24
Identification						SC76
Profile name		SYS1.PARMLIB				
Type		GENERIC				
Volume serial list						
-	Effective first qualifier	SYS1				
-	Owner	SYS1				
Installation data						
User	Access	ACL id	When	Name	InstData	
I -group-	READ	@SYSSTC			SYSTEM STA	
Safeguards						Other permissions
Erase on scratch		No		Allow all accesses	WARNING	No
Audit access success/failures		U R		Universal access authority		NONE
Global audit success/failures				Resource level		0
User to notify of violation						
Days protection provided #						
Mandatory Access Control				Statistics		

Figure 3-27 Set up timed action - Insert new permit

In Figure 3-27 on page 52, we entered I in front of one line of the current access list and pressed Enter to insert a new permit. After pressing Enter, we can specify the permit details, shown in Figure 3-28. Enter the user ID SP001 and the ACCESS level as UPDATE for this example.

zSecure Admin+Audit for RACF - RACF - New permit	
Command ==>	_____
Profile to be changed	
Class	DATASET
Profile name	'SYS1.PARMLIB'
Profile modifier . .	generic
Permit to be added	
User or group . . .	<u>SP001</u>
Access level	<u>UPDATE</u>
Optional conditions for the permit	
When class	_____
When resource/profile	_____

Figure 3-28 Set up timed action - Input permit details

After pressing Enter you will see a panel asking for confirmation of the command, as shown in Figure 3-29. To use a timed permit, select command execution option 4 and specify a Start date and Until/for with an optional Reason field that will be logged in the SMF record for this permit. The date fields can be specified in a variety of easily understood formats, such as today or number of days.

zSecure Admin+Audit for RACF - Confirm command	
Command ==> _____	
Confirm or edit the following command	
<u>permit 'SYS1.PARMLIB' generic id(SP001) access(UPDATE)</u>	

Command execution . 4	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	<u>2008-04-18</u> (ddmmmyyyy, yyyy-mm-dd or TODAY)
Until/for	<u>2008-04-19</u> (ddmmmyyyy, yyyy-mm-dd or number of days)
Reason	_____
Press ENTER to continue or END to cancel the command	

Figure 3-29 Set up timed action - Input options

Pressing Enter again will present you with the CKGRACF commands generated, as shown in Figure 3-30.

```
zSecure Admin+Audit for RACF - Press ENTER to process

Command ==> _____

Confirm or edit the following command
ckgracf cmd at 2008-04-18 until 2008-04-19 request permit 'SYS1.PARMLIB' gener
ic id(SP001) access(UPDATE)
_____

Command execution . 4 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 . . . . . 2008-04-18 (ddmmmyyyy, yyyy-mm-dd or TODAY)
Until/for . . . . . 2008-04-19 (ddmmmyyyy, yyyy-mm-dd or number of days)
Reason . . . . .

Press ENTER to continue or END to cancel the command
```

Figure 3-30 Set up timed action - Commands generated

In this example, we use option 4, and queue the command for later execution. Depending on any multiple authority settings, the command may require one or more levels of review prior to actually being executed (refer to 15.3.3, “Workflow for RACF commands” on page 376 for further information about multiple authority settings). You may be asked to confirm a SETROPTS REFRESH after pressing F3 if the class in question requires this task to activate the change.

After the command has been queued, you can see this by displaying the data set whose access list is being modified using the RA.D DATASET display window as shown in Figure 3-31, near the bottom of the panel.

zSecure Admin+Audit for RACF DATASET Overview		Line 12 of 36	
Command ==>		Scroll==> CSR	
like SYS1.PARMLIB		17 Apr 2008 12:48	
Volume			
Safeguards		Other permissions	
Erase on scratch	No	Allow all accesses	WARNING No
Audit access success/failures	U R	Universal access authority	NONE
Global audit success/failures	_____	Resource level	0
User to notify of violation	_____		
Days protection provided #	_____		
Mandatory Access Control		Statistics	
Security label	_____	Creating user's connect group	SYS1
Security level	_____	Creation date	25May94
Categories list	_____		
Timed commands waiting for execution			
Queued command (P): CMD AT 03Apr2008 UNTIL 04Apr2008 REASON('C676464') PERMIT			
Queued command (P): CMD AT 18Apr2008 UNTIL 19Apr2008 PERMIT 'SYS1.PARMLIB'			

Figure 3-31 Set up timed action - Result

Security administrators can manage these queued and, optionally, time limited commands directly from the profile display shown in Figure 3-31 or from the RA.2 menu option. An administrator who does not have the necessary authorities to request these commands may ask for them instead. This places the commands in a queue for approval or other action by another administrator. As long as a queued command has not been acted upon, the original requester can withdraw the command from the queue.

This is a workflow implementation for RACF commands that can be applied to most real-world situations where a help desk may initiate a request on behalf of a user, and a central or intermediate team will take action on them. This can all be achieved without the use of native RACF special or group special authorities and their associated risks. Additional controls over this workflow are discussed in 15.3.3, "Workflow for RACF commands" on page 376.

Using a batch process, commands approved for execution by an administrator with sufficient authorization will execute according to a nightly or more frequent schedule. This is done using the zSecure suite provided batch job C2RJXRFR, which may be scheduled at the desired frequency by your batch scheduling tool, such as IBM TWS for z/OS.

For more detailed information about the job C2RJXRFR, refer to the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12, LC27-2773*.

3.2.4 Single action to perform an access check

A regular request of security administrators is to answer the question *does user xxx have access to resource yyy?* zSecure Admin allows you to perform such an access check in one simple action using either ISPF panel option of RA.1 or the AC command placed against any user ID in a list of users under RA.U. You can quickly test the access of a user or group to a data set or general resource and see the result in a simple display, as shown in Figure 3-32.

```
***** Top of Data *****
ckgracf ACCESS ITSOTS1 DATASET ckr.sckrload
CKG582I 00 ITSOTS1 has NONE access to DATASET CKR.SCKRLOAD
        profile DATASET CKR.**
***** Bottom of Data *****
```

Figure 3-32 Quick access check

3.2.5 Complete access report

There is a much more comprehensive access report for a user ID or group provided by the zSecure Admin option of RA.3.4, or the A command against any user ID listed in RA.U. This function provides several options and filters on the access to be displayed. The main options are:

- ▶ First, direct access for the user ID or group in question, which is mostly used to reveal where a user ID is directly defined in a profile access list, a practice that is not recommended.
- ▶ Second, access received through either a direct permit, or a group that the user ID is a member of. This is the most common case.
- ▶ Third, access received through any means. This is commonly referred to as the users *scope* in RACF. Scope reports imply that the user, taking advantage of any RACF mechanism or privilege such as group special, can manipulate or otherwise access the resources that will be reported in the output.

Additionally, you can select the minimum access level that you are interested in, so that resources to which the user ID has a lower level of access are excluded from the report. You can also specify filters to reduce the amount of output returned in the report. For example, you may only be interested in the user's access to a specific class, or data set profiles matching a specific high level qualifier.

For this example, we choose to report on access through direct permits or connected groups for the user ITSOTS1, and set the minimum access to show as update. See Figure 3-33 for details.

```

zSecure Admin+Audit for RACF - RACF - Report Scope/permit
Command ==> _____

Id . . . . . ITSOTS1 _____
Specify type of authorization
  2 1. Direct permit to the Id (Id on access list)
    2. Direct permit or Connect (Id or Connect Group on access list)
    3. Scope (access or administrative authority by any means)
Report options
  Minimum access to show          Specify output options
  3 1. Execute      2. Read          _ Show datasets covered by profile
    3. Update       4. Control       _ Including datasets on scratch tapes
    5. Alter        6. Admin
    7. Owner        8. Show all      _ Output in print format
                                   _ Start each Id on a new page

Select profiles to include
Dataset HLQ . . . . * _____ (qualifier or filter, * for all, blank for none)
Dataset profile . . _____ (EGN mask)

Other class . . . . * _____ (class or filter, * for all, blank for none)
Other profile . . . _____ (EGN mask)

```

Figure 3-33 Quick access check - Permit/scope report

If no RACF class filters are specified, after pressing Enter, you will see a summary of classes where the user has access, as shown in Figure 3-34.

```

zSecure Admin+Audit for RACF Authorization for ITSOTS 0 s elapsed, 0.0 s CPU
Command ==> _____ Scroll==> PAGE
                                     3 Apr 2008 16:55

Complex Scope of Profiles HighAcc
SC76 ITSOTS1 7 UPDATE
Class Profiles HighAcc
__ FACILITY 2 UPDATE
__ XFACILIT 5 UPDATE
***** Bottom of Data *****

```

Figure 3-34 Quick access check result

In Figure 3-34 on page 58, you can select each class and will be presented with a list of the resources in that class that the user has access to, including the details of how they receive this access in the column named *via*. Via can be the user ID, indicating a PERMIT for the user, or a group, showing that a connect group has a PERMIT. If you selected the Output in print format option in the report specification panel, you will see the print formatted report that contains the same details (Figure 3-35).

***** Top of Data *****					
U S E R A U T H O R I Z A T I O N F O R I D I T S O T S 1 T E S T F O R Z Y M					
Class	Type	Profile name	Volume	Access	Via
FACILITY		IRR.DIGTCERT.LIST		UPDATE	SYS
FACILITY		IRR.DIGTCERT.LISTRING		UPDATE	SYS
XFACILIT	GENERIC	C4R.*.UACC.READ.**		UPDATE	SYS
XFACILIT	GENERIC	C4R.DATASET.=NOCHANGE.SYS1.**		UPDATE	SYS
XFACILIT	GENERIC	C4R.DATASET.ACL.=RACGPID.**		UPDATE	SYS
XFACILIT	GENERIC	C4R.DATASET.ACL.=RACUID.**		UPDATE	SYS
XFACILIT	GENERIC	C4R.USER.DFLTGRP.**		UPDATE	SYS
***** Bottom of Data *****					

Figure 3-35 Quick access check - Print format report

3.2.6 Automated verification and cleanup

Personnel and their roles in an organization are constantly changing. In security management terms this is often referred to as *Joiners, leavers, and movers processing*. Additionally, there are increasingly stringent requirements to be met to both demonstrate compliance with internal or external audit requirements, and ensure that basic security principles are actively applied in IT systems.

To help with this, zSecure Admin provides many reports and functions to assist the RACF security administrator to verify and clean up the RACF database contents. These include reports of undefined, sometimes referred to as *orphan*, permissions to user IDs or groups that have subsequently been deleted from RACF, user IDs that have not been used for long periods of time, and analysis of RACF quirks such as redundant access paths. In most cases these reports generate RACF commands to rectify these situations.

zSecure also helps automate such exercises as migration from the use of Universal Access to the safer ID(*) permit construct.

We recommended that these types of reports are scheduled for execution through batch on a regular basis and reviewed to ensure that none of these issues have re-appeared in your RACF configuration over time. The frequency with which this reporting should be done depends on your unique audit requirements and security posture.

We demonstrate a number of these reports and cleanup capabilities in Chapter 13, “Implementation phase I” on page 201.

3.2.7 Access Monitor for additional cleanup

IBM Security zSecure Admin provides the Access Monitor service. It is a started task (C2PACMON) that tracks the granting of access to RACF protected resources. This tracking information is captured and consolidated into daily, weekly, monthly, or similarly scheduled collections. After enough data has been collected, reports can be generated to see which authorization profiles are good candidates for deletion.

This service can save you time by eliminating the manual process of collecting and reviewing accesses controlled by RACF.

The process of analyzing the usage of RACF database entries has four distinct steps:

1. Which access list entries (permits) were referenced? Used to allow or deny access.
2. Which connect groups were used? Used to allow or deny access.
3. Which profiles were used?
 - Normal profiles.
 - Members of grouping profiles.
 - Global profiles.
4. Generate commands.
 - Remove unused Permits, Connects, UACC, and Profiles.
 - Clean up inconsistencies in the RACF database.

Using this four step service facilitates the clean up of your RACF database in a much quicker and safer fashion. You can combine this approach with the RACF Offline feature discussed in 3.2.2, “RACF Offline” on page 49.

3.2.8 Automated reporting using CARLa

You can use the CARLa language in zSecure Admin to generate almost any report you may need. For example, displaying users with group level authorities from RACF data, reporting access in a matrix display, with access levels of users in columns and 1 row per profile, identifying profiles that do not match your standards, and much more complex reporting if required. Many of these or similar reports are provided just by using the ISPF interface. A feature available throughout the zSecure interface is the ability to capture and modify these supplied reports to suit your own unique requirements.

You can save these reports and schedule them for regular execution and have them automatically emailed, even in XML rather than plain report format, to security administrators, auditors, and managers. For more details about CARLa reporting capabilities and example reports, refer to Appendix B, “An introduction to CARLa” on page 427.

3.2.9 Recovering from administrator errors

There are many other useful functions provided by zSecure Admin, far too numerous to list in this book, that can help you automate routine tasks and simplify administration. A practical example is recovery from accidental deletion of a user ID or user IDs. This is virtually impossible using only native RACF capabilities.

For example, you might have deleted one or more user IDs for some reason, but later you want to recreate them with the exact same characteristics as before, even down to their current password values prior to deletion. It can take a great deal of effort just to recover data set profiles and access list entries for a deleted user. However, even if you do this work, the user will still not be exactly the same as they were before they were deleted. At the very least, they will have to reset their password at next logon, and their initial password value must be communicated to them in some way.

Using zSecure Admin, you can accomplish this task quickly, safely, and easily. Assuming you are backing up your RACF databases daily using the IBM recommended best practice, that is, using the RACF IRRUT200 utility, you have in this backup an accessible copy of the user IDs you have deleted. (Accessible by zSecure Admin at least.)

To fully restore a deleted user ID, select a backup RACF copy created using IRRUT200 prior to the accidental deletion, as the input source in zSecure Admin SE.1 option. Then you can use the RA.U option to list this user information, and use the R command, as shown in Figure 3-36.

```

zSecure Admin+Audit for RACF USER ITSOTS1 overview      0 s elapsed, 0.0 s CPU
Command ==> _____ Scroll==> PAGE
Users like ITSOTS1                                     3 Apr 2008 19:05
  User      Complex Name      DfltGrp Owner   RIRP SOA gC LCX Grp
R ITSOTS1  SC76     TEST FOR ZVM  SYS1    MARIAN      X    1
***** Bottom of Data *****

```

Figure 3-36 Recreate a deleted user

After pressing Enter, you will see the panel shown in Figure 3-37. The R command provides you with three options. If you want the user ID's data set and general resource profiles to be recreated as well, you need to select the first option. If you want the user ID to be reinstated with permits on profile access lists, you need to select the second option. Select the third option if you want to generate CKGRACF commands for restoring additional information, such as USERDATA and CKGRACF data. In this case, we want to fully restore the deleted user ID and associated information, so we select all options.

```

                                zSecure Admin+Audit for RACF - RACF - User Recreate
Command ==> _____

Recreate userid . . ITSOTS1

Specify options for recreate
/ Copy Dataset and General Resource profiles for this user
/ Copy Access List entries with this user
/ Use CKGRACF to update the user profile

```

Figure 3-37 Recreate a deleted user - Specify options

You can see the commands that are generated after you press Enter in Figure 3-38.

```

***** ***** Top of Data *****
000001 /* CKRCMD file CKR1CMD complex SC76 NJE WTSC76 generated 3 Apr 2008 19:
000002 /* CKRCMD file CKR1CMD complex SC76 NJE WTSC76 generated 3 Apr 2008 19:
000003 adduser ITSOTS1 owner(MARIAN) dfltgrp(SYS1)
000004 altuser ITSOTS1 name('TEST FOR ZVM')
000005 password user(ITSOTS1) interval(100)
000006 connect ITSOTS1 group(SYS1) owner(SYS1) auth(USE) uacc(NONE)
000007 ckgracf USER ITSOTS1 pwset nopassword nonexpired; field USER IT
000008 pphrdate('FFFFFF'X) flag7('00'X)
***** ***** Bottom of Data *****

```

Figure 3-38 Recreate a deleted user - Commands for creating this user

After running these commands, the deleted user ID will be recreated exactly as they were prior to deletion, including their current password value, last logon date (and hence expiry or password change options preserved), as well as all access at the time of deletion. So this time-consuming task just takes you a couple of minutes by using zSecure Admin.

The true value of the procedure described above though is in a situation where there was accidental deletion of either a critical user ID or a large number of staff, batch, or other functional type user IDs. These can be restored virtually instantly after the error is discovered using the facilities provided by zSecure Admin. Often functional or server based RACF user IDs are used in connections from systems where either the current password for the user ID is not known, or is coded into either a script or a compiled program somewhere off host. In many cases, it is difficult or impossible to recover or otherwise change the password for the remote connection. These are usually Windows or UNIX based servers connecting to the mainframe. The ability to restore a password to its previous value when re-creating a user ID, as in the situation described here, can prevent significant service outage.

TSO details management

Other useful, time-saving and error preventing functions, such as alias and data set management, are available. You can have a new TSO user ID's alias, default RACF data set profile, and even the ISPF profile data set created automatically when you add the new user ID. In many installations, this is done using a customization of the TSO Logon Procedure. You could simplify the logon procedure or its related REXX exec by removing this customization and using zSecure Admin instead.

Similarly, you can have zSecure Admin automatically delete a TSO user's personal data sets and any data set profiles when the user ID is deleted.

3.3 Delegating RACF administration tasks

zSecure Admin provides some single panel interfaces designed for help desk use. The most common use of these interfaces is to help desk administrators reset user passwords. For this function, some options are available that native RACF cannot provide, such as restoring the previous password of the user ID, or setting the password to a default value known only to the owner of this user ID. This can be an attractive solution if you do not want passwords communicated by phone or clear text email messages.

The options available under one of the help desk interfaces are shown in Figure 3-39.

zSecure Admin+Audit for RACF - RACF - Helpdesk

Option ==> 3

1

List

List RACF profile information

2

Password

Set a new password

3

Default

Set the password to the user's default value

4

Previous

Set the password to the previous value

5

Resume

Resume a userid after too many password attempts

6

Disable

Temporarily disable logon for a userid

7

Enable

Allow user to logon after a Disable

8

Set default

Define a default password for a userid

Userid

ITS0TS1

(type userid and press enter)

New password

(type new password)

Verify password

(type new password again)

Reason

Workflow option

1

1. Request 2. Withdraw 3. Approve 4. Deny

Figure 3-39 Online help desk

Another product in IBM zSecure suite, zSecure Visual, also provides sophisticated help desk functionality and control over delegation of RACF authority. zSecure Visual uses a graphical user interface and runs under most recent versions of Microsoft Windows. Refer to Chapter 6, “IBM Security zSecure Visual” on page 111 for more information about this product.

3.4 Preventing and identifying problems to minimize threats

Errors by security administrators can cause system outages, or worse, leave your critical data exposed to theft or abuse. zSecure Admin helps prevent common errors by security administrators by showing RACF information in easily understood contexts and by generating the RACF commands rather than forcing administrators to learn complex and error prone syntaxes and conventions.

When integrated with zSecure Audit, zSecure Admin can help you quickly identify potential problems in RACF, for example, incorrect audit settings and missing or inconsistent definitions. With this information, you can correct or prevent mistakes before they become a threat to security and compliance.

3.4.1 Using reports provided by zSecure Admin

zSecure Admin provides a number of special reports that benefit from the products capabilities for cross-referencing RACF data. These reports provide both deep and fast results, and can take into account the effects of RACF scoping on the results. Usually reports are provided in a summary format, from which you can select line items for further drill down to view detailed information. We saw some of these capabilities in the example in Figure 3-33 on page 58. This is an especially valuable report because of its ability to take into account RACF scope, and the administrators ability to influence a profile, and thus influence the resulting access.

Reports of comparing users and groups

Another question that occurs regularly in security administration is *user xxx has access, and he sits next to me doing the same role, why don't I have access as well?* zSecure Admin answers this easily with the RA.3.G option - Compare Users. It can be used to compare different user IDs from multiple RACF databases, synchronizing users in preparation for merging RACF databases.

In Figure 3-40, we compare ITSOTS1 and ITSOTSA and select options to compare access through user-specific permits, group permits, and also their connected groups.

zSecure Admin+Audit for RACF - Reports - Compare users	
Command ==>	
Enter up to 4 userids to compare access and/or connects	
Userid	ITSOTS1 ITSOTSA
Select report(s)	
/	Compare access through user-specific permits
/	Include group permits
/	Compare connects

Figure 3-40 Compare users - Specify user IDs or groups

After entering the user IDs, you want to compare them by pressing Enter viewing the summary results in Figure 3-41. Because we selected both permit and connect comparisons above, we now have to drill down into the detail of these two reports.

```

zSecure Admin+Audit for RACF Display Selection          0 s elapsed, 0.0 s CPU
Command ==> _____ Scroll==> PAGE

  Name      Summary Records Title
- PERMIT          5      16 Compare PERMITs for users, including group permits
- CONNECT         3        3 Compare CONNECTs for users
***** Bottom of Data *****

```

Figure 3-41 Compare users summary

After selecting the permit report, we see the comparisons once again at a summary level, by RACF class, as shown in Figure 3-42.

```

Compare PERMITs for users, including group permits          Line 1 of 5
Command ==> _____ Scroll==> PAGE
Enter S in front of a class for more info          4 Apr 2008 15:15
  Class      Profiles ITSOTS1 ITSOTSA
- DATASET          1 NONE    READ
- FACILITY          2 UPDATE  NONE
- OPERCMDS          1 NONE    UPDATE
- TSOAUTH           5 READ    READ
- XFACILIT          6 UPDATE  NONE
***** Bottom of Data *****

```

Figure 3-42 Compare users - Permits for users

The summary here also shows the number of profiles within every class for which a permit for one of the user IDs entered is found, and the highest access within this class for these user IDs. To see the specific permissions of each user, select a class entry and the profiles that any of the user IDs being compared have access to within that class are displayed. As shown in Figure 3-43, you can now easily identify the different access rights of the two user IDs, and determine what, if any, action must be taken to correct any reported security problem.

```

Compare PERMITs for users, including group permits          Line 1 of 5
Command ==> _____ Scroll==> PAGE
Enter S in front of a class for more info          4 Apr 2008 15:27
  Class      Profiles ITSOTS1 ITSOTSA
  TSOAUTH           5 READ    READ
  Profile key
- ACCT              READ    NONE
- CONSOLE            READ    READ
- JCL                 READ    NONE
- OPER               READ    NONE
- PARMLIB            READ    NONE
***** Bottom of Data *****

```

Figure 3-43 Compare users - Profile permits detail

Comparing RACF groups: You can also use option RA.3.G to compare RACF groups; simply specify group names rather than user IDs in the initial entry panel.

3.4.2 Customizing your own report display

zSecure Admin allows you to define a custom report display that can be integrated into the ISPF interface automatically. If you have a specific need to see certain RACF resource data or fields easily and simply on one selectable panel, this tool may help.

Using option RA.C, you can enter up to three CARLa SELECT clauses and three EXCLUDE clauses to determine what data your custom report will show. You can then select which RACF fields and what order they will appear in on the panel using the DISPLAY statement near the bottom of RA.C. A number of supplied default display lists are provided, and you can simply select one of them and customize it to suit your purposes.

You can add or remove fields, and to assist you in determining the field names available, just type the command FIELDS at the top of the panel. This will show the available fields from the RACF templates and the zSecure built-in fields. Figure 3-44 shows a CARLa select clause to display user IDs and a default field list for a class user.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - User defined display				
Command ==> _____				
Enter up to 3 SELECT condition sets:				
Select	<u>class=user</u>			
Select	_____			
Select	_____			
Enter up to 3 EXCLUDE condition sets:				
Exclude	_____			
Exclude	_____			
Exclude	_____			
Enter output variables:				
Display	<u>key(8,key) complex segment pgmrname dfltgrp owner revoke(1) </u> <u>revoke_inactive(1) protected(1) spec(1) oper(1) auditor(1)</u> <u>adsp(1) grpacc(1) any_group sba(1) congrpct clcnt last_connect_date</u> <u>ljdte ltime seclabel passdate passint revokedt resumedt instdata</u> <u>flag4 class connects connects:data / cname(header) / raclink(header)</u>			
Select a set of standard report variables for one of these profile classes:				
_ User	_ Dataset	_ Connect	_ General resource with cond.acc.list	
_ Group	_ Tape		_ General resource with member list	

Figure 3-44 Customize your own report display

For more detailed information about how to customize your own reports with the RA.C option, refer to the *zSecure Admin User Reference Manual*, LC23-6552.

Of course, if this is not enough customization for you, you can also extend zSecure Admin functionality by writing your own CARLa programs and even providing simple panel interfaces for them for a user to input selection or query criteria and output options. Much of the zSecure Admin functionality is actually written using the CARLa language, an architectural choice that ensures the products consistent look and feel together with easy integration with other IBM zSecure products.

For more information about how to use CARLa to create reports, refer to Appendix B, “An introduction to CARLa” on page 427.

3.4.3 Integration with zSecure Audit

When integrated with zSecure Audit, zSecure Admin can help you quickly identify problems with RACF controls or configuration. zSecure Audit can also create reports from SMF records, showing access violations, logon attempts, and so on. If there are audit concerns for profiles that zSecure Audit has identified, these will be listed when you display the profile using zSecure Admin. Likewise, if you run zSecure Audit status auditing reports and find issues that need rectification, you can access zSecure Admin directly from zSecure Audit and correct them.

In Chapter 13, “Implementation phase I” on page 201, we work through a scenario, based on real world experiences, that will demonstrate many of the uses that these products can be put to when integrated.

3.5 Other enhancements for RACF administration

zSecure Admin provides many enhancements for RACF administration other than the user friendly interface to RACF. We introduce here just a few more of the most intriguing of our favorite functions. As they become more familiar with the product, all administrators discover their own favorite product features.

3.5.1 Storing user data and installation data in RACF

zSecure Admin allows you to store non-security related data in the RACF database, such as telephone numbers, accounting codes, and email addresses. This feature is provided for both the RACF Installation Data field and the UsrData field. For installation data, zSecure Admin can enforce structured data fields when the installation data is changed or added. For UsrData, it allows you to view and update information and provides access controls over who can manipulate the data. Figure 3-45 shows some structured UsrData entries.

```
zSecure Admin+Audit for RACF USER ITSOTS1 overview                               Line 32 of 58
Command ==> _____ Scroll==> CSR
Users like ITSOTS1                                                                4 Apr 2008 18:56

Mandatory Access Control      Privileges
Security label                _____ Security admin          SPECIAL No
Security level                _____ DASD administrator OPERATIONS No
Categories list               _____ Global audit set/list AUDITOR No
                                _____ Class authority
Safeguards
Ignore UACC/Glob/* RESTRICTED No
Log all user actions  UAUDIT No

Digital certificate labels      Digital certificate names
_____
Certificate filter label
_____

UsrNm   Flg UsrData
_ EMAIL   00 ITSOTS1@SOMEWHERE.IBM.COM
_ PHONE   00 001-512-123-4567
```

Figure 3-45 User base segment with UsrData field

Here you can see there are two entries, EMAIL and PHONE, in the UsrData field of the user ITSOTS1. These are managed using the MU line command from an RA.U display of user IDs. You can List, Add, Set value/flag, and Delete the Entry name.

You can also include information from external files in your RACF profile displays and reports. For example, you may want to match human resource information with user profiles.

3.5.2 Access list display modes

When native RACF is used to display an access list, only the user ID or group and its corresponding access level are displayed. If a user ID is a member of several RACF groups, and more than one of these groups appears in an access list, it complicates determining the actual access level that user ID will receive.

Of course, zSecure Admin has a solution for this and similar access list resolution issues. It allows you to display access lists in any of four special ways that make it easy to determine a user's real access. The four access list display modes are Exploded, Resolved, Effective, and Trust. In addition, the access lists can be further customized with the (NO)SCOPE and (NO)UNIVERSAL toggles.

The ACL EXPLODE command shows an exploded list of those with access to the profile, which may contain more than one line per user ID should the user ID have multiple access paths to the profile.

The detailed display indicates which access list entries provide what level of access for the user IDs. So if several of a user's connect groups are on the ACL, each is displayed with all entries expanded, but no logic is performed to determine the effective access for an individual user ID.

The ACL RESOLVE command shows you one entry per user ID indicating exactly what access each user has. However, access obtained by system wide and group OPERATIONS attributes are not shown.

The ACL EFFECTIVE command shows you only one entry per user indicating exactly what access each user has but can also include users who have access because they possess the OPERATIONS attribute. You have to turn on ACL UNIVERSAL first for it to show access achieved through the OPERATIONS attribute.

The ACL TRUST command shows the actual access that a user has to a profile, as well as access received through administrative privileges over profiles.

For more detailed information about access list display modes and how they can be used with (NO)SCOPE and (NO)UNIVERSAL toggles, refer to the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12, LC27-2773*.

3.5.3 RACF database merge processing

In the fast changing business world of today, mergers, acquisitions, and internal re-organizations are a frequent reality. Often these changes in business have an impact on the operations of and requirements made upon the security administrators.

zSecure Admin has capabilities to assist in the regular tasks that happen when these business processes occur, such as merge security rules from different databases, copy or move users, groups, resources, applications or even entire databases between systems, rename user ID within databases, and so on.

There is also the specific CARLa MERGE statement designed to assist in cleanly bringing multiple RACF databases into one. Generally you would use the extensive reporting capabilities of zSecure Admin to compare and correct inconsistencies between two databases, then use the MERGE function to complete the task.

Refer to Chapter 4, “Merge Usage Guide”, of the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773 for detailed usage information.

3.6 Conclusion

By showing RACF information in a user friendly context and by generating RACF commands automatically, zSecure Admin helps make administrators more effective and less prone to making mistakes. It provides many practical RACF administration capabilities, such as fast searches and comparison of profiles from multiple RACF databases, decentralized administration functions, and verification and cleanup tools and reports. It provides highly detailed and correlated reporting, including over RACF scoped administrators, and makes cloning and mass updates simple and reliable processes. zSecure Admin represents a paradigm shift forward in mainframe RACF administration.

In the next chapter, we introduce zSecure Alert, the real-time monitoring solution for RACF and CA ACF2.



IBM Security zSecure Alert

In this chapter, we introduce the features, architectures, and processing of zSecure Alert in detail. To help you better understand the zSecure Alert functions and use, we also provide implementation suggestions and integration guidelines about how to achieve effective security monitoring. We discuss a more detailed configuration scenario in Chapter 14, “Implementation phase II” on page 265.

This chapter covers the following topics:

- ▶ Product positioning and features
- ▶ zSecure Alert architecture and processing
- ▶ Implementation suggestions
- ▶ Integration guidelines

4.1 Product positioning and features

Mainframes often are the core repository for enterprise critical information, so you must protect them against from external threats and internal intruders, and undesirable configuration changes. At the same time, you want to stay ahead of potential compliance issues. Timely alerts are a critical part of monitoring because they help you respond quickly to prevent further damage. If an intrusion attempt is happening, you want to take action right away, and zSecure Alert can make this easier to accomplish.

As part of the IBM Security zSecure Suite, zSecure Alert is built on zSecure Audit but can run independently, providing real-time security monitoring capability on the mainframe, and helping you efficiently monitor for intruders and improper configurations. By combining a threat knowledge base with parameters from your active configuration, it can help identify resources that need protection and isolate relevant attack patterns. With zSecure Alert, you can go beyond conventional intrusion detection and practice intrusion prevention.

zSecure Alert provides you with a broad range of monitoring capabilities, including monitoring sensitive data for misuse on z/OS, RACF, CA ACF2, and z/OS UNIX System Services. It can help you detect multiple types of attacks and configuration threats, for example, unwanted logon and attempts, changes that violate security policy, and suspicious activity on the UNIX subsystem. It can monitor sensitive data sets to help make sure none of your privileged users copy or change critical data.

It can send alerts to enterprise audit and security administrators through different methods, such as email, cell phones, pagers, and Write to Operator (WTO) messages. These alerts are written in the easy-to-use CARLa Auditing and Reporting Language (CARLa) and are defined using a standard set of 50 supplied alerts. It can also be customized to specific application needs. In addition to real-time alerting, zSecure Alert can take automated action upon detecting an event, for example, revoking a user with excessive violations.

zSecure Alert integrates with the complete IBM zSecure Suite of enterprise-wide security administration and auditing solutions, providing a comprehensive, end-to-end workbench for RACF security management. For example, working with zSecure Admin, it can provide immediate and integrated remediation for intrusion attempts or improper configurations. It also provides the capability to integrate with compliance management solutions, such as Tivoli Security Information and Event Manager and other enterprise monitoring solutions.

4.2 zSecure Alert architecture and processing

IBM Security zSecure Alert provides real-time mainframe threat monitoring, allowing security staff to monitor intruders and identify misconfigurations that could hamper compliance efforts. It monitors SMF records and WTO messages as they are issued, and invokes the zSecure Audit engine to send out alerts as emails, text messages, WTOs, messages to the z/OS UNIX SYSLGOG daemon, or SNMP traps in real time. The WTO format can be processed by Automated Operations Control. In this section, we discuss the zSecure Alert data flow and components in detail, and we also introduce the data collection mechanism.

4.2.1 zSecure Alert data flow

zSecure Alert works by monitoring SMF records, WTO messages, and critical system tables and settings. Figure 4-1 shows the zSecure Alert data flow, explaining how zSecure Alert identifies threats and generates alerts.

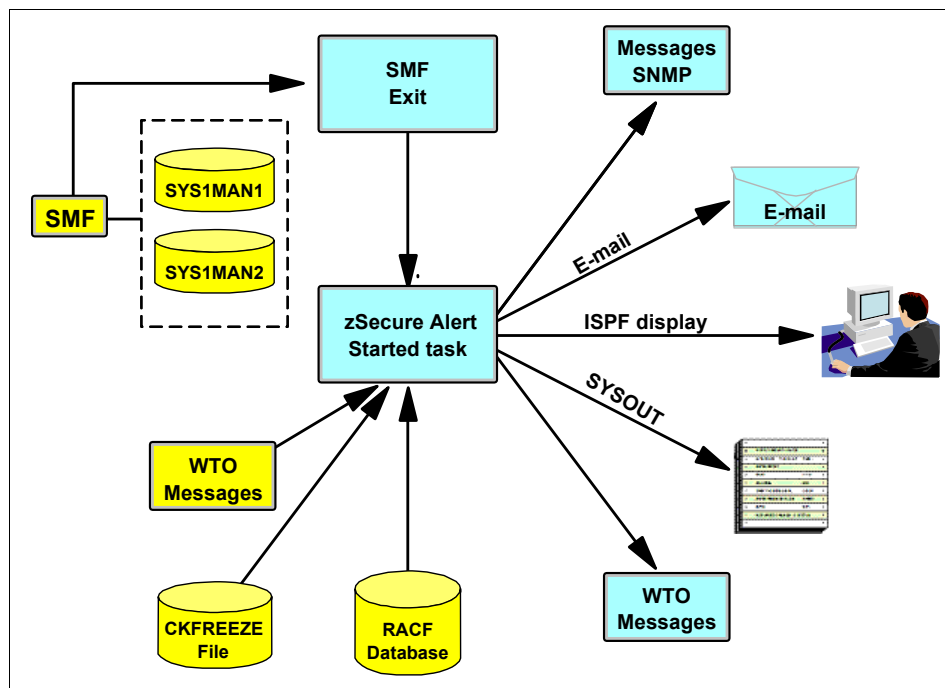


Figure 4-1 zSecure Alert data flow

In Figure 4-1 on page 75, you can see that the input sources for zSecure Alert include SMF data, WTO messages, RACF database(s), and the CKFREEZE file. There is an ISPF application that allows the user to configure the security definitions and z/OS events to be monitored. To monitor defined events, the zSecure Alert started task (C2POLICE) intercepts and filters information from the SMF Exit and the WTO messages appearing on z/OS Console. It sends alerts by WTO, SNMP, z/OS UNIX SYSLGOG daemon, or emails when the defined events are detected. It also writes the alert information into the zSecure Alert address space SYSOUT.

4.2.2 zSecure Alert components

As shown in Figure 4-1 on page 75, zSecure Alert consists of two components:

1. A long-living address space (a started task) that does the actual capturing, correlation, and alert generation.
2. An ISPF interface that allows you to specify which events are to be monitored and in which format(s).

zSecure Alert runs as a started task and collects the information needed to detect alert conditions. It dynamically defines SMF exits, can install an ICHPWX01 RACF exit on RACF systems, establishes itself as an EMCS console, and periodically starts the zSecure Collect program to obtain information about the system environment and stores it in a CKFREEZE file. We discuss this task in more detail in 4.2.3, “Address space and data collection mechanism” on page 77.

The ISPF interface for zSecure Alert configuration is supplied with other zSecure suite products and is available for use after zSecure Alert is installed and configured. It helps you to easily set up both predefined and custom alerts.

Generally, there are three types of information involved with alert configuration in the ISPF interface:

- First, there are general settings that are required for the started task, like the number and size of the data buffers.
- Second, the alert configuration specifies which alert conditions you want to monitor and how the resulting alerts should be delivered. zSecure Alert comes with many predefined Alert conditions, grouped into several categories. You can configure alerts from any or all categories for any Alert set. You can have multiple Alert sets, perhaps used by different LPARs.
- Third, and optionally, if you want to send alerts to groups of email addresses, you can create email destination groups using the ISPF interface.

4.2.3 Address space and data collection mechanism

As a real-time monitor for z/OS systems protected with RACF or CA-ACF2, zSecure Alert runs as a permanently active started task with the default name of C2POLICE. It dynamically defines SMF exits to process selected SMF records before they are written to the SMF log.

Only SMF records specified in the active SMFPRMxx parmlib member can be captured by SMF exits; also, the SMF dynamic exits are only invoked if the exits are enabled in SMFPRMxx. So, before starting the zSecure Alert started task, you should ensure that the SMF records required for your selected alerts are being created and that the necessary SMF exits are enabled in SMFPRMxx. Most predefined alerts require SMF records types 30, 80, 81, and 83 on RACF systems (by default, CA-ACF2 writes SMF 230 records). The SMF exits that should be enabled are IEFU83, IEFU84, and IEFU85. Each exit point is used in a specific environment and for specific SMF records. You should ensure that these three exit points are enabled for the entire system and for all defined subsystems by setting the following SMF options in the SMFOPTxx member(s) or parmlib:

```
SYS(EXIT(IEFU83,IEFU84,IEFU85))
```

The zSecure Alert STC also automatically establishes an EMCS console that it uses to collect WTO messages. It does not modify these messages. The captured WTO messages are optionally filtered. The remaining records and messages are saved in an in-memory buffer allocated in the private area of zSecure Alert STC for further processing. There are two buffers: history buffer and the recent buffer. These may be used to compare current messages to previous messages that by themselves do not appear noteworthy, but which over time may indicate a problem or concern. zSecure Alert can be used as part of your overall system automation capabilities to watch for specific events over time that exceeded a threshold, which may indicate a security attack or other system problem.

The zSecure Alert started task periodically starts the zSecure Collect program to obtain information about the system environment. The CKFREEZE file generated by this collection run is not a full CKFREEZE and thus uses a minimal amount of resources and elapsed time. For analysis and report generation, it invokes the zSecure CARLa engine at each reporting interval, by default every 60 seconds.

4.3 Implementation suggestions

In this section, we provide suggestions for zSecure Alert implementation to help new users of this product achieve best practices rapidly.

4.3.1 Initial setup

Before you start the zSecure Alert started task, you need to do the initial setup of general settings using the zSecure ISPF interface, option SE.A.A. There is a default alert configuration available, named C2PDFL. We recommend you create a new alert definition set by copying this default and modifying the parameters. The general parameters are required for the zSecure Alert started task. See Figure 4-2 for an example.

zSecure Admin+Audit for RACF - Setup - Alert		
Command ==> _____		
Name	<u>SC76AS</u>	(also report member)
Description	<u>ITS0 alert configuration</u>	
SMTP node	<u>WTCS76</u>	
SMTP sysout	<u>B</u>	
SMTP writer	<u>SMTP</u>	
Interval	<u>60</u>	(in seconds)
Environment refresh	<u>60</u>	(in minutes)
RACF database	<u>PRIMARY</u>	(PRIMARY or BACKUP)
Average	<u>300</u>	(in seconds)
Buffer size	<u>1024</u>	(in kilobytes)
Number of buffers	<u>10</u>	
Collect started task	<u>C2PCOLL</u>	
CKFREEZE data set	<u>CKRU.DATA.SSC76.C2POLICE.CKFREEZE</u>	
CKFREEZE Collect time	<u>0100</u>	(Time of day in hhmm)
Enter / to view/edit the global CARLa skeleton		
— Skeleton	<u>C2PSGLOB</u>	

Figure 4-2 zSecure Alert initial setup

After you finish the initial setup, verify and refresh the new configuration. You need to modify the parameter PPARM in the zSecure Alert started task PROC by using the new NAME from the general parameters above, remembering to keep a P as the suffix for this new alert definition.

Another important setup task is the establishment of your base and current zSecure Alert data sets. The C2POLICE started task compares the current and base data sets and can issue an alert when a strategic parameter changes, even without an SMF record being written or a message being sent to the z/OS console. These alerts commonly are security parameters.

4.3.2 Selecting alerts ready for use

All alerts in this product are written in the easy-to-use CARLa Auditing and Reporting Language (CARLa), which has a great deal of flexibility in selecting events and applying thresholds. CARLa also allows customization of the alert by including installation-specific data, such as user data or parts of the installation data from the security data base as well as key-based lookups. There are many predefined alert conditions supplied that you can use as selected and use without modification. These alert conditions are grouped into several alert categories. In the zSecure Alert ISPF panel shown in Figure 4-3, you can see the alert conditions selections.

```
zSecure Admin+Audit for RACF - Setup Row 1 to 8 of 8
Command ==> _____ Scroll ==> CSR

Select the alert category you want to work with
The following line commands are available: W(Who/Where), S(select)
-----
```

	Id	Category	#alerts	#selected
-	1	User alerts	15	15
-	7	Group alerts	1	1
-	2	Data set alerts	6	6
-	3	General resource alerts	3	3
-	4	UNIX alerts	8	8
-	5	RACF control alerts	3	3
-	6	System alerts	2	2
-	0	Other alerts	2	1

```
***** Bottom of data *****
```

Figure 4-3 Alerts selection

The display shows how many alerts are available in each category and how many of these have been selected by the installation. It is *not* suggested that you activate all alerts but only those that are relevant for you. The alert conditions commonly used by zSecure Alert and RACF customers are:

- ▶ Unwanted logons and attempts
 - Logon by unknown users
 - Logon with emergency user ID
 - UNIX superuser logons
- ▶ User behavior that may be a violation of security policy
 - Password recycling
 - User issued UNIX System Services command with uid0
 - Administrator propagates their authority
- ▶ Suspicious activity on the UNIX sub-system
 - File access violations
 - APF or controlled program attributes assigned
 - Global write or read specified

- ▶ Changes that may be a violation of security policy
 - Addition or removal of system authority
 - Revoking of production user IDs
 - Excessive universal access granted
 - Disable system security options (SETROPTS)
- ▶ Audit trail disabled
 - Core system resources at risk
 - Update on a system data set
 - Dynamic addition of APF data set
 - SMF buffers becoming full, risking data loss
 - Started task running with unspecified authority

For the full list and detailed description of all predefined alerts, refer to *IBM Security zSecure Alert User Reference Manual Version 1.12, SC27-2776*.

Some predefined alerts can be customized simply using the ISPF interface, so you can add your installation specific criteria to them. For example, in the *Group alerts* category, you can edit the alert for *Connect to an important group*, which allows you to add important groups defined specifically on your systems.

In Figure 4-4, you can see Y in the C flag bit of the *Connect to an important group* alert, which means it allows customization. You can select or edit by typing S or E in the front of this alert and, in the panel then displayed, you can add the group information for this alert.

```

                                zSecure Admin+Audit for RACF - Setup  Row 1 to 1 of 1
Command ==> _____ Scroll ==> CSR

Group alerts
Select the alert you want to work with.
The following line commands are available: A(Preview), C(opy), D(elete),
E(dit), I(nsert), W(Who/Where), S(elect), U(nselect), B(rowse)
-----
Alert                                Id    Sel  gECSW  C
-   Connect to an important group      1701  Yes  gE  W  Y
***** Bottom of data *****

```

Figure 4-4 Example of alerts that can be customized

As shown in Figure 4-5, you can add SYS1 or other important groups. After you finish the customization for this alert, you need to verify and refresh the new configuration.

zSecure Admin+Audit for RACF - Setup - Alert

Command ==>

Specify important group(s)

Group
SYS1

Figure 4-5 Customize a predefined alert

4.3.3 Sending alerts to their destination

zSecure Alert supports email, text messages to a cell phone, SNMP, and WTO as possible alert destinations. It allows you to customize one or multiple destinations by using the line command W(Who/Where) in the zSecure ISPF interface, option SE.A.A, for all the alerts (known as Global destination setting), one alert category, or even one specific alert condition.

The most common used destination is email. In Figure 4-6, you can see the configuration items for sending an alert to email. In this example, we use an email distribution list named ITSOMAIL.ADDRESS.

zSecure Admin+Audit for RACF - Setup - Alert

Command ==>

You may scroll forward/backward to view all recipient types

More: +

Select the alert destination

/ E-mail
/ Write e-mails to C2RSMTD DD

Specify e-mail recipient(s)

From
&jobname at &system <zsecure@us.ibm.com>

Mail to

ITSOMAIL.ADDRESS
(You may specify : to receive a list of defined recipients :setname.fields)

CC

BCC

Reply to

noreply@us.ibm.com

Output format

1 1. Normal (MIME/HTML)
2. Plain text (formatting may be lost)

Font size

_ (number in range 1-7)

Figure 4-6 Example of a destination - Email

This distributed list can be created or updated through option SE.A.E, as shown in Figure 4-7.

```

zSecure Admin+Audit for RACF - Setup - A Row 4 from 7
Command ==> _____ Scroll ==> CSR

Select zSecure Alert e-mail destination
The following line commands are available: B(rowse), C(opy), D(elete),
E(dit set), I(nsert), S(elect), V(iew)
-----
Name      Description
Data set name
-  ITSOMAIL  ITSOMail Distribution List
      'CKRU.EMAIL'
-----
***** Bottom of data *****

```

Figure 4-7 Set up the email distribution list for alerts

After you finish the initial setup, alerts selection, and destination customization, it is time to see if any event meets one of your conditions and has generated an alert. Here is an example of monitoring authorized access in real time by using zSecure Alert to send alerts to the security administrator's email box, as shown in Figure 4-8.

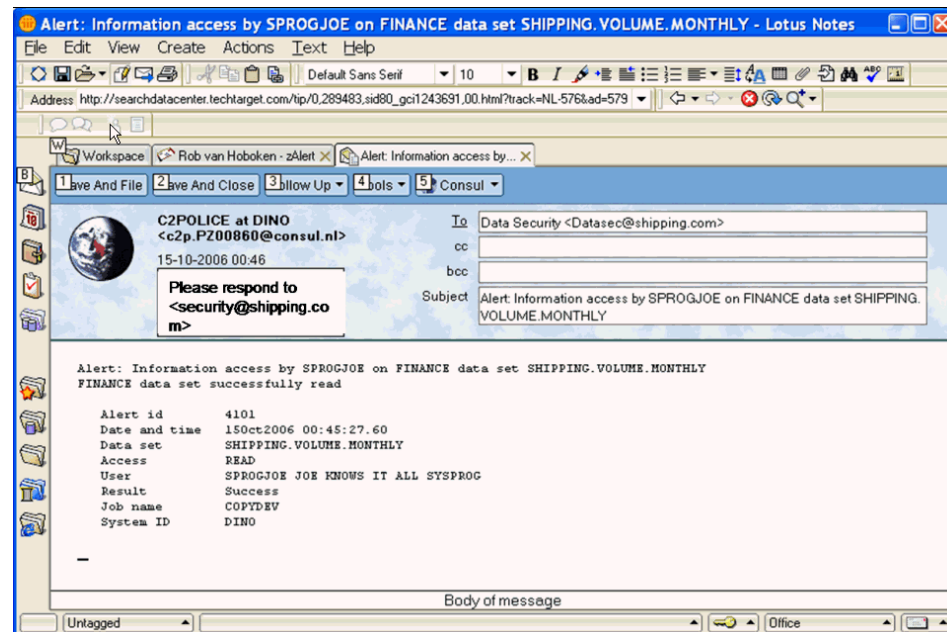


Figure 4-8 Example of alerts by email

4.3.4 Adding your own alerts

You can select to use or customize predefined alerts or create your own alerts using CARLa. Any user created alerts may also be used to generate reports using zSecure Audit. New alerts are more easily added if you copy an existing similar alert, then modify the CARLa skeleton rather than starting from an empty skeleton member. Figure 4-9 shows an example of adding an alert condition so that anybody updating a CKR prefixed data set will generate an alert. We add this new alert by copying the “Update on APF dataset” alert in the Data set alerts category.

```
zSecure Admin+Audit for RACF - Setup - Alert
Command ==> _____
Description . . Update on CKR* datasets
Member prefix IN
Alert id . . . 4001 Severity . . . I (I, W, E or S)
Data source . . SMF
Parameters . . . _____

Specify SMF records to be collected for this alert
Type Sub   Type Sub   Type Sub   Type Sub   Type Sub
80 _____

Specify WTO filters for this alert
Prefix     Prefix     Prefix     Prefix     Prefix
_____

Select allowable destination types
/ E-mail   _ Cellphone   _ SNMP       / WTO

Enter / to view/edit the skeleton for this alert
_ ISPF Skeleton INS4001
```

Figure 4-9 Example of adding one new alert

In Figure 4-9, we need to specify at least an alert ID and data source. Every alert has a unique ID; the ID range from 4000 through 4999 is reserved for user written custom alerts. In this example, we specify the ID as 4001.

To identify if anybody has updated a CKR prefixed data set, we need to specify the data source as SMF, record type 80. We also need to select allowable destinations for this new alert. Then we can enter / in the ISPF Skeleton line to edit the CARLa skeleton for this alert. You can see the skeleton panel shown in Figure 4-10.

```

EDIT          CKRU.DATA.SSC76.C2POLICE.C2PCUST(INS4001) - 01. Columns 00001 00072
Command ==> _____ Scroll ==> PAGE
000018 )CM * 030707 1.6.0 SDG QZ0306009: Changed alert titles
000019 )CM * 050606 1.7.0 MR EZ0504008: Change newlist name
000020 )CM * 061219 1.8.0 MR EZ0603014: zAlert ACF2
000021 )CM *****EndModule*****/
000022 )CM Pass one query
000023 )SEL &C2PEPASS = Y
000024 )ENDSEL
000025 )CM Alert condition
000026 )SEL &C2PEPASS = N
000027 )IM C2PSGNEW
000028     select likelist=recent,
000029         ((dataset=(CKR*.**) or,
000030             (class=DATASET resource=(CKR*.**)),
000031         )
000032     ),
000033     intent>=UPDATE
000034 )CM EMAIL sortlist
000035 )SEL &C2PERCTP = MAIL
000036 sortlist,
000037     recno(nd),
000038     'Alert: Update by' (t) user(t) 'on data set' (t) dataset(t),
000039     'Alert: Update by' user(0) 'on data set' dataset(0) /,
```

Figure 4-10 Skeleton panel

The CARLa sections within the skeleton are each marked with an identifying comment line, such as:

```

)CM Pass one query
)CM Alert condition
)CM EMAIL sortlist
```

We must at least update the script in the)CM Alert condition section when creating a new alert. As shown in Example 4-1, the scripts in this section tells zSecure Alert to generate an alert when CKR prefixed data sets are updated.

Example 4-1 Alert condition

```

)CM Alert condition
)SEL &C2PEPASS = N
)IM C2PSGNEW
    select likelist=recent,
        ( (dataset=(CKR*.**) or,
            (class=DATASET resource=(CKR*.**)),
```

```
)  
,  
intent>=UPDATE
```

In this example, we set the destination of the alert to email and WTO. We can delete the)CM Cellphone sortlist and the)CM SNMP sortlist sections, which are for cellphone and SNMP message layouts respectively. We need to update the)CM EMAIL sortlist section shown in Example 4-2 to specify our alert message in email, as well as the)CM WTO sortlist section to describe the layout of the WTO message on the console.

Example 4-2 Destination sections

```
)CM EMAIL sortlist  
)SEL &C2PERCTP = MAIL  
sortlist,  
recno(nd),  
'Alert: Update by'(t) user(t) 'on data set'(t) dataset(t),  
'Alert: Update by' user(0) 'on data set' dataset(0) /,  
'USER data set successfully updated' /,  
)ENDSEL  
)CM WTO sortlist  
)SEL &C2PERCTP = WTO  
sortlist,  
recno(nd),  
'C2P&c2pemem.&c2peflag',  
'Update by' user(0) 'on data set' /,  
dataset(0)  
)ENDSEL
```

For more information about editing an alert CARLa skeleton, refer to *IBM Security zSecure Alert User Reference Manual Version 1.12, SC27-2776*. For detailed information about how to use CARLa, refer to Appendix B, “An introduction to CARLa” on page 427.

4.4 Integration guidelines

zSecure Alert can be integrated with other tools, enabling you to send relevant alerts to your central security or network management console. For example, you can send alerts as Simple Network Management Protocol (SNMP) traps to IBM Tivoli Security Operations Manager for real-time correlation and threat monitoring. You can also send these alerts to the IBM Tivoli Enterprise Console® or other enterprise monitoring applications for integrated event monitoring.

IBM Tivoli Security Information and Event Manager gathers audit information from across the organization and compares activity to the acceptable use policies defined by both your organization and by your regulators. One of the outstanding capabilities of Security Information and Event Manager is to collect data from distributed systems such as UNIX, Linux, and Windows together with midrange event data and System z. zSecure Alert can send SNMP traps to Security Information and Event Manager through Security Operations Manager.

You can also install the Insight Enabler for z/OS to get full integration between IBM zSecure and Security Information and Event Manager. Using Security Information and Event Manager integrated with z/OS, a business auditor who is not familiar with mainframe technology can easily report on and analyze mainframe access data.

The mainframe events sent to Security Information and Event Manager are converted to an English-based language known as the W7 event model (Who, What, on What, When, Where, Where from, and to Where) for easy, platform independent interpretation. The z/OS event source, together with Security Information and Event Manager, gives business security officers an overall picture of access activity on their mainframe.

For more information about IBM zSecure Suite integration with Security Information and Event Manager, refer to Chapter 8, “IBM z/OS compliance enablers” on page 145.

For more information about IBM zSecure Suite integration with Security Information and Event Manager and Security Operations Manager, refer to *IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager*, SG24-7530.


TEC and Netcool/OMNIbus, as enterprise event consoles, can consolidate events from networks, hardware, and software throughout the environment. They have pre-configured rules that provide an enterprise event management solution. For TEC or Netcool/OMNIbus users, if you have no Security Operations Manager installed, you can also filter and send alerts as a SNMP traps directly to TEC or Netcool/OMNIbus, which can help you achieve integrated event monitoring or possibly go further towards business dashboard monitoring.

Another useful integration practice is to send alerts as a WTO to automation products such as IBM Tivoli System Automation for z/OS (SA for z/OS). Although zSecure Alert can take automated actions upon detection of an event, you can use CARLa to perform automated RACF administration tasks, such as revoking a user with excessive violations. For more complex situations, we recommend integration with SA for z/OS, which provides the capability to associate one alert with other WTOs with the system status SYSPLEX wide and execute predefined automated operations in response.

4.5 Conclusion

As one part of the IBM zSecure Suite, zSecure Alert is the result of decades of experience collected into a threat knowledge base that can quickly alert you to suspicious activities. It offers real time mainframe threat detection and prevention, with alerts and automated commands to counter attacks and misconfigurations. It integrates seamlessly with the complete IBM zSecure Suite of security and audit management solution, providing a comprehensive workbench for enterprise mainframe security management.

In the next chapter, we take a closer look at zSecure Audit.



IBM Security zSecure Audit

In this chapter, we introduce zSecure Audit and discuss its features, services, and the controls available to secure this product from unauthorized use.

zSecure Audit is a compliance and audit solution that enables security and audit personnel to automatically analyze and report on security events and detect security exposures. It is a mainframe auditor in a box, checking the protection of the Trusted Computing Base and generating prioritized audit concerns for any aberrations. It also provides reports about what the system environment's protection settings are.

This chapter covers the following topics:

- ▶ zSecure Audit architecture
- ▶ Initial setup of zSecure Audit
- ▶ System environment reporting
- ▶ Database verification and cleanup
- ▶ Event reporting
- ▶ Change tracking
- ▶ Library and sequential data set audit

5.1 zSecure Audit architecture

As an automated security vulnerability analyzer for z/OS, zSecure Audit provides the most comprehensive analysis available of a z/OS security posture by correlating data from various input sources.

zSecure Audit can correlate data from:

- ▶ Your ESM security database(s)
- ▶ The z/OS IPL parameters and other configuration information from multiple systems
- ▶ The System Management Facility (SMF) audit trail data from multiple systems
- ▶ Other sources (HTTP logs and flat files)

Information gathered about your system by zSecure Audit is stored in a database structure that can be analyzed in a variety of ways using the supplied reports. You can easily add customized reports using the CARLa Audit and Reporting Language (CARLa).

zSecure Audit works with the user in either of two modes: interactively using an ISPF based set of panels (available in both MVS/TSO and VM/CMS for zSecure Manager for RACF z/VM) and in batch mode. In Figure 5-1, we show the data flow architecture of zSecure Audit.

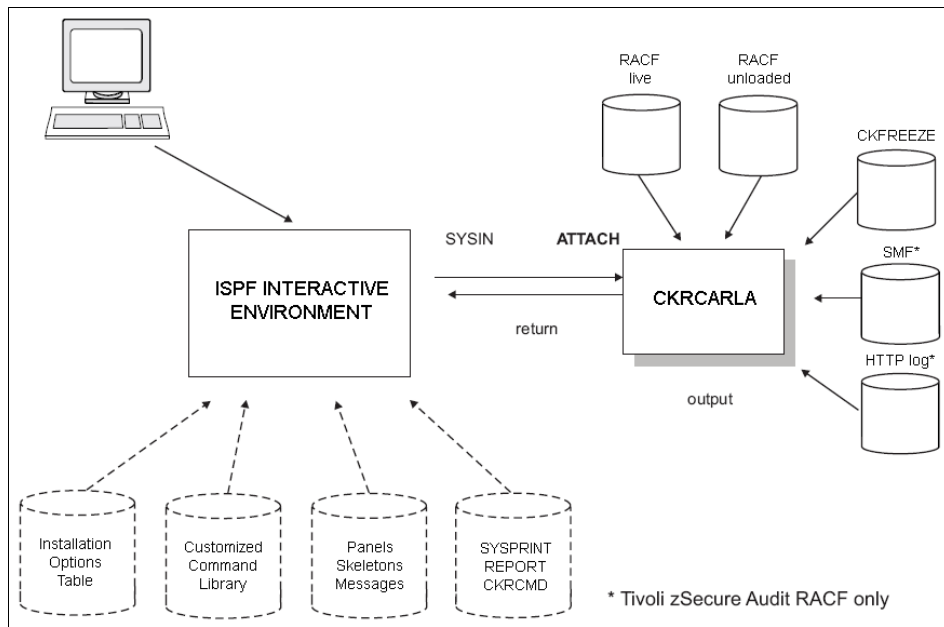


Figure 5-1 zSecure Audit data flow

Further details of this processing can be found in the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773.

It should be noted that the zSecure Compliance Insight Manager Enablers for z/OS use zSecure Audit services. This topic is covered in Chapter 8, “IBM z/OS compliance enablers” on page 145.

Also note that while zSecure Audit can report on ACF2 databases and events and the CA-TSS Audit/Tracking file, these features are beyond the scope of this book.

5.2 Initial setup of zSecure Audit

Figure 5-2 shows the zSecure main menu after the AU option has been entered, zSecure Audit features are available using the RA (RACF profile reports), EV (events), and AU (audit) options of the zSecure main ISPF menu.

zSecure Admin+Audit for RACF - Main menu		
Option ==>		
SE	Setup	Options and input data sets
RA	RACF	RACF Administration
AU	Audit	Audit security and system resources
C	Change track	Track changes to the system
L	Libraries	Library status and update analysis
S	Status	Status auditing of security and system tables/options
V	Verify	Verify and cleanup security database
EV	Events	Event reporting from SMF and other logs
CO	Commands	Run commands from library
IN	Information	Information and documentation
LO	Local	Locally defined options
X	Exit	Exit this panel

Figure 5-2 Invoking zSecure Audit from main menu

Using option SE.1, we first select the input data that is used for this zSecure Audit session. The following section discusses the input files available for processing by zSecure Audit.

5.2.1 Input data used by zSecure Audit

Before the user can report on the desired events or environments, the input sources must be selected. Depending on which input files are selected, different reports are available within zSecure Audit.

You can see a list of the supported input sources by entering a question mark (?) in the TYPE field when defining an input source. The available input types are limited by the products you have licensed for your system. With only RACF licensed, you will see a list similar to that shown in Figure 5-3.

Access required: The user running the request must have *read* access to any files or data sets requested for input.

```

zSecure Admin+Audit for RACF - Setu Row 1 to 14 of 14
Command ==> _____ Scroll ==> CSR

Select the type of data set or file

      Type           Description
-   ACT.BACK        The backup RACF database of your active system
-   ACT.PRIM        The primary RACF database of your active system
-   ACT.SMF         The live SMF data set(s)
-   ACT.SYSTEM      Live settings
-   CKFREEZE        System resource information data set
-   CKRCMD          A file for generated RACF commands
-   COPY.RACF       A copy of a single data set RACF database
-   COPY.SEC       A non-first component of a multiple data set RACF database
-   COPY.TEMP       The first component of a multiple data set RACF database
-   SMF             VSAM or dumped SMF
-   SMF.LOGSTR      SMF logstream
-   UNLOAD          An unloaded RACF database
-   WEBACCESS       IBM HTTP Server access log
-   WEBERROR        IBM HTTP Server error log
***** Bottom of data *****

```

Figure 5-3 Input file specification

As zSecure Admin works with RACF only, the widest variety of input data types supported are for RACF:

- ▶ Active RACF primary database (type: ACT.PRIM)
- ▶ Active RACF backup database (type: ACT.BACK)
- ▶ Copy of a single data set RACF database (type: COPY.RACF)
- ▶ Copy of the first data set of a RACF database (type: COPY.TEMP)
- ▶ Copy of subsequent data sets of a RACF database (type: COPY.SEC)
- ▶ RACF database unloaded by zSecure (type: UNLOAD)

There is an important distinction between the RACF database input sources prefixed with COPY and those known as UNLOAD RACF input sources. The COPY prefixed versions refer to databases created using the IBM IRRUT200 RACF copy utility or with data mover tools such as DFHSM. The UNLOAD data set is in a proprietary format and created by the zSecure Audit UNLOAD command.

A primary difference between these two kinds of RACF copies is that the UNLOAD format will never contain RACF passwords, while the standard IBM utilities for copying a RACF database produce exact copies. These exact copies are able to be used as a live RACF database and therefore contain all sensitive fields, such as the user authentication values in encrypted format.

Certain operations of zSecure Admin and Audit may only function with an appropriate type of input data. For example, when using the zSecure Admin Recreate function to generate user IDs with their passwords intact, an exact COPY of the RACF database will be required; alternatively, the group tree report RA.3.8 must be used with an UNLOAD format of the database if you want to specify the *start at* option. Refer to the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773 for complete details about restrictions such as these.

zSecure Audit uses the multiple system access capability, where you can define a zSecure network and transfer data between the connected systems for reporting on a single image to consolidate all the data.

zSecure Audit reports on security events logged to SMF. The SMF data can be read from the running system (type: ACT.SMF), previously dumped to disk or tape (type: SMF) or SMF logstream (type: SMF.LOGSTR). All security related SMF data for all supported ESMs is available for analysis by zSecure Audit. Over 60 types of SMF record are supported to enhance the analysis capability. For example, full support for SMF type 14 and 15 records (data set open and close) is provided. You can use data from the type 14 and 15 records to infer missing type 80 data. This is a unique part of the zSecure Audit capabilities for advanced security analysis. You can request archived SMF records or live SMF data from the running system or from other systems within the zSecure network.

zSecure Audit provides extensive analysis of the z/OS operating system configuration parameters that relate to security. While many of these analyses may be run using settings derived from the live system (type: ACT.SYSTEM), for the most thorough analysis, we recommend you generate the zSecure Audit snapshot database, known as a CKFREEZE file. You may also choose to request the current system configuration information from other systems within the zSecure network.

The CKFREEZE input files are a point-in-time snapshot of your system settings and additional security relevant data. It is generated using the batch program CKFCOLL or requested through the zSecure network connection (CKNSERVE). You can see the JCL to generate this file by simply entering the REFRESH primary command after having selected a CKFREEZE file as your current input source using the setup menu (SE.1). The REFRESH command will generate a batch job that uses the CKFCOLL program to populate the CKFREEZE file.

Additional input sources include IBM HTTP Server logs, the server access log (type: WEBACCESS), and the server error log (type: WEBERROR). Generally these will be HFS files. If you have the appropriate licenses, you can also analyze data from ACF2 (types: ACF2INFO, ACF2LID, ACF2RULE, ACT2.BACK, and INACT2.BACK) and the CA-TSS Audit/Tracking file (type: ATF).

ACF2 support: An unload format database is also supported for ACF2 (type: UNLOAD).

A special type of file allocation is available for CKRCMD. This file is for output rather than an input source; it will contain RACF commands generated during the zSecure session. You can also allocate your own input file types for processing external data.

There is no need to specify a data set name when using any of the input file types prefixed with ACT, as the data for these will be drawn from the active system rather than any predefined file. You can see an example of the dynamically generated names that zSecure Audit will assign when using the active system as its input data in Figure 5-4. All other input sources require a data set to be specified that contains the expected data. An error will occur should unexpected data be found in an input source, for example, you select a type of UNLOAD but allocate its input data set to a file containing SMF data.

```

zSecure Admin+Audit for RACF - Setup - I Row 2 from 3
Command ==> _____ Scroll ==> CSR

Description . . . . Active backup RACF data base and live SMF data sets
Complex . . . . . SC76
RRSF node . . . . . Local node for RRSF

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: B                  Type REFRESH to submit unload job.

      Data set or Unix file name          Type      NJE node
_ -DATA SET NAME DYNAMICALLY OBTAINED FROM COMMON STORAGE ACT.BACK
- -DATA SET NAME DYNAMICALLY OBTAINED FROM COMMON STORAGE ACT.SMF
***** Bottom of data *****

```

Figure 5-4 zSecure Audit fills in a literal for data set name when using active (ACT)

5.2.2 Security controls for zSecure Audit

Users of the zSecure Audit ISPF interface require access to an XFACILIT profile that matches the menu option entered, for example, option AU.S may be protected by CKR.OPTION.AU.S.** or a similar profile. Users of the zSecure Audit CKFCOLL program require access to a profile covering CKF.AUDIT to use the FOCUS=AUDIT parameter.

READ access to all of the input files used in your zSecure session is also required. Update access to the relevant data sets is required to use the UNLOAD for RACF or CKFCOLL processes.

Additionally, users of zSecure Audit can be restricted to view only the information available using their normal RACF access or privileges. This is known as scoped or restricted mode and can be implemented in various ways.

- ▶ Access to a profile covering the resource CKR.READALL gives full access to view the entire RACF database contents.
- ▶ Option SE.5 can be used to switch a user between restricted and un-restricted view of RACF data. You can control a user's ability to change this setup option; see "RACF security for IBM zSecure" on page 208.
- ▶ Installing zSecure in PADS mode results in all users receiving restricted views of RACF data by default.

If a profile covering the resource CKR.READALL is defined, then restricted mode is enforced. One of the following messages will be present in the SYSPRINT for any execution of zSecure Audit if a profile covering CKR.READALL exists. The message indicates your level of access to the profile:

- ▶ CKR0031 00 Restricted mode active, no READ access to XFACILIT CKR.READALL
- ▶ CKR0031 00 Unrestricted mode active, READ access to XFACILIT CKR.READALL

If no profile covers CKR.READALL then the following message will be shown:

CKR1092 00 CKR.READLL in class XFACILIT no defined. Using defaults

If the output of any zSecure Audit report is not as complete as expected, examine the SYSPRINT output for these messages and their explanatory text to see if you might be running in restricted mode.

Additionally, if you want to control access to the RACF database without setting up Program Access to Data Sets (PADS), set up a zSecure network consisting of only the local system.

5.3 System environment reporting

The zSecure Audit menu option AU.S is the primary point to begin analysis of your security settings. Reports available under this option range from simple listings of data in a readable manner through sophisticated vulnerability scans of the entire operating system environment. Generally, use a previously created CKFREEZE file and a current RACF database copy for most uses of the AU.S menu; however, using just the active system information also works for many of these reports.

A unique feature of zSecure Audit is its ability to *correlate data from multiple sources*, which is used to great effect in the more complex reports available here. For example, zSecure Audit can compare the definitions within your security database, with the data sets used as configuration files in the current operating system. In doing so, the product is able to dynamically determine for any specific operating system configuration whether the appropriate definitions also exist in the security database to ensure that there is adequate protection for this instance of the operating system. This is a truly unique strength, made possible by the underlying architecture and product design.

The AU.S menu provides an initial menu listing five categories of reports, as shown in Figure 5-5. Each of these categories contains a number of specific reports. You can select multiple categories and multiple reports within each category for a single reporting run. You can also select to run in either foreground or batch mode.

```
zSecure Admin+Audit for RACF - Audit - Status
Command ==> _____

Enter / to select report categories
/ MVS tables           MVS oriented tables (reads first part of CKFREEZE)
- MVS extended         MVS oriented tables (reads whole CKFREEZE)
- RACF control          RACF oriented tables
- RACF user             User oriented RACF tables and reports
- RACF resource         Resource oriented RACF tables and reports

Select options for reports:
- Select specific reports from selected categories
/ Concise (short) report
- Output in print format
- Run in background
- Include audit concern overview, higher priorities only

Audit policy
/ zSecure
- C1
- C2
- B1
```

Figure 5-5 Selection menu for System Status reports in the Audit section

Using a combination of various reports, it is possible to discover particular security issues that are combining to provide an exposure. We demonstrate a number of approaches and techniques for doing this task starting with Chapter 13, “Implementation phase I” on page 201 with our customer scenario.

5.3.1 Audit priorities and policies

A major feature of zSecure Audit’s reporting and analysis is its ability to associate an audit finding with a priority. The priority is simply a numerical weighting assigned to any specific audit item to indicate a severity level. Some audit findings can combine, so that when found in the same analysis, their combined presence causes one or both of the original findings to have a higher severity than if found alone. Again, we see zSecure Audit’s ability to correlate what might seem irrelevant information at first glance, into something that reveals complex and non-intuitive audit concerns or vulnerabilities.

Audit priorities reported by zSecure Audit are broadly categorized as follows:

- ▶ 0 through 9: Housekeeping. Usually of informational interest only.
- ▶ 10 through 19: May also represent housekeeping or normal system settings; however, these should be reviewed.
- ▶ 20 through 39: Usually indicates concerns, vulnerabilities, or dangerous security setting, which must be reviewed.
- ▶ 40 and above: Indicates a serious exposure and must be reviewed and either corrected or otherwise mitigated.

Audit priorities are assigned to specific findings using a built-in knowledge base of potential mainframe security issues. Priorities are additionally weighted according to an audit policy that you select when running an analysis. The supplied policies are zSecure, C1, C2, and B1. The zSecure policy is a collection of best practices found in commercial operating environments. The other policy settings come directly from the US DoD Orange Book definitions and standards for trusted computing systems. For detailed information about these other standards, refer to:

<http://csrc.nist.gov/publications/history/dod85.pdf>

The use of priorities and policies allows the user to chose a security posture they want to attain, for example, C1, then easily view the list of audit findings they would need to address to achieve this standard. Ranking the audit findings by priority informs us which issues need to be dealt with and what sequence may be appropriate. We recommend you rectify the highest priority audit findings first.

For an example of this weighting process in action, we look at the SMF subsystem configuration parameters LASTDS and NOBUFFS. Usually in commercial z/OS environments these settings are allowed to default to the value of MSG. These settings cause a message to be issued should SMF fill up its last available data set or run out of temporary buffer space. Should these conditions occur, audit data may be irretrievably lost.

In a commercial z/OS environment, this is typically acceptable, as the alternative is to stop processing system work, that is, halt all online and batch processing. Most commercial installations would prefer to lose some SMF records than stop the entire system.

However, in a military system the same might not be true. It may be acceptable to stop processing should the system be compromised in any way that the audit trail is threatened. So, for military installations it is not uncommon to see LASTDS=HALT and NOBUFFS=HALT.

Therefore, when we analyze a system where these settings are specified as MSG, and use the zSecure default audit policy, these settings appear in our audit report with a priority of only 5, that is, for information. However, analyzing the same system using a B1 policy, which mandates that under no circumstances can the audit trail be lost, these settings are ranked with a priority of 45. An additional audit finding message is also indicated stating that the system is not B1 compliant.

Choosing the right policy to audit your system is something you must decide to suit your company's security posture. It is also worth occasionally checking your system against a stricter policy to discover areas you might like to improve.

5.3.2 Trust, profile audit concerns, and sensitive data trustees

Another extremely powerful feature of zSecure Audit is its ability to analyze *trust*. In the context of zSecure Audit, we use the term trust to mean *privileges* granted to a user that may allow a destructive level of access within the system. So a *trusted user* is one who can damage or undermine the processing of your system. It is likely that these users can also access sensitive or privileged data, perhaps even covering their tracks at the same time.

Trust is an extremely important concept to bear in mind when analyzing the security posture of your system. Ideally, you want to attain a state where the minimum number of staff with a high trust levels exist. Additionally, the access to sensitive resources or data by these trusted users should be segregated such that no one individual has too wide a level of trust.

Trust accumulates; the more access to sensitive data a user has, the higher their accumulated trust level in the related reports. It is sometimes better to have a larger number of users with relatively low trust levels than to have a handful of widely trusted users.

Trust analysis reports are found under both the RACF User and the RACF Resource categories shown in Figure 5-5 on page 97. The first lists the users that are trusted, the second starts with the data sets and privileges. Both allow you to drill down and thus find details. The RACF Resource category houses another important report, the RACF profile audit concerns report.

Often you find that users appearing in the sensitive data trustees and the trusted users reports are receiving their accumulated high trust level virtually by accident. Using the profile audit concerns report to find RACF profiles that have too high UACC or other issues allowing too many users access can reveal quick wins to reduce overall trust levels safely. In Chapter 13, “Implementation phase I” on page 201, we describe in more detail some scenarios for using these reports in this manner.

5.3.3 Automated vulnerability analysis

zSecure Audit provides several reports that may be categorized as vulnerability scans for z/OS and the security management products you are using. These are found under the RACF Control, RACF Resource, and the IBM MVS™ related report categories. There are equivalent options for zSecure ACF2 users.

In the RACF Control category, the SETROPAU report provides a prioritized list of audit concerns found in the RACF system settings. Addressing these audit findings helps you achieve compliance with the declared audit policy and may also reduce overall trust levels granted to system users.

Under the RACF Resources category, the profile audit concerns report, among others, provides analysis of profiles that may allow too wide a level of access, thus introducing a vulnerability to the overall system security.

In the MVS related report categories, the System audit concerns report is a good starting point. Other reports that often reveal vulnerabilities introduced by system configuration settings include the writable storage and SVC related reports.

By analyzing each of the reports, you can quickly come to understand and prioritize the particular audit concerns present in your unique z/OS system configuration.

A deep technical audit by a specialist who has spent a career working in this field should reveal many of the audit concerns automatically identified by just one reporting run using zSecure Audit. However, these deep technical audits are rarely performed, and even more rarely by someone with the necessary skills and experience. zSecure Audit brings to your security management processes the ability to access a deep technical audit on demand, with whatever frequency your security posture demands.

5.3.4 Data sets used by System Status

The MVS Reports category relies primarily on having a CKFREEZE file or files allocated in the current input data. The MVS Extended category uses RACF data and CKFREEZE information to provide its analysis. The RACF related categories use both input sources as well, depending on the specific report selected. If the necessary input files are not present, zSecure Audit will present a message and continue processing. Only reports that can be derived from the live system will be available in such a case.

Use the SETUP FILES command to allocate the input data required by the reports you want to run.

Consideration: Only one RACF database with the same Complex name can be allocated at the same time. If you want to process more than one source of RACF data, allocate the second source with a different value for the Complex name than the initial database. This applies to all RACF input types: active, unloaded, or copy.

5.4 Database verification and cleanup

The reports available under the AU.V Verify menu option are mostly oriented around cleanup activities and verification of appropriate definitions within RACF. Many reports provide features not available with native RACF alone. Others offer enhanced processing in addition to that available with native RACF.

For example, the report option in the RACF Remove ID utility program, IRRRID00, is superseded by the Permit report under AU.V. The Permit report can identify additional places in the RACF database that must be updated to completely remove all references to a user ID, such as class JESSPOOL and RACFVARS profiles that contain the user ID as part of the profile name.

Another example from the Verify reports is the ability to compare all DASD defined data sets with the RACF data set profiles defined for the system, the Protect all report. It is easy then to determine which data sets are not protected, and which RACF data set profiles are not protecting any data sets.

Other useful reports perform tasks such as identifying unnecessary or obsolete profiles (for example, protected resources no longer exist, or an undercutting profile provides the same protections as its superseding profile). Other checks include users with inadequate protective measures in place, and that data sets that are considered *sensitive* have appropriate protection.

It is a useful security measure to run all these reports on your system regularly and review the generated commands. Hopefully many of the reports produce no recommended RACF commands, implying that no action is suggested. Where RACF commands are generated by a verify report, you should review these commands and the messages in SYSPRINT to ensure they are appropriate for your system before executing them.

5.4.1 Requirements for using verify reports

All database verification reports require a RACF input source to be allocated. zSecure Audit will generate a message, and the reports cannot be used until a RACF input file is allocated using the SETUP FILES command.

The verify password report cannot be performed against an UNLOAD copy of RACF and must have a real RACF database or an IRRUT200 generated copy available.

The user must have access to a profile covering CKR.OPTION.AU.V.** to use the ISPF menu for verification. Also, READ access to the RACF input file is required.

5.5 Event reporting

Option EV from the main zSecure ISPF panel invokes the event reporting feature of zSecure Audit. The event reporting application primarily reports on SMF data; however, other sources are supported, including the HTTP server logs, as shown in Figure 5-8 on page 105.

zSecure Admin+Audit for RACF - Main menu		
Option ==>		More: +
SE	Setup	Options and input data sets
RA	RACF	RACF Administration
AU	Audit	Audit security and system resources
EV	Events	Event reporting from SMF and other logs
U	User	User events from SMF
G	Group	Group events from SMF
D	Data set	Data set events from SMF
R	Resource	General resource events from SMF
F	Filesystem	Unix filesystem events from SMF and other logs
I	IP	IP events from SMF and other logs
1	SMF reports	Predefined analysis reports
2	RACF events	RACF logging for specific events
4	DB2	DB2 events from SMF
C	Custom	Custom report
CO	Commands	Run commands from library

Figure 5-6 Menu showing event reporting options

While most reports are drawn from the RACF records, details can be reported from over 60 SMF record types. The broad categories of events to report on are listed in Figure 5-6 on page 103. These include reports specifically tailored to present all required data for user, group, resource, data set, z/OS UNIX file system, TCP/IP, DB2, and other categories.

The selection panel for User reports shown in Figure 5-7 illustrates the many options available to filter the reported data and thus reduce the volume of SMF processed.

```

zSecure Admin+Audit for RACF - Events - User Selection
Command ==> _____ _ start panel

Show records that fit all of the following criteria:
Userid . . . . . _____ (userid or EGN mask)
Owned by . . . . . _____ (group or userid, or EGN mask)
System . . . . . _____ (system name or EGN mask)
Name . . . . . _____ (name/part of name, no filter)
Installation data . . . . . _____ (scan of data, no filter)
Jobname . . . . . _____ (job name or EGN mask)
Terminal . . . . . _____ (Terminal id or EGN mask)

Advanced selection criteria
- User actions          - User attributes          - Date and time
- Data set selection    - HFS selection          - Resource selection
- DB2 selection

Output/run options
/ Include detail        - Summarize          - Specify scope
- Output in print format  Customize title      Send as e-mail
  Run in background      Sort differently

```

Figure 5-7 Filtering panel for user records within event reporting

In addition to basic and advanced selection criteria, this panel allows the user to choose various output and execution options. Output may be emailed, reports wider than a standard panel may be generated for printing, and the report may be executed in batch.

The inclusion of detailed data from the SMF record can also be controlled here. Single line output can be requested for print format, or, for ISPF display format, a single line with the ability to select the line and drill down for greater detail in a multi-line display.

The default reporting of events includes a time stamp and descriptive line with variable text (depending upon event specifics). Appendix B, “An introduction to CARLa” on page 427 provides details about preparing customized user defined event reporting. See 14.1.1, “Using supplied reports” on page 266 for more details about and some examples of event reporting.

Additional requirements: For event reporting to report on a specific SMF type, the data must be generated and retained first. Ensure your SMFPRMxx member is configured to generate the SMF records you want, and that your batch processing of these records makes them available for zSecure Audit to process. The AU.S report SMFSUBOP can be useful to review your SMFPRMxx settings.

5.5.1 HTTP reporting

zSecure Audit supports creating reports for predefined HTTP logs or site defined log types. The installation adds the log data set or file to an input file set, as shown in Figure 5-8, using the appropriate type: WEBACCESS or WEBERROR.

```

zSecure Admin+Audit for RACF - Setup - I Row 2 from 3
Command ==> _____ Scroll ==> CSR

Description . . . . HTTP logs
Complex . . . . .
RRSF node . . . . . Local node for RRSF

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 I R D            Type REFRESH to submit unload job.

      Data set or Unix file name          Type          NJE node
- 'HAHNSM.HTTP.D080330.LOG'              WEBACCESS
- '/u/hahnsm/httplogs/errolog'          WEBERROR

```

Figure 5-8 Allocating HTTP log files for reporting

To report on the content of the HTTP log files, you should examine the CARLa definition statements contained in your SCKRSAMP data set members CKASWBAC and CKASWBER. You will then need to write a custom CARLa program to report on the fields defined here that are of interest to you.

5.5.2 Requirements for using event and HTTP reports

Users of the event reporting panels must have access to a profile covering the resource CKR.OPTION.EV.** in the XFACILIT class. The user must also have read access to the SMF input data sets; any zFS or HFS files and any HTTP log files are allocated as input data.

Most of the input for event reporting comes from the SMF data sets, either live VSAM data sets or from dumped SMF data on tape or DASD.

Many installations maintain generation data groups (GDGs) of their SMF data, not only within a day, but also sets of weekly, monthly, or longer period groupings. SETUP FILES allows the specification of live SMF, or dumped data sets that may include single data sets, a single GDG data set (for example, SMF.DATA(0)), or an entire collection of the GDG (for example, SMF.WEEKLY (without the relative generation number)).

Consideration: If the generation data group is on DASD, all data sets can be allocated; if the generation data group is on tape, the input set will be available for batch operations. We recommend that you update the JCL before submission to use UNIT=AFF=xxxxx for all but the first tape to minimize impact on the tape drive pool.

New to z/OS with V1.9 is the SMF logstream (type: SMF.LOGSTR). When SMF.LOGSTR is specified, zSecure Audit defaults to requesting records for the past 24 hours from the logstream service. This can be modified by changing the data set name specification, for example specifying IFASMF.TYPE80('DURATION=(24,HOURS)') in the data set name column of the input set.

It is also possible to request an SMF logstream from a different date by using the CKRCARLA allocate statement.

5.6 Change tracking

By comparing information from RACF and from CKFREEZE data sets with the values collected the previous day, zSecure Audit can report on differences in the system. This feature is used to provide *change tracking* of z/OS parameters, RACF profiles, and user defined indicators.

The primary intent of the change tracking is to establish a verified base set of parameters and advise management if changes to the established values occurs. Changes can then be selectively approved, rejected, or deferred thorough the zSecure Audit ISPF interface shown in Figure 5-9.

CKREPROD Change Tracking - SC76 Changes		Row 1 of 17
Command ==>		Scroll==> CSR
Enter A(ccept),D(efer),R(eject) or S(how), or p 31 MAR 2008		
Msg	Description	Detail
— C210001	Deletion, System special	AMONSAL
— C210001	Deletion, System special	DEBMC
— C210002	Deletion, System operations	AMONSAL
— C210002	Deletion, System operations	CONWAYM
— C210002	Deletion, System operations	DEBMC
— C210002	Deletion, System operations	HAHNSM
— C210002	Deletion, System operations	JPEASE

Figure 5-9 Determining which exceptions to accept, defer, or reject

A more detailed description and scenarios for use of the change tracking feature can be found in 14.2.1, “Using the change tracking feature” on page 283.

5.6.1 Data sets used by change tracking

Multiple data sets are used by the change tracking process:

1. The verified base data set, which stores the verified base. This verified base data set should be included in the changes track list.
2. Two input data sets. These are the current images against which the trusted base is compared.
 - a. A periodically refreshed CKFREEZE (commonly generated daily)
 - b. A periodically refreshed RACF unload (commonly generated daily)
3. The master data set, which stores those systems under change tracking administration.
4. Local setup tables updated by the ISPF-transactions within the change processing under the AU.C option.
5. The exceptions data set, which stores the exceptions to the verified base.
6. The defer data set, which stores the deferred exceptions to the verified base.

5.6.2 Security controls for change tracking use

RACF profiles in the XFACILIT class control the actions available to a user who is performing change tracking. If a user does not have READ access to these profiles, the options do not appear in the menu of available options, and they may not be specified in a fast path menu command such as =AU.C.

► CKR.OPTION.AU.C

This resource controls access to the main AU panel from the zSecure primary ISPF menu. The user will also require READ access to profiles that cover the following resources to inspect reported changes and perform follow up actions:

- CKR.ACTION.CH.* for actions on Exceptions overview
- CKR.ACTION.CT.* for actions on Systems overview

► CKR.OPTION.SE.C

This resource controls access to the setup option for change tracking. READ access to a profile covering this resource allows a user to maintain the control tables used by the process, for example, the list of data set names to be considered “sensitive”. Figure 5-10 shows the addition of a new set of data to be considered as sensitive for change tracking purposes using this option.

Note: Additions made to this table are included in the standard CARLa REPORT SENSITIVE list of sensitive data sets.

zSecure Admin+Audit for RACF - Change Row 1 to 1 of 1		
Command ==> _____	Scroll ==> <u>PAGE</u>	
Insert, Delete or Select (I/D/S) a data set mask		
Data set mask	Access level	System
CKRU.**	READ	SC76

Figure 5-10 Maintaining the list of sensitive data sets

5.7 Library and sequential data set audit

The library audit feature of zSecure Audit is used to track member level modifications to sensitive data sets and configuration files for z/OS. It may also be used to track this type of change for additional user specified files.

A library audit works by performing a checksum operation on each member of a library and recording the unique digital signature generated as a result. This process can be applied to both normal files and load libraries with equal effect. It is then possible to compare the digital signature between two copies of the original data and thus verify that the data has not been changed, or reveal the fact that it has changed should the signatures now differ.

A similar operation is also available from zSecure Audit for sequential data sets on DASD or tape and is known as *fingerprinting*.

Some preparatory work is required to establish a base line against which subsequent signature comparisons will be made. At least two images are required for changes to be reported. The signature CKFREEZE can be smaller than the normal CKFREEZE usually generated nightly for full analysis.

A sample CKFREEZE job that might be scheduled for nightly, weekly, or monthly running can be generated with option AU.L.0 and is shown in Figure 5-11.

```
//SIGNATS1 JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID
//*
//*****
//* CREATE SIGNATURE FILE
//*****
//*
//CKFREEZE EXEC PGM=CKFCOLL,REGION=64M
//STEPLIB DD DISP=SHR,DSN=CKR.SCKRLOAD
//SYSPRINT DD SYSOUT=*
//CKFREEZE DD DISP=(NEW,CATLG),
//          DSN=CKRU.DAILY.CKFREEZE.SIGNATUR(+1),
//          UNIT=SYSDA,SPACE=(27998,(30,30),RLSE,,ROUND)
//SYSIN DD *
DASD=N,TAPE=N,SWCH=N,HFS=N,PATH=N,SMS=N,CAT=MCAT
CHECK=Y,PDS=YES
SCANSTR=('SECURITY','BYPASS','KEY=ZERO','MODE=SUP')
//
```

Figure 5-11 Jobstream to create the next signature file

5.7.1 Data sets used by data set audit

The library audit function uses at least two CKFREEZE data sets. Usually these would represent the beginning of a reporting interval, paired with end of the interval, for example, yesterday compared with today or last week compared with today. The frequency with which you run library analysis depends on your unique security requirements; however, it is not uncommon to see library analysis performed as part of a daily verification suite.

Figure 5-12 shows two generations of a signature CKFREEZE file allocated for processing by library analysis.

```
zSecure Admin+Audit for RACF - Setup - I Row 2 from 3
Command ==> _____ Scroll ==> CSR

Description . . . . daily signature qdg input file
Complex . . . . . SC76
RRSF node . . . . . Local node for RRSF

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                  Type REFRESH to submit unload job.

      Data set or Unix file name          Type          NJE node
_   'CKRU.DAILY.CKFREEZE.SIGNATUR(0)'      CKFREEZE
_   'CKRU.DAILY.CKFREEZE.SIGNATUR(-1)'      CKFREEZE
***** Bottom of data *****
```

Figure 5-12 Input set of files for library change audit

zSecure Audit can also report on the updated z/OS tracking of members with PDS and PDS/E data sets using SMF type 42 records.

5.7.2 Security controls for using library analysis

Users must have access to a profile covering CKR.OPTION.AU.L.** and READ access to the CKFREEZE signature data sets.

5.8 Conclusion

In this chapter, we discussed the power and capabilities of the zSecure Audit product. Not only can it report on your live system (or snapshots from a previous time), but also live or unloaded SMF data sets, live or unloaded RACF databases, and HTTP logs and logged events from other security systems.

Using automated vulnerability analysis and verification and cleanup capabilities, zSecure Audit becomes an essential part of your security tool set to ensure that z/OS and RACF stay the way you intend them to be.

zSecure Audit provides a simple and deeply detailed user interface to explore and document your security environment.



IBM Security zSecure Visual

In this chapter, we introduce the Security zSecure Visual product. zSecure Visual allows your security staff to perform mainframe administrative tasks from a Microsoft Windows-based GUI for RACF administration. It is a GUI for basic drag and drop RACF administration. It provides a Windows interface for decentralized RACF administration. The Windows clients talk to the server on the mainframe, which interacts with the zSecure Admin engine and RACF itself.

We show some main features of the product and demonstrate the available efficiency gains using capabilities such as drag and drop, which are native to the Windows environment. We briefly describe the software architecture and implementation.

A major feature of zSecure Visual is its ability to provide a view of RACF data without requiring high level security privileges on the mainframe. Additionally it can provide *scoped* or limited views of security data depending on the security privileges granted to its users. This feature will be covered in some detail using real world examples.

This chapter covers the following topics:

- ▶ zSecure Visual architecture and implementation
- ▶ Usage scenarios
- ▶ RACF scoping rules and examples

6.1 zSecure Visual architecture and implementation

zSecure Visual is implemented as a client-server application. The client is what is sometimes referred to as a *thick client*, meaning that it is an application specifically installed on the users Windows desktop rather than a web based or lightweight deliverable. The server is a long running address space, usually referred to as a *Started Task*, on the mainframe and takes advantage of the UNIX System Services for z/OS component of the z/OS operating system.

zSecure Visual leverages components of the zSecure suite by using CARLa (the CARLa Auditing and Reporting Language) queries and CKGRACF scoping rules, although you do not require licensed copies of other zSecure suite products on your system to take advantage of zSecure Visual.

Figure 6-1 shows the architecture and data flow of the zSecure Visual client and server.

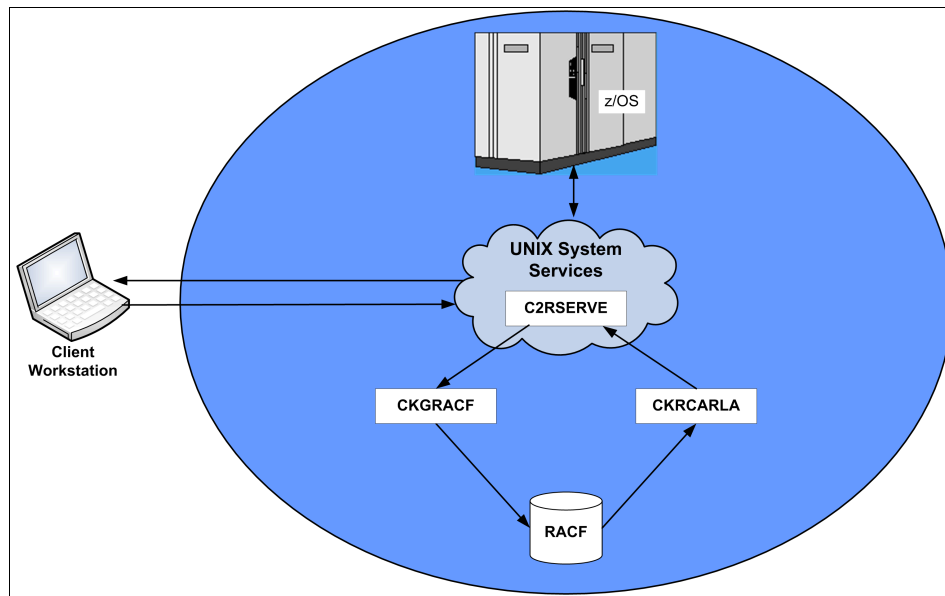


Figure 6-1 zSecure Visual client and server architecture

Using zSecure Visual, an administrator can perform all the common functions necessary for user and resource management in RACF without having access to the TSO environment normally used by RACF administrators. zSecure Visual administrators do require a z/OS UNIX System Services UID and GID, though. This can be provided either individually through an OMVS segment in their RACF user ID definition or through the UNIX System Services default UID/GID facility. The user of zSecure Visual does not need a home directory in z/OS UNIX.

Several roles representing common types of security administrators are built-in to the product. We will explain how to set up and use these roles in the following sections. Use of the roles is not mandatory, although it may assist in initially setting up and getting used to the way zSecure Visual controls RACF authorities.

The zSecure Visual client should use a fixed IP address rather than a DHCP allocated address that may change with desktop restarts, if possible; however, a DHCP provided address will work as well.

When a client is first defined to the server, an initial encryption key is generated for that specific connection. The key must be transmitted from the server to the client in a secure manner and entered into the client/server connection configuration dialog to establish the initial connection. Then the client and server negotiate their own session keys. The initial key is time limited and must be used within 24 hours or it will be no longer valid for that IP address to connect to the server. There are facilities in the zSecure based ISPF dialog to generate keys en-mass for a rollout to many clients and to list the IP addresses used by already configured clients.

6.2 Usage scenarios

zSecure Visual may be used to control or limit the administrative privileges of RACF security administrators. It implements these controls using profiles in the RACF database under the controls provided by the CKGRACF component of zSecure Admin. As such, the zSecure Admin CKGRACF rules effectively control the behavior of the zSecure Visual GUI.

6.2.1 zSecure Visual default roles

The *IBM Tivoli zSecure Visual Server Manual Version 1.11*, SC23-6549 documents six commonly used roles and provides a table indicating recommended access permissions and CKG profiles for these roles.

We converted this table into native RACF commands to easily implement the recommended defaults. These commands are listed in Appendix C, “User roles for IBM Security zSecure Visual” on page 455 for your convenience. It is possible to tweak the access privileges associated with the roles, but doing this task requires a thorough understanding of the entire CKGRACF language. Refer to 15.1, “Delegated RACF administration” on page 322 for more information about using CKGRACF. If you use the recommended roles, you will also have to supplement them with the scoping profiles described in 6.2.3, “CKGRACF ID level scoping” on page 115.

The recommended roles are described as:

- ▶ Helpdesk: Reset passwords, and enable and disable user IDs.
- ▶ Connect: Manage user connection and removal from groups.
- ▶ User: Create, delete, and manage user IDs.
- ▶ Access List: Permit and delete access list entries of profiles.
- ▶ Group: Add, delete, and change groups.
- ▶ Full: All administrator functions.

6.2.2 RACF setup to support zSecure Visual

To get the most from zSecure Visual, it is essential to have a good understanding of the functions provided by the CKGRACF program, especially its ability to scope RACF access. Native RACF scoping is a complex and generally not well understood concept. zSecure Visual simplifies this complexity by using CKGRACF scoping instead. CKGRACF scoping is implemented using RACF profiles that describe the ownership of RACF definitions. The use of these RACF profiles is easier to understand for many administrators than traditional RACF scoping rules.

The CKGRACF scoping profiles are implemented in three distinct methods:

- ▶ ID level scoping profiles
- ▶ GROUP level scoping profiles
- ▶ USER level scoping profiles

You may chose to exploit a combination of these profiles as necessary to suit your particular RACF ownership structure. Additionally, zSecure Visual allows for an administrator’s access to take into account any native RACF scope they also possess.

Before we discuss these three implementation methods for zSecure Visual scoping, we will first review the basic concepts of RACF ownership in general. Awareness of RACF ownership is a prerequisite for understanding zSecure Visual scoping rules.

In RACF, all User IDs, Groups, and Dataset or General resources (referred to generically as *RACF profiles*) have an owner. The owner is a field in the RACF profile that is set by the administrator when the profile is defined. By default, the administrator who defined the profile is the owner. Unfortunately, this leads to a situation common in many RACF installations where many profiles are owned by a small number of administrators who were involved in the profiles' initial definition. Often this ownership by the administrators makes the use of native RACF scoping difficult, usually requiring correction of the owner after the profile has been defined. zSecure Visual allows you to gain the practical benefits of scoped RACF administration without having to restructure your RACF group tree or change the way that administrators define new profiles.

We recommend that you have a well defined RACF group tree and resource ownership structure and conventions. zSecure Admin can help to establish these settings, and zSecure Command Verifier can help ensure that structural and naming conventions remain in place and are not accidentally altered during normal administrative processes.

For the remainder of this chapter, we will be referring to group tree structures in our examples that are defined and explained more fully in 13.5.3, "Implementing an improved RACF group tree structure" on page 241.

6.2.3 CKGRACF ID level scoping

RACF profiles defined in the XFACILIT class starting with CKG.SCP.ID provide ID level scoping. Access to these profiles allow a zSecure Visual administrator authority over RACF resources specifically owned by an ID. The RACF profiles that are used to define ID level scoping are in the following format:

```
CKG.SCP.ID.user.owner.default-group
```

or

```
CKG.SCP.ID.group.owner
```

You may use any combination of the *user*, *owner*, and *default-group* or *group* qualifiers. These qualifiers may also contain generic characters. It is important to remember when defining these profiles that normal RACF generic processing applies. Thus, only the most fully qualified profile will apply for any access check.

For example, if we have a user ID TESTUSER, who is owned by a group TESTOWNG and has a default group of TESTACCG, the following resource name will be used by zSecure Visual in any calls to RACF:

```
CKG.SCP.ID.TESTUSER.TESTOWNG.TESTACCG
```

Access to a RACF profile matching this resource will effectively place the TESTUSER user ID within the scope of the zSecure Visual administrator. Any of the following profile definitions are a match, from most fully qualified first, to least fully qualified last:

1. XFACILIT CKG.SCP.ID.TESTUSER.*.*
2. XFACILIT CKG.SCP.ID.*.TESTOWNG.*
3. XFACILIT CKG.SCP.ID.*.*.TESTACCG

Depending on your naming conventions, you could use any of these profiles to bring the user ID TESTUSER within the scope of a zSecure Visual administrator. Profile 3 will provide CKGRACF scope over users who have the RACF default group of TESTACCG. Profile 2 covers user IDs who are owned by the group TESTOWNG. Access to profile 1 grants scope over only the user ID TESTUSER.

ID profiles may also be used to provide CKGRACF scope over RACF groups as well. Using the TESTACCG and TESTOWNG example above, to bring these groups within a zSecure Visual administrators scope, the following profiles might be used:

```
XFACILIT CKG.SCP.ID.TESTACCG.*
XFACILIT CKG.SCP.ID.TESTOWNG.*
```

ID level scoping is useful when you need to grant zSecure Visual administrative scope over a specific user, group of users, or profiles owned by a specific RACF group or user ID. Using generic masking in ID scoping profiles provides even greater flexibility if your user IDs or groups adhere to a naming convention that works well with RACF generic pattern matching, for example:

```
XFACILIT CKG.SCP.ID.TEST*
```

This profile will provide CKGRACF scope over both the TESTACCG and the TESTOWNG group, as well as the TESTUSER user ID profiles (provided a more fully qualified profile is not defined).

Getting the CKG profiles correct when setting up zSecure Visual and the CKGRACF scoping feature in general can become quite complex. Refer to Chapter 15, “Implementation phase III” on page 321 for more examples and detailed descriptions of the approach we took to simplify these issues.

6.2.4 CKGRACF USER and GROUP level scoping

For a different level of flexibility for the ID scoping profiles discussed above, CKGRACF provides two further methods to control an administrator’s scope within RACF. These are known as *group and user level scoping*. Access to these profiles provides CKGRACF scope for a zSecure Visual administrator over user IDs, groups, and resources owned by a user or group.

The difference between ID level scoping and Group/User level scoping is in the method they use to determine if a RACF profile is within their scope. With ID level scoping, the choice is either *owner*, *default-group*, or *user ID*, specifically. With Group/User level scoping, the RACF ownership chain itself is used, in the same manner as native RACF scoping.

Access to Group and User level scoping profiles flows down the RACF ownership chain. Resources take the form of:

```
CKG.SCP.G.group.owner.owner....  
CKG.SCP.U.userid.owner.owner....
```

You could define profiles, such as:

```
XFACILIT CKG.SCP.G.**  
XFACILIT CKG.SCP.U.**
```

Access to these would effectively provide zSecure Visual administrators with scope over all user IDs, groups and resources in RACF (assuming no more fully qualified profile denied access).

An example serves to best demonstrate these profiles. Figure 6-2 shows an example group tree.

Note: The group in the CKG.SCP.G.group type profiles must be a direct subgroup of SYS1. In some circumstances, it may be easier to implement profiles of the format CKG.SCP.G.**.TESTOWNG.**.

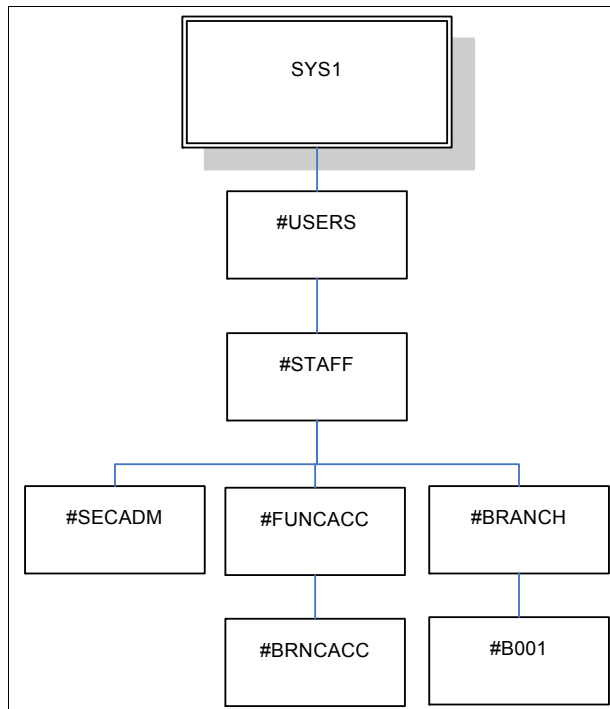


Figure 6-2 Group tree example

The connecting lines represent RACF ownership:

- ▶ #SECADM owns the security administrators.
- ▶ #B001 owns some branch users.
- ▶ #STAFF owns no users, but owns the whole staff ownership part of the tree.
- ▶ #FUNCACC owns the groups that are used to grant access to RACF profiles.

We define the following scoping profiles, with the listed intentions:

- ▶ `XFACILIT CKG.SCP.G.#USERS.**.#SECADM`

Use this to prevent zSecure Visual administrators from looking at the user IDs of the global security administrators. This profile covers any user ID owned by `#SECADM` where `#SECADM` is somewhere in the subgroup tree of `#USERS`, a subgroup of `SYS1`.

- ▶ `XFACILIT CKG.SCP.G.*.#STAFF.#BRANCH.**`

Use this to allow the zSecure Visual administrators to look at any users owned under the `#BRANCH` group subtree. Note that this profile will apply if the group `#STAFF` is owned by any subgroup of `SYS1`. This is the effect of the `*` in the profile.

- ▶ `XFACILIT CKG.SCP.G.*.#STAFF.**`

Use this to grant zSecure Visual administrators scope over users under the `#STAFF` subtree. These administrators cannot work with the security administrators user IDs unless they are granted access to the `CKG.SCP.G.#USERS.**.#SECADM` profile as well.

The User level scope profiles work similarly to the group level ones, except that the *userid* qualifier after `CKG.SCP.U.userid` is the user ID who owns that part of the RACF group tree. This may be most useful if you already have delegated administrative rights to specific users, who currently use direct ownership or RACF group special to perform their admin tasks. Using zSecure Visual, this type of user no longer requires group special, and can be more precisely controlled in the range of RACF commands they may issue.

6.2.5 Using RACF group special

One more usage scenario for zSecure Visual is also provided. If you are already exploiting RACF native group scoping with group special, you can allow zSecure Visual administrators to use their existing group special authorities through the windows GUI.

This is implemented with a RACF resource:

`CKG.SCP.RACF`

By granting administrators access to a profile covering this resource, the administrators scope within zSecure Visual will be extended to include their native RACF group special scope.

6.2.6 Access level on scoping profiles

Two access levels are distinguished by CKGRACF and zSecure Visual. The administrator who has READ access on a scoping profile has the ability to see profiles within the scope, similar to the (group) auditor attribute in RACF.

Administrators with UPDATE access on a scoping profile have the ability to change profiles within the scope, similar to (group) special.

6.3 RACF scoping rules and examples

You can think of ID level scoping as being most appropriate when you know exactly the resource or profile you want to control administration over, and you can tightly define its name and ownership. Use ID level scoping to control specific profiles in RACF. While group and user level scoping may be more useful when you have an ownership structure, you may want to delegate responsibility for the resources and profiles within that ownership structure.

Besides scoping rules, CKGRACF and zSecure Visual use CMD rules to specify the actions that an administrator may perform. The access required for CMD profiles is READ for operations that list information, and UPDATE for operations that may change information.

6.3.1 Local password administrator

In Figure 6-3, we show a sample screen capture of a zSecure Visual administrator who is only able to perform basic password management on users in a specific part of the RACF group tree.

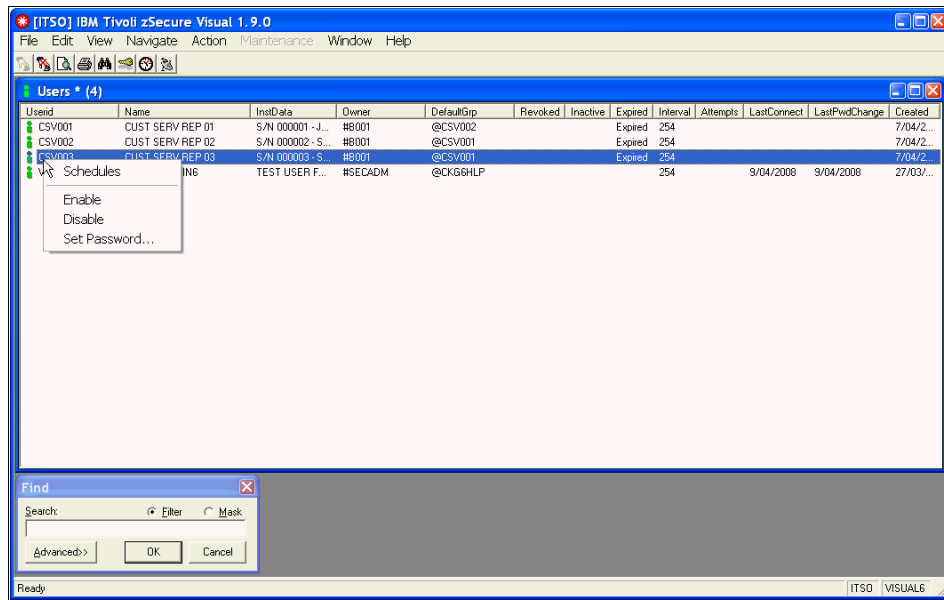


Figure 6-3 Basic password management

The users whom this administrator has authority over are all owned by the RACF group #B001, and the functional profiles the administrator has access to are:

CKG.CMD.COMMENT
CKG.CMD.LIST
CKG.CMD.SHOW.MYACCESS
CKG.CMD.USER.REQ.PWRESET
CKG.CMD.USER.REQ.PWSET
CKG.CMD.USER.REQ.RESUME
CKG.CMD.USER.REQ.SCHEDULE

In addition, the password administrator is scoped using the following profile:

XFACILIT CKG.SCP.ID.*.#B001.*

This profile ensures they can only use their password reset capabilities against user IDs that are directly owned by the RACF group #B001.

6.3.2 Branch wide group connections administrator

In Figure 6-4, we show a sample screen capture of a zSecure Visual administrator who is able to perform connections of users to groups, across all branches defined in the group tree. You can see a Create Connect window, which appears when the administrator simply drags a user from the top window, over to one of the groups in the second window, demonstrating simple drag and drop RACF administration.

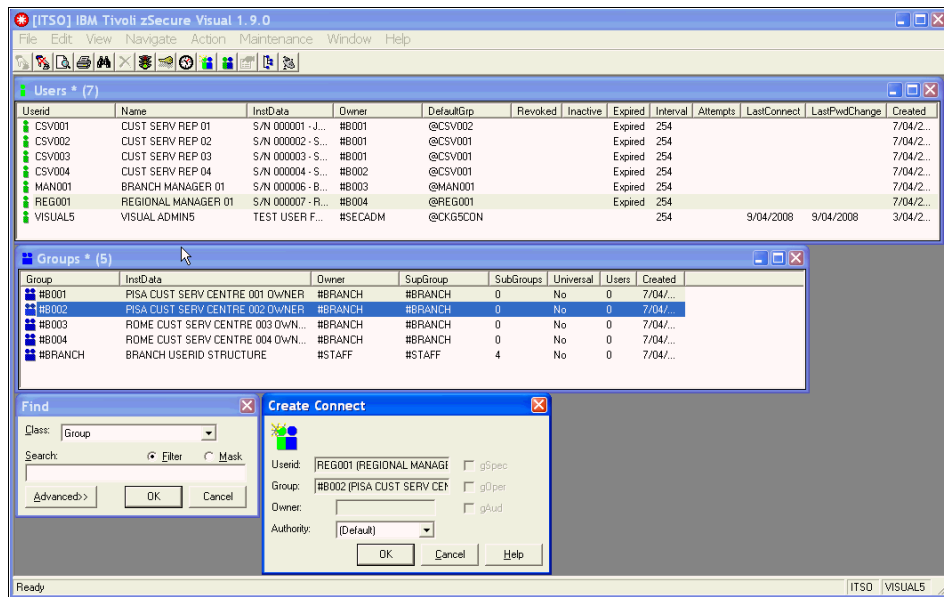


Figure 6-4 Group level administration

The functional profiles this administrator has access to are:

```
CKG.CMD.CMD.REQ.CONNECT
CKG.CMD.CMD.REQ.REMOVE
CKG.CMD.COMMENT
CKG.CMD.LIST
CKG.CMD.SHOW.MYACCESS
CKG.CMD.USER.REQ.PWRESET
CKG.CMD.USER.REQ.PWSET
CKG.CMD.USER.REQ.RESUME
CKG.CMD.USER.REQ.SCHEDULE
```


The users whom this administrator has authority over are all owned somewhere under the RACF group tree starting at #BRANCH. Thus the administrators access has been scoped to cover only these users, using the following RACF profiles:

```

CKG.RAC.SCP.CONNECT.BASE.AUTH.USE
CKG.SCP.G.#FUNCACC.#BRNACC.**
CKG.SCP.G.#USERS.#STAFF.#BRANCH.**
CKG.SCP.ID.*.#B001.*

```

6.3.3 Staff wide user administrator

Figure 6-5 shows an administrator who can manage user IDs across the entire branch of the RACF group tree for staff. They can request user IDs to be created or deleted and manage password resets and other basic user ID provisioning tasks.

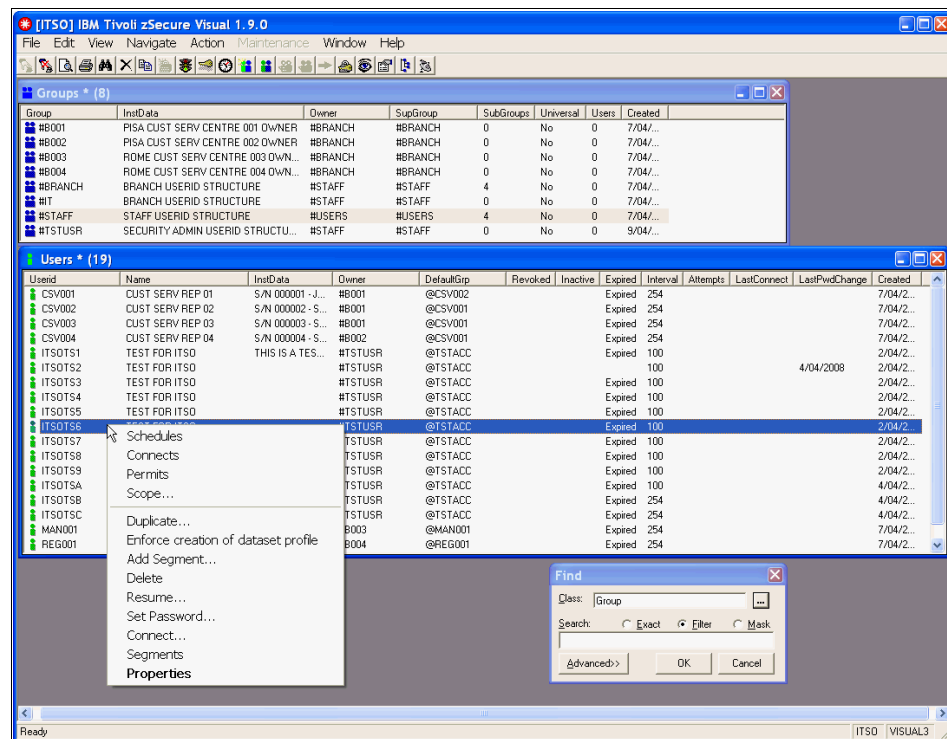


Figure 6-5 Staff wide user administration role

The functional profiles this administrator has access to are:

```
CKG.CMD.CMD.EX.ADDSD
CKG.CMD.CMD.EX.ADDUSER
CKG.CMD.CMD.EX.ALTDSD
CKG.CMD.CMD.EX.ALTUSER
CKG.CMD.CMD.EX.PERMIT
CKG.CMD.CMD.EX.RALTER
CKG.CMD.CMD.EX.RDEFINE
CKG.CMD.CMD.EX.SETROPTS
CKG.CMD.CMD.REQ.CONNECT
CKG.CMD.CMD.REQ.PERMIT
CKG.CMD.CMD.REQ.REMOVE
CKG.CMD.COMMENT
CKG.CMD.LIST
CKG.CMD.SHOW.MYACCESS
CKG.CMD.USER.REQ.PWDEFAULT
CKG.CMD.USER.REQ.PWRESET
CKG.CMD.USER.REQ.PWSET
CKG.CMD.USER.REQ.PWSET.DEFAULT
CKG.CMD.USER.REQ.PWSET.EXPIRED
CKG.CMD.USER.REQ.PWSET.NONEXP
CKG.CMD.USER.REQ.PWSET.PASSWORD
CKG.CMD.USER.REQ.PWSET.PREVIOUS
CKG.CMD.USER.REQ.RESUME
CKG.CMD.USER.REQ.SCHEDULE
CKG.RAC.SCP.CONNECT.BASE.AUTH.USE
CKG.RAC.SCP.CONNECT.BASE.AUTH.*
CKG.RAC.SCP.*.BASE.*
```

Additionally, the administrator has been scoped to work only with users and groups from the functional access and staff sections of the RACF group tree:

```
CKG.RAC.SCP.*.BASE.*
CKG.RAC.SCP.CONNECT.BASE.AUTH.USE
CKG.SCP.G.#FUNCACC.#BRNCACC.**
CKG.SCP.G.#FUNCACC.#USERACC.**
CKG.SCP.G.#USERS.#STAFF.**
CKG.SCP.G.#USERS.#STAFF.#BRANCH.**
CKG.SCP.ID.*.#B001.*
```

6.3.4 Applications data administrator

Figure 6-6 shows the groups available for processing by an administrator restricted to managing specific applications related groups and data set profiles. This administrator can create new groups, and data set profiles for these, as well as permit users or groups to these data set profiles. This can only be done within their defined administrative scope.

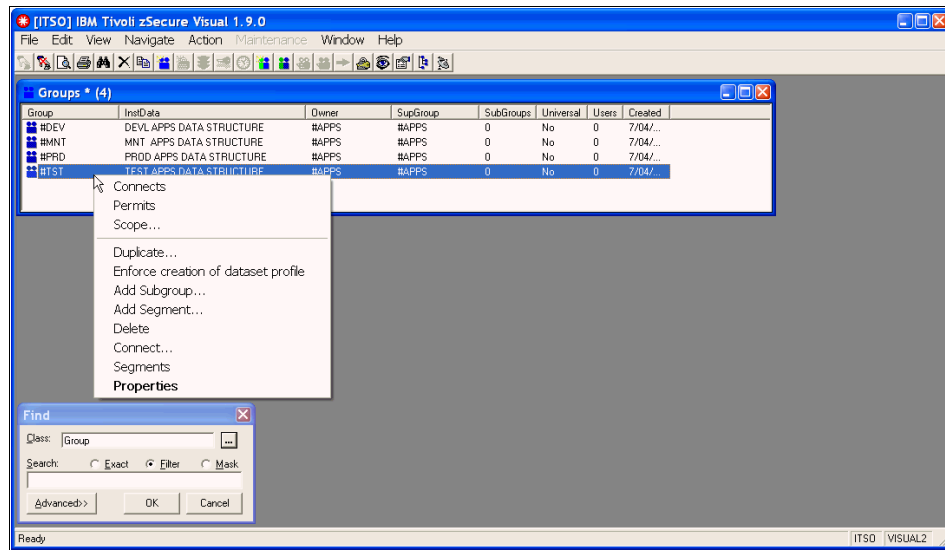


Figure 6-6 Applications data access administrator role

This administrator has access to the following functional profiles:

CKG.CMD.CMD.EX.ADDGROUP
CKG.CMD.CMD.EX.ADDSD
CKG.CMD.CMD.EX.ADDUSER
CKG.CMD.CMD.EX.ALTDSD
CKG.CMD.CMD.EX.ALTGROUP
CKG.CMD.CMD.EX.ALTUSER
CKG.CMD.CMD.EX.DELDSD
CKG.CMD.CMD.EX.DELGROUP
CKG.CMD.CMD.EX.PERMIT
CKG.CMD.CMD.EX.RALTER
CKG.CMD.CMD.EX.RDEFINE
CKG.CMD.CMD.EX.RDELETE
CKG.CMD.CMD.EX.SETROPTS
CKG.CMD.CMD.REQ.CONNECT
CKG.CMD.CMD.REQ.PERMIT

```

CKG.CMD.CMD.REQ.REMOVE
CKG.CMD.COMMENT
CKG.CMD.LIST
CKG.CMD.SHOW.MYACCESS
CKG.CMD.USER.REQ.PWDEFAULT
CKG.CMD.USER.REQ.PWRESET
CKG.CMD.USER.REQ.PWSET
CKG.CMD.USER.REQ.PWSET.DEFAULT
CKG.CMD.USER.REQ.PWSET.EXPIRED
CKG.CMD.USER.REQ.PWSET.NONEXP
CKG.CMD.USER.REQ.PWSET.PASSWORD
CKG.CMD.USER.REQ.PWSET.PREVIOUS
CKG.CMD.USER.REQ.RESUME
CKG.CMD.USER.REQ.SCHEDULE
CKG.RAC.SCP.CONNECT.BASE.AUTH.USE
CKG.RAC.SCP.CONNECT.BASE.AUTH.*
CKG.RAC.SCP.*.BASE.*

```

They have then been scoped to allow them to administer the access list and resources only under the applications owning part of the RACF group tree:

```

CKG.RAC.SCP.*.BASE.*
CKG.RAC.SCP.CONNECT.BASE.AUTH.USE
CKG.SCP.G.#APPS.**

```

6.3.5 Resource access list administrator

The resource access list administrator can permit users or groups to general resource profiles within their defined administrative scope. Figure 6-7 on page 127 shows the difference between resources within and outside their scope.

At the top of the window is a list of all resources in the class XFACILIT. Only some of these are within our administrator's scope. You can see that the first profile the administrator displayed, CKG.CMD.USER.REQ.PWSET.PREVIOUS, does not display an access list for the administrator to work with. With the second profile, CKG.SCP.G.*.#STAFF.#BRANCH, our administrator can see and work with the profile access list.

The resource administrator is scoped to only work with profiles owned by the RACF group #SYSTECH in this case. You can see the owner of the profiles displayed in the top window of Figure 6-7 on page 127. The administrator cannot work with the profiles that are owned by group SYS1.

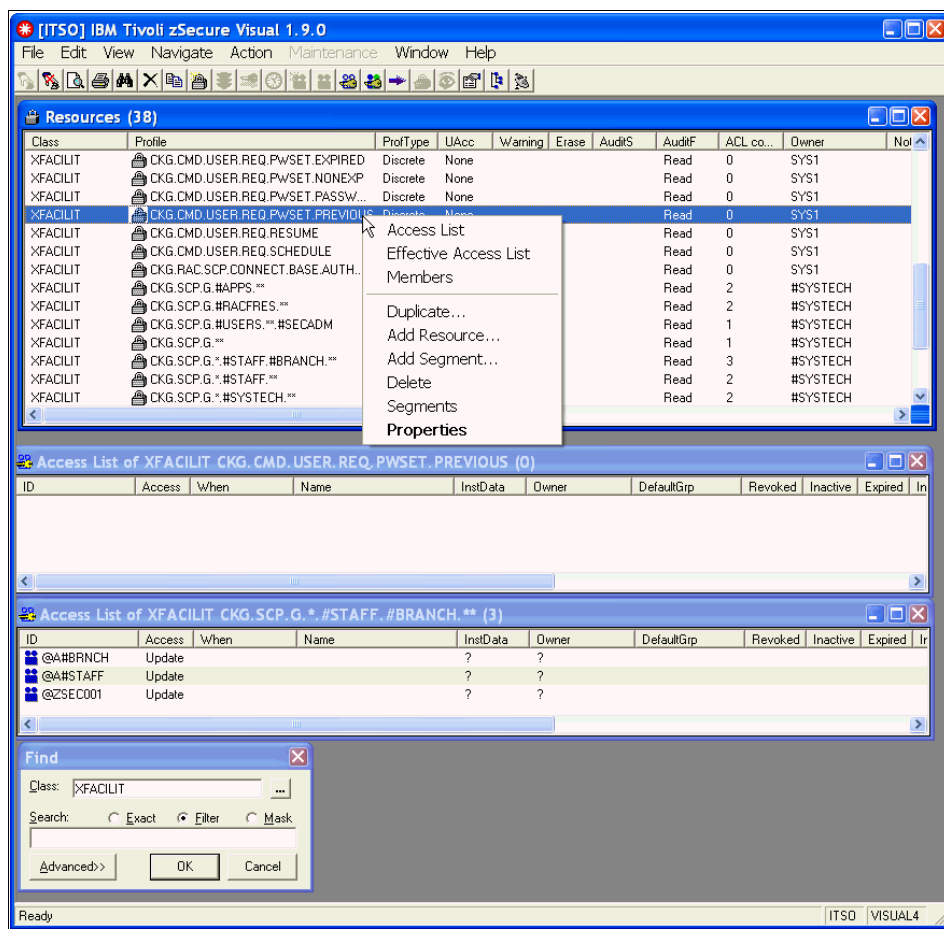


Figure 6-7 Restricted RACF classes administrator role

The resource administrator has access to the following functional profiles:

CKG.CMD.CMD.EX.ADDSD
 CKG.CMD.CMD.EX.ADDUSER
 CKG.CMD.CMD.EX.ALTDSD
 CKG.CMD.CMD.EX.ALTUSER
 CKG.CMD.CMD.EX.PERMIT
 CKG.CMD.CMD.EX.RALTER
 CKG.CMD.CMD.EX.RDEFINE
 CKG.CMD.CMD.EX.SETROPTS
 CKG.CMD.CMD.REQ.CONNECT
 CKG.CMD.CMD.REQ.REMOVE
 CKG.CMD.COMMENT

```
CKG.CMD.LIST
CKG.CMD.SHOW.MYACCESS
CKG.CMD.USER.REQ.PWDEFAULT
CKG.CMD.USER.REQ.PWRESET
CKG.CMD.USER.REQ.PWSET
CKG.CMD.USER.REQ.PWSET.DEFAULT
CKG.CMD.USER.REQ.PWSET.EXPIRED
CKG.CMD.USER.REQ.PWSET.NONEXP
CKG.CMD.USER.REQ.PWSET.PASSWORD
CKG.CMD.USER.REQ.PWSET.PREVIOUS
CKG.CMD.USER.REQ.RESUME
CKG.CMD.USER.REQ.SCHEDULE
CKG.RAC.SCP.CONNECT.BASE.AUTH.USE
CKG.RAC.SCP.CONNECT.BASE.AUTH.*
```

Additionally, the administrator is scoped to allow access to resources and users within the following sections of our group tree:

```
CKG.RAC.SCP.CONNECT.BASE.AUTH.USE
CKG.RAC.SCP.XFACILIT.BASE
CKG.RAC.SCP.XFACILIT.BASE.ACCESS.*
CKG.SCP.G.#FUNCACC.#BRNCACC.**
CKG.SCP.G.#FUNCACC.#USERACC.**
CKG.SCP.G.#RACFRES.#SYSTECH
```

6.3.6 Multiple system support

In addition to supporting the z/OS system on which the server is running, zSecure Visual can also access RACF databases that are on other systems within the zSecure network. While working in multi-system mode, the user can specify with which systems within the network they will be working. Additionally, the built commands can be routed to one or more of the connected systems. This action reduces the time required for maintaining the systems, and reduces the risk of typographical errors by eliminating the need to enter a command multiple times.

This capability can make the maintenance of multiple RACF databases easier and more fail safe than it has been in previous years.

6.4 Conclusion

zSecure Visual provides a full function RACF administration interface, with the added benefit of being able to control the Administrative authorities in a detailed way. This helps to properly secure an infrastructure from both accidental and malicious damage by your most trusted internal staff: the privileged users.

The additional controls implemented by zSecure Visual operate in two important ways:

- ▶ Reducing the specific commands and options available to a RACF administrator, in a way that is difficult or impossible using only native RACF.
- ▶ Reducing the scope of authority in terms of which users, groups, and resources an administrator can perform actions against. Again, this is difficult or impossible with only RACF in use.

In 15.1, “Delegated RACF administration” on page 322, we will provide additional details about the scoping capabilities provided by the CKGRACF function, referring back to the user IDs defined above for zSecure Visual use.



IBM Security zSecure Command Verifier

In this chapter we introduce the Security zSecure Command Verifier. The zSecure Command Verifier provides fine grained control over RACF commands and parameters on certain commands. Additionally, it is designed to replace many of the typical RACF customized exits that are sometimes developed by customers. It can also be configured to maintain a *command audit trail* of modifications made to RACF resources.

We explain how zSecure Command Verifier can be used as a proactive enforcer of site security standards, providing default values or overriding undesirable parameters on RACF commands. It compares RACF commands against defined security policy and adapts or blocks noncompliant ones. We also cover typical user exit replacement scenarios and the use of the Command Audit Trail to enhance RACF auditing.

This chapter covers the following topics:

- ▶ zSecure Command Verifier architecture
- ▶ Controlling RACF commands
- ▶ Replacing user exits
- ▶ Command audit trail feature
- ▶ Alerting and action capabilities

7.1 zSecure Command Verifier architecture

zSecure Command Verifier is implemented as a z/OS dynamic exit using the *RACF common command exit point* (IRREVX01). As such, the Command Verifier supplements any existing site exit that uses the same exit point. The main advantage of using the Command Verifier over traditional RACF exit points is the reduction in complexity involved in effectively coding the combination of RACF exits necessary to control any particular action.

For example, to control data set naming conventions using standard RACF exits, up to three RACF exits may be required. Even then, the solution may not cover all possible ways of defining a RACF data set profile, as this is a non-trivial task. The Command Verifier allows for control of almost all RACF commands and the controls are implemented using standard RACF profiles and access list entries, something familiar to all RACF administrators and not requiring systems programming or assembler skills.

zSecure Command Verifier also generates SMF audit records for actions it may take against a RACF command. Again, this is a non-trivial task in a RACF exit and most user written exits do not supply audit records documenting the exit's processing. It is possible to further modify the processing of RACF commands using a Command Verifier exit point, although this is an advanced feature and will not be explored here.

Another advantage of using the common command exit is that all entry points where RACF commands may be issued are passed through this exit. This includes RACF commands entered through the System Operators console, the UNIX for z/OS RACF callable services (R_Admin), RACF Remote Sharing Facility (RRSF), and the standard RACF TSO commands.

Figure 7-1 shows a conceptual data flow of RACF command processing where zSecure Command Verifier is in use.

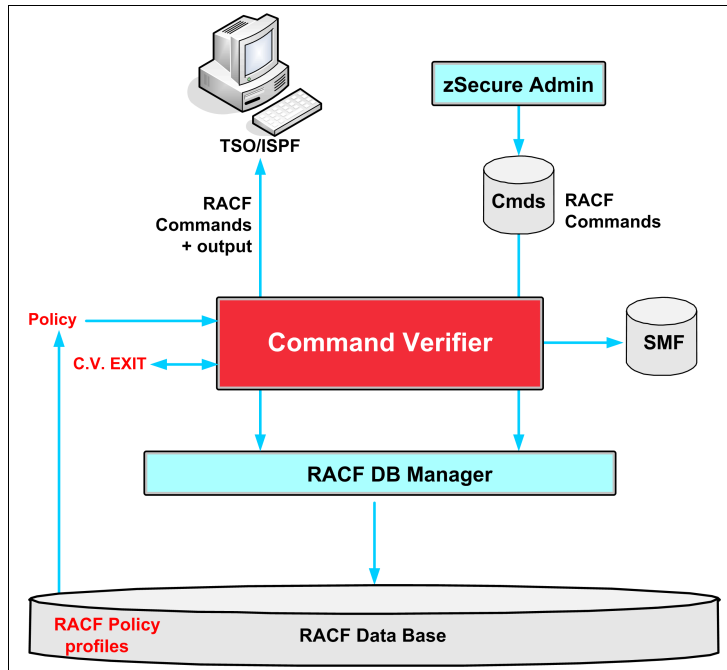


Figure 7-1 zSecure Command Verifier data flow

7.2 Controlling RACF commands

RACF command control is implemented using RACF profiles prefixed with the string “C4R.” In the class XFACILIT, they are referred to as *policy profiles*. The RACF class used may be altered during product installation; however, throughout this book, we continue to refer to the default supplied class of XFACILIT.

The format and significance of qualifiers in the RACF profiles depend on the type of RACF command being processed. You can think of this as the naming convention for Command Verifier profiles. We do not discuss the particulars of this naming convention here; refer to the *IBM Security zSecure Command Verifier User Reference Manual Version 1.12*, SC27-2779 for these details.

Before describing some example policy profiles, it is important to understand the nature of the Command Verifier implementation. In most cases, Command Verifier is implementing control over the *target value* of RACF commands rather than their *source value*.

The term *target value* represents the new value the administrator assigns whatever aspect of a RACF profile they are attempting to change, in general, the value of the target is not governed by RACF, while the choice of *source value* in any RACF command is entirely governed by RACF attributes and scope.

The source value is the RACF profile (user, group, or resource) that the administrator is attempting to change, using whatever RACF command. Without appropriate RACF authority, the administrator will not be permitted to issue the RACF command against the source profile.

zSecure Command Verifier adds the capability to control the target value in a RACF command. Thus, it is possible to control exactly what values an administrator may, for example, specify as the new owner of a RACF profile. This capability extends over all standard RACF commands and their operands, thereby allowing the site security policy to be enforced by profiles defined in RACF itself (prefixed with “C4R.”).

RACF *access levels* apply to policy profiles and determine Command Verifier behavior based on the specific profile involved and the access that the user issuing the commands has to the profile. Different access levels to Command Verifier profiles may imply different actions on the part of Command Verifier. In general, if no matching RACF profile is defined for a specific policy, Command Verifier takes no action and normal RACF processing of the command occurs.

Before going on to explain various ways in which zSecure Command Verifier can be used to prevent or control RACF changes, it is worth mentioning how to avoid being subject to the controls that Command Verifier implements. This is important to avoid a *lockout situation*, in which no RACF commands may be issued. This can happen if some administrator creates a Command Verifier related profile that has too wide or broad an effect, and thus prevents other administrators from being able to do any RACF work, even to correct the “bad” profile. We recommend that prior to implementing any Command Verifier controls, the profile C4R.EXEMPT should be defined in the class XFACILIT, and some user or users granted UPDATE access to this. These users will never be subject to Command Verifier controls, and thus may be used to recover from any potential RACF lockout situation; these might be your emergency or other “firecall” user IDs.

Note: The Command Verifier does not implement any controls over the RRVARY, RACLINK, RACDCERT, or RACPRIV commands.

7.2.1 Example policy profiles to control RACF changes

The following paragraphs describe and demonstrate a number of zSecure Command Verifier capabilities over RACF commands that are often used to enforce security standards and practices.

Preventing the granting of access to self

A common security requirement is to prevent an administrator granting themselves additional access within the system. Administrators especially should be subject to the same authorization process as the user community. The following command policy profiles may be used to implement this control.

First, define the command policy profiles in the class XFACILIT with an appropriate RACF owner for your system and a Universal Access of NONE. Universal Access of NONE is assumed throughout this book unless otherwise specified. We recommend the use of RACF installation data, shown with the profile name below, to provide a meaningful description of the function being implemented:

```
C4R.CONNECT.ID.*.=RACUID.** 'PREVENT AN ADMINISTRATOR FROM CONNECTING  
THEIR OWN USER ID TO A RACF GROUP'  
C4R.DATASET.ACL.=RACGPID.** 'PREVENT AN ADMINISTRATOR FROM PERFORMING A  
SELF PERMIT'  
C4R.DATASET.ACL.=RACUID.** 'PREVENT AN ADMINISTRATOR FROM PERFORMING A  
SELF PERMIT'
```

Grant access at UPDATE level to the users whom you do not want to be subject to these controls. The profiles above apply only to the DATASET class. You could use a generic in the second qualifier, but we advise caution when doing this to avoid locking yourself out of the ability to change C4R related profiles in general.

At this point, only users who are permitted these profiles may grant access to their own user IDs or groups of which they are a member. Even this may be considered unacceptable in a high security environment and you could consider having a batch only user ID authorized to perform these kinds of actions, removing access to these profiles from all other administrators.

Preventing the disablement of security controls

Administrators may accidentally or maliciously change critical security controls such as RACF System Settings, activating warning mode on profiles or defining user IDs with default passwords. zSecure Command Verifier can prevent these types of actions from exposing your critical assets to unauthorized use.

Default password values

To prevent the issuing of a RACF ADDUSER command with a default password, the following RACF profiles would be defined in the class XFACILIT:

```
C4R.USER.PASSWORD.=DFTGRP 'PREVENT USE OF DEFAULT GROUP AS PASSWORD'  
C4R.USER.PASSWORD.=USERID 'PREVENT USE OF USERID AS PASSWORD'
```

Again, you can permit users access to these profiles should you want the controls to be bypassed, although this is not recommended.

Warning mode on RACF profiles

To prevent an administrator activating RACF WARN mode on a profile, use the following Command Verifier profile in the class XFACILIT:

```
C4R.*.ATTR.WARNING.** 'PREVENT USE OF WARN MODE'
```

Once again, permit users whom you want to be able to use WARN mode.

Control RACF SETROPTS command use

To prevent both activation and de-activation of RACF classes, define the following profile in the class XFACILIT. Deny access to users whom you do not want to issue the (NO)CLASSACT operand of SETROPTS:

```
C4R.RACF.*.CLASSACT 'PREVENT SETR CLASSACT COMMAND'
```

Additionally, you may want to define the following profiles to control system settings with more granularity:

```
C4R.RACF.*.RACLIST 'PERMIT RACLIST ACCESS'  
C4R.RACF.** 'CONTROL ALL OTHER SETR COMMAND OPTIONS'  
C4R.RACF.LIST 'CONTROL SETR LIST COMMAND'
```

Users who need to issue SETR RACLIST REFRESH commands require access to the first profile. Deny all but the most senior security administrators access to the second profile above, and permit those who may need to issue the SETROPTS LIST command to the third.

7.3 Replacing user exits

Sometimes a RACF user exit is used to further control or restrict changes to the protection afforded critical operating system data sets or resources. A common example is the use of the *profile level indicator*, as specified with the ALTDSD or RALTER parameter LEVEL(nn). This field is usually not used for anything other than validation by a RACF exit of some kind.

7.3.1 Profile locking

zSecure Command Verifier implements a similar feature where policy profiles may be used to prevent any changes to a data set or resource profile. In the examples below, we control access to any data set profile beginning with the string "SYS1.":

```
C4R.DATASET.=NOCHANGE.SYS1.** 'PREVENT DATASET PROFILE CHANGES FOR
PROFILES WITH A LEVEL MATCHING THAT SPECIFIED IN PROFILE APPLDATA'
```

For this control to function, the XFACILIT profile must have a text string indicating a RACF level in its APPLDATA field, for example,
`appldata('LEVEL=99')`

Now we need to make the link between the LEVEL=99 in the profile appldata above and the actual RACF data set profiles we would like to prevent changes to:

```
ALTDSD 'SYS1.**' LEVEL(99)
```

Using the ALTDSD command, add LEVEL(99) to the RACF data set profiles for which you want this control to be active. This technique is also applicable to RACF *general resources*.

When someone tries to change the access list of the SYS1.RACF.** profile, the following message is displayed:

```
ICH408I USER(MCAIRNS ) GROUP(ZSEC001 ) NAME(MICHAEL CAIRNS      )
      C4R.DATASET.=NOCHANGE.SYS1.++ CL(XFACILIT)
      INSUFFICIENT ACCESS AUTHORITY
      FROM C4R.DATASET.=NOCHANGE.SYS1.** (G)
      ACCESS INTENT(UPDATE ) ACCESS ALLOWED(NONE )
***
```

If the command was issued via zSecure Admin, the panel shown in Figure 7-2 is also displayed.

```
COMMAND OUTPUT BROWSE ----- LINE 00000 Command failed
COMMAND ==> _                                SCROLL ==>
***** Top of Data *****
permit 'SYS1.**' id(ZSEC001) access(READ)
Management of locked profiles not allowed, command terminated
Command failed, return code 8 (decimal)
***** Bottom of Data *****
```

Figure 7-2 Output in zSecure Admin when zSecure Command Verifier locks the profile

7.3.2 Temporary system special

Another commonly implemented exit or system customization is a program to temporarily give system wide administrative rights in case of emergency. zSecure Command Verifier can replace this customization with a RACF controlled and monitored profile. There are two types of *temporary system special* implemented by Command Verifier: controlled or unconditional. The example below demonstrates the use of the controlled version. Define profiles in the class XFACILIT as follows:

```
C4R.PERMIT.=CTLSPEC 'ALLOW TEMP SPECIAL ON THE PERMIT COMMAND ONLY'  
C4R.DATASET.ACL.*.*.SYS1.** 'CONTROL CHANGES TO DATASETS STARTING WITH  
SYS1 - ANY USERID, ANY ACCESS LEVEL'
```

Granting access at UPDATE or a higher level to the two profiles above would allow the user to issue the PERMIT command against any “SYS1.” prefixed data set profile and grant access to any user ID or group, at any access level. You could also define a profile to allow similar PERMIT commands against general resource classes:

```
C4R.*.ACL.** 'CONTROL CHANGES TO ALL GENERAL RESOURCES NOT COVERED BY A  
MORE SPECIFIC PROFILE - ANY USERID, ANY ACCESS LEVEL'
```

Granting your systems programming staff access to these resources may be an acceptable alternative to system exits or even the retention of a system wide or group level special attribute. With this access, a systems programmer would have the capability to correct any access problem that was affecting system operations, but could not change anything else in RACF. There would also be an audit record generated for later review. You could also provide for a notification message to be sent to the system administrator whenever these authorities were exercised; see 7.5, “Alerting and action capabilities” on page 141 for more information.

7.4 Command audit trail feature

zSecure Command Verifier provides an auditing feature that alleviates the considerable effort usually necessary to discover when and how a RACF change was made. Often the task of determining when a RACF profile was modified, for example, access granted or removed, can only be performed by processing large amounts of the SMF audit trail data. The basic problem is that unless you have a good idea when the alteration was made, you have to process all SMF data from today going back in history until you find the date on which the change occurred.

This processing can represent a significant amount of data and processing, sometimes literally taking days or even weeks just to determine when and how an event occurred. The zSecure Command Verifier *command audit trail* feature can completely negate the requirement for this kind of processing and provide the answer to the *who did what and when* question within seconds instead of days.

The command audit trail feature works by storing a list of changes to a profile within the actual profile itself, so by displaying the profile, it is possible to see a list of changes the profile has undergone, that is, a history of the profile. This implies that when attempting to determine who changed the profile and when it was changed all you need do is examine the profile in question. Note this does not address the question of who gained access to a resource protected by the profile; that is still a question only answerable by reporting on SMF data.

7.4.1 Activating command audit trail for profiles

To activate command audit trail for a RACF class such as FACILITY, define the following profiles in the class XFACILIT:

```
C4R.FACILITY.=CMDAUD.=SEGMENT.**
C4R.FACILITY.=CMDAUD.=ATTR.**
C4R.FACILITY.=CMDAUD.=ACL.**
C4R.FACILITY.=CMDAUD.=MAINT.**
```

The double asterisk at the end of the profile may be replaced by a further qualified resource name if you do not want to maintain the command audit trail for all resources in the class.

To do the same for a RACF grouping class such as GXFACILI, define profiles similar to the above, substituting GXFACILI for FACILITY and use an additional profile with the fourth level qualifier of =MEMBER:

```
C4R.GXFACILI.=CMDAUD.=MEMBER.**
```

For auditing the connection or removal of a user from a group, the fourth qualifier =CONNECT is used in an XFACILIT class profile with the following format:

```
C4R.USER.=CMDAUD.=CONNECT.**
```

The double asterisk wildcard in this profile can be further qualified to limit the audit trail to a subset of RACF users.

7.4.2 Auditing changes failed by zSecure Command Verifier

Controls are now available for logging notifications related to command outcomes:

- ▶ DEFAULTS: Due to default policy
- ▶ MANDATORY: Due to mandatory policy
- ▶ SUPPRESS: Due to violated policy

These controls provide an additional level of command controlling activities.

7.4.3 Reviewing profile changes

To list changes made to a profile after a command audit trail is active, use the following TSO command:

```
C4RCATMN LIST CLASS(racf-class-name) PROFILE(racf-profile-name) GENERIC
```

The GENERIC keyword is optional and is relevant to fully qualified data set profiles in general. The profile name must be specified precisely; this is not a generic mask.

Depending on the executing users access to the command audit trail maintenance profile (see 7.4.4, “Maintaining the command audit trail information” on page 141), the audit trail data may also be displayed at the terminal as part of the output from standard RACF resource listing commands. For example, if the user has access at READ to C4R.*.=CMDAUD.=MAINT.** and issues the following RACF command, the audit trail data is also displayed (the display of usual RACF data has been trimmed here for brevity):

```
RLIST FACILITY STGADMIN.** ALL
```

```
Command Audit Trail for FACILITY STGADMIN.**
```

```
Attrib:  OWNER   Added on 08.091/20:10 by MCAIRNS
         UACC    Added on 08.091/20:10 by MCAIRNS
         Changed on 08.091/20:11 by MCAIRNS
Access:  ZSEC001 access READ on 08.091/20:10 by MCAIRNS
         WELLIE2 access Removed on 08.091/20:11 by MCAIRNS
```

7.4.4 Maintaining the command audit trail information

The C4RCATMN command has other operands and parameters used to manage the information maintained in the RACF profiles. You should be aware that the audit trail information stored in the RACF profiles will grow over time and cause growth in the overall size of your RACF database. This should be monitored during normal operations and occasionally you may decide to clean up older data in the command audit trail for individual profiles.

You should also understand that in general you do not grant access to command audit trail related profiles. The existence or not of a profile is sufficient to activate command auditing, and access as such is irrelevant. There is one exception to this guideline: Command audit trail profiles with the fourth qualifier of =MAINT are used to control who can or cannot maintain the command audit trail in a profile. You can use the following profile to control who can issue the C4RCATMN command:

```
C4R.*.=CMDAUD.=MAINT.** 'MAINTAIN AUDIT TRAIL DATA FOR ALL CLASSES AND ALL RESOURCES'
```

Access to this profile at CONTROL level is required to delete the audit trail data using the C4RCATMN command with the REMOVE operand. Access at UPDATE allows the use of the C4RCATMN command with the LIST operand to list the audit trail data. Access at READ allows a user who can list the RACF profile, using normal RACF commands, to also see the audit data, as shown in 7.4.3, “Reviewing profile changes” on page 140.

Considerations: For both Access List entries and Grouping class Member lists, a maximum of the previous 64 entries will be maintained.

7.5 Alerting and action capabilities

zSecure Command Verifier can also be used to generate some action both before and after any specific RACF command or replace a command entirely. For example, send messages using the TSO SEND command or issue other RACF commands whenever a specific RACF command is used. This function is implemented using a concept referred to as *Pre-Command* and *Post-Command* processing in zSecure Command Verifier.

To receive a message upon granting system special to any user, define the following profile in the class XFACILIT:

```
C4R.ALTUSER.=PSTCMD.SPECIAL 'SEND MESSAGE ON GRANTING OF SYSTEM SPECIAL'
```

Add the following appldata to the RACF profile:

```
appldata('SEND ''SPECIAL GRANTED TO &PROFILE BY'' USER(SECADM)')
```

A TSO SEND message will be sent to the nominated user, SECADM in this case, when system special authority is granted.

Another use of pre- and post-command processing is to add to or modify the processing involved. The profile below will cause the RACF group STORADM to be added to the access list of any new data set profile with the access level of ALTER (this is not recommended and should not be necessary if RACF storage administration controls are in place; this is just an example):

```
C4R.ADDSD.=PSTCMD.CLASS.DATASET  
appldata('PERMIT ''&PROFILE''' ID(STORADM) ACCESS(ALTER)')
```

Replacement of a command could be used to prevent the use of direct permits to user IDs and replace the target user ID with a predefined RACF group. Define the following profile in the class XFACILIT with a universal access of UPDATE:

```
C4R.PERMIT.=PRECMD.CLASS.SDSF
```

Add application data containing the desired CONNECT command to the profile:

```
appldata('CONNECT &ACLID GROUP(SDSF#ACLACC(1))')
```

Define the =REPLACE profile to the class XFACILIT, also with universal access of UPDATE:

```
C4R.PERMIT.=REPLACE.CLASS.SDSF
```

Now, if a user issues the following RACF PERMIT command, granting a user ID access to a profile in the class SDSF:

```
PERMIT profile CLASS(SDSF) ID(IBMUSER) ACCESS(READ)
```

the PERMIT command will be replaced by the =PRECMD established by the first profile above, as such:

```
CONNECT IBMUSER GROUP(SDSF#R)
```

This kind of processing assumes that you already have RACF groups established with a naming convention derived from the respective access levels they are used to grant. In this case, a group with the name SDSF#R was already defined and used to grant READ level access to resources in the class SDSF. The group name of SDSF#R was derived using a one character suffix extracted from the access level specified in the original PERMIT command. This could be considered an advanced use of the Command Verifier functions, but serves as an excellent example of the level of flexibility achievable using this sophisticated product.

Alerting: The alerting capabilities of Command Verifier are relatively simple and not a substitute for the comprehensive alerting available using Tivoli zSecure Alert.

7.6 Conclusion

In this chapter, we introduced the major functions of the zSecure Command Verifier. There are many more detailed capabilities in the product as supplied, and the possibility to further customize processing for advanced users by exploiting the product exit point. For complete information regarding these capabilities, refer to the *IBM Security zSecure Command Verifier User Reference Manual Version 1.12*, SC27-2779.

In demonstrating the capability to control RACF commands, provide an enhanced audit trail, reduce complex system customization requirements, and enhance system integrity and security through these capabilities, we are certain you can see the long term benefits of using zSecure Command Verifier in your environment.



IBM z/OS compliance enablers

In this chapter, we introduce the concept of the IBM Security zSecure Compliance Insight Manager Enablers for z/OS. We describe what they are, the different types of enablers, why you would want to use them, and how they work. Finally, we show you a few reports that showcase the value that the enablers provide to the organization with regards to auditing, monitoring, and compliance.

This chapter covers the following topics:

- ▶ What enablers are
- ▶ Currently available Enablers for z/OS
- ▶ Why you would want to use Enablers for z/OS
- ▶ How Enablers for z/OS work
- ▶ Sample screen captures of Enablers for z/OS in action

If you are looking for more detailed information about IBM Tivoli Security Information and Event Manager, including a practical z/OS integration scenario, you may want to refer to *IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager*, SG24-7530.

8.1 What enablers are

Any discussion about the Enablers for z/OS should begin with a brief introduction to the IBM Tivoli Security Information and Event Manager software offering.

Tivoli Security Information and Event Manager is a Windows Server based application that provides an easy-to-use *security compliance dashboard* to summarize disparate log files into one patented W7 viewing format. Tivoli Security Information and Event Manager can also take this data and run it through various compliance modules to provide reports against specific standards and regulations, such as the Sarbanes-Oxley Act (SOX) or the Payment Card Industry Data Security Standard (PCI DSS). Figure 8-1 shows a graphical representation of the Tivoli Security Information and Event Manager architecture and flow of data through the system.

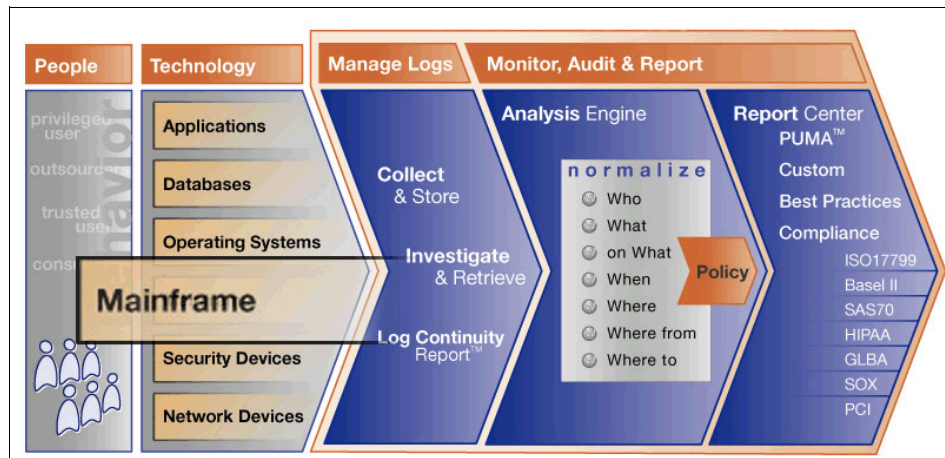


Figure 8-1 Security Information and Event Manager architecture

Also included in Tivoli Security Information and Event Manager are a Log Management tool, a Policy Generator tool, and a Scoping tool to manage views of collected data.

Security Information and Event Manager's data sources include:

- ▶ System data from IBM System z, IBM System i®, IBM AIX, Sun Solaris, HP-UX, Microsoft Windows, and Linux.
- ▶ Application audit trails gathered from files or database tables.
- ▶ Database data from IBM DB2, IBM Informix® Dynamic Server, and Sybase ACE for enhanced auditing capabilities.
- ▶ Security device logs through syslog and SNMP.

To assist with establishing a full audit and compliance infrastructure, Tivoli Security Information and Event Manager has also been integrated into many IBM Tivoli identity and access management products. These include IBM Tivoli Federated Identity Manager and IBM Tivoli Directory Server.

The IBM Security zSecure Compliance Insight Manager Enablers for z/OS are the software components that retrieve the various log sources and required information from the audited systems. In the case of z/OS, the enabler code is installed directly on the mainframe. From there, it interacts with zSecure Audit to collect the SMF data and sends this back to the Tivoli Security Information and Event Manager server installation for analysis at the server.

8.2 Currently available Enablers for z/OS

A little confusion: The integration component that is needed between zSecure and Tivoli Security Information and Event Manager is still called IBM Tivoli Compliance Insight Manager Enabler for z/OS. Be aware that IBM Tivoli Compliance Insight Manager was renamed to IBM Tivoli Security Information and Event Manager, but the zSecure Enablers still bear the old name. This change, however, does not impact functionality.

Tivoli Security Information and Event Manager currently reports on the following type of z/OS data:

- ▶ z/OS event and configuration data
- ▶ RACF events and user information including group membership
- ▶ CA-ACF2 events and user information
- ▶ CA-Top Secret (TSS) events
- ▶ DB2 events

8.3 Why you would want to use Enablers for z/OS

Tivoli Security Information and Event Manager, using its W7 methodology, normalizes audit and compliance data from various platforms into a reliable, integrated browser based view. Additionally, it provides sophisticated analysis and reporting tools.

The Compliance Insight Manager Enabler for z/OS works through zSecure Audit to collect data from the mainframe platform and securely transfer this data to the main Tivoli Security Information and Event Manager server. Once at the server, Tivoli Security Information and Event Manager combines mainframe data with information from other operating systems, applications, and databases. After the information has been processed, Tivoli Security Information and Event Manager can be used to:

- ▶ Provide collect history and log continuity reports to show what you are monitoring and that you are doing so continuously. This kind of report demonstrating collection evidence and procedures for exceptions is often required for audit purposes.
- ▶ Represent z/OS technical information in a more readily understood format using W7 categorization, expressing data and access patterns in English language terms. It can also present exception or unusual event information only for less technical auditors or generic security administrators to note for further investigation.
- ▶ Monitor the actions of internal, privileged users with high access rights. It can also create policies to define exceptions or unexpected behaviors, such as tracking activity to distinguish test system changes from updates to production systems.
- ▶ Quickly review your compliance posture and identify potential compliance issues through the Tivoli Security Information and Event Manager dashboards and reporting tools. This helps reduce the cost, remediation effort, and exposure of any non-compliant behavior discovered in the system logs.

8.4 How Enablers for z/OS work

While we refer to a Compliance Insight Manager Enabler for z/OS as one component, it is in fact the collaboration of a number of distinct software elements that provide the end to end data process of passing data to the Tivoli Security Information and Event Manager server for analysis. Let us now take a closer look at these distinct components.

8.4.1 Enabler

The Compliance Insight Manager Enabler for z/OS is the ordering term for the event source connection between Tivoli Security Information and Event Manager and zSecure Audit. Note that you do not require a licensed copy of zSecure Audit to be installed on your z/OS. However, certain components that make up a zSecure Audit installation will be installed, if not present already, as part of the enabler installation.

As mentioned in 8.2, “Currently available Enablers for z/OS” on page 147, there are different Enablers for z/OS that may be ordered, depending on the type of information to be monitored and audited.

8.4.2 Agent

After the Enabler for z/OS has been ordered and installed on z/OS, the next piece to run is the agent JCL. This agent is the primary UNIX System Services process and interacts with the Tivoli Security Information and Event Manager server over a TCP/IP network connection. It receives requests from the Tivoli Security Information and Event Manager server known as *collect requests*, and in response gathers and transmits the required data back to the Tivoli Security Information and Event Manager server. The agent UNIX System Services process initiates the actuator processes, specifically the User Information Source (UIS) collect process and the Event Source (ES) collect process.

8.4.3 Actuator

The actuators are secondary processes started by the agent UNIX System Services primary process. These processes do the actual work of invoking zSecure Audit to collect the data.

8.4.4 The data

The actuators collect data from several sources:

- ▶ SMF records from either SYS1.MANxx data sets or from SMF dump data sets; SMF record types for z/OS, RACF, CA-ACF2, CA-Top Secret (TSS), DB2 and UNIX System Services are predefined. Additional data may be gathered by modifying supplied exit code samples using the CARLa language.
- ▶ CKFREEZE and the security database provide the z/OS system information, such as user IDs, name, connect groups, user attributes, and z/OS Trusted Computing Base standards such as APF, linklist, and parmlib settings.

8.4.5 The process

Using the Tivoli Security Information and Event Manager Management Console, the z/OS machine to be audited and the z/OS event sources to be collected are defined. Part of this definition includes setting schedules for gathering and processing this data.

At the scheduled collect interval, Tivoli Security Information and Event Manager sends a collect request to the agent currently running on z/OS. The agent, in turn, initiates an actuator process to collect the desired data.

After the collection has completed successfully, the collected data is compressed and encrypted for transmission back to the Tivoli Security Information and Event Manager server. The Tivoli Security Information and Event Manager server uses the term *chunk* to refer to the results of any specific collect request. The chunk of data returned is centrally stored on the Tivoli Security Information and Event Manager server for subsequent processing. This processing is referred to as the *load schedule* for the data. Following the specified load schedule, one or more chunks of data is processed by the Tivoli Security Information and Event Manager server for representation in the W7 format and additional reporting capabilities.

At the point where the load has completed successfully, the data is now ready for viewing and reporting.

8.4.6 At a glance

Figure 8-2 represents the data flow from z/OS to Tivoli Security Information and Event Manager through zSecure Audit and the Compliance Insight Manager Enabler for z/OS, actuator, and agent processes.

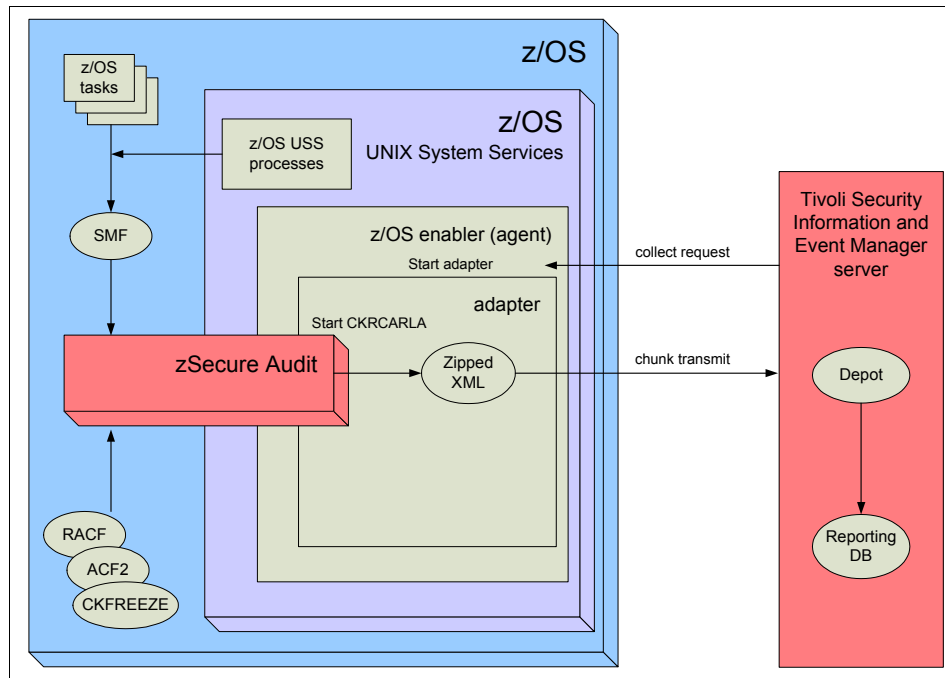


Figure 8-2 How Compliance Insight Manager gets the mainframe data

8.5 Sample screen captures of Enablers for z/OS in action

The Tivoli Security Information and Event Manager Portal is a web-based application that provides the tools to review logging and event information. The Portal presents four different tools to assist with audit and compliance:

- ▶ Compliance Dashboard: A reporting tool allowing drill down access to the data.
- ▶ Log Manager: A reporting tool for log management.

- ▶ Policy Generator: A wizard that walks you through creating policies (policy and grouping rules).
- ▶ Scoping: A tool to manage views of different users of Tivoli Security Information and Event Manager to different sets of data.

We can see what is happening in respect to security on our mainframe system using the Compliance Dashboard.

The first window is the dashboard, showing the databases that are housed on this particular Tivoli Security Information and Event Manager server, what data they contain, the status of the last load, and when it took place. In our example shown in Figure 8-3, we highlight a database named simply GEM (for Generalized Event Model), as this is the database that contains the z/OS and DB2 data we are interested in.

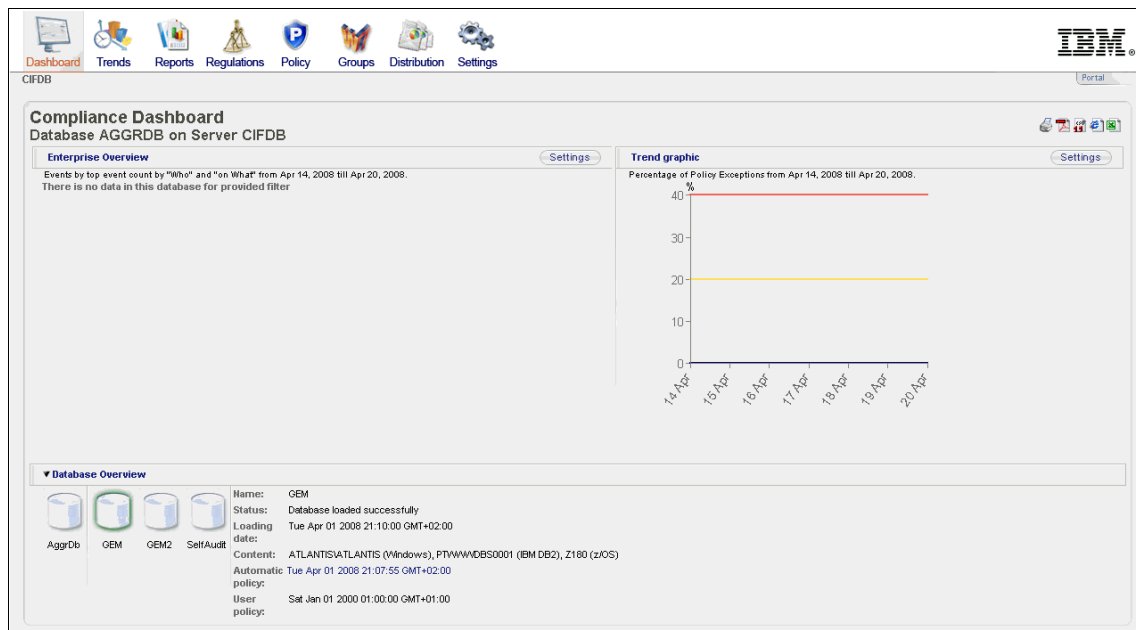


Figure 8-3 Compliance Dashboard

After double clicking the GEM database, the Compliance Dashboard presents more detailed information, such as where the data came from, when it was loaded, and how many events were logged. See the lower portion of the display in Figure 8-4. In the upper right hand section, this display also shows the number of Policy Exceptions, Special Attention Events, and Failures.

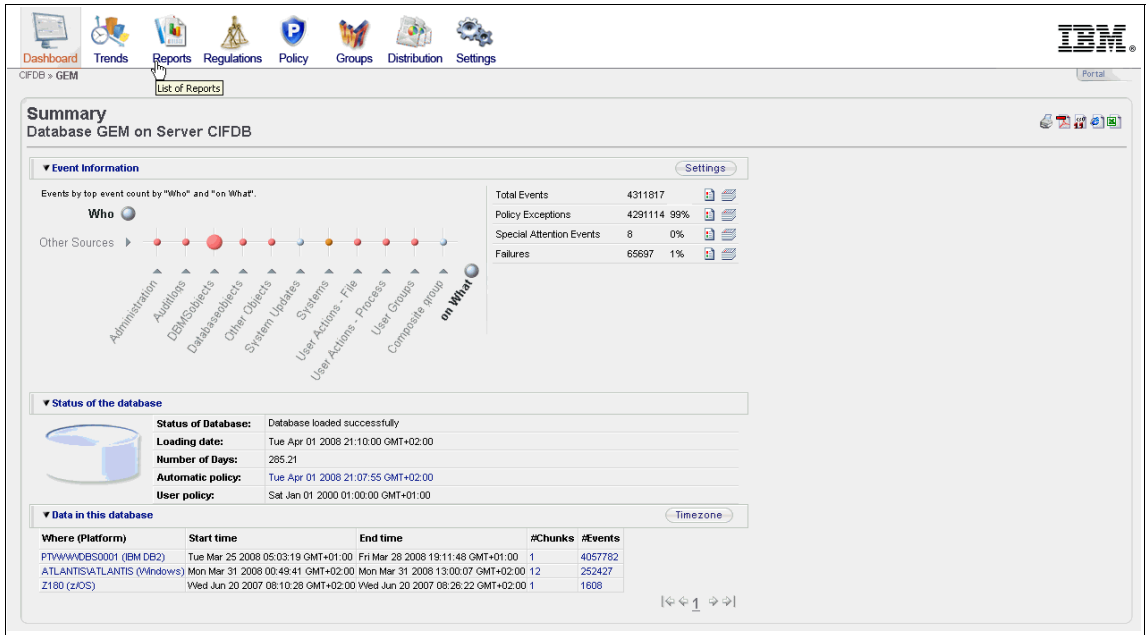


Figure 8-4 More detailed view of GEM database contents

We investigate the Special Attention Events by clicking **Reports** at the top of the window and then choosing **Events by type** from the list of reports available. The display shown in Figure 8-5 categorizes the types of events, the number of events recorded, and how many of those had policy exceptions, special attention events, or failures.

Event type	#Events	#Pol. Exp.	#Spec. Att	#Fail.
Convert : Auditlog / Success	167	167	0	0
Delete : Auditlog / Success	4	4	4	0
Inspect : Dbinstance / Success	271	271	0	0
Modify : Dblogger / Success	68	68	0	0
Verify : User / Failure	65332	65332	0	65332
Authenticate : User / Success	20626	0	0	0
Askexecute : Dbroutine / Success	435062	435062	0	0
Access : Dbmsobject / Success	3470161	3470161	0	0
Read : Name / Success	1823	1823	0	0
List : Groups / Success	276	276	0	0
Askaccess : Dbinstance / Success	1673	1673	0	0
Connect : Database / Success	5111	5111	0	0
Execute : Dbpackage / Success	19012	19012	0	0
Update : Datable / Success	1838	1838	0	0
Select : Datable / Success	1857	1857	0	0
Verify : User / Success	14924	14924	0	0
Askconnect : Database / Success	14945	14945	0	0
Backup : Database / Success	3	3	0	0
Backup : Database / Failure	1	1	0	1
Askexecute : Datable / Success	2960	2960	0	0
Insert : Datable / Success	486	486	0	0
Select : Dbview / Success	407	407	0	0
Configure : System / Success	3	3	3	0
Inspect : Databasespace / Success	144	144	0	0
Read : Databasespace / Success	621	621	0	0

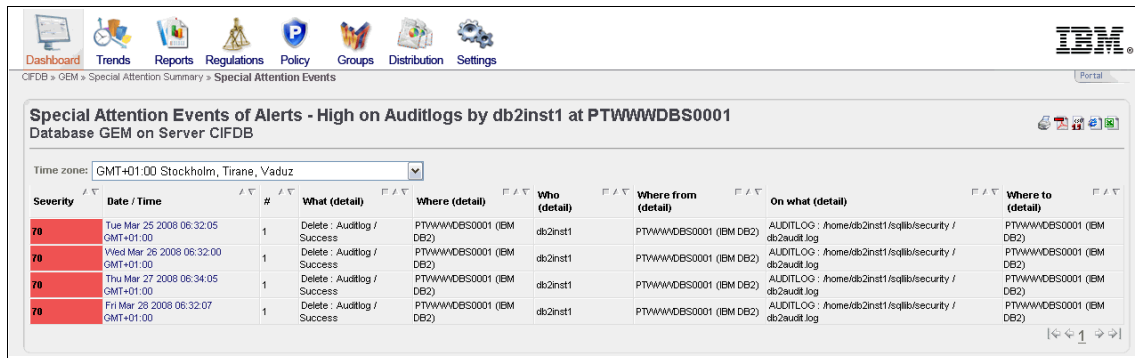
Figure 8-5 GEM events listed by type

Rather than continuing to drill down through the Reports option, we return to the initial GEM database summary view (Figure 8-4 on page 153) and choose the **Special Attention Events** link from here. We see in Figure 8-6 that several DB2 and some z/OS events were flagged with high (red) severity warnings.

Severity	Logon name	Where (Platform)	On What (Object group)	What (Event group)	#SpecAtt
70	db2inst1	PTWWWW/DBS0001 (IBM DB2)	Auditlogs	Alerts - High	4
70	db2inst1	PTWWWW/DBS0001 (IBM DB2)	Auditlogs	Alerts	4
20	fernando	PTWWWW/DBS0001 (IBM DB2)	Systems	Alerts - Low	3
20	fernando	PTWWWW/DBS0001 (IBM DB2)	Systems	Alerts	3
30	.CRMBFT1	Z180 (z/OS)	Other Objects	Access via user privilege	1
30	.CRMBFT1	Z180 (z/OS)	Other Objects	Read Data	1

Figure 8-6 Complete list of Special Attention Events

Looking at the first of these events, which have four high alerts on auditlogs, by double clicking the number 4 produces a table listing the four events and their W7 details, as shown in Figure 8-7.



CFDB > GEM > Special Attention Summary > Special Attention Events

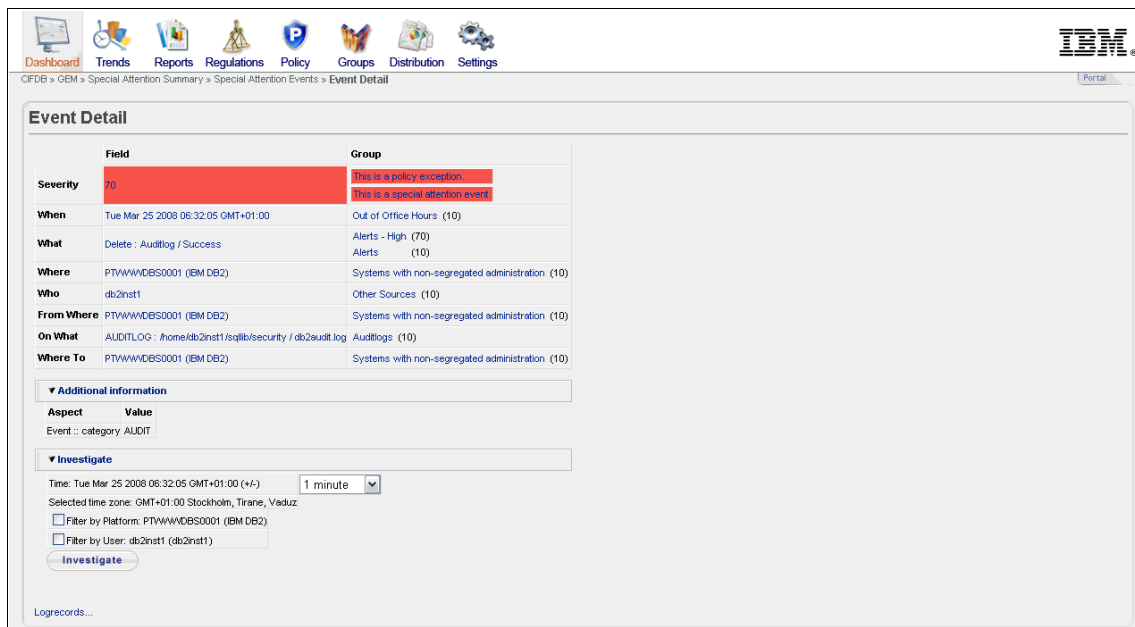
Special Attention Events of Alerts - High on Auditlogs by db2inst1 at PTWWWDBS0001
Database GEM on Server CIFDB

Time zone: GMT+01:00 Stockholm, Tirane, Vaduz

Severity	Date / Time	#	What (detail)	Where (detail)	Who (detail)	Where from (detail)	On what (detail)	Where to (detail)
70	Tue Mar 25 2008 06:32:05 GMT+01:00	1	Delete : Auditlog / Success	PTWWWDBS0001 (IBM DB2)	db2inst1	PTWWWDBS0001 (IBM DB2)	AUDITLOG : /home/db2inst1/sqllib/security / db2audit.log	PTWWWDBS0001 (IBM DB2)
70	Wed Mar 26 2008 06:32:00 GMT+01:00	1	Delete : Auditlog / Success	PTWWWDBS0001 (IBM DB2)	db2inst1	PTWWWDBS0001 (IBM DB2)	AUDITLOG : /home/db2inst1/sqllib/security / db2audit.log	PTWWWDBS0001 (IBM DB2)
70	Thu Mar 27 2008 06:34:05 GMT+01:00	1	Delete : Auditlog / Success	PTWWWDBS0001 (IBM DB2)	db2inst1	PTWWWDBS0001 (IBM DB2)	AUDITLOG : /home/db2inst1/sqllib/security / db2audit.log	PTWWWDBS0001 (IBM DB2)
70	Fri Mar 28 2008 06:32:07 GMT+01:00	1	Delete : Auditlog / Success	PTWWWDBS0001 (IBM DB2)	db2inst1	PTWWWDBS0001 (IBM DB2)	AUDITLOG : /home/db2inst1/sqllib/security / db2audit.log	PTWWWDBS0001 (IBM DB2)

Figure 8-7 Detailed information about Special Attention Events - High Alerts

By double-clicking the Date/Time field for a particular event instance, we can drill down even further to see the full event details, as shown in Figure 8-8.



CFDB > GEM > Special Attention Summary > Special Attention Events > Event Detail

Event Detail

Field	Group
Severity	70
When	Tue Mar 25 2008 06:32:05 GMT+01:00
What	Delete : Auditlog / Success
Where	PTWWWDBS0001 (IBM DB2)
Who	db2inst1
From Where	PTWWWDBS0001 (IBM DB2)
On What	AUDITLOG : /home/db2inst1/sqllib/security / db2audit.log
Where To	PTWWWDBS0001 (IBM DB2)

Additional information

Aspect Value

Event :: category AUDIT

Investigate

Time: Tue Mar 25 2008 06:32:05 GMT+01:00 (+/-) 1 minute

Selected time zone: GMT+01:00 Stockholm, Tirane, Vaduz

☐ Filter by Platform: PTWWWDBS0001 (IBM DB2)

☐ Filter by User: db2inst1 (db2inst1)

[Investigate](#)

Logrecords...

Figure 8-8 Detailed event information

Although this detailed information shows several minor (10) alerts, we are really interested in what flagged this event as Special Attention (70). An auditor, not knowing our policies, may wonder why this was flagged. By clicking the red box containing the words “This is a special attention event”, a window appears that explains that this event actually represented the deletion of audit logs, obviously a noteworthy event. See Figure 8-9 for this explanatory window.

Dashboard

Trends

Reports

Regulations

Policy

Groups

Distribution

Settings

IBM

CFDB > GEM > Special Attention Summary > Special Attention Events > Event Detail > Explanation of Severity

Portal

Explanation of Severity

Previous

This event is a Special Attention Event and a Policy Exception. It matches an Attention Rule and does not comply with the Security Policy, because it does not match any of the Policy Rules.

The event is given the higher of the Policy Exception Severity and the Special Attention Severity:

Policy Exception Severity is determined from Group Significance.

For every event, IBM Tivoli Compliance Insight Manager shows into which W7 groups it falls.

When you configured IBM Tivoli Compliance Insight Manager you gave each W7 group a significance (see the numbers in brackets).

The most significant group (Eventtype group "Alerts - High" in this case) determines the Policy Exception Severity.

event:

When group : Period	What group : Event type	Where group : Platform	Who group : Source	From Where group : Origin	On What group : Object	Where To group : Target
Tue Mar 25 2008 06:30:00 GMT+01:00	Delete : Auditlog / Success	PTVWWDBS0001 (IBM DB2)	db2inst1	PTVWWDBS0001 (IBM DB2)	AUDITLOG : /home/db2inst1/sqllib/security / db2audit.log	PTVWWDBS0001 (IBM DB2)

groups:

When group : Period group	What group : Eventtype group	Where group : Platform group	Who group : Source group	From Where group : Origin group	On What group : Object group	Where To group : Target group
Out of Office Hours (10)	Alerts - High (70) Alerts (10)	Systems with non-segregated administration	(10) Other Sources (10)	Systems with non-segregated administration (10)	Auditlogs (10)	Systems with non-segregated administration (10)

Policy Exceptions have a severity (in the range 10-99), equal to the Group Significance.

In this case, Eventtype group "Alerts - High" is the most significant W7 group, with significance 70.

Therefore, the Policy Exception Severity is 70.

Special Attention Severity is determined from Attention Rule Severity.

When you configured IBM Tivoli Compliance Insight Manager, you defined the Attention Rules and their severity.

Attention Rules have a severity in the range 10-99.

This is a Special Attention Event, because it matches the rule:

Who (Source group)	What (Event group)	When (Period group)	Where (Platform group)	On What (Object group)	fromWhere (Origin group)	WhereTo (Target group)	Description	Severity
ANY	Alerts - High	_ANY_	_ANY_	_ANY_	_ANY_	_ANY_	Requires immediate attention	70

The severity defined by the Attention Rule is 70, therefore, the Special Attention Severity is 70.

Event Severity is the higher of the Policy Exception Severity (70) and the Special Attention Severity (70), so the Event Severity is 70.

Figure 8-9 Explanation of High Alert Special Attention Event

In Figure 8-6 on page 154, we also noticed some flagged z/OS events. By drilling down on the “Access via user privilege” event report and clicking the explanation box, we are informed that this event matches a rule we established at a severity of 30. See Figure 8-10 for this explanatory window.

Dashboard

Trends

Reports

Regulations

Policy

Groups

Distribution

Settings

IBM

CFDB > GEM > Special Attention Events > Event Detail > Explanation of Severity

Portal

Explanation of Severity

Previous

This event is a Special Attention Event and a Policy Exception. It matches an Attention Rule and does not comply with the Security Policy, because it does not match any of the Policy Rules.

The event is given the higher of the Policy Exception Severity and the Special Attention Severity:

Policy Exception Severity is determined from Group Significance.

For every event, IBM Tivoli Compliance Insight Manager shows info which W7 groups it falls.

When you configured IBM Tivoli Compliance Insight Manager you gave each W7 group a significance (see the numbers in brackets).

The most significant group (Period group "Out of Office Hours", Eventtype group "Access via user privilege", Eventtype group "Read Data", Platform group "Systems with non-segregated administration", Source group "Other Sources", Origin group "Systems with non-segregated administration", Object group "Other Objects", Target group "Systems with non-segregated administration" in this case) determines the Policy Exception Severity.

event:

When group : Period	What group : Event type	Where group : Platform	Who group : Source	From Where group : Origin	On What group : Object	Where To group : Target
Wed Jun 20 2007 08:15:00 GMT+02:00 Read : File / UserPriv	Z180 (z/OS)	CRMBFT1	Z180 (z/OS)	SAF.DATASET : - / CICSDS.CICSTS31.SDFHLOAD	Z180 (z/OS)	

groups:

When group : Period group	What group : Eventtype group	Where group : Platform group	Who group : Source group	From Where group : Origin group	On What group : Object group	Where To group : Target group
Out of Office Hours (10)	Access via user privilege Read Data (10)	Systems with non-segregated administration (10)	Other Sources (10)	Systems with non-segregated administration (10)	Other Objects (10)	Systems with non-segregated administration (10)

Policy Exceptions have a severity (in the range 10-99), equal to the Group Significance.

In this case, Period group "Out of Office Hours", Eventtype group "Access via user privilege", Eventtype group "Read Data", Platform group "Systems with non-segregated administration", Source group "Other Sources", Origin group "Systems with non-segregated administration", Object group "Other Objects", Target group "Systems with non-segregated administration" is the most significant W7 group, with significance 10.

Therefore, the Policy Exception Severity is 10.

Special Attention Severity is determined from Attention Rule Severity.

When you configured IBM Tivoli Compliance Insight Manager, you defined the Attention Rules and their severity.

Attention Rules have a severity in the range 10-99.

This is a Special Attention Event, because it matches the rule:

Who (Source group)	What (Event group)	When (Period group)	Where (Platform group)	On What (Object group)	fromWhere (Origin group)	WhereTo (Target group)	Description	Severity
ANY	Access via user privilege	_ANY_	_ANY_	_ANY_	_ANY_	_ANY_	Success through a user attribute or privilege that allows access in spite of a security rule saying the user has no access	30

The severity defined by the Attention Rule is 30, therefore, the Special Attention Severity is 30.

Event Severity is the higher of the Policy Exception Severity (10) and the Special Attention Severity (30), so the Event Severity is 30.

Figure 8-10 Explanation of z/OS Special Attention Event

Next, we want to provide collection evidence to the auditors. We do this by clicking the **Portal** icon (top right hand corner) to get out of the Compliance Dashboard and select the **Log Manager** component of Tivoli Security Information and Event Manager. By clicking the Continuity icon on the next page, we are presented with a display of how often and when logs were collected and loaded, as shown in Figure 8-11.

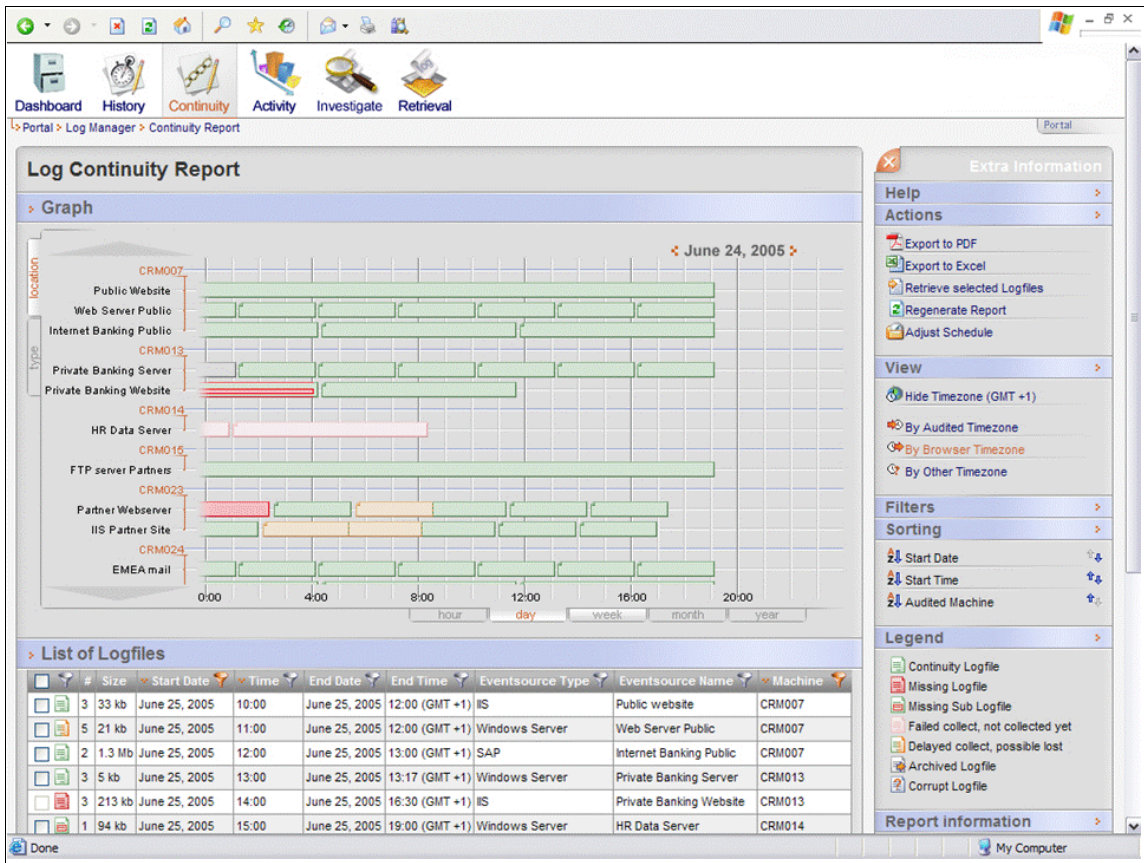


Figure 8-11 Providing collection evidence to the auditors

8.6 Conclusion

In this chapter, we showed you the power, flexibility, and advantages of using the IBM Security zSecure Compliance Insight Manager Enabler for z/OS in combination with zSecure Audit. Automated auditing and compliance reporting for mainframe data just got easier.

To find more detailed information about IBM Tivoli Security Information and Event Manager, including a practical z/OS integration scenario, you may want to refer to *IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager*, SG24-7530.

The next chapter discusses the zSecure CICS Toolkit.



IBM Security zSecure CICS Toolkit

In this chapter, we describe the functions and features of the Security zSecure CICS Toolkit. We have already provided an overview of this product in 2.4, “zSecure CICS Toolkit” on page 18, which we encourage you to review prior to reading this chapter to familiarize yourself with this component. The zSecure CICS Toolkit allows users to perform mainframe administrative tasks from within a CICS environment, freeing up native RACF resources. It provides a CICS interface to administer RACF. It has no interaction with the other products in the zSecure suite.

This chapter covers the following topics:

- ▶ zSecure CICS Toolkit architecture
- ▶ Command interface usage
- ▶ Application programming interface usage
- ▶ Conclusion

9.1 zSecure CICS Toolkit architecture

In this section, we describe the architecture for zSecure CICS Toolkit. There are two architecture diagrams to explain the data flow for both the command interface and the application programming interface (API). This provides a deeper insight as to how zSecure CICS Toolkit can be used for both RACF administration and externalizing of security in your CICS application.

9.1.1 Command interface architecture

In Figure 9-1, we show the data flow for the command interface using a password reset scenario. We have expanded on the flow diagram after Figure 9-1.

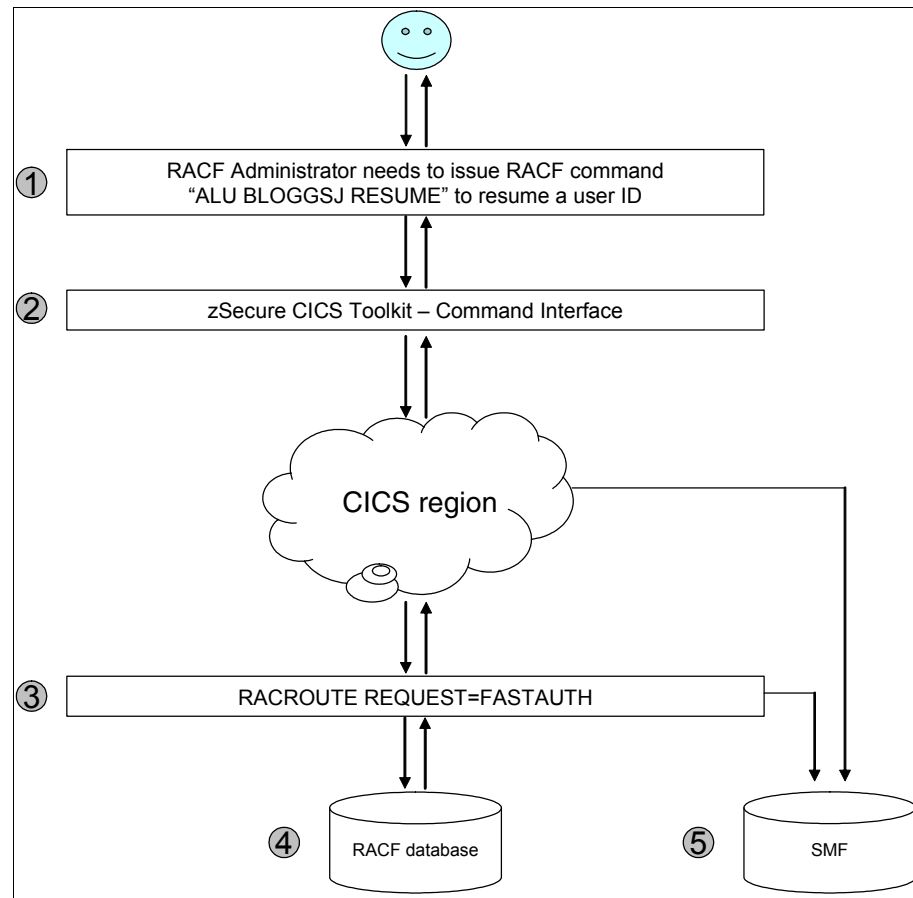


Figure 9-1 zSecure CICS Toolkit command interface data flow

1. A RACF administrator needs to restore a user ID in RACF because it is currently revoked.
2. The RACF administrator signs on to a CICS application with the RACF user ID and password and invokes the transaction RTMM to display the command interface. The administrator proceeds to enter the user ID that requires a restore.
3. zSecure CICS Toolkit issues a RACROUTE REQUEST=FASTAUTH to establish whether the RACF administrator is authorized to execute the ALTUSER function through checking the resource profile TOOLKIT.AUSR. We also check to see whether the affected user ID is within the RACF administrator's scope through checking the resource profile AUSR.dfltgrp (where dfltgrp = affected user's default group).
4. RACF returns a return code, which the CICS application will use to make a decision as to whether the RACF administrator is authorized to perform the requested action. If access is allowed, the ALTUSER function is issued.
5. SMF type 80 record is written, to log successful or failed access (provided that the correct audit settings are in place).

9.1.2 Application programming interface architecture

In this section, we describe the data flow for the application programming interface to specifically cover a resource access checking scenario using RACF as the external security manager (ESM). We have expanded on the flow diagram after Figure 9-2.

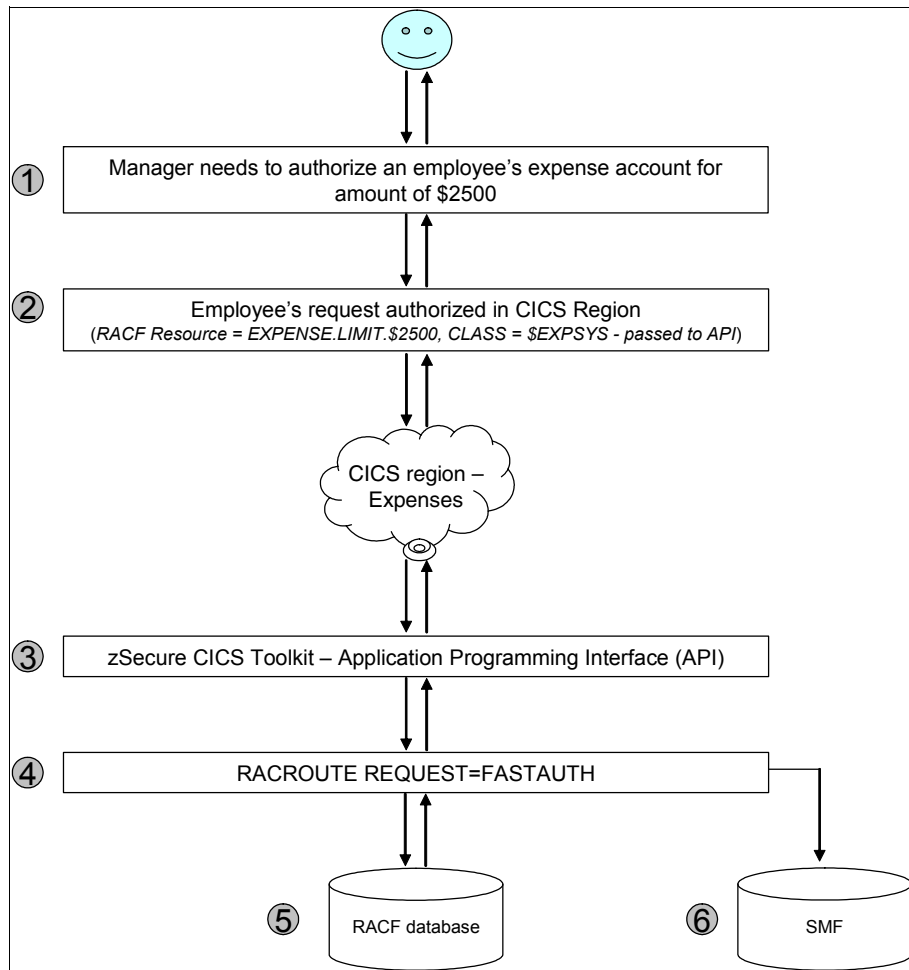


Figure 9-2 zSecure CICS Toolkit application programming interface data flow

1. In this scenario, a company's expense system is based on a CICS application and a manager needs to log on to this application to review an employee's expense account.

2. The Manager authorizes the employee's request in the CICS application. In this case, an installation defined RACF class containing RACF resources will be checked to establish whether the manager can authorize the request for the expense amount claimed by the employee.
3. The zSecure CICS Toolkit API is invoked by the CICS application, which passes certain parameters to it in a COMMAREA.
4. The API issues a RACROUTE REQUEST=FASTAUTH to determine whether the manager has access to the RACF resource.
5. RACF returns a return code, which the CICS application will use to make a decision as to whether the manager can authorize the request.
6. SMF type 80 record is written, to log successful or failed access (provided that correct audit settings are in place).

This particular scenario demonstrates how you can use the API to externalize your application security to RACF. Many companies still use internal application security, such as OP-ID, to perform security checking. This can result in an impact to the maintenance of separate tables or databases, security issues around the protection of these tables or databases, issues with segregation of duties, lack of auditing, and inflexible security and audit reporting.

Further information about using the application programming interface (API) can be found in the *zSecure CICS Toolkit User's Guide*, SC23-6551.

9.2 Command interface usage

As described in 2.4, “zSecure CICS Toolkit” on page 18, you can issue RACF commands from CICS using the zSecure CICS Toolkit from either the IBM supplied command interface or customized panels. For a full list of supported commands (which includes USRDATA management), refer to the *IBM Security zSecure CICS Toolkit User Guide Version 1.12*, SC27-2780.

Additional support information: zSecure CICS Toolkit does not support the RACDCERT, RACPRIV, RVARY, or SETROPTS commands.

You can control RACF commands that a RACF administrator is allowed to issue by defining and permitting access to TOOLKIT prefixed RACF profiles. zSecure CICS Toolkit uses RACF profiles in the class TCICSTRN unless otherwise noted. You may use profiles in the grouping class GCICSTRN with Toolkit resources. Using these profiles, you can add further granular control, for example, management of segments (TSO, OMVS, and so on).

zSecure CICS Toolkit also allows you to stipulate which profiles are within an administrator's scope. For example, if you wanted to allow a RACF administrator to issue the ALTUSER command for user's that have a default group of HRDEPT and PAYROLL, you can define sample profiles, as shown in Table 9-1.

Table 9-1 Sample RACF profiles for zSecure CICS Toolkit

RACF profile	Description
TOOLKIT.AUSR	Issue ALTUSER command.
AUSR.HRDEPT	Issue ALTUSER command for users with a default group of HRDEPT.
AUSR.PAYROLL	Issue ALTUSER command for users with a default group of PAYROLL.

A comprehensive set of sample RACF profiles for zSecure CICS Toolkit can be found in the ACQTSAMP library.

9.2.1 Customized panels

zSecure CICS Toolkit allows you to customize panels. This is achieved by customizing the BMS files (panel maps), which can be found in the ACQTSAMP library.

9.3 Application programming interface usage

As described in 2.4.2, "Application programming interface" on page 20, the zSecure CICS Toolkit API allows a user to interface with RACF from a CICS application. The API gives you additional features and flexibility that cannot be found in the CICS EXEC SECURITY function.

Note: No special knowledge of RACF or the RACF database format is required and the application(s) do not need to run APF authorized.

Using the API is a simple procedure that only requires the CICS application program to call the interface module and pass certain parameters to it in the COMMAREA.

9.3.1 Using the API for resource authorization checking

The zSecure CICS Toolkit API is commonly used to allow CICS applications to perform resource authorization checking in RACF. This is the best practice method of access control rather than using internal security mechanisms, such as referencing a DB2 table or VSAM file to look up authorization levels. We have already described the benefits of using RACF as the external security manager in 2.4.2, “Application programming interface” on page 20.

It should be noted that the API allows you to perform one or more resource authorization checks. For example, if you need to build a menu for a user when they log on to a CICS application, you can perform the required number of authorization checks in one call to construct the menu. This is a powerful control feature that helps enforce strong controls and auditing within your application by utilizing the security features of RACF.

A sample program CQTXAPIR is available that demonstrates how the API may be used to perform resource authorization checks. This can be found in the SCQTSAMP library.

An additional feature of the API is the ability to verify a user ID / password for another user other than the current session ID. Note that this VERIFY function does not perform a sign on for the user. An example of where you might use the VERIFY function is to enforce segregation of duties. Suppose you have a CICS application that allows you to raise checks. Your organization would not want to be in a situation where a person can request and authorize a large financial transaction without the appropriate review and authorization by a different person. A solution that some organizations adopt to enforce adequate segregation of duties is documented in the following scenario:

1. A clerk inputs a request into a CICS application to raise a check for the sum of \$5000 to settle a claim.
2. A message is returned to the clerk's panel informing them that approval is required to proceed with the transaction. A pop-up window also appears asking for the supervisor's user ID and password.
3. The supervisor indicates their authorization of the request by entering their own user ID and password on the clerk's panel.
4. If the supervisor's password is validated by the VERIFY function, the transaction continues and the check for \$5000 is raised.
5. The request is completed and the CICS application stores an audit trail containing the user ID who authorized the request, captured from the VERIFY process.

The zSecure CICS Toolkit API also allows you to perform a third-party authorization check through a RACROUTE REQUEST=AUTH to enable the CICS application to specify a user ID (other than the signed on user ID) during resource authorization checking. An example of where this may prove useful is for a simple workflow application. Before you read on, refer back to Figure 9-2 on page 164 to refresh your memory on the original scenario for access control. The following steps show you how workflow can be generated using a third-party authorization check:

1. When the employee originally submitted their expense account for review, they were requested to enter the user ID of a person who can review and authorize their expenses. This information is collected to generate workflow.
2. The employee entered their manager's user ID, which resulted in this user ID being used during a resource authorization check to verify access to RACF profile EXPENSE.LIMIT.\$2500 in class \$EXPSYS. This is referred to as a third-party authorization check.
3. The manager's user ID has access to this RACF profile, which resulted in confirmation to the employee that:
 - a. The specified manager can authorize an expense of up to \$2500.
 - b. Their request has been submitted and sent to the specified manager for review.

9.3.2 Using the API for RACF administration

Suppose your installation has developed (or plans to develop) an application for the purpose of allowing the help desk to manage password resets. You can use the zSecure CICS Toolkit API to execute RACF functions in addition to performing the appropriate resource authorization checks that are documented in 9.2, "Command interface usage" on page 165. This could replace current methods for interfacing with RACF, such as triggering batch jobs or adding RACF commands to a flat file for execution by your job scheduler or other methods.

We have provided an example of a CICS application, as shown in Figure 9-3 on page 169, which was developed to enable password resets to be managed by the help desk. In this example, the company has used the zSecure CICS Toolkit API to interface with RACF.

Necessary resources: IBM does not supply the code to accompany these panels. These panels are examples, which have been taken from a company that has used the zSecure CICS Toolkit API as part of their in-house developed user provisioning system.

If your organization has a requirement to develop a CICS application using the zSecure CICS Toolkit API, at a minimum you will require a developer who has Cobol/CICS programming skills. A RACF administrator will also be required to assist with designing and implementing the appropriate RACF controls.

In Figure 9-3, we show the main menu for the help desk, which is dynamically built based on the user's access to resources in RACF. We have used the zSecure CICS Toolkit API to perform multiple resource authorization checks in one call.

```
          Delft Transportation Authority - Services for Helpdesk

          1 Password - Password Reset / Resume / Revoke
          X Exit      - Exit

          Select option . . . _

          * Please contact SYSSEC for Technical Support *

          F1 = Help   F3 = Exit  Enter = Proceed
```

Figure 9-3 Sample password reset menu

In Figure 9-4, we use the zSecure CICS Toolkit API to:

1. Validate that the user ID entered by the operator exists in RACF.
2. The user ID is within the administrators scope.

```
      Delft Transportation Authority - Password Services

Enter the user ID to Reset / Resume / Revoke . . . J&MIE


      * Please contact SYSSEC for Technical Support *

F1 = Help   F3 = Exit   Enter = Proceed
```

Figure 9-4 Sample password reset menu to collect a user ID

In Figure 9-5, we use the zSecure CICS Toolkit API to:

1. Extract information from the RACF database to display information about the affected user ID to the Help desk operator.
2. Check that the Help desk operator is authorized to issue the ALTUSER function.
3. Execute the ALTUSER function.

```

Delft Transportation Authority - Password Services

Action for user ID JAMIE (JAMIE CARRINGTON)

Default group . . HRDEPT
Owner group . . . HRDEPT
Revoked . . . . . Yes

Reset User . . . ____ New password . . ____ Confirm . . ____

Resume only . . . ____

Revoke only . . . ____

          * Please contact SYSSEC for Technical Support *

F1 = Help   F3 = Exit   Enter = Proceed
```

Figure 9-5 Sample password reset panel to select desired action

In Figure 9-6, the CICS application processes the response from the zSecure CICS Toolkit API and provides the Help desk operator with confirmation of the result from the ALTUSER function.

```
Delft Transportation Authority - Password Services

Confirmed action for user ID JAMIE (JAMIE CARRINGTON)

Status . . . The password was successfully reset for the user ID


* Please contact SYSSEC for Technical Support *

F1 = Help   F3 = Exit   Enter = Proceed
```

Figure 9-6 Sample password reset panel to show result from previous action

For further information about the API, refer to the *IBM Security zSecure CICS Toolkit User Guide Version 1.12, SC27-2780*.

9.4 Conclusion

zSecure CICS Toolkit is a powerful product that can help an organization to perform effective and controlled RACF administration from CICS. It can also help to significantly improve application access controls and auditing for both new and well established CICS applications. Many organizations have found that weaknesses in application access controls and auditing can be rectified through the implementation of zSecure CICS Toolkit so that applications can utilize the robust and long established z/OS security system, RACF.

We recommend that zSecure Admin is also deployed within the organization to enable the central RACF administration team to perform end to end RACF administration and reporting. It should be noted that some of the functions and features available in zSecure Admin are not available in zSecure CICS Toolkit. Further information about zSecure Admin can be found in 2.2, “zSecure Admin” on page 14.



Planning for deployment

In this chapter, we describe the business, planning, and technical aspects of a deployment project with zSecure and we outline the items for consideration when planning for a deployment. We do not attempt to address all situations or solution scopes, but the purpose is to provide a framework for crafting project plans, deliverable checklists, and environmental installation and configuration considerations for a successful engagement.

This chapter covers the following topics:

- ▶ Services engagement preparation
- ▶ Solution descriptions
- ▶ Services engagement overview
- ▶ Conclusion

10.1 Services engagement preparation

Defining the scope of a project to be started is a critical path for success on a zSecure project. After the scope of the project is clearly defined and well documented with the appropriate project controls in place, the successful execution and delivery of the project is likely.

To help position the type of services engagement that is to be designed, it is critically important to determine the External Security Manager (ESM) that is currently deployed. After the ESM is known, be it RACF, CA-ACF2, or CA-Top Secret, then the engagement design and approach can be provided appropriately.

10.1.1 Implementation skills

When planning an engagement, one of the first items to be decided is who should be on the team and what are the required roles needed for the duration of the project. The identification of the appropriate architecture and engineering skills are cornerstones for a successful project.

We now describe the key skill areas that are needed for a zSecure engagement. These areas should help you identify the required skills for your particular situation. Depending on the components that will be installed, only some of these skills may be needed:

- ▶ System Programmer
 - Performs software download and SMP/E install
 - Has access to and a good understanding of TCP/IP and associated tasks
 - SMTP
 - SNMP
 - Possesses general UNIX (UNIX System Services) knowledge and authorities, along with file systems access
 - Understands JES, Procs, JCL, ISPF, TSO, and so on
- ▶ CICS support technician
 - Runs zSecure CICS Toolkit install
- ▶ The Security Architect understands the following:
 - RACF design
 - RACF Remote Sharing Facility (RRSF)
 - Security with z/OS UNIX System Services

- DB2 security with RACF
- CICS transactions
- RACF in a basic or parallel sysplex
- RACF program control
- Backup and recovery of the RACF database
- Network Job Entry (NJE) security controls
- ▶ Security Administrator
 - Determines which RACF functions to use
 - Identifies the level of RACF protection
 - Identifies which resources RACF has to protect
 - Defines administrative structures and users
- ▶ Security Auditor
 - Understands auditing parameters and requirements
- ▶ Windows/Tivoli Security Information and Event Manager owner
(required if deploying Compliance Insight Manager Enablers for z/OS)
 - Understands real-time alerts and audit capabilities

10.1.2 Available resources

The following lists of education classes, library reference materials, and product documentation highlight the resources that are available during your planning and design phase of a zSecure deployment.

Education plans

The following courses are available for skill development and for client education. In order to locate education availability in your geography visit the following IBM Global training site:

<http://www.ibm.com/software/tivoli/education/>

- ▶ IBM Tivoli zSecure Admin Basic Administration and Reporting
- ▶ IBM Tivoli zSecure RACF Management Workshop
- ▶ IBM Tivoli zSecure RACF and SMF Auditing
- ▶ IBM Tivoli zSecure Auditing and Reporting Language (CARLa) - 3 day class
- ▶ z/OS UNIX System Services Security Overview - 1 day class

IBM Security zSecure Suite library

The following list of zSecure library materials are available on the IBM Security zSecure Information Center, which is accessible from the zSecure support web pages. For licensed documentation, follow the Licensed Publications link on the zSecure Information Center web page. A good starting point to access this documentation can be found at:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp>

All zSecure documents are located under the product category *zSecure for z/OS* and *zSecure for z/VM* in the left hand pane:

- ▶ *IBM Security zSecure Release Information Version 1.12*
- ▶ *IBM Security zSecure Suite: CARLa-Driven Components Installation and Deployment Guide Version 1.12, SC27-2772*
- ▶ *IBM Security zSecure Messages Guide Version 1.12, SC27-2783*
- ▶ *IBM Security zSecure Audit for RACF Getting Started Guide Version 1.12, GI11-9355*
- ▶ *IBM zSecure Audit for ACF2 Getting Started Guide Version 1.12, GI11-9356*
- ▶ *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12, LC27-2773*
- ▶ *IBM Security zSecure Audit for ACF2 User Reference Manual Version 1.12, LC27-2774*
- ▶ *IBM Security zSecure Audit for Top Secret Version 1.12, LC27-2775*
- ▶ *IBM Security zSecure Alert User Reference Manual Version 1.12, SC27-2776*
- ▶ *IBM Security zSecure CICS Toolkit User Guide Version 1.12, SC27-2780*
- ▶ *IBM Security zSecure Command Verifier User Reference Manual Version 1.12, SC27-2779*
- ▶ *IBM Tivoli zSecure Visual Server Manual Version 1.11, SC23-6549*
- ▶ *IBM zSecure Visual Client Manual Version 1.12, SC27-2778*

10.2 Solution descriptions

When planning for a zSecure engagement, the scope of your requirements need to be analyzed in detail and a plan drafted to assure the appropriate and adequate coverage of the requirements.

During this phase, it is critically important to detail the aspects of the engagement proposal because a lack of a clear, detailed description can lead to a misunderstanding of the technical requirements and mismatched expectations. When detailed descriptions are prepared and planned, it will become clear early in the project where your expectations may differ from the implementation proposal.

In some cases a combination of solution proposals is desired so a composite solution can be drafted and proposed to you for acceptance.

The amount of time and detail invested in this phase of the project can directly affect the cost and sizing efforts produced for your zSecure project.

Next we take a look at the different possible solution approaches:

- ▶ 10.2.1, “Audit and compliance solution” on page 177
- ▶ 10.2.2, “Administration solution” on page 178
- ▶ 10.2.3, “Monitoring solution” on page 178
- ▶ 10.2.4, “Reporting solution” on page 179

10.2.1 Audit and compliance solution

The *audit and compliance solution* identifies key aspects of your needs in this area. You should help identify the areas that you want to target for the audit and which areas of the RACF database you want to perform a gap analysis on (for the compliance aspects of this solution). You must provide a detailed document outlining the security policy and requirements in order for the compliance solution (gap analysis) to be executed. If you do not provide a clear definition of the security policy and baselines to be audited against, the engagement cannot be successful.

The following audit points can be areas for consideration:

- ▶ Verify SETROPTS settings.
- ▶ Check Resource Class Settings (CDT).
- ▶ Audit user IDs.
- ▶ Review RACF groups.
- ▶ General resource and data set profiles.
- ▶ APF Library review.
- ▶ UNIX environment validation.
- ▶ ACL validation.

An analysis should be conducted from the above list of audit points (against the security policy you have provided), which identifies and recommends any actions necessary to become compliant or to be documented for future engagements. Scope content should be tightly controlled.

10.2.2 Administration solution

The *administration solution* identifies the key areas where the zSecure Administration product can be utilized to perform the day-to-day operational aspects of your RACF database management. The Administration solution can also identify the areas where the predefined reporting capabilities of zSecure Admin can be customized to produce a deeper analysis based upon your RACF database policy and goals determined in the planning phases of this solution. You must provide a clear document outlining the tasks required to be performed by the administrators, operators, and auditors so that the appropriate engagement solution can be proposed.

The following topics should be used to refine a proposal for a zSecure Administration solution:

- ▶ Current user administration practices
- ▶ Delegated administration possibilities for the future
- ▶ RACF group tree structure and possible enhancements
- ▶ RACF resource/profile naming conventions
- ▶ Role Based Access Control (RBAC) analysis and recommendations

An analysis should be conducted that will identify and recommend any actions necessary to create custom reports that can then be automated to allow for daily/weekly/monthly reporting to assist with compliance and management of the RACF database. Any recommendations should be documented for future engagements. Scope content must be tightly controlled.

10.2.3 Monitoring solution

The *monitoring solution* delivers the instrumentation and automation of alerts and design considerations to enable zSecure to participate in an enterprise monitoring solution (such as IBM Tivoli Security Operations Manager integration). Your monitoring architecture needs to be clearly documented for the implementation of the integration of zSecure capabilities so that a comprehensive solution can be valuable to the business and that your security requirements are met. Lack of an overall architecture and strategic direction for your monitoring needs can hinder the business value of this solution. Scope content must be tightly controlled during this phase of the project.

Your solution must consider the following elements

- ▶ Alerts
 - Definition of Alerts to be monitored
 - Setup of email destinations
 - Low/medium/high level of violation
- ▶ Changes to groups/roles
- ▶ Access to sensitive groups
- ▶ PADS compliance
- ▶ System Special, Operations, Auditor, and Group Specials

10.2.4 Reporting solution

The *reporting solution* delivers the automation and customization aspects of an engagement with zSecure. This solution area needs to be detailed and complete in the requirements definition. You should clearly document the baseline to be measured against for report generation. The information to be reported on needs to be clearly documented. Lack of detail in this phase of the solution can expose the completion criteria by not meeting your goals. During the definition phase of this solution, it will become clear what reporting can be accomplished within the scope of this project and which requirements need to be identified for future projects. Scope content must be tightly controlled.

- ▶ User and group reports to validate access rights
- ▶ Batch jobs identified
- ▶ Baseline reports
- ▶ CARLa customization based upon project plan definition

10.3 Services engagement overview

In order to successfully implement any of the aforementioned zSecure solutions, it is important to clearly identify the necessary tasks, the dependencies each task has on available resources, system outages, workload considerations, and funding availability. All these factors can influence a project and its timely completion and success.

The major phases of an engagement can be identified as follows:

- ▶ 10.3.1, “Executive assessment” on page 180
- ▶ 10.3.2, “Demonstrating the system setup” on page 180

- ▶ 10.3.3, “Analyzing solution tasks” on page 181
- ▶ 10.3.4, “Creating a contract” on page 181
- ▶ 10.3.5, “Defining solution tasks” on page 182
- ▶ 10.3.6, “Deployment tasks” on page 185

10.3.1 Executive assessment

During an *executive assessment*, your needs are discussed and documented. By interviewing the stakeholders and understanding their business requirements, a compelling business case can be prepared to help propose a solution offering that can deliver business value.

This is a business discussion, not a technical discussion (the technical aspects will be clearly identified and documented in later phases). The target audience for this part are the business owners, line-of-business executives, marketing and sales managers, and the IT managers. The most likely decision maker will be the business owner or line-of-business executive.

During this phase, together with the service provider, you will determine who will be involved in the project. This group of people will define what needs to be accomplished, when the solution can be deployed, what is mission-critical in their minds, and how this project will be funded. After these aspects of the business case are well understood, building a strong proposal will be attainable and you should be able to better understand the value of the proposal presented to you.

An effective executive assessment can demonstrate that the service provider has the right technology and the right people for their mission.

10.3.2 Demonstrating the system setup

Depending upon your needs, a product demonstration or proof-of-technology (POT) may be desired. This aspect of an engagement allows you to view the solution and actually have an opportunity to use and ask specific questions about the products and particular features of the products.

Determining the systems to be used, the products to install and the scripts to use to drive the demonstration should be tailored for your particular environment, based upon the overall assessment of what you are most interested in viewing or having demonstrated to you.

10.3.3 Analyzing solution tasks

This phase of the project involves looking at the proposed implementation (the details that are planned to be delivered for you) and figuring out at a high level what needs to be documented and described in a contract or statement-of-work (SOW).

The details in this phase need to be clear and concise, but allow for some design and implementation flexibility. The goal here is to have enough detail so that you and the service provider understand what is going to be done, how long it should take, and how much it will cost. To summarize this information requires planning and investigation for areas that are not well defined. Minimizing the risk is knowing the details and the requirements, without which the knowledge and experience of a project may become exposed.

10.3.4 Creating a contract

A contract or statement-of-work is a binding contractual agreement between you and your service provider that defines the engagement that is to be performed and the results you are expecting from the engagement. The contract should leave nothing in doubt.

A statement-of-work needs to include the following information:

- ▶ Executive summary of the solution, which is typically a short (less than a page) summary of the solution and its benefit. The service provider must specify any major restrictions of the implementation, such as:
 - The solution is only implemented for finance applications.
 - The solution will be implemented in phases.
- ▶ Solution description, which includes the major components and solution building blocks that will be implemented. It should cover the conceptual architecture of the solution and the solution scope in general. This description is aimed for technical personnel to understand the implementation scope.
- ▶ Assumptions, which lists all the assumptions that are to be used to prepare the contract and provide task estimation. Any deviation from the assumptions that are used will definitely impact the scope of the engagement and must be managed using a change management procedure. Typical changes would include cost changes or scope changes.
- ▶ Business Partner responsibilities, which lists all the responsibilities or major tasks that will be performed by the service provider to implement the solution.
- ▶ Client responsibilities, which lists all the responsibilities or items that you must provide for you or your team to perform the engagement.

- ▶ Staffing estimates, which lists the estimated personnel that should implement the solution.
- ▶ Project schedule and milestones, which describes the major steps, schedule, and achievement calendar that can be used to check the project progress.
- ▶ Testing methodology, which lists the test cases to ensure that the project implementation is successful.
- ▶ Deliverables, which provides tangible items that you will receive at the end of the engagement, including:
 - Machine installation
 - Documentation
 - Training
- ▶ Completion criteria indicates that the engagement is successfully completed. For most engagements, this is probably the most delicate to define. Completion criteria can be too general and you will most likely have difficulty signing off on acceptance. Alternatively, an inadequate completion criteria is often rejected initially because it leaves too much ambiguity. This is a balance that must be agreed upon prior to the beginning of a project.

10.3.5 Defining solution tasks

The key to a positive engagement is to identify the tasks that need to be completed and consider options adequately to assess impacts to the overall project scope and deliverables. The following outline identifies key areas to be considered when documenting the scope and details of the project. The outline is not meant to be all inclusive, but is to be used as a general planning guide to assist in the definition and the scope of the project.

- ▶ Installation
 - Distribution considerations.
 - Single or multiple z/OS images.
 - Can be installed on a *dead* system to allow for replication or connected (through catalog pointers).
 - Allows for data set names to differ.
 - Maintenance patch application.
 - Development systems.
 - Production systems.
 - Workload balancing.
 - Maintenance considerations.

- z/OS and z/VM require installation on both.
- ▶ Maintenance considerations
 - Patch application.
- ▶ Upgrade migration
 - Installation defined customization.
 - Data set name considerations.
 - Name conflicts for future releases.
 - User catalog for data sets.
 - Easy access from multiple z/OS images.
- ▶ Configuration
 - Distributed consideration for z/OS images.
 - Configuration information is represented in partitioned data sets that are not part of the installed software.
 - Image dependent data.

Dedicated configurations for special purposes, for example, Help Desk or Central Administrators (different levels of functions available).
- ▶ Sensitive data sets
 - Identification and security policy.
- ▶ Trusted Computing Base
 - Design considerations and security policy.
- ▶ APF authorization

The program object library containing zSecure needs to be APF authorized.
- ▶ Batch jobs and started tasks
 - ISPF transactions inherit the software location from the TSO session.
 - Configuration data set definitions.
 - z/OS image considerations.
 - Visual.
 - Compliance Insight Manager Enablers for z/OS.
- ▶ Setting up security for zSecure
 - APF authorized functions of zSecure protected by resources in the XFACILIT class.
 - Profile creation needed or equivalent resource rules.

- SAF calls used to configure menus and limit use of authorized functions. Located in the XFACILIT class.
- ▶ Restricted and unrestricted mode
 - Restricted.
Only data within the scope of the user can be displayed or reported.
 - Unrestricted.
All information in the RACF database can be displayed or reported (similar to RACF AUDITOR, but cannot change any global options or auditor settings). An unrestricted zSecure administrator can make RACF changes within the RACF scope that cannot be made by an Auditor, who can only administer Audit related settings.
- ▶ Establishing PADS access
 - Conditional access or PADS mode.
 - Requires definition of profiles in the program class and activating RACF program control.
May be set up as part of the implementation of UNIX System Services. PADS mode implies that all zSecure users are in restricted mode, unless this is overruled by having READ access on the CKR.READALL profile in the XFACILIT class.
- ▶ Change tracking
 - Monitors changes in system parameters and security settings against a baseline.
 - Single z/OS image or multiple images into a single view for centralized Change Tracking administration.
 - Interface to external change management systems, for example, IBM Information Management.
- ▶ Daily batch suite
 - Change tracking.
 - Audit reports.
 - CKFREEZE process.
 - UNLOAD process.

10.3.6 Deployment tasks

Before beginning your deployment, read *IBM Security zSecure Version 1.12 Quick Reference*, SC27-2785, which describes many of the considerations and steps to be performed for a deployment. A thorough and complete review of this book and subsequent study of the related materials regarding the specific engagement requirements should be fully understood and planned before beginning the actual deployment tasks.

The following steps should be completed for the deployment:

- ▶ SMP/E installation of the zSecure product.
- ▶ RACF security profiles to be defined/built for the product(s), including the class definitions.
- ▶ TCP/IP domain name resolution (DNS).
- ▶ Input sources identified for audit and customized to capture desired SMF records (for example, DB2 IFCIDs).
- ▶ Using a fresh CKFREEZE and UNLOAD.
- ▶ Verify the installation.
 - Checking the base ISPF interface functions and menu configuration
 - Checking the zSecure Collect function and the base batch operation of zSecure
 - Displaying reports
 - Checking CKGRACF
- ▶ Testing and validation.
- ▶ Completion and exit criteria met.

10.4 Conclusion

In summary, planning for deployment should explore the entire scope of the project and engagement being considered. Clear and accurate documentation is critical to the success of a project. The scope management of the project is the cornerstone to keeping a project on track and successful. Defining the scope of a project is a key contributor to the overall successful planning and delivery of an engagement.

You and your team, in concert with an adequate service provider, need to work closely together through every step outlined.



Part 2

Customer scenario

In this part, we discuss a business scenario for the Delft Transportation Authority (DTA), which is facing a merger with Tivoli Rome Airlines.



Delft Transport Authority

In this chapter, we introduce the company Delft Transport Authority. Delft Transport have been growing their business both organically and through strategic acquisitions. Their most recent acquisition is a company called Tivoli Rome Airlines (TRA).

We briefly discuss the company profile of Delft Transport, as well as their current IT architecture, corporate business vision, and objectives. We will then provide an overview of the acquisition project and an outline of the business challenges and strategic objectives for the merger of Delft Transport and Tivoli Rome Airlines IT systems.

Note: All company names and references to businesses used in this chapter are fictional. Any similarities to real corporations are completely coincidental.

This chapter covers the following topics:

- ▶ Delft Transport company profile
- ▶ Delft Transport IT security architecture
- ▶ Corporate business vision and objectives
- ▶ Acquisition project
- ▶ Conclusion

11.1 Delft Transport company profile

The IT department of Delft Transport are experienced and efficient. Having been through previous acquisitions, they have developed processes to assist in speeding the integration of disparate companies' systems into their own. Delft Transport's IT management are able to control expenses and operational costs efficiently using state of the art software and processes. These are essential ingredients to the companies' successful integration of previous acquisitions.

Their security management policies and procedures are well documented and integrated into business processes using as much automation as possible. The design of their RACF database is in line with accepted best practice, and they have been using the IBM Security zSecure Suite over many years to maintain best practice for security management. Regulatory compliance and audit readiness are automatic at Delft Transport, a result of having clear and documented processes for all change processes, including security and automatic exception reporting and monitoring in place.

With the acquisition, the IT staff at Delft Transport has an opportunity to review the current IT architecture in place at TRA. During this initial assessment and audit of TRA systems, it became apparent that the integration of TRA into the Delft Transport IT infrastructure was going to be somewhat of a challenge. In particular, the security management processes currently in use at TRA would not pass the strict audit guidelines in place already within Delft Transport.

Integrating any two business IT infrastructures presents significant challenges, but the gap between where TRA is in terms of IT security maturity and where Delft Transport desires them to be highlights some common challenges facing many businesses today.

Now, let us explore how Delft Transport begins the integration of TRA IT systems into their own by focusing on the RACF system, and the methodology used to address these common security challenges using the IBM Security zSecure Suite of products.

11.2 Delft Transport IT security architecture

Delft Transport's RACF database is on their System z server at Delft and TRA's RACF database resides on a System z server in Rome. There are no immediate plans to merge the two RACF databases into one, although this is a medium to long term goal.

Delft Transport currently has a comprehensive suite of daily batch jobs that provide the reporting files needed for historical and baseline analysis. Using these data sources, and live system information, they implement the following processes and controls in their own systems:

- ▶ Daily reporting to both administrators and data owners
- ▶ Real-time alerting of security related events
- ▶ Automated controls over changes to sensitive system settings
- ▶ Sensitive data sets access reporting
- ▶ Sensitive data set change tracking
- ▶ Trusted computing base validation
- ▶ Trusted user reporting
- ▶ Audit profile settings verification

The same processes and controls will be required for TRA systems as a prerequisite for successful integration. The following chapters of this book document the process of implementing these controls in a live system.

Contributing to Delft Transport's efficient RACF management is the use of a defined and controlled RACF group tree structure. Analysis of the group tree structure in use at TRA revealed serious deficiencies in administrative controls and segregation of access rights. Most users had a high level of unnecessary access.

Converting the TRA RACF group tree access structure into something that will satisfy the minimum audit requirements of Delft Transport is a major transformation. We will describe many of the steps required in a typical transformation of such a large scale and how we used zSecure to address them in the following chapters.

11.3 Corporate business vision and objectives

During the early phases of acquisition planning, several *tiger teams* of IT and business specialists from Delft Transport were selected to spend time at TRA facilities and evaluate their IT operations and environment. Teams were formed for Application Integration, Infrastructure Integration, and Business Process analysis. As a part of the Infrastructure Integration team, a security specialist was assigned to review TRA systems and estimate the effort required to integrate these with Delft Transport.

The initial business objective for security management is to bring TRA systems into line with Delft Transport security policy. Subsequent objectives include establishing an auditing and monitoring framework to match that of Delft Transport, followed by the implementation of structures and processes required to support a fully delegated security administration framework eventually.

After these initial integration objectives are complete, an internal audit of TRA security systems will be performed. The aim being, should the audit results be satisfactory, to give Delft Transport IT management the confidence and assurance that the required regulatory and compliance initiatives have been successfully deployed at TRA. Successful achievement of this goal will feed into overall governance and management programs for the entire IT integration, giving confidence that the eventual full integration of the two companies systems will not expose either to unnecessary business risk.

Over time, as the two businesses integrate both support systems and business processes more closely, opportunities to reduce duplication of effort by merging similar business units and IT functions will occur. Eventually, as has occurred with previous Delft Transport acquisitions, both companies will be running on identical platforms and infrastructures and total merging of all IT functions will occur, bringing significant cost savings.

Strategic business objectives delivered by the security integration we are about to describe include:

- ▶ Readiness for eventual merging of the two RACF databases into a single image.
- ▶ Positioning to exploit enterprise wide monitoring across both mainframe and distributed environments.
- ▶ Readiness to deploy additional identity and access management solutions.

Overall, taking the opportunity to make significant changes to the way Tivoli Rome Airlines systems are managed today, and doing this as part of an acquisition program of activities, is a brave step for any acquiring company. It would be easier to just leave the existing systems in place. Easier, but more costly in the long term, and exposing the combined group to greater business risk. TRA would be the weak link in the overall groups IT security, and thus the most likely point for introduction of security related risks.

The Delft Transport management and IT teams have done this before, though, and have the experience to do it again. Previous projects such as this have reaped good returns on the investment, and continued to assure management that the required IT governance and controls are in place across the entire enterprise.

11.4 Acquisition project

The overarching strategy that Delft Transport brings to all their acquisitions is one of aligning the acquired company to the Delft Transport established IT management processes and infrastructure configuration. This is usually done as early as possible after official acquisition. The aim of this being to ease the later processes of integrating the IT and business systems and processes into the combined business group.

The overall integration consists of three distinct stages.

- ▶ Alignment of acquired company with Delft Transport standards
- ▶ Physical and logical integration of business systems
- ▶ Final integration of all company functions

From the security management perspective, the most challenging parts of this process occur in the first stage. Thus, a project team is formed consisting of RACF and security specialists from Delft Transport to work with TRA security management to align their systems and processes with Delft Transport standards.

The team chosen to perform this work will be referred to as the RACF *tiger team*. They will implement a program of rapid improvement in TRA RACF configuration, auditing, monitoring, and reporting. The remainder of this book will describe exactly how they achieve these goals using IBM Security zSecure.

11.5 Conclusion

The challenges faced during a business acquisition today can be daunting. Delft Transport has considerable experience in this area, though, and approaches the acquisition of TRA with a solid project management framework that has proven its usefulness previously.

The RACF tiger team will bring their years of experience, using the IBM Security zSecure Suite of tools, to the TRA systems. They will quickly and safely migrate these systems to be in line with Delft Transport's well defined and managed corporate security architecture, policies, and practices. As part of this, a skills transfer will also occur as TRA staff are exposed to and learn from the modern processes that this security transformation brings.

In the following chapters, we describe the integration processes for IT security between the two companies. You will see how these same strategies and practices can be applied to the day-to-day operational aspects of your business.



Project requirements and design

In this chapter, we give an overview of the RACF Security Improvement Project in place at TRA as a result of their acquisition by Delft Transport.

The project is divided into three phases, each concerned with specific areas of functional improvement in security and audit management to make the TRA systems comply with Delft Transport security policy.

This chapter covers the following topics:

- ▶ Business requirements
- ▶ Functional requirements
- ▶ Design approach
- ▶ Implementation approach
- ▶ Conclusion

12.1 Business requirements

Using a globally accepted Risk Management framework and an organizational culture and structure with a focus on Group Security, Delft Transport had been successful in an aggressive expansion strategy over the last several years while introducing no additional risk profile to the group. This was part of Delft Transport's overall business strategy, and contributed significantly to their ability to gain competitively structured finance deals and other corporate benefits to assist their business growth initiatives.

The overall driving business requirements of the integration program were to:

- ▶ Minimize cost.
- ▶ Minimize both operational and financial risk.
- ▶ Ensure as smooth an integration as possible with no disruption to either businesses' capability to continue delivering service to customers.
- ▶ Ensure as tight an integration as possible with minimal duplication of business functions in the resulting organization.

It is obvious that IT has a key role to play in achieving some of these business outcomes. In order to achieve these aims, some stages of integration were defined:

1. Initial co-location of functionally similar business units to gain scaling cost benefits of using shared facilities.
2. Subsequent merging of similar business units into a shared, group wide resource. Candidates for this include such operational units as IT, HR, Group Management, and so on.
3. Final cross branding and collaboration of disparate business units from the two entitled to leverage new, group wide, business relationships.

As the focus of this book is on management of enterprise data center security, we will now discuss the issues that arise from stage 1, and in preparation for stage 2, that is, the security work required prior to collocation and merging of the Delft Transport and TRA IT infrastructure facilities.

12.2 Functional requirements

With the rapid expansion of non-mainframe based facilities in the TRA IT infrastructure over the previous decade, the mainframe environment had received a lower level of funding and management attention in general. In particular, security management of mainframe resources had been allowed to fall well behind what was now considered industry common practices.

Remediation of the RACF security was considered to be on the critical path for the entire integration project. The security controls that Delft Transport had been maintaining for many years were mandated by Delft Transport Security Management Group and monitored by their Internal Audit Group, who together controlled the company wide security policies and exemption management processes. If we could achieve the same standards in TRA as were currently in place at Delft, then the entire integration project could proceed with less overall risk, avoid the requirement for CEO or Board level exemption signoff, and likely avoid incurring additional costs to provide risk mitigation processes that might be mandated by Group Security.

The current Delft Transport Mainframe Security Policies and Procedures were used as a baseline against which to measure the acceptability of the TRA systems.

These guidelines mandated specific controls that must either exist or have documented and accepted exemption status, such as:

- ▶ All access to mainframe data is controlled by RACF definitions.
- ▶ All user IDs on the system are current and in use. No unused accounts are to be defined or remain.
- ▶ All high level privileges issued are appropriately monitored for abuse.
- ▶ Monitoring and escalation processes are in place for any use of high level privileges.
- ▶ High level privileges cannot be used to bypass security controls.
- ▶ All mainframe data must have a documented owner who is responsible for approving access to their data and reporting on this action.
- ▶ Strict separation of duties between systems programming, IT security management, and change control must be in place.
- ▶ Trusted system tasks and configuration settings are monitored for change, and system privileges are only granted when a documented requirement is available.

It was decided to aim for avoiding a required exemption status for any of these main security requirements, and thus to bring the above standards into practice at TRA prior to other integration work beginning.

In the following section, we discuss the approach decided upon by the tiger team to achieve these goals.

12.3 Design approach

The Delft Transport RACF team already had extensive practical experience in the use of the IBM Security zSecure Suite, having been a customer for many years and having recently migrated to the latest release: IBM Security zSecure V1.12. This software would be used during the project to clean up the TRA mainframe systems security configuration.

Rather than spend time in research, documentation, and project planning for this emergency rectification work, it was decided to take advantage of the automated analysis tools available in the IBM Security zSecure Suite. The team felt that using the reports and rectification tools provided by the suite would significantly accelerate their task. In some cases, these tools may provide an outcome that exceeds the stated objectives, and in others, meeting the objectives may still require extra effort. However, using zSecure, the team was assured that an acceptable baseline for best practice mainframe security could be more easily achieved in the tight time frame available to them.

It should be noted at this point that this scenario is rare in real businesses, and often delays to security improvement programs such as this are more influenced by political decisions and processes within a company than by purely technical issues. We were fortunate in writing this book to have no such political issues obstructing our ability to implement major security changes in a short amount of time.

For the remainder of this book, we will be using and demonstrating the practices employed by this mainframe security tiger team as they progress through the TRA RACF setup.

12.4 Implementation approach

The project was broken down into three distinct, yet possibly overlapping, phases. The main reason for these phases was the critical urgency of this overall task as a prerequisite to successful integration of the two enterprises. It was decided that, if necessary, more advanced security implementation suggested in the final project phases could be deferred should any complications be discovered early on and delay the entire project. Thus, the project phases became:

- ▶ Phase 1: Discovery and initial cleanup.
- ▶ Phase 2: Establish audit reporting and monitoring.
- ▶ Phase 3: Prepare for and establish a delegated security administration framework.

Phase one was defined as essentially setting up the zSecure suite of software and configuring it to gather the appropriate data about the TRA system. Where the software could provide automated or semi-automated cleanup of security definitions in a totally harmless manner, this too would be performed as part of this phase.

Phase two was defined as the establishment of automated audit and monitoring tasks designed to satisfy the requirements of the Internal Audit and Integration Management committees.

Phase three will implement delegation of security management to employees closer to the actual resources being secured, a prerequisite for this being both improved security controls and more detailed monitoring and audit reporting.

12.5 Conclusion

This chapter has now set the scene for a project involving massive RACF change in a short period of time. In the following chapter, we will describe the processes that our RACF tiger team used to migrate TRA from an essentially low level of security management in their mainframe environment to one where they had best practices implemented.



Implementation phase I

In this chapter, we describe phase one of the RACF improvement project for TRA. In this phase, we will be completing configuration and customization of the IBM Security zSecure Suite and describing the process of gathering data about the system configuration.

We will also be covering the implementation of several of the recommendations that the zSecure reports and analysis tools provide.

This chapter covers the following topics:

- ▶ Post systems programmer installation setup
- ▶ Running initial analysis reports
- ▶ Implementing initial improvements in system security posture
- ▶ Post implementation verification reports
- ▶ Conclusion

13.1 Post systems programmer installation setup

The TRA systems programmer has completed the installation of products from the zSecure suite. However, there are additional steps required by the security administrator to best utilize the tools.

Delft Transport’s tiger team made the following recommendations:

- ▶ Run a daily batch jobs to refresh the zSecure unload, CKFREEZE, and signature files, and to refresh CKG; the following screen captures, Figure 13-1 and Figure 13-2, show an example of a zSecure daily suite set up through the batch scheduling tool IBM Tivoli Workload Scheduler and what the suite would include.

```
----- LIST OF APPLICATIONS ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> CSR

Enter the CREATE command above to create a new application, or,
enter the GRAPH command above to view the list graphically, or,
enter any of the row commands below:
B - Browse, M - Modify, C - Copy, D - Delete,
P - Print, A - Calculate and print run days,
L - Modify LTP (external dependencies are not resolved)

Row Application Valid T S Owner id
cmd id text from date
- CKRDAILY zSecure Daily Maint 11/04/08 A A SYSPROG
***** Bottom of data *****
```

Figure 13-1 Sample daily zSecure scheduled application in IBM TWS

```
----- BROWSING OPERATIONS ----- Row 1 to 6 of 6
Command ==> _                               Scroll ==> CSR

Enter the PRED command above to include predecessors in this list, or,
enter the GRAPH command above to view operations graphically.
Enter the row command S to select the details of an operation.
Enter the row command J to browse the JCL

Application : CKRDAILY zSecure Daily Maint

Row Oper Duration Job name Operation text
cmd ws no. HH.MM.SS
**** NONR 001 00.00.01 START Start Daily Run
**** CPU1 002 00.01.00 CKRUNL Recreate zSecure Unload
**** CPU1 003 00.01.00 CKRFRE Recreate CKFREEZE File
**** CPU1 004 00.01.00 CKRSIG Create Signature GDG
**** CPU1 005 00.01.00 CKRCKG CKG Refresh
**** NONR 006 00.00.01 END End Daily Run
***** Bottom of data *****
```

Figure 13-2 Sample jobs within daily batch zSecure suite

- ▶ For the purposes of recoverability and auditability, the tiger team also recommends a daily run of IRRUT200 (a copy of the RACF database) and the creation of GDGs, as we describe in 13.2, “CKFREEZE, Signature, and UNLOAD generation data groups” on page 203.
- ▶ As TRA uses RACF as their ESM, and specific security measures are suggested, as described in 13.3, “RACF security for IBM zSecure” on page 208.

Along with these major tasks, the tiger team also verifies the final steps of the installation:

- ▶ Updating IFAPRDxx to enable the zSecure products
- ▶ Including SCKRLOAD in LNKLSTxx (Admin)
- ▶ Making SCKRLOAD APF authorized
- ▶ Adding CKGRACF to the TSO AUTHCMD authorized command table
- ▶ Adding CKGRACF to the ISPF TSO command table
- ▶ Copying the C2R\$PARM member into a JES2 proclib
- ▶ Ensuring that SMF exits IEFU83, IEFU84, and IEFU85 are enabled for SYSTEM and for all subsystems (Alert)
- ▶ Customizing TCP/IP for SNMP or SMTP use
- ▶ Schedule the provided batch job C2RJXRFR for timed RACF actions

13.2 CKFREEZE, Signature, and UNLOAD generation data groups

To enable TRA to use some of the functions within IBM zSecure, such as zSecure Audit, there are some input files that should be readily available.

As part of your daily security batch suite, you should aim to create the following input files:

- ▶ An unload of your RACF database, generated by the zSecure Admin unload utility
- ▶ A copy of your RACF database, generated by the IBM supplied RACF utility IRRUT200
- ▶ Dumped SMF data
- ▶ CKFREEZE file
- ▶ CKFREEZE file containing digital signatures

These daily files should be created as standard flat files using the supplied unload utilities. We recommend you also create these files as generation data groups (GDGs) so that you can quickly perform analysis and recovery actions using data captured over a period of days, weeks, and months. We recommend you define these GDGs as (default) input files in zSecure option SE.D.1. This section aims to show you how to create a GDG base, how to create new GDGs as part of your daily batch suite, and how to reference them in option SE.D.1.

You should aim to retain a minimum of seven daily GDGs for each of the standard flat files. As part of your weekly and monthly batch processing, you should aim to retain five weekly and 13 monthly GDGs respectively. Here are some of the reasons why creating these GDGs may prove valuable to your organization:

- ▶ You have accidentally deleted RACF profiles or you are in a system recovery situation and need to restore RACF profiles at a point in time.
- ▶ You are trying to resolve access issues for a user and need to analyze SMF records and RACF definitions at a point in time.
- ▶ Changes have been applied to the system and you need to compare the current configuration with the old configuration to make sure no security exposures were created.
- ▶ You need to run the Library Analysis feature in option AU.L.
- ▶ You are required to provide access control lists and audit trail evidence as part of a special investigation.
- ▶ An auditor needs to report on configuration settings or audit records at a point in time.
- ▶ You are implementing role based access controls or other security improvement programmes and need to analyze data over a long period of time.

There are many other reasons why you need to retain historical records relating to your security database(s), SMF data, and system configuration data. Security Analysts are normally required to act quickly to provide information or fix security issues. Thus, it is important to have historical information readily available with a sufficient retention period.

In Example 13-1 we show you how to create a GDG base. For the purpose of this example, we are creating a GDG base to retain up to seven daily CKFREEZE files (GDGs).

Example 13-1 JCL to define a GDG base

```
//*****
//* BUILD GDG
//*****
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
  DEF GDG (NAME(CKRU.DAILY.CKFREEZE) -
          LIMIT(7) -
          NOEMPTY -
          SCRATCH
/*
```

Note: If you want to create additional GDG bases for your RACF unloads, signature files, and dumped SMF, simply repeat lines 7 through to 10 and edit the repeated define statement.

Having created the GDG base, you should be able to see it from option 3.4 (DSLIS), as shown in Figure 13-3.

Menu Options View Utilities Compilers Help		
DSLIS - Data Sets Matching CKRU.DAILY.CKFREEZE		Row 1 of 7
Command ==> _____		Scroll ==> CSR
Command - Enter "/" to select action		Message
-----		Volume
CKRU.DAILY.CKFREEZE		??????

Figure 13-3 GDG base listed from DSLIS (option 3.4)

The next step is to create some JCLs for the purpose of defining a new CKFREEZE file as a GDG. The required JCL is shown in Example 13-2.

Example 13-2 JCL to create a new CKFREEZE file as a GDG

```
//*****
//* CREATE NEW CKFREEZE FILE AS GDG
//*****
//CKFREEZE EXEC PGM=CKFCOLL,REGION=64M
//STEPLIB DD DISP=SHR,DSN=CKR.SCKRLOAD
//SYSPRINT DD SYSOUT=*
```

```
//CKFREEZE DD DISP=(NEW,CATLG),VOL=SER=BH6ST5,
//          DSN=CKRU.DAILY.CKFREEZE(+1),SPACE=(CYL,(50,30))
//SYSIN    DD *
//
```

When the new CKFREEZE file is available, it should be added to zSecure option SE.D.1 as an input file. Figure 13-4 shows you how to do this for GDGs. Use the insert line command in option SE.D.1 to insert a new input file.

Note: The CKFREEZE file generated by zSecure Alert's collect started task should not be set up as a GDG. This file is purely for use by zSecure Alert, which does not need to reference historical CKFREEZE files.

Menu	Options	Info	Commands
zSecure Admin+Audit for RACF - Setup - Row 11 from 20			
Command ==> _____		Scroll ==> <u>CSR</u>	
Description test daily ckfreeze gdg input file			
Complex SC76			
RRSF node Local node for RRSF			
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: B		Type REFRESH to submit unload job.	
Data set or Unix file name		Type	NJE node
_ 'CKRU.DAILY.CKFREEZE(0)'		CKFREEZE	
***** Bottom of data *****			

Figure 13-4 Adding your GDG to option SE.D.1 as an input file

You can concatenate two GDG files in option SE.D.1. An example of when you will need to do this is for the zSecure Audit library analysis feature in option AU.L. You need to compare two signature files (GDGs) to find changes. Figure 13-5 provides an example of GDG concatenation. To add more than one file, you can use the I (insert) or R (repeat) line command.

```

Menu    Options    Info    Commands
-----
                                zSecure Admin+Audit for RACF - Setup - Row 8 from 20
Command ==> _____ Scroll ==> CSR

Description . . . . test daily signature gdg input file
Complex . . . . . SC76
RRSF node . . . . . Local node for RRSF

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: B                  Type REFRESH to submit unload job.

      Data set or Unix file name          Type          NJE node
_   'CKRU.DAILY.CKFREEZE.SIGNATUR(0)'    CKFREEZE
_   'CKRU.DAILY.CKFREEZE.SIGNATUR(-1)'   CKFREEZE
***** Bottom of data *****

```

Figure 13-5 Concatenate two GDGs as input

Having generated the input files, Figure 13-6 shows the minimum number of input files that should be available to zSecure Admin and zSecure Audit users.

```

Menu    Options    Info    Commands
-----
                                zSecure Admin+Audit for RACF - Setup - I Row 1 from 7
Command ==> _____ Scroll ==> CSR

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

      Description          Complex
_   Active backup RACF data base and live SMF data sets    SC76    selected
_   Active primary RACF data base                          SC76
_   test daily signature gdg input file                    SC76
_   test daily ckfreeze gdg input file                     SC76
_   test daily unload gdg input file                       SC76
_   test daily dumped smf gdg input file                   SC76
_   Active backup RACF data base                            SC76
***** Bottom of data *****

```

Figure 13-6 Selectable input files in option SE.1 or SETUP FILES

If you decide to adopt GDGs for some of your input files in option SE.D.1 batch programs that invoke the IBM zSecure cataloged procedure C2RC will continue to use the input file format as defined in members C2RI0IOC, C2RI0SMF, C2RI0UNL, C2RI1IOC, C2RI1SMF, and C2RI1UNL within SCKRPROC. We recommend you leave these members unmodified so that you continue to create and reference the standard input files whenever the cataloged procedure C2RC is used.

The GDGs are an enabler for performing analysis on historical data and are therefore a supplement to your existing set of input files.

13.3 RACF security for IBM zSecure

As part of the implementation of the IBM zSecure products on TRA's mainframe, there is a requirement to implement controls for the products to protect them against unauthorized use.

IBM zSecure allows you to protect menu options, line commands, and other functions using RACF profiles in the XFACILIT class.

Note: Your system programmer may have changed the default class of XFACILIT to a different class in the site module CKRSITE. In this section, we will continue to refer to class XFACILIT.

When installing IBM zSecure, we recommend that you set up some base RACF profiles to protect the various resources within the product. This should be a short term security implementation and we recommend you consult the product manuals to assist with extending RACF security protection for the installed products. In many organizations, a large number of people from various departments will use IBM zSecure. There are functions and features that should be restricted to security administrators and auditors. We highly recommend that you plan and implement the required RACF security to ensure that the most appropriate access is granted to users of the products.

Table 13-1 lists the base profiles we recommend you set up. Unless otherwise stated, READ access is required to the appropriate XFACILIT RACF profile.

Table 13-1 Recommended base XFACILIT profiles

Profile	Description
CKF.**	Catch-all for zSecure Collect functions.
CKG.**	Controls access to CKGRACF functions. <i>We recommend granting UACC(NONE).</i>
CKR.**	Catch-all profile. <i>We recommend the access control list be left empty for this profile.</i>
CKR.OPTION.**	Controls access to menu options for zSecure Admin, zSecure Audit, and zSecure Alert. <i>We recommend granting UACC(READ).</i>
CKR.OPTION.AU.**	Controls access to audit options.
CKR.OPTION.CO.**	Controls access to command options.
CKR.OPTION.EV.**	Controls access to event options.
CKR.OPTION.RA.**	Controls access to RACF administration options.
CKR.OPTION.RA.H.**	Controls access to help desk options.
CKR.OPTION.SE.A.**	Controls access to zSecure Alert options.
CKR.OPTION.SE.D.**	Protects the use of menu option SE.D, used for setting system defaults for IBM zSecure. <i>Access to this profile should be restricted to people who are responsible for configuring the product.</i>
CKR.ACTION.**	Controls access to line commands for zSecure Admin, zSecure Audit, and zSecure Alert. <i>We recommend granting UACC(READ).</i>
CKR.ACTION.CH.**	Controls access to actions on the exceptions overview with change tracking.
CKR.ACTION.CT.**	Controls access to actions on the system overview within change tracking.

Profile	Description
CKR.READALL	<p>Allows a user to run in unrestricted mode if read access is granted.</p> <p><i>You should refer to the IBM Tivoli zSecure Suite: CARLa-Driven Components Installation and Configuration Manual Version 1.9.1, SC23-6556 for further details about this resource.</i></p> <p><i>We recommend initially granting UACC(READ).</i></p>
CKR.CKRCARLA.APF	<p>Allow CKRCARLA to run in APF mode.</p> <p><i>We recommend granting UACC(NONE). Access should be granted only to the zSecure Alert started task ID.</i></p>

You can protect IBM zSecure programs using the PROGRAM class. We recommend you set up some base profiles, as shown in Table 13-2.

Table 13-2 Recommended base PROGRAM profiles

Profile	Description
CKRCARLA	<p>Program access for CKRCARLA.</p> <p>You must add your.prefix.SCKRLOAD to this profile as a member.</p>
C2P*	<p>Program access for zSecure Alert programs.</p> <p>You must add your.prefix.SCKRLOAD to this profile as a member.</p>

We also recommend that you protect IBM zSecure software data sets and your own installation defined data sets where you have stored:

- ▶ CARLa programs
- ▶ Reports
- ▶ zSecure unloads
- ▶ RACF database unloads
- ▶ Dumped SMF
- ▶ CKFREEZE files
- ▶ External files

Some of these data sets may contain sensitive information. These should have appropriate access control lists and auditing in place.

In Table 13-3, we recommend some base data set profiles to implement. Where you have APF authorized the load module libraries, the VERIFY SENSITIVE report in zSecure Audit will check to see if these libraries are properly protected:

Table 13-3 Recommended base data set profiles

Profile	Description
your.prefix.SCKRLOAD	IBM zSecure library containing load modules. <i>This library is normally APF authorized and update access should be restricted. Read access is required for IBM zSecure users.</i>
your.prefix.SC4RLNK	IBM zSecure library containing load modules for zSecure Command Verifier. <i>This library is normally in the linklist and does not require users on the access list.</i>
your.prefix.**	All other IBM zSecure related libraries. <i>Access should be restricted to zSecure users.</i>

13.3.1 Program Access to Datasets

Your installation should consider implementing *Program Access to Datasets* (PADS) to enhance security within your environment. For example, you can implement conditional access to the RACF database(s) and other data sources when using IBM zSecure. In this context, conditional means that a user can only open the RACF database for read access when using IBM zSecure; they will not be able to read the database under any other circumstances.

This helps to further restrict access to these sensitive data sources and prevent users from copying them. The implementation of PADS requires an experienced RACF administrator to help plan and implement the necessary controls. They should refer to Appendix C, “Restricted mode”, in *IBM Security zSecure Suite: CARLa-Driven Components Version 1.12 Installation and Deployment Guide*, SC27-2772 for further guidance on this subject.

13.3.2 Conclusion for RACF security

IBM zSecure has some powerful reporting and security administration functions. Access to IBM zSecure products should be restricted to those who require it as part of their job function. Whether you are installing IBM zSecure for the first time, or you are a well established IBM zSecure user, you should take time to review the security for the products to ensure that appropriate controls and auditing are in place.

You should also take time to review the security for your RACF databases, SMF data sources, and any other sensitive files that are used by IBM zSecure.

We recommend that you refer to Appendix B, “Tivoli zSecure-specific security resources”, in *IBM Security zSecure Suite: CARLa-Driven Components Version 1.12 Installation and Deployment Guide*, SC27-2772 for further information about IBM zSecure specific security resources.

13.4 Running initial analysis reports

When first faced with a new system, it is useful to run a number of the supplied reports from the zSecure suite to learn how this system is set up and secured. This section describes the most commonly used reports and the information they provide; we will use this information as the basis for our subsequent cleanup of the TRA security environment.

13.4.1 Status audit reports

Most of the reports in zSecure Audit require a CKFREEZE file as input. So before we can use them, we must select an appropriate set of input data. As shown in Figure 13-7, we select the current CKFREEZE file and the RACF database, using either option SE.1 or the SETUP line command, to use the full range of zSecure Audit status and audit reports.

```
zSecure Admin+Audit for RACF - Setup - I Row 1 from 6
Command ==> _____ Scroll ==> CSR

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

  Description                                Complex
- test daily ckfreeze gdg input file         SC76      selected
- Active primary RACF data base               SC76      selected
- test daily signature gdg input file         SC76
- test daily unload gdg input file            SC76
- Active backup RACF data base                SC76
- Active backup RACF data base and live SMF data sets SC76
***** Bottom of data *****
```

Figure 13-7 Select input files

zSecure Audit provides many predefined reports, divided into five categories: MVS tables, MVS extended, RACF control, RACF user, and RACF resource oriented tables and reports.

zSecure Audit also provides several *audit policies* against which you can measure your compliance: zSecure, C1, C2, and B1. The default audit policy of zSecure is appropriate for most commercial users of z/OS and associated subsystems. For detailed information about C1, C2, and B1, refer to the US Department of Defence (DoD) orange book standards at:

<http://csrc.nist.gov/publications/history/dod85.pdf>

You might want to test your system against these alternative standards to evaluate possible higher levels of security for your installation, but the zSecure standard is based on real world achievable security standards that are commonly accepted as common practice worldwide. We have chosen to use the zSecure standard as our basis for the following audit reports.

Figure 13-8 shows the report categories and audit policy choices.

```

zSecure Admin+Audit for RACF - Audit - Status
Command ==> _____

Enter / to select report categories
/ MVS tables           MVS oriented tables (reads first part of CKFREEZE)
- MVS extended         MVS oriented tables (reads whole CKFREEZE)
- RACF control          RACF oriented tables
- RACF user             User oriented RACF tables and reports
- RACF resource         Resource oriented RACF tables and reports


Select options for reports:                                Audit policy
- Select specific reports from selected categories          / zSecure
- Concise (short) report                                    - C1
- Output in print format                                     - C2
- Run in background                                         - B1
- Include audit concern overview, higher priorities only

```

Figure 13-8 Report categories and audit policy choices

MVS tables

First, we want to review the MVS tables report category. There are 15 reports available in this category, as shown in Figure 13-9. This is the summary level panel for all subsequent reports in this category.

```

zSecure Admin+Audit for RACF Display Selection                Line 1 of 15
Command ==> _____ Scroll==> PAGE

  Name      Summary Records Title
- SYSTEM      1      1 System settings and software levels
- SYSTEMAU    1      3 System settings - audit concerns
- IPLPARM     1      1 Effective system IPL parameters
- SMFSUBOP    1      2 SMF subsystem-dependent settings
- SUBSYS      1     12 Subsystem Communication Vector Tables
- VSM         1     19 Virtual storage map
- WRITABLE    0      0 Globally Writable Common Storage
- MPFMSG      1     14 Message Processing Facility message intercepts
- JOBCCLASS   1     36 JES2 Job Class parameters (e.g. MVS command auth / B
- CONSOLE     1     30 Operator Consoles
- PPT         1     95 Program Property Table
- SVC         1    148 Supervisor Call Audit Display
- PC          2   1217 Program Call Audit Display
- TAPE        1      1 Tape protection settings (RACF)
- IOAPP       0      0 Authorized I/O Appendage table
***** Bottom of Data *****

```

Figure 13-9 MVS system status audit reports

We reviewed each report and investigated the noted audit concerns on the TRA system. The audit concerns highlight where the current system settings and security definitions do not meet the selected audit policy level. Every audit concern is assigned a priority number. Audit priorities are classified as follows:

- ▶ Priorities of 40 or higher indicate a serious vulnerability or configuration error that should be rectified urgently.
- ▶ Priorities 20 through 39 indicate a potentially serious configuration or security error that should be investigated or mitigated as soon as possible.
- ▶ Priorities 10 through 19 indicate a problem that can possibly be fixed by adjusting RACF profile settings or cleanup activity.
- ▶ Priorities lower than 10 indicate housekeeping activity and may be for information value only.

In this section, we show many typical audit reports used to determine the current security posture of the TRA system.

System settings

In Figure 13-9 on page 214, we select the SYSTEMAU report, which contains three items of interest for audit purposes. You can see in Figure 13-10 that there are two SMF related audit findings and one for TSO. All are of low priority in respect of our stated audit policy of zSecure.

System settings - audit concerns				Line 1 of 2
Command ==>				Scroll==> PAGE
				2 Apr 2008 19:46
Pri	Complex	System	Count	
5	SC76	SC76	3	
Pri	Area	Count		
5	SMF	2		
5	TSO	1		
***** Bottom of Data *****				

Figure 13-10 SYSTEMAU report

There are two SMF audit records and one TSO audit record in this report. The priorities of SMF and TSO records are both 5, indicating a minor audit concern. For the SMF findings in this case, these would be considered a 40 or higher concern if we had audited against a B1 policy.

B1 mandates that SMF is never to be lost under any conditions, even to the point of shutting down z/OS rather than losing the any of SMF audit trail. This is considered a *military grade* security requirement, and not appropriate in normal z/OS business environments. This kind of requirement is partly why we chose to audit against the supplied zSecure policy rather than any of the alternatives.

Figure 13-11 shows detailed information for the TSO related audit concern.

```

System settings - audit concerns                                     Line 1 of 13
Command ==> _____ Scroll==> PAGE
                                     2 Apr 2008 19:46

System
Complex name                SC76
System name                 SC76

System setting
Parameter area              TSO
Parameter name              TSOCONFTXT
Parameter value             No

Audit concern
Relative audit priority      5
Audit concern               VTAM buffers are not confidential
***** Bottom of Data *****

```

Figure 13-11 TSO audit concern

This audit concern VTAM buffers are not confidential means the data in the IBM VTAM® buffers could be overwritten with zeros after it is has been sent to the terminal and thus the buffer content could not be traced. Again, a minor concern, but demonstrative of the level of detail that zSecure Audit is capable of extracting from the system configuration information.

SMF settings

In Figure 13-12, we have selected SMFSUMOP report from the main summary level for MVS tables. This shows the audit concerns for SMF subsystem-dependent settings. We again see some minor audit issues, in this case for STC and SYS SMF definitions.

SMF subsystem-dependent settings		Line 1 of 2
Command ==>		Scroll==> PAGE
2 Apr 2008 19:46		
Complex	System	SMF subsystems Audit concerns Priority
SC76	SC76	2 2 5
Pri	Subs	Su# Wr# Pa# Ex# Det Interval Recording activity summary
<u>s</u>	5 STC	211 45 0 6 No 00:10:00 Write 4:5 7 14:15 20 23 30 35:36 42 60
—	5 SYS	211 45 0 10 No 00:10:00 Write 4:5 7 14:15 20 23 30 35:36 42 60
***** Bottom of Data *****		

Figure 13-12 SMFSUBOP report

Selecting STC in Figure 13-13, we see that the audit concern is Dataset activity not recorded, which is highlighted at the bottom of the report. This implies that some SMF records related to data set activity, such as SMF type 14, 15, and 60 through 69, are being suppressed in the current PARMLIB SMFPRMxx member. We can discover precisely which records are being suppressed by reviewing the list of records flagged in the Act report column.

```

SMF subsystem-dependent settings
Command ==> _____
Line 409 of 423
Scroll==> PAGE
2 Apr 2008 19:46
Complex System SMF subsystems Audit concerns Priority
SC76 SC76 2 2 5
Pri Subs Su# Wr# Pa# Ex# Det Interval Recording activity summary
5 STC 211 45 0 6 No 00:10:00 Write 4:5 7 14:15 20 23 30 35:36 42 60
Exit Address Record Act Record description
243 No
244 No
245 No
246 No
247 No
248 No
249 No
250 No
251 No
252 No
253 No
254 No
255 No
Audit concern
Dataset activity not recorded
***** Bottom of Data *****

```

Figure 13-13 STC audit concerns

JES settings

Figure 13-14 shows the detail level report provided for one JES2 class after selecting the summary report JOBCLASS. This reveals potential audit concerns regarding JES2 job class parameter settings.

```

JES2 Job Class parameters (e.g. MVS command auth / BLP)                               Line 5 of 19
Command ==> _____ Scroll==> CSR
                                     2 Apr 2008 19:46
Complex System   Subsys Classes Audit concerns Priority
SC76      SC76   JES2      36      36      34
Pri C Command Auth commands      BLP HOLD ACCT Time      Regio SWA   PL UJP US
 34 B DISPLAY ALL                Yes No   No  000450,00 0002M ABOVE 00 Yes Ye

Jobs held until released HOLD No
Account number required ACCT No
Time limit              TIME 000450,00
Region size             REGION 0002M
SWA ctrl block residency SWA ABOVE
PROCxx suffix          PROCLIB 00
Job purge exit taken    IEFUJP Yes
SYSOUT limit exit actv  IEFUS0 Yes
SMF Type 6 written      TYPE6 Yes
SMF Type 26 written     TYPE26 Yes

Audit concern
BLP allowed; TAPEVOL not active, will not test ICHBLP, MVS Modify commands
allowed, RACF-protected but UACC>=UPDATE, not verified by operator, No account
numbers required
***** Bottom of Data *****

```

Figure 13-14 JES2 job class audit concerns

The priority for this jobclass related issue is ranked as 34, implying that this deserves closer examination. The detailed information reveals that BLP allowed; TAPEVOL not active, will not test ICHBLP, MVS Modify commands allowed, RACF-protected but UACC>=UPDATE, not verified by operator, No account numbers required. It seems there are a number of improvements that could be made to the JES2 jobclass definitions before this system can be said to implement a best practice.

Other system configuration parameters

Some other important reports in the MVS table category that we should review in any detailed system audit are CONSOLE, PPT, SVC, and PC related reports. We do not show these here, and instead briefly review the MVS extended tables reporting category.

MVS extended tables

We now select the MVS extended tables category of reports. There are nine reports currently provided here, concerned with pure MVS configuration. Each report should be reviewed and any audit findings investigated. The SENSALL report is a combination of the SENSAFP, SENSLINK, and SENSLPA reports together with reports on other specific sensitive configuration files. In Figure 13-15, we see all the sensitive data sets and any audit concerns grouped by data set type.

All sensitive data sets by priority and type

Line 1 of 27

Command ==> Scroll==> CSR

2 Apr 2008 19:46

Complex	System	Priority	Sensitive data sets	Audit concerns
SC76	SC76	5	3151	5
Priority	Sensitivity		Sensitive data sets	Audit concerns
—	5 NoAPFnoDsn		5	5
—	APF lib+Lnk		54	0
—	APF library		41	0
—	APF Linklst		19	0
—	APF LPAlist		8	0
—	Catalog		6	0
—	Couple Alt		3	0
—	Couple Prim		3	0
—	HFS dataset		2973	0
—	IPL Nucleus		1	0
—	JES2 Ckpt		1	0
—	JES2 Spool		1	0
—	LPA list		15	0
—	MSTR prmlib		4	0
—	MSTR STClib		1	0
—	Pagedataset		3	0
—	RACF back		1	0
—	RACF prim		1	0

Figure 13-15 SENSALL audit report

The audit concern NoAPFnoDsn record shown in Figure 13-16 indicates that these data sets are in the Authorized Program Facility (APF) list but do not exist on DASD.

All sensitive data sets by priority and type					Line 1 of 5	
Command ==> _____					Scroll==> <u>CSR</u>	
					2 Apr 2008 19:46	
Complex	System	Priority	Sensitive data sets	Audit concerns		
SC76	SC76	5	3151	5		
Priority	Sensitivity	Sensitive data sets		Audit concerns		
	5 NoAPFnoDsn	5		5		
Pri	Dataset	VolSer		Risk	Audit concer	
—	5 EJES.SPHXLMD0	Z19RE1		ALTER	In APFlist b	
—	5 EUVF.SEUVFLNK	Z19RE1		ALTER	In APFlist b	
—	5 GLD.SGLDLNK	Z19RE1		ALTER	In APFlist b	
—	5 GSK.SGSKLOAD	Z19RE1		ALTER	In APFlist b	
—	5 ISF.SISFLOAD	Z19RE2		ALTER	In APFlist b	
***** Bottom of Data *****						

Figure 13-16 SENSALL audit report - Audit concern list

RACF control

We now select the RACF control report category. There are 11 reports available here, focusing on RACF configuration tables and settings. Again, every report should be reviewed and the audit items investigated. The SETROPAU is a good starting point, and will be a major focus of the TRA cleanup activities.

Figure 13-17 shows this report. You can see that there are many high priority audit concerns that must be addressed on this system to achieve compliance with Delft Transport standards.

Pri	Complex	System	Count	
35	SC76	SC76	17	
Pri	Parameter		Value	
—	35	PROTECTALL	No	
—	30	BATCHALLRACF	No	
—	30	REVOKE	No	
—	25	TAPEVOL	No	
—	21	SAUDIT	No	
—	20	OPERAUDIT	No	
—	15	AUDIT_GROUP	No	
—	15	AUDIT_USER	No	
—	15	CMDVIOL	No	
—	15	ERASEONSCRATCH	None	
—	15	HISTORY	No	
—	15	INACTIVE	No	
—	11	MINCHANGE	No	
—	11	RVARYSTATUSPWSET	No	
—	10	GENERICOWNER	No	
—	10	INACTIVE	No	
—	10	RVARYSWITCHPWSET	No	

Figure 13-17 SETROPAU report - Summary list

We also carefully review the two reports related to system Started Tasks, STARTED and STCTABLE. In this system, the RACF started procedures table, ICHRIN03, was not in use, so the only relevant report was STARTED.

Profiles in the RACF class STARTED are used to provide a RACF user and (optionally) group to a started task. They can also specify attributes such as privileged or trusted that allow the task to bypass RACF protection, which is why careful analysis of these definitions is important.

In Figure 13-18, we have sorted the display using the line command SORT TRU DESC to list tasks with the trusted attribute at the top. If you have zSecure Admin installed, you can manipulate the RACF profile directly from this panel (use the forward slash character / to see the available options).

RACF Profiles in Started Class - sorted by procedure						Line 1 of 72	
Command ==>						Scroll==> CSR	
						9 Apr 2008 15:51	
Complex	Timestamp	Count					
SC76	9Apr2008 15:51	72					
Profile key	Userid	Group	Pri	Tru	Tra		
— TS0.*	<u>TS0</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— XCFAS.*	<u>XCFAS</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— BLSJPRMI.*	<u>BLSJPRMI</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— ANTMMAIN.*	<u>ANTMAIN</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— MVS*	<u>=MEMBER</u>	—	<u>YES</u>	<u>YES</u>	—		
— ZFS.**	<u>DFS</u>	—	—	<u>YES</u>	—		
— APSWPROC.*	<u>APSWPROC</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— WLM.*	<u>WLM</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— VLF.*	<u>VLF</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— SMF.*	<u>SMF</u>	<u>STCGROUP</u>	<u>YES</u>	<u>YES</u>	<u>YES</u>		
— EZAZSSI.*	<u>EZAZSSI</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— NET.*	<u>NET</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— CATALOG.*	<u>CATALOG</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— JES2.*	<u>JES2</u>	<u>STCGROUP</u>	<u>YES</u>	<u>YES</u>	<u>YES</u>		
— JESXCF.*	<u>JESXCF</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— ANTAS000.*	<u>ANTAS000</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— IOSAS.*	<u>IOSAS</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		
— INIT.*	<u>INIT</u>	<u>STCGROUP</u>	—	<u>YES</u>	<u>YES</u>		

Figure 13-18 RACF profiles in started class

RACF user

We now select the RACF user category of reports. The reports available here will be the main focus of the security improvement cleanup at TRA. There are 56 reports currently available in this category. Many of these are variations on common password oriented reports often requested by auditors.

The report of most interest in our cleanup activity is the TRUSTUSR report. It is the aim of this project to reduce the trust levels currently provided to the user community on this system to an acceptable level.

Trusted user IDs

In this context, *trust* means that these users could compromise the system using their high level of access to parts of the *trusted computing base*. To meet any kind of detailed audit, the reduction of *trust* throughout the system is essential. Some users will always retain high trust levels, which is essential to system maintenance and management. However, this must be restricted to as small a number of users as possible, and the level of trust invested in these users must also be the minimum required to perform their duties. This is the real *art* of modern security in any information technology environment, and we will demonstrate how to achieve these goals using the zSecure suite.

All the reports in this category should be reviewed for their audit findings. Special attention should be applied to the reports listing users who have high level privileges, such as AUTHSYS for users with system special, operations, auditor, or any RACF class authorities. AUTHUID0 lists users with UNIX System Services UID(0), and AUTHGRP is for users with group level attributes.

The password related reports provide a wealth of information that is useful in determining who is actually using their user IDs, which can be used to determine who may be removed from the system.

We select the TRUSTUSR report shown in Figure 13-19 to see what levels of trust are invested in our user community (Figure 13-20 on page 223).

zSecure Admin+Audit for RACF Display Selection			Line 1 of 56
Command ===> _____			Scroll==> <u>CSR</u>
Name	Summary	Records	Title
= TRUSTUSR	1	6144	Trusted userids (may bypass security)
- AUTHSYS	1	99	Users with system-wide special, operations, auditor,
- AUTHUID0	1	25	Users with uid 0
- AUTHGRP	1	1	Users with group level special, operations, auditor
- SHRDUIDS	1	119	OMVS UIDs shared between RACF users
- OMVSNUID	0	0	RACF users with OMVS segment but no UID
- SHRDGIDS	1	43	OMVS GIDs shared between RACF groups
- OMVSNPID	0	0	RACF groups with OMVS segment but no GID
- PROTECT	1	14	Protected users
- PWNONE	0	0	Users who can logon without password
- PWUID	0	0	Users who can logon with OIIdcard
- PWINNONE	1	12	Users without password interval
- PWINLONG	1	234	Users with password interval > 60 days
- PWEXPIRE	1	208	Users with expired passwords
- PHEXPPIRE	1	1	Users with expired password phrases
- PWNOCHG	1	191	Users that never changed password
- PHNOCHG	0	0	Users that never changed password phrase
- PWAGESUM	1	277	RACF password age overview
- PHAGESUM	1	1	RACF password phrase age overview
- PWAGEALL	1	277	User Password or Phrase Age: All users

Figure 13-19 RACF user audit report - Summary

Trusted userids (may bypass security)						Line 1 of 108
Command ==> _____						Scroll==> <u>CSR</u>
						15 Apr 2008 13:10
Pri Complex Trusted userids						
49 SC76 108						
Pri	Reasons	Userid	Name	RIP	DfltGrp	InstData
— 49	217	TKRAUS	THOMAS KRAUS		SYS1	
— 49	212	CFZADM	CFZADM		SYS1	
— 49	110	????????				
— 10	1544	SYSPROG	SYSPROG		SYS1	
— 10	1498	DFS			DFSGRP	
— 10	115	RC76	RICH CONWAY		SYS1	
— 10	111	WELLIE2			SYS1	
— 10	109	HAIMO	HAIMO		SYS1	
— 10	107	ALEX	ALEX L		SYS1	
— 10	107	MCAIRNS	MICHAEL CAIRNS		@ZSEC001	
— 10	107	RMETH	RICARDO METH		SYS1	
— 10	107	ROGERS	PAUL ROGERS		SYS1	
— 10	106	VAINI	JUHA VAINIKAINEN		SYS1	
— 10	106	WHITE	WHITE		SYS1	
— 10	105	CEA			SYS1	
— 10	105	CIMUSR1			SYS1	
— 10	105	FRANCK	FRANCK		SYS1	
— 10	105	FRANCK1	FRANCK1		SYS1	

Figure 13-20 Trusted users

We can see that high priority audit concerns are present for a number of users. Of special interest is the user ID marked as ????????, which represents the JES2 undefined user ID, and indicates that trust is available for all users in the system, for example, through UACC. This situation is highly undesirable, and we will have to make significant changes to the overall security settings to rectify this.

We would like to see what specific audit findings are generating such high level priorities, so we select one of the users with this level of trust to review the detailed audit issues, as shown in Figure 13-21.

Trusted userids (may bypass security)						Line 1 of 106		
Command ==> _____						Scroll==> <u>CSR</u>		
						15 Apr 2008 13:10		
Pri	Complex	Trusted userids						
49	SC76	108						
Pri	Reasons	Userid	Name	RIP	DfltGrp	InstData		
49	212	CFZADM	CFZADM		SYS1			
Pri	Cnt	Audit concern						
==	49	107	Can submit jobs for trusted user					
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					BPXROOT
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					CEA
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					CIMUSR1
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					FRANCK
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					FRANCK1
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					IBMUSER
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					ITDI
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					KMT1
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					KMT2
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					OMVSKER
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					OSA
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					PAGENT
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					ROGERS
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					TCPIP
__	10	1	Can use Trojan attacks via the homedirectory of trusted user					WEBSRV

Figure 13-21 Trusted users detailed access

We can see that the primary audit finding is that this user may submit jobs on behalf of some other trusted user, and that there are 107 ways that this might be achieved. A significant audit issue, especially as this concern is likely to apply to a number of the users shown in the previous level report. To better understand this issue, we again drill down into the report by selecting this line item. The resulting panel is shown in Figure 13-22.

```

Trusted userids (may bypass security)
Command ==> _____
Line 1 of 107
Scroll==> CSR
15 Apr 2008 13:10

Pri Complex Trusted userids
49 SC76 108

Pri Reasons Userid Name RIP DfltGrp InstData
49 212 CFZADM CFZADM SYS1

Pri Cnt Audit concern
49 107 Can submit jobs for trusted user

Pri Complex Sensitivity Resource Risk
— 49 SC76 Surrogate BPX.SRV.????????? READ
— 10 SC76 Surrogate BPX.SRV.ALEX READ
— 10 SC76 Surrogate BPX.SRV.ALEX01 READ
— 10 SC76 Surrogate BPX.SRV.ALEX02 READ
— 10 SC76 Surrogate BPX.SRV.ALEX03 READ
— 10 SC76 Surrogate BPX.SRV.AXELB READ
— 10 SC76 Surrogate BPX.SRV.BARI READ
— 10 SC76 Surrogate BPX.SRV.BDH01 READ
— 10 SC76 Surrogate BPX.SRV.BLDSEG READ
— 10 SC76 Surrogate BPX.SRV.BPXROOT READ
— 10 SC76 Surrogate BPX.SRV.BROGERS READ
— 10 SC76 Surrogate BPX.SRV.BRUINSM READ
— 10 SC76 Surrogate BPX.SRV.BWILSON READ
— 10 SC76 Surrogate BPX.SRV.CAGOOD READ

```

Figure 13-22 Trusted user - Detailed privileges

Selecting the line item with audit priority 49 revealed that a RACF profile BPX.SRV.** was defined in the class SURROGAT. This profile covered *any* user ID without a matching BPX.SRV profile more fully qualified, thus allowing a user with access to assume the identity of almost any UNIX System Services user ID for access purposes. Reducing the number of users with access to this profile will help remediate this high priority audit finding. Access reduction measures such as this will be described in 13.5, “Implementing initial improvements in system security posture” on page 231.

RACF profiles

The last category of status audit reports we look at is the RACF resource oriented reports.

Upon selecting this category, we are presented with a panel with a request to decide whether JES2 procedure libraries and Linklist data sets should be considered sensitive, as well as some other output options (Figure 13-23). In general, we recommend that both JES2 procedures and Linklisted data sets should be selected. Access to updating a JES2 procedure library could be used to introduce trojan code into the environment, which is possible with Linklisted data sets, especially if Linklist is running as APF authorized; these are the usual sensitive libraries due to these concerns.

zSecure Admin+Audit for RACF - Audit - Status Specifications

Command ==> _____

Customize output for sensitive resource reporting

Enter "/" to select option(s)

☒ JES2 JOB proclibs considered sensitive
☒ All linklist data sets considered sensitive
☐ Show all datasets covered by profiles for sensitive data

Customize output for started task (STC) reporting

Sort order

☒ 1. Userid 2. Member
 Enter "/" to select option
☐ Suppress undefined STC userids (Id=* and +++++++)

Minimum access to show

☐ 1. Show all 2. Hidden 3. Copy 4. Execute 5. Loadexe
 6. Readlpa 7. Read 8. Update 9. Control 10. Alter

Figure 13-23 RACF resource reports customization

There are 18 reports in this category, covering issues such as access to sensitive data sets, TSO authorized commands, UNIX System Services privileged file settings, and globally writable data, as shown in Figure 13-24.

```

zSecure Admin+Audit for RACF Display Selection
Command ==> _____
Line 1 of 18
Scroll==> CSR

```

Name	Summary	Records	Title
= RACPRAUD	1	180	RACF profile audit concerns
- SENSTRUS	1	6144	Sensitive data trustees
- SENSPROF	1	847	Profiles covering sensitive data sets
- ENTITY#S	1	1888	Entity segment summary by descending number of bytes
- SEGMENT	1	1888	Class segment summary by descending number of bytes
- TSOAUTH	1	134	TSO authorized commands
- LPAPROT	1	149	LPA module protection overview
- APFPROT	1	1201	APF module protection overview (except UNIX files)
- UNIXAPF	1	656	UNIX files with APF authorization
- UNIXCTL	1	5138	UNIX files that are program controlled (daemons etc)
- UNIXSUID	1	44	UNIX files with SETUID authorization
- UNIXSGID	1	11	UNIX files with SETGID authorization
- PADS	1	2	PADS module protection overview
- STCPROT	1	733	Started task overview
- GLBWDSN	1	6	Data sets vulnerable to trojan horse & back door att
- GLBWUNIX	1	64	UNIX files vulnerable to trojan horse & back door at
- UIDNOUSR	1	32	UIDs not defined in the complex
- GIDNOGRP	1	34	GIDs not defined in the complex
***** Bottom of Data *****			

Figure 13-24 RACF resource audit reports list

The RACPRAUD report shows audit concerns for RACF profiles, such as having a UACC setting above READ access. The SENSTRUS report displays all sensitive resource categories and the people who have access to them. The SENSPROF report shows the profiles protecting sensitive resources, as shown in Figure 13-25.

Profiles covering sensitive data sets					Line 1 of 847	
Command ==> _____					Scroll==> <u>CSR</u>	
16 Apr 2008 12:47						
Complex	Timestamp	Profiles Audit concerns		Priority		
SC76	16 Apr 2008 12:47	847		846	70	
Pri	Profile key			UACC	Era	S/F Audit conce
—	70 SYS1.**			<u>ALTER</u>	<u>NO</u>	<u>R</u> No read aud
—	60 \$AHX110.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$ARS.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$ASN.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$BIP.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$BIP21.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$CICS310.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$CQM.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$CTG.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DDA.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DHB.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DNE710.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DNF.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DNI.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DOL710.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DSN610.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DSN710.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected
—	60 \$DSN810.*.**			<u>ALTER</u>	<u>NO</u>	Unprotected

Figure 13-25 Sensitive profile concerns

During our work to improve the overall security of the TRA system, we will make extensive use of the RACPRAUD, SENSTRUST, and SENSPROF reports. Between them, these reports quickly highlight issues where we can achieve substantial improvement in our overall security posture with the minimal number of system changes and attendant risk (using the 80/20 rule).

By reviewing the RACF resources at risk against the levels of trust assigned to users in the trust analysis reports from “RACF user” on page 221, we quickly discover which resources provide the greatest system exposures, and which users have access to them. This information becomes the basis for our first major changes to truly improve the security on this system.

Now we will review the RACF group tree structure in place at TRA.

13.4.2 Reviewing the current RACF group tree

Using zSecure Admin option RA.3.8, we examine the RACF group tree structure in use on the TRA system, starting at the top group SYS1. You can see in Figure 13-26 that there is little or no hierarchy or structure in place. Almost all RACF groups are direct subgroups of SYS1. A hierarchy or a consistent naming convention are not absolutely necessary. However, using a hierarchy and naming standards does make security and systems management much easier and more consistent in the long term, thus easing the burden of managing the environment.

zSecure Admin+Audit for RACF GROUP TREE DISPLAY						Line 1 of 116
Command ==> _____						Scroll==> PAGE
						10 Apr 2008 15:01
Complex Groups						
SC76 116						
Group structure						
— SYS1	Lvl	Subgrp	Connct	SupGroup	Owner	X
— CMNGRP	1	38	203	IBMUSER	X
— CPAC	2	0	0	SYS1	ARMIGES	X
— DFSGRP	2	0	0	SYS1	WELLIE2	X
— EMPLOYEE	2	0	1	SYS1	SYSPROG	X
— EXTERNAL	2	0	1	SYS1	RC76	X
— GADM	2	0	1	SYS1	IBMUSER	X
— GBIN	2	0	1	SYS1	IBMUSER	X
— GNOBODY	2	0	1	SYS1	IBMUSER	X
— GSYS	2	0	1	SYS1	IBMUSER	X
— HFS	2	0	0	SYS1	SYSPROG	X
— IMWEB	2	0	2	SYS1	RC76	X
— ITSC	2	0	0	SYS1	WELLIE2	X
— KINGS	2	1	0	SYS1	WELLIE	X
— NOGID	3	0	0	KINGS	ELVIS	X
— LDAPGRP	2	0	1	SYS1	RC76	X
— MAIL	2	0	0	SYS1	IBMUSER	X
— OMVS	2	0	0	SYS1	SYSPROG	X

Figure 13-26 A listing of subgroups of SYS1

13.5 Implementing initial improvements in system security posture

Given what was discovered in 13.4.1, “Status audit reports” on page 213 and 13.4.2, “Reviewing the current RACF group tree” on page 229, we now realize just how much effort might be required to align the TRA system with accepted RACF best practices and the Delft Transport security standards. A number of major issues presented themselves, among them:

- ▶ Almost all user IDs are connected to group SYS1 and receive access through this group. In other words, there is little or no segregation of user privileges; all users have the same access. Almost all other RACF groups are direct SubGroups of SYS1, providing no access structure or meaning in the group tree.
- ▶ The lack of the RACF SETROPTS setting PROTECTALL implies that unprotected data is highly likely to exist on this system, access to this data is not under RACF control, and all users effectively have ALTER level access to these data sets.
- ▶ A large number of RACF SETROPTS changes are required to enforce even basic auditing and user access control.
- ▶ Many users possess RACF system level privileges such as SPECIAL and OPERATIONS. This increases their trust status to cover virtually the entire system. All of these users represent system integrity risks to both data and overall security.

A plan for rectification of as many of these problems as possible was quickly drawn up:

1. Implement as many SETROPTS auditing and control functions as possible with immediate effect. Without adequate auditing (SMF) records we would have no information on which to base further access control decisions. Refer to 13.5.1, “Implementing SETROPTS improvements” on page 232.
2. Implement a RACF group tree structure that allows for segregation of control and access rights. Move all users currently in SYS1 into one or more groups in the new access structure. Remove all users except IBMUSER from SYS1 and move all SubGroups of SYS1 into a appropriate part of the new group tree. Refer to 13.5.3, “Implementing an improved RACF group tree structure” on page 241.
3. Outline a plan for implementation of PROTECTALL allowing for no impact to the system users, that is, however we implement this, we cannot remove access that currently exists, thus threatening the availability of service on this system. Refer to 13.5.4, “Planning for PROTECTALL implementation” on page 249.

4. Use zSecure Audit automated verification reports to determine the next steps for security improvements. Refer to 13.6, “Post implementation verification reports” on page 254.

13.5.1 Implementing SETROPTS improvements

From the SETROPAU report in “RACF control” on page 220, we know that there are a number of simple improvements needed in the RACF SETROPTS settings. In this section, we show how we made these changes using zSecure Admin option RA.S. Figure 13-28 and Figure 13-29 show the overtypable fields that we use to issue the appropriate SETROPTS commands.

Auditing options	—	Mandatory Access Control options	
Audit SPECIAL users	<u>No</u>	Require SECLABEL	MLACTIVE <u>No</u>
Audit OPERATIONS users	<u>No</u>	Prevent declassify	MLS <u>No</u>
Audit USER profile changes	<u>No</u>	Stabilize labels	MLSTABLE <u>No</u>
Audit GROUP profile changes	<u>No</u>	Label maintenance	MLQUIET <u>No</u>
Audit SECLABELed resources	<u>No</u>	No SECLABEL tolerate	COMPAT <u>No</u>
Audit command violations	<u>No</u>	Special required SECL.CONTROL	<u>No</u>
Audit from security level	<u>None</u>	Req. labels UNIX fs	MLFSOBJ <u>No</u>
Real datasetnames in SMF	<u>No</u>	Req. labels IPC obj	MLIPCOBJ <u>No</u>
Dataset logoptions	<u>Profile</u>	Name hiding active	MLNAMES <u>No</u>
APPLAUDIT is active	<u>No</u>	Labels by system	SECLBYSYSTEM <u>No</u>

Figure 13-28 SETROPTS display - Part 1

Identification/Authentication options		Job Entry Subsystem options	
Remember dates INITSTATS	<u>Yes</u>	Batch userid req	BATCHALLRACF <u>No</u>
Prevent logon if unused days	<u>No</u>	Monitor userid req	XBALLRACF <u>No</u>
Revoke after password attempt	<u>No</u>	Call router exit	EARLYVERIFY <u>No</u>
Old passwords forbidden	<u>No</u>	Default uid remote	NJEUSERID <u>????????</u>
Password change wait days	<u>No</u>	Default uid local	UNDEFINEDU <u>++++++</u>
Password change interval	<u>254</u>		
Password change warning day	<u>No</u>		
Mixed case passwords allowed	<u>No</u>		
Key change required day	<u>30</u>		
RVARY SWITCH password set	<u>No</u>		
RVARY STATUS password set	<u>No</u>		

Figure 13-29 SETROPTS display - Part 2

After overtyping the fields:

- Audit SPECIAL users - yes

RACF will log all RACF commands issued by users with the SPECIAL or group SPECIAL attribute.

- ▶ Audit OPERATIONS users - yes
RACF will log all actions allowed only because a user has the OPERATIONS or group OPERATIONS attribute.
- ▶ Audit USER profile changes - yes
RACF will log all RACF commands and DEFINE requests affecting profiles in the USER class.
- ▶ Audit GROUP profile changes - yes
RACF will log all RACF commands and DEFINE requests affecting profiles in the GROUP class.
- ▶ Audit command violations - yes
RACF will log violations detected by RACF commands.
- ▶ Real datasetnames in SMF - yes
RACF logging uses the real SMF data set name for logging purposes.
- ▶ Batch user ID req BATCHALLRACF - yes
JES will only accept jobs containing either a valid RACF user ID and password or propagated RACF information.
- ▶ Monitor user ID req XBMALLRACF - yes
JES2 will test jobs to be run with an execution batch monitor for either a valid RACF user ID and password or propagated RACF information.
- ▶ Call router exit EARLYVERIFY - yes
JES will call the System Authorization Facility (SAF) for jobs that do not qualify for user ID propagation. Thus, jobs will verify the user ID, group, and password at job submission time.

We see the following SETROPTS command generated in Figure 13-30. We issue this command.

zSecure Admin+Audit for RACF - Confirm command	
Command ==> _____	
Confirm or edit the following command	
<u>SETROPTS</u> <u>saudit</u> <u>operaudit</u> <u>audit(user)</u> <u>audit(group)</u> <u>cmdviol</u> <u>realdsn</u> <u>JES(batchal</u>	
<u>lracf</u> <u>xbmallracf</u> <u>earlyverify)</u>	
Command execution . <u>1</u>	<ol style="list-style-type: none">1. EXECUTE RACF command2. EXECUTE CKGRACF command (allows use of Reason)3. ASK administrator to execute CKGRACF command4. REQUEST CKGRACF command for later execution5. WITHDRAW CKGRACF command
Reason _____	
Press ENTER to continue or END to cancel the command	

Figure 13-30 SETROPTS basic settings

We then set the basic password and user ID access controls by overtyping the fields from Figure 13-29 on page 232.

- ▶ Prevent logon if unused days - 90
User IDs that are inactive for 90 days will be marked as inactive. An inactive user is revoked by RACF the next time it tries to log on.
- ▶ Revoke after password attempt - 6
A user ID will be automatically revoked after six invalid password attempts.
- ▶ Old passwords forbidden - 12
Users are forbidden to use any of their 12 previous passwords.
- ▶ Password change wait days - 1
A user is forbidden to change their password more than once on the same day.

- ▶ Password change interval - 35
A user is required to change their password every 35 days.
- ▶ Password change warning day - 3
RACF will issue a warning message to a user if their password is due to expire in 3 days or less.

We then issue the generated command shown in Figure 13-31.

<div>zSecure Admin+Audit for RACF - Confirm command</div> <div>Confirm or edit the following command</div> <div>SETROPTS inactive(90) PASSWORD(revoke(6) history(12) minchange(1) interval(35) warning(3))</div>
--

Figure 13-31 SETROPTS password and logon settings

13.5.2 Cleaning up badly defined data set profiles

While reviewing data set profiles defined on the system using zSecure Admin option RA.D, we discovered that many of these were not properly defined.

The system uses RACF Enhanced Generic Naming (EGN), and the data set profiles shown in Figure 13-32 were not providing the intended level of security. Some of these profiles would only cover data sets that had two qualifiers. Any other data sets matching these high level qualifiers were effectively unprotected.

We could have written a small CARLa program to find and replace these profiles. However, there were only a few of them defined, and we decided to just fix these using the zSecure Admin panels.

zSecure Admin+Audit for RACF DATASET Overview				Line 85 of 104
Command ==>				Scroll==> CSR
All profiles				10 Apr 2008 16:34
Profile key	Type	UACC	Owner	S/F W
___ ROGERS.*	GENERIC	NONE	ROGERS	R _
___ RRDCUST.*	GENERIC	NONE	RRDCUST	R _
___ SAHEEM.**	GENERIC	NONE	SAHEEM	R _
___ SJON.*	GENERIC	NONE	SJON	R _
___ SMCHUGH.*	GENERIC	NONE	SMCHUGH	R _
___ SOWERS.*	GENERIC	NONE	SOWERS	R _
___ STAN.**	GENERIC	NONE	STAN	R _
___ SYSPROG.*	GENERIC	NONE	SYSPROG	R _
___ SYS1.PARMLIB	GENERIC	READ	WELLIE2	R _
___ SYS1.PARMLIB	NONVSAM	UPDATE	WELLIE2	R _
___ SYS1.**	GENERIC	ALTER	SYS1	R _
___ SYS2.*	GENERIC	ALTER	WELLIE2	R _
___ TKRAUS.**	GENERIC	NONE	TKRAUS	R _
___ TROWELL.*	GENERIC	NONE	TROWELL	R _
___ UCAT.VBOOK01	VSAM	READ	HAIMO	R _
___ USER1.*	GENERIC	NONE	USER1	R _
___ VAINI.*	GENERIC	NONE	VAINI	R _
___ VELLOSO.**	GENERIC	NONE	VELLOSO	R _
___ WELLIE3.*.*	GENERIC	NONE	WELLIE3	R Y
___ WHITE.*	GENERIC	NONE	WHITE	R _

Figure 13-32 Incorrectly defined data set profiles

To speed up this work a little, we first changed our setup confirmation options so that we would not have to press Enter more than once for each panel of commands we were generating. Using option SE.4, we changed Confirmation from the default value of 4 to 1, the effect being to disable all confirmation of the commands we were about to generate.

Figure 13-33 shows the default settings prior to our change.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - Setup - Confirm				
Command ==> _____				
Action on command . . .	<u>1</u>	1. Queue	2. Execute	3. Not allowed
Confirmation	<u>4</u>	1. None	2. Deletes	3. Passwords 4. All
Command generation				
Enter "/" to select option(s)				
/ Overtime fields in panels				
/ Change generated commands				
/ Specify start/end date				
- Generate SETROPTS REFRESH commands				
- Issue prompt before generating SETROPTS REFRESH commands				
Commands to generate				
/ RACF commands				
/ CKGRACF commands				
/ CKGRACF ASK for later execution				
/ CKGRACF REQUEST for later execution				
- CKGRACF WITHDRAW queued commands				
/ CKGRACF RDELETE commands				

Figure 13-33 Setup confirm options - SE.4

We recommend you only do this when you are completely confident of the changes you intend to make and comfortable with the flow of panels that zSecure Admin presents. After correcting the data set profiles in question, we changed this option back to the default confirmation for all actions.

In a production environment, you might want to consider setting the confirmation setting to 3 (Passwords). This suppresses confirmation of commands that contain a password and could be beneficial if you work on a large office floor where someone could potentially look over your shoulder to see the generated RACF commands that contain a readable password.

After doing this task, we were ready to generate copy commands for all the data sets on the panel shown in Figure 13-34.

zSecure Admin+Audit for RACF DATASET Overview					Line 85 of 104
Command ==>					Scroll==> <u>CSR</u>
All profiles					10 Apr 2008 16:45
Profile key	Type	UACC	Owner	S/F	W
<u>c</u> ROGERS.*	GENERIC	<u>NONE</u>	<u>ROGERS</u>	<u>R</u>	<u>-</u>
<u>c</u> RRDCUST.*	GENERIC	<u>NONE</u>	<u>RRDCUST</u>	<u>R</u>	<u>-</u>
<u> </u> SAHEEM.**	GENERIC	<u>NONE</u>	<u>SAHEEM</u>	<u>R</u>	<u>-</u>
<u>c</u> SJON.*	GENERIC	<u>NONE</u>	<u>SJON</u>	<u>R</u>	<u>-</u>
<u>c</u> SMCHUGH.*	GENERIC	<u>NONE</u>	<u>SMCHUGH</u>	<u>R</u>	<u>-</u>
<u>c</u> SOWERS.*	GENERIC	<u>NONE</u>	<u>SOWERS</u>	<u>R</u>	<u>-</u>
<u> </u> STAN.**	GENERIC	<u>NONE</u>	<u>STAN</u>	<u>R</u>	<u>-</u>
<u>c</u> SYSPROG.*	GENERIC	<u>NONE</u>	<u>SYSPROG</u>	<u>R</u>	<u>-</u>
<u> </u> SYS1.PARMLIB	GENERIC	<u>READ</u>	<u>WELLIE2</u>	<u>R</u>	<u>-</u>
<u> </u> SYS1.PARMLIB	NONVSAM	<u>UPDATE</u>	<u>WELLIE2</u>	<u>R</u>	<u>-</u>
<u> </u> SYS1.**	GENERIC	<u>ALTER</u>	<u>SYS1</u>	<u>R</u>	<u>-</u>
<u>c</u> SYS2.*	GENERIC	<u>ALTER</u>	<u>WELLIE2</u>	<u>R</u>	<u>-</u>
<u> </u> TKRAUS.**	GENERIC	<u>NONE</u>	<u>TKRAUS</u>	<u>R</u>	<u>-</u>
<u>c</u> TROWELL.*	GENERIC	<u>NONE</u>	<u>TROWELL</u>	<u>R</u>	<u>-</u>
<u> </u> UCAT.VBOOK01	VSAM	<u>READ</u>	<u>HAIMO</u>	<u>R</u>	<u>-</u>
<u>c</u> USER1.*	GENERIC	<u>NONE</u>	<u>USER1</u>	<u>R</u>	<u>-</u>
<u>c</u> VAINI.*	GENERIC	<u>NONE</u>	<u>VAINI</u>	<u>R</u>	<u>-</u>
<u> </u> VELLOSO.**	GENERIC	<u>NONE</u>	<u>VELLOSO</u>	<u>R</u>	<u>-</u>
<u>c</u> WELLIE3.*.**	GENERIC	<u>NONE</u>	<u>WELLIE3</u>	<u>R</u>	<u>Y</u>
<u>c</u> WHITE.*	GENERIC	<u>NONE</u>	<u>WHITE</u>	<u>R</u>	<u>-</u>

Figure 13-34 Generate the copy commands

Some manual effort was required, because we had to enter the new profile name for each copy command when copying from HLQ.* to HLQ.**. However, this was still far simpler than the alternative of creating RACF commands in batch and then editing (or using a similar process).

Figure 13-35 shows the CKRCMD file containing the commands. We now execute these commands as normal by placing an R next to the CKRCMD file.

File	Edit	Edit_Settings	Menu	Utilities	Compilers	Test	Help
EDIT		CONWAYM.C2R1294.CKRCMD			Columns 00009 00080		
Command ==>					Scroll ==> CSR		
000107	addsd	'SJON.**'	GENERIC	owner(SJON)	uacc(NONE)	audit(failures(READ	
000108	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:01 */					
000109	addsd	'SMCHUGH.**'	GENERIC	owner(SMCHUGH)	uacc(NONE)	audit(failure	
000110	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:01 */					
000111	addsd	'SOWERS.**'	GENERIC	owner(SOWERS)	uacc(NONE)	audit(failures(
000112	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:01 */					
000113	addsd	'SYSPROG.**'	GENERIC	owner(SYSPROG)	uacc(NONE)	audit(failure	
000114	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:01 */					
000115	addsd	'SYS2.**'	GENERIC	owner(WELLIE2)	uacc(ALTER)	audit(failures(R	
000116	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:01 */					
000117	addsd	'TROWELL.**'	GENERIC	owner(TROWELL)	uacc(NONE)	audit(failure	
000118	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:02 */					
000119	addsd	'USER1.**'	GENERIC	owner(USER1)	uacc(NONE)	audit(failures(RE	
000120	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:02 */					
000121	addsd	'VAINI.**'	GENERIC	owner(VAINI)	uacc(NONE)	audit(failures(RE	
000122	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:02 */					
000123	addsd	'WELLIE3.**'	GENERIC WARNING	owner(WELLIE3)	uacc(NONE)	audit(
000124	permit	'WELLIE3.**'	GENERIC id(WELLIE4)	access(NONE)			
000125	/*	CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 17:02 */					
000126	addsd	'WHITE.**'	GENERIC	owner(WHITE)	uacc(NONE)	audit(failures(RE	

Figure 13-35 Generated commands

The results can be seen by displaying the list of RACF data set profiles again using option RA.D. We then needed to delete the original badly defined profiles, which we did by issuing the D line command against a panel full of profiles at a time, as shown in Figure 13-36.

This also queued commands into the CKRCMD command buffer, which we subsequently executed in the usual way. With just a few panels and some repeated commands, we had simply and easily replaced the badly defined data set profiles with their correct alternatives.

zSecure Admin+Audit for RACF DATASET Overview				Line 143 of 161	
Command ==> _____				Scroll==> CSR	
All profiles		10 Apr 2008 17:14			
	Profile key	Type	UACC	Owner	S/F W
d	SYSPROG.*	GENERIC	<u>NONE</u>	<u>SYSPROG</u>	<u>R</u>
__	SYSPROG.**	GENERIC	<u>NONE</u>	<u>SYSPROG</u>	<u>R</u>
__	SYS1.PARMLIB	GENERIC	<u>READ</u>	<u>WELLIE2</u>	<u>R</u>
__	SYS1.PARMLIB	NONVSAM	<u>UPDATE</u>	<u>WELLIE2</u>	<u>R</u>
__	SYS1.**	GENERIC	<u>ALTER</u>	<u>SYS1</u>	<u>R</u>
__	SYS2.*	GENERIC	<u>ALTER</u>	<u>WELLIE2</u>	<u>R</u>
__	TKRAUS.**	GENERIC	<u>NONE</u>	<u>TKRAUS</u>	<u>R</u>
d	TROWELL.*	GENERIC	<u>NONE</u>	<u>TROWELL</u>	<u>R</u>
__	TROWELL.**	GENERIC	<u>NONE</u>	<u>TROWELL</u>	<u>R</u>
__	UCAT.VBOOK01	VSAM	<u>READ</u>	<u>HAIMO</u>	<u>R</u>
d	USER1.*	GENERIC	<u>NONE</u>	<u>USER1</u>	<u>R</u>
__	USER1.**	GENERIC	<u>NONE</u>	<u>USER1</u>	<u>R</u>
d	VAINI.*	GENERIC	<u>NONE</u>	<u>VAINI</u>	<u>R</u>
__	VAINI.**	GENERIC	<u>NONE</u>	<u>VAINI</u>	<u>R</u>
__	VELLOSO.**	GENERIC	<u>NONE</u>	<u>VELLOSO</u>	<u>R</u>
d	WELLIE3.*.**	GENERIC	<u>NONE</u>	<u>WELLIE3</u>	<u>R Y</u>
__	WELLIE3.**	GENERIC	<u>NONE</u>	<u>WELLIE3</u>	<u>R Y</u>
d	WHITE.*	GENERIC	<u>NONE</u>	<u>WHITE</u>	<u>R</u>
__	WHITE.**	GENERIC	<u>NONE</u>	<u>WHITE</u>	<u>R</u>

Figure 13-36 Deleting badly defined profiles

Having deleted the badly defined profiles, we also propose to review the ownership of the data set profiles at a later date. For example, group data sets such as SYS1.PARMLIB should be owned by a group. In Figure 13-36, the data set profile SYS1.PARMLIB is currently owned by WELLIE2, which is a user ID. We could change the owner for this data set profile by overtyping the current value with the desired owner, for example, SYS1. This action will generate an ALTDSD RACF command to change the owner.

13.5.3 Implementing an improved RACF group tree structure

A design proposal for a new RACF group tree structure was agreed upon, based on the structure in place already at Delft Transport. A basic concept behind the group structure implemented was to allow for segregation between security administration functions and security access functions. This segregation allows administrators to control users and their authorities within the system, without the capability for administrators to gain access to the resources used by the staff under their control. This was seen as an essential separation of duties in a modern security infrastructure.

This segregation was achieved by implementing separate branches for ownership and access in the RACF group tree. The #FUNCACC branch of the tree shown in Example 13-3 on page 242 is used only for access provisioning. The #USERS branch is used for ownership of user IDs. Administrators exercise their RACF scope, controlled through CKGRACF scoping rules, only over the user ID ownership part of the tree, that is, #USERS or its subgroup #STAFF. Administrators cannot therefore connect themselves to the groups used for access to RACF resources, as these are outside their administrative scope in the #FUNCACC branch of the tree. A naming convention was chosen to reflect and help enforce this segregation of authority.

Group naming convention

A group naming convention was decided upon and will follow these guidelines:

- ▶ Groups prefixed with a hash character (# prefix) represent Structural or Resource/Userid Owning groups. User IDs are never connected to these and these groups should never appear in any access lists.
- ▶ Groups prefixed with a dollar (\$) prefix represent Data Owning groups (as a reminder, you can think of “Dollar = Data”). User IDs are never connected to these and these groups should never appear in any access lists. SubGroups of these groups are used as the actual data set HLQs.
- ▶ Groups prefixed with an At character (@ prefix) represent Access Granting groups (as a reminder, you can think of “At = Access”). These groups are how user IDs are granted access to resources and data sets, and should be the only groups seen in profile access lists.
- ▶ If a group has no prefix, then it should be a Dataset High Level Qualifier. User IDs are never connected to these and these groups should never appear in any access lists, nor should these ever have SubGroups.

Whatever group naming and group tree structural choices you make in your system, the most important factor is to document these conventions and maintain them.

The RACF group tree diagram shown in Example 13-3 is only an overview of the design and detailed implementation we chose. This is not necessarily recommended as a best practice or representative of any necessary requirements for proper RACF security management. Likewise, this is not a complete listing of all groups we defined during this scenario. This is purely a convenient structure that we chose for the purposes of this scenario. Due to the inherent flexibility in the RACF group tree structure, literally infinite variations on this theme are equally valid, depending on your unique security concerns and requirements.

Group tree structure

In Example 13-3, we show the group tree structure we defined for this step of the security improvement project.

Example 13-3 Sample of a group tree structure

```

SYS1
  #USERS      Userid Owning/Structural/Administrative
  #STAFF      Staff Structural/Administrative
  #BRANCH     Branch Staff Owning
  #B001       Branch dept Owning
  #B002       Branch dept Owning
  #IT         IT Department Staff Owning
  #BATCH      Batch Userid Structural/Administrative
  #APP1       Batch Application Userid Owning
  #TASKS      System Task Structural/Administrative
  #APPS       Applications Structural/Administrative
  #PRD        Production App Structural/Administrative
  #TST        Test App Structural/Administrative
  #DEV        Development App Structural/Administrative
  #MNT        Maintenance App Structural/Administrative
  #SYSTEM     System Datasets Structural/Administrative
  $ZOS        z/OS Data Owning
  SYS2        System related HLQ (SYS2.***)
  BDT1        System related HLQ (BDT1.***)
  $TPP        Third Party Product Data Owning
  TIVOLI      Product related HLQ (TIVOLI.***)
  $CICS       CICS Data Owning
  CICSTS      CICS HLQ (CICSTS.***)
  #RACFRES    RACF General Resource Structural/Admin
  #SYSAPPL    CICS and APPL Resource Owning
  #SYSTECH    z/OS Resource Owning
  #SYSUSER    Other Resource Owning
  #FUNCACC    Access Granting Structural/Administrative
  #BRNCACC    Branch Access Administrative

```


@CSV001	Senior Customer Services access
@CSV002	Junior Customer Service access
#USERACC	Staff Access Administrative
@SP001	Senior Sysprog Access
@SP002	Junior Sysprog Access
#TASKACC	System Task Access Administrative
@SYSSTC	System Started Tasks Access
@TPPSTC	Third Party Product Tasks Access
@CICSP	CICS Regions Access PROD
@CICST	CICS Regions Access TEST
#BATACC	Batch Userids Administrative
#SRGTACC	Surrogate Userids Administrative

RACF commands to generate groups

We used RACF commands similar to the ones shown in Example 13-4 to define our new group tree structure.

Example 13-4 RACF commands to generate groups

```
ag #users  owner(sys1) supgroup(sys1) data('userid structure')
ag #apps   owner(sys1) supgroup(sys1) data('application structure')
ag #system owner(sys1) supgroup(sys1) data('system structure')
ag #racfres owner(sys1) supgroup(sys1) data('racf resource structure')
ag #funcacc owner(sys1) supgroup(sys1) +
           data('functional access structure')

ag #staff  owner(#users) supgroup(#users) +
           data('staff userid structure')
ag #batch  owner(#users) supgroup(#users) +
           data('batch userid structure')
ag #tasks  owner(#users) supgroup(#users) +
           data('started task userid structure')

ag #branch owner(#staff) supgroup(#staff) +
           data('branch userid structure')
ag #it     owner(#staff) supgroup(#staff) +
           data('branch userid structure')

ag #app1   owner(#batch) supgroup(#batch) +
           data('application 1 userid structure')

ag #prd    owner(#apps) supgroup(#apps) data('prod apps structure')
ag #tst    owner(#apps) supgroup(#apps) data('test apps structure')
ag #dev    owner(#apps) supgroup(#apps) data('devl apps structure')
```

```

ag #mnt      owner(#apps) supgroup(#apps) data('mnt apps structure')

ag $zos      owner(#system) supgroup(#system) +
              data('system data structure')
ag $tpp      owner(#system) supgroup(#system) +
              data('products data structure')
ag $cics     owner(#system) supgroup(#system) +
              data('cics data structure')

ag sys2      owner($zos) supgroup($zos) data('sys2 hlq owner')
ag bdt1      owner($zos) supgroup($zos) data('bdt1 hlq owner')
ag newgrps   owner($zos) supgroup($zos) data('newgrps hlq owner')
ag tivoli    owner($tpp) supgroup($tpp) data('tivoli hlq owner')
ag cicsts    owner($cics) supgroup($cics) data('cicsts hlq owner')

ag #sysappl  owner(#racfres) supgroup(#racfres) +
              data('cics and appl racf resource owning structure')
ag #systech  owner(#racfres) supgroup(#racfres) +
              data('zos racf resource owning structure')
ag #sysuser  owner(#racfres) supgroup(#racfres) +
              data('user related racf resource owning structure')

ag #useracc  owner(#funcacc) supgroup(#funcacc) +
              data('staff user access owner')
ag #brncacc  owner(#funcacc) supgroup(#funcacc) +
              data('branch staff user access owner')
ag #taskacc  owner(#funcacc) supgroup(#funcacc) +
              data('started task user access owner')
ag #batacc   owner(#funcacc) supgroup(#funcacc) +
              data('batch user access owner')
ag #srgtacc  owner(#funcacc) supgroup(#funcacc) +
              data('surrogat user access owner')

ag @sp001    owner(#useracc) supgroup(#useracc) +
              data('senior systems programmer access')
ag @sp002    owner(#useracc) supgroup(#useracc) +
              data('junior systems programmer access')
ag @sysstc   owner(#taskacc) supgroup(#taskacc) +
              data('system started task access')
ag @tppstc   owner(#taskacc) supgroup(#taskacc) +
              data('product started task access')
ag @cicspt   owner(#taskacc) supgroup(#taskacc) +
              data('cics prod task access')

```

Populating a group tree with users and groups for testing

We used commands similar to the ones shown in Example 13-5 to populate some of our group tree with definitions for testing the security administration framework.

Example 13-5 Populate group tree with users and groups for testing

```
addgroup @CSV001 +
    data('senior customer service rep access') +
    owner(#BRNCACC) supgroup(#BRNCACC)
addgroup @CSV002 +
    data('junior customer services rep access') +
    owner(#BRNCACC) supgroup(#BRNCACC)
addgroup @man001 +
    data('branch manager access') +
    owner(#BRNCACC) supgroup(#BRNCACC)
addgroup @reg001 +
    data('regional manager access') +
    owner(#BRNCACC) supgroup(#BRNCACC)

addgroup #B001 +
    data('pisa cust serv centre 001 owner') +
    owner(#branch ) supgroup(#branch )
addgroup #B002 +
    data('pisa cust serv centre 002 owner') +
    owner(#branch ) supgroup(#branch )
addgroup #B003 +
    data('rome cust serv centre 003 owner') +
    owner(#branch ) supgroup(#branch )
addgroup #B004 +
    data('rome cust serv centre 004 owner') +
    owner(#branch ) supgroup(#branch )

adduser CSV001 password(start) +
    name('CUST SERV REP 01') +
    data('S/N 000001 - JUNIOR cust serv rep') +
    owner(#B001) +
    dfltgrp(@CSV002)
adduser CSV002 password(start) +
    name('cust serv rep 02') +
    data('S/N 000002 - SENIOR cust serv rep') +
    owner(#B001) +
    dfltgrp(@CSV001)
adduser CSV003 password(start) +
    name('cust serv rep 03') +
```

```

data('S/N 000003 - SENIOR cust serv rep') +
owner(#B001) +
df1tgrp(@CSV001)
adduser CSV004 password(start) +
name('cust serv rep 04') +
data('S/N 000004 - SENIOR cust serv rep') +
owner(#B002) +
df1tgrp(@CSV001)
adduser CSV005 password(start) +
name('cust serv rep 05') +
data('S/N 000005 - JUNIOR cust serv rep') +
owner(#B002) +
df1tgrp(@CSV002)
adduser MAN001 password(start) +
name('BRANCH MANAGER 01 ') +
data('S/N 000006 - BRANCH MANAGER') +
owner(#B003) +
df1tgrp(@MAN001)
adduser REG001 password(start) +
name('REGIONAL MANAGER 01 ') +
data('S/N 000007 - REGIONAL MANAGER') +
owner(#B004) +
df1tgrp(@REG001)

connect CSV001 group(@CSV002) owner(@CSV002)
connect CSV002 group(@CSV001) owner(@CSV001)
connect CSV003 group(@CSV001) owner(@CSV001)
connect CSV004 group(@CSV001) owner(@CSV001)
connect CSV005 group(@CSV002) owner(@CSV002)
connect MAN001 group(@MAN001) owner(@MAN001)
connect REG001 group(@REG001) owner(@REG001)

```

Copying users from SYS1 into the correct owner and access groups

In this step, we prepare for the implementation of PROTECTALL (see 13.5.4, “Planning for PROTECTALL implementation” on page 249) by taking all user IDs currently connected to group SYS1 and connecting them to the more appropriately named RACF groups defined for their ownership and access rights in “Group tree structure” on page 242.

All user IDs connected to SYS1 need to be connected to their new ownership group, generally either a subgroup of #TASKS or one of the staff related ownership groups under #STAFF. They then need to be connected to an access granting group, generally either @SP001, @SP002, or one of the @ prefixed subgroups of #TASKACC in the case of a started task user ID. The access granting groups are prefixed with the @ character and will become the user IDs' new RACF default group instead of SYS1. The order of these steps is important, as we cannot change the users default group to an @ prefixed one until they are connected to it. Only then can we remove the user IDs from their current default group of SYS1.

The removal of users from SYS1 is planned to occur only after our first test IPL of the new structure, but we will show how we generate the relevant RACF commands in this topic.

In this case, we were lucky. In general, the user ID definitions without RACF Name fields seemed to be started task related user IDs. This made our job of separating the staff based user IDs from the task related ones relatively simple. However, if you have any reasonable method of distinguishing tasks from staff, then a similar CARLa program could be used. It may even be worth the effort to manually populate your task user ID definitions with some recognizable string to subsequently process them in code, such as in Example 13-6.

Example 13-6 CARLa to split STCs from staff user IDs

```
newlist type=racf name=firstp
s c=user mask=* cggrpnm=sys1 s=base
sortlist key(8) name(20) creadate last_connect_date

newlist type=racf dd=ckrcmd nopage
s likelist=firstp exists(name)
exclude key=ibmuser
sortlist 'CONNECT ' key(0) ' GROUP(@sp002) OWNER(@sp002)'
sortlist 'ALTUSER ' key(0) ' DFLTGRP(@sp002) OWNER(#useracc)'

newlist type=racf dd=ckrcmd nopage
s likelist=firstp missing(name)
exclude key=ibmuser
sortlist 'CONNECT ' key(0) ' GROUP(@sysstc) OWNER(@sysstc)'
sortlist 'ALTUSER ' key(0) ' DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD
NOOIDCARD'
```

The commands and initial report generated looked similar to Example 13-7.

Example 13-7 Reports and commands from CARLa

```
P R O F I L E   L I S T I N G   13 Apr 2008 18:54
Profile  Name                                CreateDate  LastConDate
$ALLOC$                                24 Jul 2007
$DIRECT$                               24 Jul 2007
$PAGE$                                 24 Jul 2007
$SPool$                                24 Jul 2007
$SYSCKP$                               24 Jul 2007
$SYSWRM$                               24 Jul 2007
$TDISK$                                24 Jul 2007
ADM                                     24 Jul 2007
ADMSERV                                24 Jul 2007
ALEX      ALEX L                           19 Sep 2005 16 Jul 2007
ALEX01    ALEX RESIDENT 1                   18 Sep 2005 26 Feb 2006
ALEX02    ALEX RESIDENT 2                   18 Sep 2005 14 Oct 2005
ALEX03    ALEX RESIDENT 3                   18 Sep 2005 11 Apr 2006
P R O F I L E   L I S T I N G   13 Apr 2008 18:54

      Profile key
CONNECT  ALEX  GROUP(@sp002) OWNER(@sp002)
CONNECT  ALEX01 GROUP(@sp002) OWNER(@sp002)
CONNECT  ALEX02 GROUP(@sp002) OWNER(@sp002)
CONNECT  ALEX03 GROUP(@sp002) OWNER(@sp002)
ALTUSER  ALEX  DFLTGRP(@sp002) OWNER(#sp)
ALTUSER  ALEX01 DFLTGRP(@sp002) OWNER(#sp)
ALTUSER  ALEX02 DFLTGRP(@sp002) OWNER(#sp)
ALTUSER  ALEX03 DFLTGRP(@sp002) OWNER(#sp)

P R O F I L E   L I S T I N G   13 Apr 2008 18:54

      Profile key
CONNECT  $ALLOC$ GROUP(@sysstc) OWNER(@sysstc)
CONNECT  $DIRECT$ GROUP(@sysstc) OWNER(@sysstc)
CONNECT  $PAGE$ GROUP(@sysstc) OWNER(@sysstc)
CONNECT  $SPool$ GROUP(@sysstc) OWNER(@sysstc)
CONNECT  $SYSCKP$ GROUP(@sysstc) OWNER(@sysstc)
CONNECT  $SYSWRM$ GROUP(@sysstc) OWNER(@sysstc)
CONNECT  $TDISK$ GROUP(@sysstc) OWNER(@sysstc)
CONNECT  ADM  GROUP(@sysstc) OWNER(@sysstc)
CONNECT  ADMSERV GROUP(@sysstc) OWNER(@sysstc)
ALTUSER  $ALLOC$ DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
ALTUSER  $DIRECT$ DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
```

```
ALTUSER $PAGE$ DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
ALTUSER $SPOOL$ DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
ALTUSER $SYSCKP$ DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
ALTUSER $SYSWRM$ DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
ALTUSER $TDISK$ DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
ALTUSER ADM DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
ALTUSER ADMSERV DFLTGRP(@sysstc) OWNER(#sysstc) NOPASSWORD NOOIDCARD
```

You can see that we wrote the code to automatically connect the user IDs that had a RACF name field to the @SP002 group, and the user IDs without a name, assumed to be started tasks, were connected to the @SYSSTC group.

We then used the CARLa shown in Example 13-8 to create commands to remove all these user IDs from the SYS1 group. These commands will be run at a later stage after some testing of the new access structures (refer to 13.5.4, “Planning for PROTECTALL implementation” on page 249).

Example 13-8 CARLa to remove user IDs from SYS1 group

```
newlist type=racf dd=ckrcmd nopage
s c=user mask=* cggrpnm=sys1 s=base
exclude mask=ibmuser
sortlist 'REMOVE ' key(0) ' GROUP(SYS1)'
```

You can see the RACF REMOVE command being built in the CARLa sortlist statement, and you can see that we used a CARLa exclude statement to filter out IBMUSER.

In the next section, we describe the steps and processes we used to implement RACF SETROPTS PROTECTALL(WARN) and then PROTECTALL(FAIL).

13.5.4 Planning for PROTECTALL implementation

The basic steps of the plan for PROTECTALL implementation is outlined below:

1. Implement HLQ.** style data set profiles for all data sets where no profile already exists. This implies the existence of a RACF group or user ID for these HLQs as well, so it must also be added if it is not already present. These profiles will initially be implemented with UACC(NONE) and in WARN mode to have no damaging effect. See “Defining high level qualifier (HLQ) data set profiles” on page 250.
2. Place the new RACF access groups into the data set profiles with an access level of ALTER. See “Copying users from SYS1 into the correct owner and access groups” on page 246 for more information.

3. Connect users to new access groups. See “Copying users from SYS1 into the correct owner and access groups” on page 246 for more information.
4. IPL the system to ensure that all tasks and automated functions still retain the necessary access.
5. Remove users from group SYS1. See “Copying users from SYS1 into the correct owner and access groups” on page 246 for more information.
6. Use zSecure Audit Verify Sensitive reports to identify system critical data at risk and reduce the UACC on these data sets and resources. See 13.6, “Post implementation verification reports” on page 254 for more information.
7. Issue SETROPTS PROTECTALL(WARN).
8. Perform a second IPL test.
9. If the IPL test succeeded with little or no RACF access issues, activate SETROPTS PROTECTALL(FAIL).

We now go on to explain how we achieved each of these steps. Some of these steps have been prepared for already, and we will refer to these sections as appropriate.

Defining high level qualifier (HLQ) data set profiles

As a first step, we needed to know what data sets existed on this system. To do this, we used a custom CARLa report to discover all DASD data sets, cataloged or not. We were only interested in the high level qualifier shown in the initial report in Example 13-9.

Example 13-9 Initial CARLa data set analysis report

```
newlist type=dsn name=dsnlist
summary qual qual_is_dataset_profile qual_is_user qual_is_group
```

Using this report, we now had a better idea of how many RACF groups and data set profiles were required. We extended this CARLa program by adding the sections in Example 13-10 to generate the required commands.

Example 13-10 CARLa to create missing user ID data set profiles

```
newlist type=dsn nopage dd=ckrcmd
select likelist=dsnlist qual_is_user=yes qual_is_dataset_profile=no
summary qual(nd) "ADDSD '" | qual(0) | "'.**' OWNER(" | qual(0) | ,
              ") UACC(NONE)" count(nd)
```

```
summary qual(nd) "PERMIT '" | qual(0) | ".*'" ID(@SP002, @SP001, " |,  
"@SYSSTC) ACCESS(ALTER)" count(nd)
```

The code in Example 13-10 on page 250 generates the ADDSD commands for existing user IDs with no data set profile.

Example 13-11 CARLa to create missing group data set profiles

```
newlist type=dsn nopage dd=ckrcmd  
select likelist=dsnlist qual_is_group=yes qual_is_dataset_profile=no  
summary qual(nd) "ADDSD '" | qual(0) | ".*'" OWNER(" | qual(0) |,  
") UACC(NONE) WARN" count(nd)  
summary qual(nd) "PERMIT '" | qual(0) | ".*'" ID(@SP002, @SP001, " |,  
"@SYSSTC) ACCESS(ALTER)" count(nd)
```

The code in Example 13-11 also generates the ADDSD profiles for existing groups.

Example 13-12 generates the ADDGROUP commands and ADDSD commands where no group or user ID previously existed.

Example 13-12 CARLa to create data set profiles where no group or user exists

```
newlist type=dsn nopage dd=ckrcmd  
select likelist=dsnlist qual_is_group=no qual_is_user=no,  
qual_is_dataset_profile=no  
summary qual(nd) "AG " | qual(0) | " OWNER(NEWGRPS) ",  
"SUPGROUP(NEWGRPS)" count(nd)  
summary qual(nd) "ADDSD '" | qual(0) | ".*'" OWNER(" | qual(0) |,  
") UACC(NONE) WARN" count(nd)  
summary qual(nd) "PERMIT '" | qual(0) | ".*'" ID(@SP002, @SP001, " |,  
"@SYSSTC) ACCESS(ALTER)" count(nd)
```

The resulting RACF commands are shown in Example 13-13.

Example 13-13 RACF command examples from CARLa reports

Generated by the user IDs with no dataset profiles code

```
ADDSD 'ADM.*' OWNER(ADM) UACC(NONE)  
ADDSD 'ASSR1.*' OWNER(ASSR1) UACC(NONE)  
ADDSD 'DB.*' OWNER(DB) UACC(NONE)
```

```
PERMIT 'ADM.*' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)  
PERMIT 'ASSR1.*' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)  
PERMIT 'DB.*' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)
```

```

Generated by the groups with no dataset profiles code
ADDSD 'BDT1.**' OWNER(BDT1) UACC(NONE) WARN
ADDSD 'CICSTS.**' OWNER(CICSTS) UACC(NONE) WARN
ADDSD 'ITSC.**' OWNER(ITSC) UACC(NONE) WARN

PERMIT 'BDT1.**' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)
PERMIT 'CICSTS.**' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)
PERMIT 'ITSC.**' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)

```

```

Generated by the no group, no user ID and no dataset profiles code
AG $$DRIVER OWNER(NEWGRPS) SUPGROUP(NEWGRPS)
AG $$IDX OWNER(NEWGRPS) SUPGROUP(NEWGRPS)
AG $$PSL OWNER(NEWGRPS) SUPGROUP(NEWGRPS)

```

```

ADDSD '$$DRIVER.**' OWNER($$DRIVER) UACC(NONE) WARN
ADDSD '$$IDX.**' OWNER($$IDX) UACC(NONE) WARN
ADDSD '$$PSL.**' OWNER($$PSL) UACC(NONE) WARN

PERMIT '$$DRIVER.**' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)
PERMIT '$$IDX.**' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)
PERMIT '$$PSL.**' ID(@SP002, @SP001, @SYSSTC) ACCESS(ALTER)

```

As these commands were directed to the CKRCMD DD by using the DD operand on the NEWLIST statement, we could directly run them from the ISPF interface after we verified them.

We used zSecure Audit report AU.S Status, RACF Resource, and the STARTED class to check that no started task was relying on a connection to group SYS1. Several STARTED class definitions were changed to reflect the move of these user IDs into @SYSSTC as their default group. We also had to ensure that this default group had the same UNIX GID as SYS1 to reduce the possibility of access problems in the UNIX System Services file system. Again, this was easily done using the zSecure Admin panel option RA.U and selecting to display segment data. We then had to overtype the UID field value.

As we had already moved all user IDs from the group SYS1 into alternative access granting groups, @SP001, @SP002, and @SYSSTC, we were confident that no user would lose access as a result of these commands. We were now ready for our first IPL of the system using a totally redesigned RACF group tree and access groups.

A test IPL of the system was performed and it came up with no security problems. We were now ready to see what zSecure Audit had to report about our new system.

Note: An alternative approach to planning PROTECTALL is documented below. You may want to consider this approach if you have more time allocated to the project.

1. Implement 'HLQ.**' style data set profiles with UACC(NONE), and add the WARNING attribute.
2. Talk to representatives of Tivoli Rome Airlines (RACF administrators, Auditors, Information system owners, and line management) to see what permissions they already know are required for the involved job roles to Tivoli Rome Airlines resources, and populate the access control lists accordingly. Optionally you could make an exception for started task user IDs and batch job user IDs (and give them ALTER access) to avoid started tasks or batch jobs from abending. In that case, it would also be a good idea to log successful updates (and higher) to verify the actual access used by started tasks and batch jobs.
3. Connect users to new access groups.
4. Remove users from the SYS1 group.
5. Issue SETROPTS PROTECTALL.
6. From now on, make sure that regular users only have READ access to the Master Catalog. This prevents them from adding new high level qualifiers in the master catalog, which means they cannot introduce new data sets to the system that are not protected by a proper RACF data set profile.
7. Next, use the generated SMF records to verify which access has occurred by using the WARNING attribute. The CARLa select and sortlist statements that do this would be something similar to the following:

```
select type=(80,81,83) event=access(warning)
sortlist date time user intent class resource
```
8. Verify with Tivoli Rome Airlines representatives (RACF administrators, Auditors, and information system owners) whether the reported access by means of the WARNING attribute was legitimate. If it was legitimate, start populating the involved ACLs with the appropriate permit(s) for the involved access groups. I
9. The first SMF report will probably generate a significant number of WARNING access events. However, every subsequent run of the SMF report will (if all is well) have (much) less warning events because the WARNING attribute can no longer be used for those accesses that already have been permitted. Remember that when access is permitted with an insufficient access level, access is denied and the WARNING attribute is no longer checked.

10. So after three to six months, it is most likely that you will not find any or few accesses by means of WARNING anymore and you can safely remove the WARNING attribute from the involved profiles. After removing the WARNING attribute, you can also remove the now obsolete PERMITs with access NONE that you have entered during the last three to six months. This is because if UACC=NONE and NOWARNING, the access that users get that are not on the ACL is NONE anyway.

13.6 Post implementation verification reports

In this section, we describe the reports and processes we used to verify the security status of the system now that all data sets have RACF profiles, and take steps to reduce the high levels of access initially granted.

To help with this task, we primarily used data from the zSecure Audit status reports, and used option AU.S together with the verify reports option AU.V. An initial status report of trusted user IDs was generated and showed an alarming number of trust vectors now on the system as compared to the past. Figure 13-37 shows over 600,000 trust vectors compared to only around 6000 in our initial run of this report, referring back to Figure 13-19 on page 222.

zSecure Admin+Audit for RACF Display Selection				14 s elapsed, 8.3 s CPU
Command ==> _____				Scroll==> <u>CSR</u>
Name	Summary	Records	Title	
_ TRUSTUSR	1	684221	Trusted userids (may bypass security)	
_ AUTHSYS	1	99	Users with system-wide special, operations, auditor,	
_ AUTHUID0	1	25	Users with uid 0	
_ AUTHGRP	0	0	Users with group level special, operations, auditor	
_ SHRDUIDS	1	119	OMVS UIDs shared between RACF users	
_ OMVSNUID	0	0	RACF users with OMVS segment but no UID	
_ SHRDGIDS	1	56	OMVS GIDs shared between RACF groups	
_ OMVSNPID	0	0	RACF groups with OMVS segment but no GID	
_ PROTECT	1	14	Protected users	
_ PWNONE	0	0	Users who can logon without password	
_ PWUID	0	0	Users who can logon with QIDcard	
_ PWINNONE	1	12	Users without password interval	
_ PWINLONG	1	261	Users with password interval > 60 days	
_ PWEXPIRE	1	247	Users with expired passwords	
_ PHEXPIRE	1	1	Users with expired password phrases	
_ PWNOCHG	1	189	Users that never changed password	
_ PHNOCHG	0	0	Users that never changed password phrase	
_ PWAGESUM	1	277	RACF password age overview	
_ PHAGESUM	1	1	RACF password phrase age overview	
_ PWAGEALL	1	277	User Password or Phrase Age: All users	

Figure 13-37 RACF user audit summary

We reviewed the detailed level report for TRUSTUSR, as shown in Figure 13-38, and found that basically the same users were trusted and had similar, that is, far too high, levels of access than before.

Trusted userids (may bypass security)						Line 1 of 249
Command ==> _____						Scroll==> <u>CSR</u>
						15 Apr 2008 13:10
Pri	Complex	Trusted userids				
48	SC76	249				
Pri	Reasons	Userid	Name	RIP	DfltGrp	InstData
— 48	3469	TKRAUS	THOMAS KRAUS	I	@SP002	
— 48	3330	CFZADM	CFZADM	I	@SYSSTC	
— 48	25	????????				
— 10	5942	WELLIE3		I	@SP001	
— 10	4521	SYSPROG	SYSPROG		@SYSSTC	
— 10	3174	WELLIE2		I	@SP001	
— 10	3099	HAIMO	HAIMO		@SP002	
— 10	3092	RC76	RICH CONWAY		@SP002	
— 10	3088	VAINI	JUHA VAINIKAINEN	I	@SP002	
— 10	3085	ALEX	ALEX L	I	@SP002	
— 10	3085	ROGERS	PAUL ROGERS	I	@SP002	
— 10	3084	RMETH	RICARDO METH	I	@SP002	
— 10	3083	FRANCK	FRANCK	I	@SP002	
— 10	3083	WHITE	WHITE	I	@SP002	
— 10	3082	CEA			@SYSSTC	
— 10	3082	CIMUSR1		I	@SYSSTC	
— 10	3082	FRANCK1	FRANCK1		@SP002	
— 10	3082	KMT1	KMT1	I	@SP002	

Figure 13-38 TRUSTUSR report details

After we understood that basically all we had achieved so far was to document and control our users' access using RACF groups rather than system level OPERATIONS, it seemed that the initial analysis report had not taken into account the data sets where *no* RACF definition existed.

After reviewing the AU.S, RACF Resource, SENSTRUST report shown in Figure 13-39, we began to understand what zSecure Audit was trying to tell us.

Sensitive data trustees				Line 1 of 31	
Command ==>				Scroll==> CSR	
				15 Apr 2008 13:10	
Pri	Complex	System	Trust relations		
48	SC76	SC76	684221		
Pri	Sensitivity	Class	Resources	Trust relations	
— 48	Resource	FACILITY	29	87	
— 48	Resource	TSOAUTH	2	2	
— 48	Resource	UNIXPRIV	2	8	
— 48	Surrogate	SURROGAT	250	755	
— 10	Privilege	System	1	207	
— 10	TrustedHome	FSOBJ	2	459	
— 10	TrustedProg	FSOBJ	1	2254	
— 9	APF lib+Lnk	DATASET	54	24694	
— 9	APF library	DATASET	41	9358	
— 9	APF LPAlst	DATASET	8	3652	
— 9	MSTR prmlib	DATASET	4	687	
— 9	MSTR STClb	DATASET	1	229	
— 9	RACF prim	DATASET	1	467	
— 9	System REXX	DATASET	1	229	
— 9	STC joblib	DATASET	1	229	
— 8	APF Linklst	DATASET	19	4337	
— 8	IPL Nucleus	DATASET	1	229	
— 8	LPA list	DATASET	15	3427	

Figure 13-39 SENSTRUST report

We first noted the large number of trust associations for the APF lib+Lnk line item in the report, and realized that our decision to define RACF profiles for all data sets where the profile had previously been missing was a major cause of the increase in trust vectors on this system.

Closer examination, especially by running the AU.V verify sensitive report, also revealed that we had inadvertently granted our user base ALTER level access on over 3000 data set profiles related to HFS file definitions. zSecure Audit considers all HFS files as potentially sensitive data, and had reported a huge increase in trust vectors for these previously undefined data sets.

The process we went through in discovering these factors is itself a good example of zSecure Audit's ability to zero in on system security concerns. Were it not for the wide variety of reports, each looking at the overall security problems from different points of view, this would have been a difficult issue to diagnose. Using a combination of the RACPRAUD, SENSTRUST, SENSPROF, and Verify reports, we quickly discovered the actions we would need to take to properly reduce the trust levels on our new and improved RACF database.

13.6.1 Reducing trust levels

The *verify sensitive* report lists all data sets that zSecure Audit considers sensitive from the point of view of trust. Only highly trusted users should have the access to update, or in some cases, even read the contents of these data sets. Refer to 14.1.2, “Sensitive data set analysis” on page 273 for an explanation of sensitivity levels and auditing requirements.

Verify sensitive report

We ran the *verify sensitive* report and generated the recommended RACF profiles for over 3000 data sets. An important point to appreciate here is that although we initially created many of the new HLQ.** data set profiles in WARN mode (see “Defining high level qualifier (HLQ) data set profiles” on page 250), the *verify sensitive* process copies these profiles, reducing the UACC and removing WARN where necessary and adding the correct audit settings. This means that while our user community may still have ALTER access to non-sensitive data, they no longer have this potentially damaging level of access to system critical files. Using this approach, we have been able to preserve user access, and whatever business requirements this was previously used for, and at the same time properly secure the critical resources of this system.

We also took this opportunity to reduce the access levels granted to most system users, leaving only the @SP001 Senior Systems Programmers group with ALTER access. This was done with some CARLa similar to the one shown in Example 13-14.

Example 13-14 CARLa to reduce access levels to sensitive data sets

```
NEWLIST TYPE=SENSDSN DD=CKRCMD NOPAGE
Sortlist  "PE '' | DSN(0) | '' GEN id(@sysstc) ac(read)"
Sortlist  "PE '' | DSN(0) | '' GEN id(@sp002) ac(read)"
```

It seemed that by default the TYPE=SENSDSN report did not list HFS data sets, so another CARLa code snippet was created to specifically reduce access to HFS data sets by including the line:

```
select sensitivity='HFS dataset'
```

We also coded an exclusion for the currently mounted HFS files with code similar to the following:

```
exclude dsn=(OMVS.ZOSR19.Z19RE1.ROOT,
             OMVS.SC76.PP)
```

We used the zSecure Audit option AU.S, MVS extended, MOUNT report to determine the list of mounted file systems and coded these into the CARLa from Example 13-14 on page 257 using the exclude syntax above. We then ran the generated commands to reduce access levels from ALTER to READ.

Profile audit concerns

We then addressed a number of RACF profile audit concerns highlighted by the SENSPROF and RACPRAUD from the AU.S RACF resource reports. By removing users from the access lists and changing UACC on these profiles, we once again substantially reduced the trust levels.

In Figure 13-40, you can see that the trust vectors for users is now under 10,000, down from over half a million with only a small number of changes made to critical profiles.

zSecure Admin+Audit for RACF Display Selection			9 s elapsed, 3.3 s CPU
Command ==> _____			Scroll==> CSR
Name	Summary	Records	Title
_ TRUSTUSR	1	9477	Trusted userids (may bypass security)
_ AUTHSYS	1	10	Users with system-wide special, operations, auditor,
_ AUTHUID0	1	6	Users with uid 0
_ AUTHGRP	0	0	Users with group level special, operations, auditor
_ SHRDUIDS	1	110	OMVS UIDs shared between RACF users
_ OMVSNUID	1	9	RACF users with OMVS segment but no UID
_ SHRDGIDS	1	57	OMVS GIDs shared between RACF groups
_ OMVSNPID	0	0	RACF groups with OMVS segment but no GID
_ PROTECT	1	14	Protected users
_ PWNONE	0	0	Users who can logon without password
_ PWOID	0	0	Users who can logon with OIcard
_ PWINNONE	1	12	Users without password interval
_ PWINLONG	1	261	Users with password interval > 60 days
_ PWEXPIRE	1	269	Users with expired passwords
_ PHEXPIRE	1	1	Users with expired password phrases
_ PWNOCHG	1	211	Users that never changed password
_ PHNOCHG	0	0	Users that never changed password phrase
_ PWAGESUM	1	299	RACF password age overview
_ PHAGESUM	1	1	RACF password phrase age overview
_ PWAGEALL	1	299	User Password or Phrase Age: All users

Figure 13-40 Reduced trust vectors

We had also removed all UACC and changed this to ID(*) permits using the CARLa code in Example 13-15 for data sets, and the code in Example 13-16 for general resources.

Example 13-15 CARLa to replace UACC with ID() access list entries in data set class*

```
newlist type=racf f=ckrcmd nopage
s c=dataset s=base mask=** uacc>none
sortlist "permit ' ' | key(0) | '' generic id(*) access(" | uacc(0) | ")"
sortlist "altdsd ' ' | key(0) | '' generic uacc(none)"
```

Example 13-16 CARLa to replace UACC with ID() access list entries in general resource classes*

```
newlist type=racf f=ckrcmd nopage
s c=general s=base mask=** uacc>none
x c=(ejbrole,digtcert,started,xfacilit,nodes)
sortlist "permit " key(0) " class(0) id(*)" , "access(" | uacc(0) | ")"
sortlist "ralt" class(0) key(0) "uacc(none)"
```

These two steps removed all trust status issues for the JES undefined user ID ????????, substantially reducing our total trust vector count again.

System privileges

We had also started to reduce the use of system OPERATIONS and SPECIAL attributes, and remove UID(0) from as many user IDs as possible. You can see in Figure 13-41 that not only had the overall number of trust vectors been drastically reduced, but perhaps more importantly, the audit priority of those remaining was now 10 or lower, indicating housekeeping required or otherwise normal access levels. Our previous high priority items had been at audit level 49, representing serious exposures.

Also, you will see from the list of users now assigned high trust levels, that there are only systems programmers (default group @SP001), and the main security administration team, ourselves, plus highly trusted system tasks, as would be expected.

Trusted userids (may bypass security)						Line 1 of 245	
Command ==> _____						Scroll==> CSR	
						16 Apr 2008 16:53	
Pri Complex		Trusted userids					
10 SC76		245					
Pri	Reasons	Userid	Name	RIP	DfltGrp	InstData	
— 10	1507	DFS			DFSGRP		
— 10	538	HAIMO	HAIMO		@SP001		
— 10	531	RC76	RICH CONWAY		@SP001		
— 10	381	CFZADM	CFZADM	I	@SYSSTC		
— 10	122	TCPIP		P	TCPGRP		
— 10	114	WEBSRV			IMWEB		
— 10	113	OMVSKERN		P	OMVSGRP		
— 10	112	BPXROOT		IP	OMVSGRP		
— 10	11	CONWAYM	MONIQUE CONWAY		@ZSEC001		
— 10	11	HAHNSM	MARK HAHN		@ZSEC001		
— 10	11	JPEASE	JAMIE PEASE		@ZSEC001		
— 10	11	LILIXIE	LILI XIE		@ZSEC001		
— 10	11	MEADOWS	REESE MEADOWS		@ZSEC001		
— 10	10	MCAIRNS	MICHAEL CAIRNS		@ZSEC001		
— 9	1458	SYSPROG	SYSPROG		@SYSSTC		
— 9	29	LDAPSRV	LDAP ID		@SYSSTC		
— 9	27	SMS			STCGROUP		
— 9	26	C2PSUSER		P	@ALERT		

Figure 13-41 Reduced trust priority findings

As the latest TRUSTUSR and SENSTRUST reports show, our trust levels were starting to come down.

RACF system settings report

In 13.4.1, “Status audit reports” on page 213, we reported on the RACF SETROPTS settings and revealed a number of vulnerabilities. We have addressed almost all of these now, as shown in Figure 13-42.

SETROPTS settings - audit concerns					Line 1 of 1	
Command ==> _____					Scroll==> <u>CSR</u>	
					17 Apr 2008 19:41	
Pri	Complex	System	Count			
34	SC76	SC76	1			
Pri	Parameter		Value	Audit concern		
__	34 PROTECTALL		Warning	Warnings do not prevent unauthorized a		
***** Bottom of Data *****						

Figure 13-42 Improved RACF SETROPTS

All that remains is to conduct another test IPL just to make sure we had not removed a necessary authority from any critical system service. Then we can implement SETROPTS PROTECTALL(FAIL) and close this last vulnerability.

13.6.2 RACF group structure

After implementing the structures described in 13.5.3, “Implementing an improved RACF group tree structure” on page 241 and removing the user IDs described in “Copying users from SYS1 into the correct owner and access groups” on page 246, we can see our new high level RACF group tree structure in Figure 13-43 and Figure 13-44 on page 263.

```

zSecure Admin+Audit for RACF GROUP TREE DISPLAY                                0 s elapsed, 0.1 s CPU
Command ==> _____ Scroll==> CSR
                                                                 15 Apr 2008 17:04

Complex Groups
SC76          3351

Group structure
  SYS1
  #APPS
  #DEV
  #MNT
  #PRD
  #TST
  #FUNCACC
  #BATACC
  #SECACC
  @A#APPS
  @A#BRNCH
  @A#B001
  @A#STAFF
  @A#STECH
  @CKG1FLL
  @CKG2GRP
  @CKG3USR
  @CKG4ACL

```

Figure 13-43 High level group tree structure

```
zSecure Admin+Audit for RACF GROUP TREE DISPLAY                               Line 1 of 11
Command ==> _____ Scroll==> CSR
                                     15 Apr 2008 17:04
Group structure                               Lvl Subgrp Connct SupGroup Owner   X
SYS1                                         1      9      1 ..... IBMUSER  X
User      Auth      R SOA AG Uacc      Name                               InstData
- IBMUSER  JOIN      -   -   - READ      -                               -
- SubGroup
- SYSCTLG
- VSAMDSET
- #APPS
- #FUNCACC
- #RACFRES
- #SYSTEM
- #USERS
- NEWGRPS
- TMPGRPS
***** Bottom of Data *****
```

Figure 13-44 Subgroups of SYS1 and user IDs connected to SYS1

13.7 Conclusion

Although this system is still not ideal, it is now approaching a level where ongoing monitoring and audit reporting can be implemented. Using information from periodic audit reports, it will now be possible to actually start reducing access and controlling users' authority in a way that was just not available prior to the RACF controls described above being put into place.

In Chapter 14, "Implementation phase II" on page 265, we show how to implement these necessary auditing and ongoing monitoring processes.



Implementation phase II

In this chapter, we describe phase two of the TRA security improvement program. In this phase, we focus primarily on enhancing the auditing and monitoring processes in place for the z/OS and RACF configuration.

This chapter covers the following topics:

- ▶ Audit reporting
- ▶ Ongoing monitoring
- ▶ Conclusion

14.1 Audit reporting

In the following sections, we describe additional uses of zSecure to provide enhanced auditing and real-time monitoring capabilities for z/OS and RACF.

In our current age of complex and sophisticated IT systems interconnecting on many levels, the ability to audit and monitor activity on each participating system is of critical importance.

The mainframe in your organization is one of the systems participating fully in this information systems network, and typically contains a historical repository of the most critical business data that is essential to your ongoing operations. It is obviously essential that mainframe based systems implement leading edge audit, intrusion prevention, and general security monitoring procedures. The Security zSecure Suite can help you implement these processes, and continue to support your ongoing security management in a flexible and extensible manner to easily meet new requirements.

Companies tend to fall into one of two categories in respect to their audit reporting regimes:

- ▶ Report on nothing, or not enough
- ▶ Report on everything

The first category is rapidly becoming an unacceptable business practice, especially as companies integrate more closely with their suppliers and partners. The second category is impractical, and leads to issues such as data management problems with the ever growing volume of audit data, needle in the haystack problems when you need to analyze the data, and system performance issues introduced by generating so much data in the first place.

zSecure helps you to generate just the right data and reports, and in this chapter we will show you how.

14.1.1 Using supplied reports

After establishing the basic audit settings in 13.5, “Implementing initial improvements in system security posture” on page 231, we started to see relevant SMF records generated. We can now use these to begin reducing access on this system to an acceptable level.

To review these SMF records, we used the zSecure suite main ISPF panel option EV. The report categories provided with this menu option are shown in Figure 14-1.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - Main menu				
Option ==> _____				
				More: +
SE	Setup	Options and input data sets		
RA	RACF	RACF Administration		
AU	Audit	Audit security and system resources		
EV	Events	Event reporting from SMF and other logs		
U	User	User events from SMF		
G	Group	Group events from SMF		
D	Data set	Data set events from SMF		
R	Resource	General resource events from SMF		
F	Filesystem	Unix filesystem events from SMF and other logs		
I	IP	IP events from SMF and other logs		
1	SMF reports	Predefined analysis reports		
2	RACF events	RACF logging for specific events		
4	DB2	DB2 events from SMF		
C	Custom	Custom report		
CO	Commands	Run commands from library		

Figure 14-1 EV - Event reporting main categories

The categories for User, Group, Data set, Resource, Filesystem, TCP/IP events and DB2 events, all present a similar panel where filters and selection criteria for a report from SMF can be selected. The selection panel for User events is shown in Figure 14-2.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - Events - User Selection				
Command ==> _____ _ start panel				
Show records that fit all of the following criteria:				
Userid	_____	(userid or EGN mask)		
Owned by	_____	(group or userid, or EGN mask)		
System	_____	(system name or EGN mask)		
Name	_____	(name/part of name, no filter)		
Installation data	_____	(scan of data, no filter)		
Jobname	_____	(job name or EGN mask)		
Terminal	_____	(Terminal id or EGN mask)		
Advanced selection criteria				
<input type="checkbox"/> User actions	<input type="checkbox"/> User attributes	<input type="checkbox"/> Date and time		
<input type="checkbox"/> Data set selection	<input type="checkbox"/> HFS selection	<input type="checkbox"/> Resource selection		
<input type="checkbox"/> DB2 selection				
Output/run options				
<input type="checkbox"/> Include detail	<input type="checkbox"/> Summarize	<input type="checkbox"/> Specify scope		
<input type="checkbox"/> Output in print format	Customize title	Send as e-mail		
Run in background	Sort differently			

Figure 14-2 EV - User events selection

This panel is similar to other RA panels for specifying report criteria on user IDs, profiles, or groups. Similar options for advanced selection criteria and for various output formats, such as email or batch submission, are available.

If you run a report from here using online rather than batch submission, you will quickly gain an appreciation of just how efficient the SMF reporting component of IBM Security zSecure is. In many cases, reports that can take hours using alternative methods have their results returned literally in seconds. For analyzing large volumes of SMF data, for example, daily or weekly tape summaries, we would still recommend batch submission, but even when using batch submission, these reports run quickly.

The categories for SMF reports, RACF reports, and Custom have more specific characteristics. The SMF option presents a panel of standard reports, as shown in Figure 14-3, while the RACF option provides the panel shown in Figure 14-4.

Menu	Options	Info	Commands	Setup	StartPanel
zSecure Admin+Audit for RACF - Events - SMF reports					
Option ==> _____					
1	Exceptions	RACF exception report			
2	Stat hour	RACF statistics by hour (very wide report)			
3	Stat time	RACF statistics by time			
4	Stat day	RACF statistics by weekday			
5	Revoke/resume	RACF revoke/resume summary			
9	Job viols	Dataset violations by batch jobs			
A	APPC conv	APPC conversation summary			

Figure 14-3 EV - Event reporting SMF selection

Menu	Options	Info	Commands	Setup	StartPanel
zSecure Admin+Audit for RACF - Events - RACF events					
Command ==> _____					
Enter "/" to select report(s)					
=	All events	Overview of all following RACF events (except IPL)			
-	Logging	RACF logging of all events except RACINIT			
-	Not normal	RACF access not due to normal profile access			
-	Warnings	RACF access due to profiles in warning modes			
-	Violations	RACF access violations			
-	Commands	RACF command auditing			
-	CKGRACF	zSecure Admin CKGRACF commands			
-	IPL RACF	RACF initialization			

Figure 14-4 EV - Event reporting RACF selection

A difference between the SMF and RACF sets of the reports is the way that the output reports are presented. In most of the SMF and RACF report sub-categories, you will be presented with a dynamically generated set of reports that you can then drill down into for additional details.

Figure 14-5 shows the RACF report for *All events* in summary form. Each summary report may now be reviewed for more detail, as shown in Figure 14-6.

SMF record RACF processing and audit records			2 s elapsed, 0.4 s CPU	
Command ==>			Scroll==> CSR	
			15Apr08 15:30 to 15Apr08 21:50	
Event	Q	Count	Event description	
__ RACINIT	0	24	Racinit (Success:Successful initiation)	
__ RACINIT	8	23	Racinit (Success:Successful termination)	
__ ACCESS	0	339	Resource access (Success:Successful access)	
__ ACCESS	1	4	Resource access (Failure:Insufficient authority)	
__ DEFINE	0	1	Define resource (Success:Successful definition)	
__ ADDSD	0	1	Addsd command (Success:No violations detected)	
__ ADDGROUP	0	1	Addgroup command (Success:No violations detected)	
__ ALTDSD	0	1392	Altdd command (Success:No violations detected)	
__ ALTGROUP	0	39	Altgroup command (Success:No violations detected)	
__ ALTUSER	0	186	Altuser command (Success:No violations detected)	
__ CONNECT	0	2	Connect command (Success:No violations detected)	
__ PERMIT	0	6495	Permit command (Success:No violations detected)	
__ RALTER	0	14	Ralter command (Success:No violations detected)	
__ RDELETE	0	1	Rdelete command (Success:No violations detected)	
__ REMOVE	0	4	Remove command (Success:No violations detected)	
__ SETROPTS	0	43	Setropts command (Success:No violations detected)	
__ GENERAL	0	11	General auditing (Unclear:General audit record write	
***** Bottom of Data *****				

Figure 14-5 EV - RACF all events SMF summary

SMF record RACF processing and audit records				Line 1 of 4
Command ==> _____				Scroll==> <u>CSR</u>
				15Apr08 15:30 to 15Apr08 22:14
Event	Q	Count	Event description	
ACCESS	1	4	Resource access (Failure:Insufficient authority)	
Date	Time	Description		
__ 15Apr2008	20:00:00	RACF ACCESS violation for IBMUSER: (READ,NONE) on DATASET		
__ 15Apr2008	21:07:19	RACF ACCESS violation for C2PSUSER: (READ,NONE) on DATASE		
__ 15Apr2008	21:11:12	RACF ACCESS violation for C2PSUSER: (READ,NONE) on DATASE		
__ 15Apr2008	22:07:19	RACF ACCESS violation for C2PSUSER: (READ,NONE) on DATASE		
***** Bottom of Data *****				

Figure 14-6 EV - RACF access violation summary

We can then drill down even further to see full details of the SMF event; the first panel of details is shown in Figure 14-7. There are usually at least two panels of detail available for this kind of report.

```

SMF record RACF processing and audit records                                     Line 1 of 47
Command ===> _____ Scroll===> CSR
                                                                 15Apr08 15:30 to 15Apr08 22:14

Description
RACF ACCESS violation for IBMUSER: (READ,NONE) on DATASET SYS1.PARMLIB

Record identification
Jobname + id: SMFDUM76
SMF date/time: Tue 15 Apr 2008 20:00:00.55
SMF system:    SC76      record type: 80    record no: CKR1SM00 75

Event identification
RACF event description      Resource access (Failure:Insufficient
RACF event description      authority)
RACF event qualifier        1
RACF descriptor for event   Violation
RACF reason for logging     Resource
SAF authority used          Normal
Access intent               READ
Access allowed              NONE
Unix Audit Function Code
Unix Access Intent

```

Figure 14-7 EV - RACF events - Specific violation detail

We now demonstrate a simple use of EV.C Custom SMF reporting after selecting the live SMF data sets as our input source. You must select some SMF data for you to report on before you can run any of the EV reports. Do this using either option SE.1 or the SETUP line command,

In the first panel presented after selecting EV.C, we specify some SMF data selection criteria, as shown in Figure 14-8.

```

Menu  Options  Info  Commands  Setup
-----
                                zSecure Admin+Audit for RACF - Events - Custom
Command ==> _____

Newlist type . . . . . SMF _____

Enter up to 3 SELECT condition sets (use EGN masks)
Select  racfauth=operations
Select  _____
Select  _____

```

Figure 14-8 EV - Custom event selection

We then specify some variables to display in the output report shown in Figure 14-9.

```

Menu  Options  Info  Commands  Setup
-----
                                zSecure Admin+Audit for RACF - Events - Custom
Command ===> _____

Enter output variables
Display  user recorddesc
        _____
        _____

Enter summary variables
Summary  user date time
        _____
        _____

Or select one or more default reports
_  RACF processing and audit
_  Job activity
_  Data set activity
_  ICF catalog activity
_  VSAM catalog activity
_  Basic report for unsupported records
_  DB2 audit
_  Firewall activity
_  Unix filesystem activity
_  IP connection activity

```

Figure 14-9 EV - Custom display variables

Upon pressing Enter to process the report, we see a summary level as requested by user ID and time stamps in Figure 14-10.

zSecure Audit user defined SMF report				2 s elapsed, 0.1 s CPU
Command ==> _____				Scroll==> <u>CSR</u>
				15Apr08 22:26 to 15Apr08 22:26
User	Date	Time	Count	
— MCAIRNS	15 Apr 2008	22:26	2	
— MCAIRNS	15 Apr 2008	22:26	1	
— MCAIRNS	15 Apr 2008	22:26	1	
— MCAIRNS	15 Apr 2008	22:26	1	
— MCAIRNS	15 Apr 2008	22:26	1	
***** Bottom of Data *****				

Figure 14-10 EV - Custom report summary

Upon selecting a line item from the report, we see the *recorddesc* variable displayed for the selected SMF event in Figure 14-11.

zSecure Audit user defined SMF report		Line 1 of 1
Command ==> _____		Scroll==> <u>CSR</u>
		15Apr08 22:26 to 15Apr08 22:26
Description		
RACF ACCESS success for MCAIRNS: (READ,ALTER) on DATASET SYS1.PARMLIB		
***** Bottom of Data *****		

Figure 14-11 EV - Custom report details

While this is a simple CARLa report, it serves to demonstrate the power of this reporting tool to create reports that are specific to the unique requirements of your organization or project. In later sections, we expand on some of the basics presented here, to provide audit reporting solutions for the TRA systems.

14.1.2 Sensitive data set analysis

As a processor of credit card payments, TRA is subject to Payments Card Industry (PCI) audit reporting and security implementation requirements. Unfortunately, they have been quite lacking in this respect, and to date have been paying fines to their payment card service providers in lieu of rectifying this situation.

Delft Transport implements PCI standards in all their internal IT systems, and has expertise in this respect that can immediately be applied to TRA mainframe system. A specific requirement of PCI is the monitoring of access to sensitive system configuration data, and reporting on this access to management.

In a z/OS system, there are two necessary components to any report on access to sensitive configuration data sets. First, we need to determine at any point in time what the names of each configuration data set are. Second, we need to review the SMF data for access to these data sets.

There is another consideration as well. Sensitive data may be accessed quite legitimately and innocuously much of the time. A user reading some sensitive data may not in itself be a reportable incident, but a user updating the same sensitive data would be. Hence, we use the term *risk level* in context of sensitive data. You can think of risk level as the level of access that is a risk in terms of any specific piece of sensitive data.

For example, it may not be considered sensitive if a user reads the panel libraries that make up the base ISPF product interfaces. Reading these data sets is a normal part of any user logging onto the system. However, if some user were to update these data sets, then they might change the system behavior, even introducing trojan code. Hence, for the ISPF environment libraries in general, the sensitivity or risk level is update, not read.

The situation with the RACF database serves as another example. With the RACF database, read is the risk level. Any user who can read the database can therefore make a copy of the database, and subject this copy to offline brute force password attacks, eventually cracking the password of at least some defined user IDs. So the risk level for the RACF database is read access.

Any report that lists access to sensitive data must take all of this into account. It must determine which data sets are sensitive, what their risk level is, and then report only on access that has occurred at or above the associated risk level for each data set. Remember, the list of sensitive resources is dynamic by nature, and can change dynamically or with system maintenance. Any hard coded list of sensitive resources will soon be out of date.

Fortunately, this is easy with zSecure. The JCL below implements a 2-pass CARLa report that does exactly what is required. A 2-pass CARLa report first generates CARLa statements in the first pass and then executes that generated CARLa in the second pass, as shown in Example 14-1.

Example 14-1 A 2-pass CARLa to automate reporting of access to sensitive resources

```
//STEP1 EXEC PGM=CKRCARLA,REGION=OM
//STEPLIB DD DISP=SHR,DSN=CKR.SCKRLOAD
//SYSPRINT DD SYSOUT=*
//CKFREEZE DD DISP=SHR,DSN=CKRU.DAILY.CKFREEZE(0)
//CKR2PASS DD DISP=(NEW,PASS),DSN=##PASSFILE,
//          UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN DD *
```



```
NEWLIST OUTLIM=1 DD=CKR2PASS NOPAGE
SORTLIST "SUPPRESS CKFREEZE",
  / "N TYPE=SMF",
  / " S EVENT=ACCESS (,"
NEWLIST TYPE=SENSDSN DD=CKR2PASS NOPAGE
EXCLUDE MISSING(RISK)
SORTLIST "(INTENT>=" | RISK "DSN=' " | DSN(0) | "') OR,"
NEWLIST OUTLIM=1 DD=CKR2PASS NOPAGE
SORTLIST " DATE=NEVER)",
  / "SORTLIST INTENT DESC(EXPLODE,9) DATE TIME JOBNAME",
  / "SUM DSN(30) * USER USER:PGMRNAME"
//STEP2 EXEC PGM=CKRCARLA,REGION=OM
//STEPLIB DD DISP=SHR,DSN=*.STEP1.STEPLIB
//CKR2IN DD DISP=OLD,DSN=*.STEP1.CKR2PASS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALLOC SMF
INCLUDE DD=CKR2IN
/*
```

This job produced the report shown in Figure 14-12.

Dataset	User	Name	Intent	Dsc	Date	Time	Jobname
SYS1.PARMLIB	MCAIRNS	MICHAEL CAIRNS					
			UPDATE	Success	14 Apr 2008	01:25	MCAIRNS
			UPDATE	Success	14 Apr 2008	01:33	MCAIRNS

Figure 14-12 Sensitive data set access report

Once again, you can see the real power of the CARLa reporting engine, combined this time with the CKFCOLL utility to determine the set of sensitive resources on your system. Any alternative approach to solving this problem would require regular maintenance to maintain a list of sensitive resources, and would need to incorporate knowledge of their risk levels to avoid creating too much irrelevant data in the output.

14.1.3 XML format audit reporting using CARLa

As part of the TRA security improvement program, there is a requirement to introduce a suite of security reports in a user friendly format.

IBM zSecure can produce reports in XML format using the CARLa language. This enables you to produce web-based reports instead of the normal text output that is sometimes difficult for those unused to traditional mainframe systems to interpret. Reports that are produced in XML format can also be viewed by other programs, such as Microsoft Excel and other XML services.

IBM zSecure reports that are produced in XML format can be routed as follows:

1. As an email attachment.
2. Stored within a directory in z/OS UNIX. This enables your organization to view reports directly through a web browser if you have a web server running in your z/OS system.
3. Stored in a mainframe data set.

To produce reports in XML format, you need to specify some options in your CARLa program. In the following three examples, we will show you how to produce XML reports using the routing methods mentioned.

The CARLa program shown in Example 14-2 generates a report and sends it as an email attachment using Simple Mail Transfer Protocol (SMTP).

Example 14-2 CARLa to generate an XML report and send as an email attachment

```
fileoption dd=C2REMAIL fileformat=XML xml_datadict,  
    xml_stylesheet=imbed(m=C2RXSL01) encoding=UTF-8  
option dd=C2REMAIL mailto=jamie_pease@uk.ibm.com,  
    from=itsecops@uk.ibm.com outputformat=ATTACH  
  
newlist dd=C2REMAIL type=RACF name=CSV001,  
    tt="Please re-validate the following group connections",  
    subtitle='Notes: Edit this table with Excel; add your comments in the  
    field provided; save the table; email it to ITSECOPS'  
  
select class=GROUP segment=BASE key=@CSV001  
    define comment('') boolean  
sortlist key('Group',8) instdata('Description',40),  
    userid('User ID',8) userid:name userid:dfltgrp userid:owner,  
    comment('Reviewers comments',20)
```

You need to be aware of some XML related keywords that you can use in your CARLa program. These are documented in Table 14-1.

Table 14-1 XML related keywords

XML Keyword	Description
FILEOPTION	Sets file options for output DDname.
FILEFORMAT	Specifies the format of the output file.
XML_DATADICT	Turns on data dictionary generation.
XML_DTD	Turns on document type definitions generation.
XML_STYLESHEET	Transforms the content of XML data.
ENCODING	Specifies character encoding.
OUTPUTFORMAT	Specifies the method of including results in an email.

Note: You must have SMTP active on your z/OS system to use the email function. SMTP is set up by your system programmer.

The report will be sent as an email attachment, as shown in Figure 14-13.

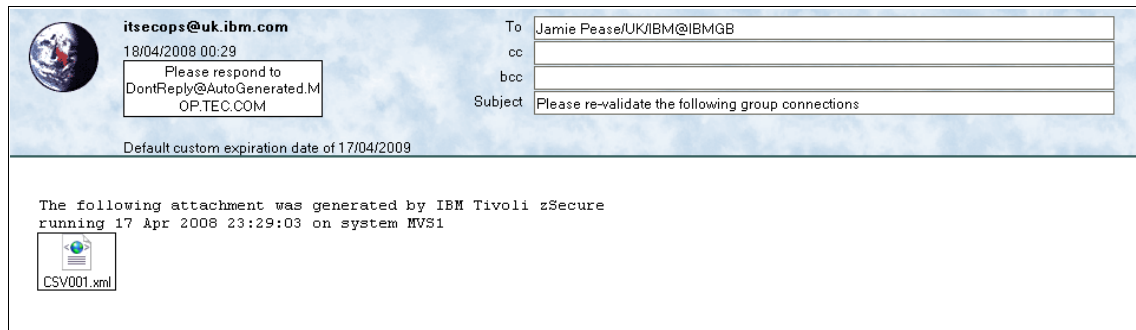


Figure 14-13 Email containing the XML report attachment

When the attachment is opened, the report will be displayed in your web browser, as shown in Figure 14-14.

Please re-validate the following group connections - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites RSS Print Mail Bluetooth

Please re-validate the following group connections

Notes: Edit this table with Excel; add your comments in the field provided; save the table; e-mail it to ITSECOPS

Group	Description	User ID	Name	DfltGrp	Owner	Reviewers comments
@CSV001	SENIOR CUSTOMER SERVICE REP ACCESS	CSV002	CUST SERV REP 02	@CSV001	#B001	
		CSV003	CUST SERV REP 03	@CSV001	#B001	
		CSV004	CUST SERV REP 04	@CSV001	#B002	
		BRNCH05	BRANCH USER 05	@CSV001	#B002	

My Computer

Figure 14-14 IBM zSecure XML report

You can edit the table by exporting it to Microsoft Excel just by right-clicking the table. You can see the Excel report in Figure 14-15. We have added a comments field to the table to allow the reviewer to add any comments to justify the re-validation of this access. This field was defined as a variable (called *comment*) in the CARLa program shown in Example 14-2 on page 276.

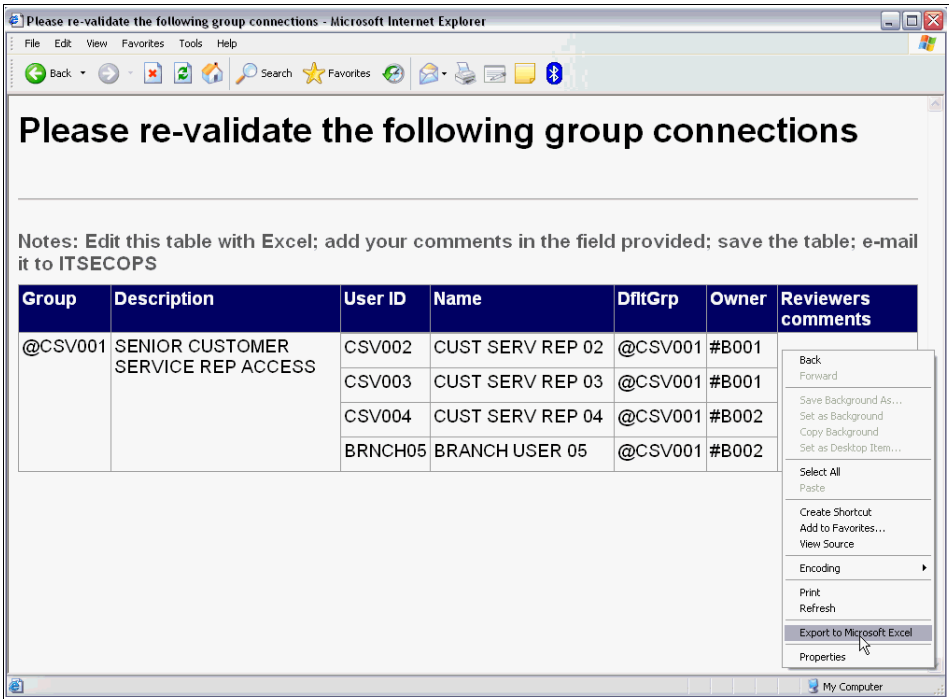


Figure 14-15 Exporting the report to Microsoft Excel

Figure 14-16 shows the Excel spreadsheet containing the data from the table. When the reviewer has completed the re-validation, they can simply send the Excel spreadsheet to the security administration team with a list of changes required or other comments.

	A	B	C	D	E	F	G
	Group	Description	User ID	Name	DfltGrp	Owner	Reviewers comments
1	@CSV001	SENIOR CUSTOMER SERVICE REP ACCESS	CSV002	CUST SERV REP 02	@CSV001	#B001	
2			CSV003	CUST SERV REP 03	@CSV001	#B001	
3			CSV004	CUST SERV REP 04	@CSV001	#B002	
4			BRNCH05	BRANCH USER 05	@CSV001	#B002	
5							
6							
7							

Figure 14-16 Report exported to Microsoft Excel

This is just one of many examples of how IBM zSecure can produce user friendly and easily accessible security reports. The XML report shown in Figure 14-14 on page 278 is a good method of sending data owners a regular report on which groups or users have access to resources they own. This assists the data owner to make an informed and responsible decision when they re-validate or cancel these access privileges.

The next CARLa program, shown in Example 14-3, shows you how to produce a CARLa report and store it within a directory in z/OS UNIX System Services. If you have a web server running on your z/OS system, you can view these reports directly by accessing the appropriate URL from a web browser. The web server is set up by your systems programmer.

You may want to consider creating an index web page that contains links to selectable IBM zSecure reports and allowing authorized users such as auditors and security administrators to view them. The reports can be created and updated automatically on a daily basis by scheduling your CARLa programs to run using IBM Tivoli Workload Scheduler for z/OS or your equivalent job scheduler package. You could also easily establish pages or links for historical data from previous periodic reports using this methodology.

Example 14-3 CARLa program to route an XML report to z/OS UNIX

```
alloc dd=REPORT type=OUTPUT path='/u/zsecure/reports/racfvio1.xml'
fileoption dd=REPORT fileformat=XML xml_datadict,
    xml_stylesheet=imbed(m=C2RXSL01) encoding=utf-8
```

```
newlist dd=REPORT type=SMF name=VIOLS tt="Daily RACF Violations",
    subtitle='Notes: Please review and refer to SECCOMP with any queries'
```

```
select type=(80,81,83) event=access(VIOLATIONS)
sortlist date time user resource(44) class profile(44) intent access
```

In Figure 14-17, the output from the CARLa program (as shown in Example 14-3 on page 280) is now stored in z/OS UNIX System Services and is being viewed directly through a web browser.

Daily RACF Violations

Notes: Please review and refer to SECCOMP with any queries

Date	Time	User	Resource	Class	Profile	Intent	Allowed
14 Apr 2008	12:27:11.97	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	12:27:27.27	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	13:27:11.97	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	13:27:27.31	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	14:27:11.97	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	14:27:27.34	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	15:48:36.05	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	15:48:41.00	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	16:08:22.83	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE
14 Apr 2008	16:08:27.79	C2PSUSER	CKR.READALL	XFACILIT	CKR.READALL	READ	NONE

Figure 14-17 Daily violations report stored in z/OS UNIX

The CARLa program in Example 14-4 routes output to a mainframe data set.

Example 14-4 CARLa program to route an XML report to a mainframe data set

```
alloc dd=MYXML type=OUTPUT dsn=ZSECURE.ADMINS.XML
fileoption dd=MYXML fileformat=XML xml_datadict,
xml_stylesheet=imbed(m=C2RXSL01) encoding=utf-8
```

```
newlist dd=MYXML type=RACF name=ADMINS,
tt="List of users with system-wide attributes - please re-validate",
subtitle='Notes: Edit this table with Excel; add your comments in the
field provided; save the table; email it to SECADM'
```

```
select class=USER segment=BASE special or operations or auditor
```

```

define comment('') boolean
define unused('Unused',8,hb) boolean where
    last_connect_date<today-300

sortlist key('Userid',10) name(22) dfltgrp(10) owner(10),
    spec(1,hb) oper(1,hb) auditor(1,hb),
    last_connect_date unused passdate revoke('Revoked',7,hb),
    comment('Reviewers Remarks',20)

```

For further information about XML support, please refer to Chapter 1, “XML support within IBM Security zSecure”, in *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773.

14.2 Ongoing monitoring

In the previous section, we have described some processes to implement enhanced audit reporting for z/OS and RACF. This allows you to identify security related actions and changes, report on important security definitions, and compare your current settings against your accepted baselines.

To achieve the next level of enterprise security management, we need not only reporting, but also automated protection and reaction to unacceptable security changes. Automation allows you to move from auditing, to monitoring and real-time compliance, rather than purely reactive security.

In this section, we demonstrate how to implement this ongoing monitoring with IBM zSecure, using detailed examples from this scenario to help you gain a better understanding of security monitoring methodology. Before we start implementing this security monitoring, we need to research and plan for our requirements.

The main aspects of the enterprise security monitoring include:

- ▶ Validating configuration settings, such as SETROPTS, CDT, Exits, Started class, profiles in Warning mode, and sensitive data sets.
- ▶ Monitoring privileged users activity, including system or group level special, operations and auditor users, users with class authority (CLAUTH), and UNIX superusers (UID=0).
- ▶ User account monitoring, such as inactive users, users with weak passwords, and unexpected logon activities from accounts with non-expiring passwords.
- ▶ Security violations, such as resource access violations.

- ▶ Successful access to critical files, such as PARMLIB updates.
- ▶ System event monitoring, such as SMF data lost, changes to the APF list, and IPL or PARMLIB settings changed and refreshed.

Many of these controls will be implemented as part of our daily security monitoring batch jobs and scheduled using Tivoli Workload Scheduler for z/OS. For additional information about recommended daily batch jobs for zSecure, see 13.2, “CKFREEZE, Signature, and UNLOAD generation data groups” on page 203.

Other controls will be implemented using the zSecure Audit Change Tracking function, zSecure Alert functions, and custom CARLa reports. Any security monitoring process requires ongoing maintenance, as requirements change over time. Our intention here is to provide a starting point for you to move from the static process of auditing, to the dynamic one of monitoring.

14.2.1 Using the change tracking feature

The *change tracking* (CT) feature of zSecure Audit is an essential part of any comprehensive monitoring and auditing framework for z/OS systems. In essence, this feature tracks all modifications to your base operating system and security related definitions. You can find a more detailed description of the data sets and configuration required to use CT in 5.6, “Change tracking” on page 106.

An important concept to understand in the context of change tracking is that of the *trusted computing base*, (TCB). You can consider an IT system to be made up of all the constituent parts that provide the user service. These are things such as the operating system itself, the security database definitions, the associated system utilities, and many more components. These components make up your base computing environment. Many other components are involved in delivering the final services of IT to the business, such as application systems, network connections, and data.

The TCB is considered to consist of the base components only, that is, the parts of the system that are trusted to provide a reliable and secure infrastructure upon which the additional components rely for their own processing. As the TCB is trusted, any changes to this must be recorded, monitored, and verified if you are to continue to trust that the overall system is reliable and secure.

As part of ensuring your TCB is reliable and secure, IBM provides the *z/OS Statement of Integrity*¹. This essentially states that should you find a way of compromising the integrity of the base z/OS using a normal, non-authorized, program or utility, IBM guarantees to rectify this problem. No other vendor of operating systems makes such a guarantee, which is one of the reasons that IBM z/OS systems are the most trusted and reliable systems in use today. When properly configured, an IBM z/OS system is literally hacker proof.

At TRA, we need to ensure that the massive change program we have put in place to improve the security of this system remains in place. To do this, we will exploit the full functioning of the zSecure Audit CT feature.

CT involves a concept referred to as the *verified base*. This is the system configuration and security settings that you have accepted as correct and secure for your organization. At TRA, after implementing all the changes discussed in Phase I, we ran the CT jobs that established this new system configuration as our accepted verified base.

We schedule a nightly job from the zSecure Audit suite that compares our accepted verified base to the settings that the system has at the time this nightly analysis is performed. This process is used to identify any variations from our verified base and report these using the CT ISPF panel option AU.C. We review these reports on a daily basis to identify and process any changes to our TCB.

Changes to the TCB can be dealt with in a number of ways. It is important to understand the implications of possible actions against such changes.

The possible actions for any change to the TCB are:

- ▶ Accept: This implies that the change was intentional, and will be included in any subsequent analysis of the system's compliance with the declared TCB.
- ▶ Defer: This accepts the change for the time being. Changes that are deferred will be reported on if they are still present in a subsequent analysis. Taking this action implies that the change might be acceptable, but you want to prevent it from becoming a part of the accepted TCB until this is confirmed, presumably with the person making the change.
- ▶ Reject: This action is taken against any change that you know to be incorrect, and the change should be backed out. Any changes rejected that appear in a subsequent analysis will continue to appear as new changes to be accepted, deferred, or rejected.

¹ To learn more about the z/OS Statement of Integrity, see http://www.ibm.com/servers/eserver/zseries/zos/racf/zos_integrity_statement.html.

To begin using change tracking, it is necessary that your systems programmer has established the necessary data sets and scheduled jobs; see 13.1, “Post systems programmer installation setup” on page 202 for more details about this topic.

Assuming this has been done, we use the zSecure Audit AU.C ISPF panels to display the current verified base. In Figure 14-18, the verified baseline from the previous CT run is displayed. Notice that the Days since last run is 0 to show that we are looking at the data from today. If the CT daily jobs are run for any period of time, this Days since last run field will increment accordingly. This documents the number of days since our last validation of the TCB.

We recommend that change tracking be performed on daily basis through your job scheduler, although your organization might choose to perform this validation on a weekly or other periodic basis.

The security officer invokes the change tracking interface to see if there are any exceptions by using the zSecure Audit option AU.C panel shown in Figure 14-18. This is a requirement of the daily security monitoring checklist now implemented at TRA. The security officer may also be notified of any exceptions through email.

Menu	Options	Info	Commands	Setup						
zSecure Admin+Audit for RACF - Audit Row 1 to 1 of 1										
Command ==> _____			Scroll ==> <u>CSR</u>							
Enter (E)xceptions, (D)eferred, (V)erified, (S)ystem info or (R)emove system _ Only select site defined message ids.										
<table><tr><th>System ID</th><th>Date of last run</th><th>Days since last run</th></tr><tr><td><u>E</u> SC76</td><td>19 APR 2008</td><td>0</td></tr></table>					System ID	Date of last run	Days since last run	<u>E</u> SC76	19 APR 2008	0
System ID	Date of last run	Days since last run								
<u>E</u> SC76	19 APR 2008	0								
***** Bottom of data *****										

Figure 14-18 This verified base is from today, as shown by the 0 in Days since last run

A list of exceptions to the verified baseline are shown in Figure 14-19. The security officer at TRA must review these exceptions, perform any necessary investigation, and take the appropriate action, such as accepting, rejecting, or deferring these changes.

CKREPROD Change Tracking - SC76 Changes		Row 1 of 21
Command ==> _____		Scroll==> <u>CSR</u>
Enter A(ccept),D(efer),R(eject) or S(how), or p 19 APR 2008		
Msg	Description	Detail
___ C210001	Addition, System special	ITS0TSA
___ C210002	Addition, System operations	ITS0TS2
___ C210003	Addition, System auditor	ITS0TS3
== C251020	Addition, SETROPTS Password change interval	30
___ C251020	Deletion, SETROPTS Password change interval	35

Figure 14-19 A list of exceptions generated since the last change tracking update

In Figure 14-20 the security officer has entered the S line command against the Addition line for the exception relating to a change in the SETROPTS password change interval to 30 days. This displays more information about the exception. This is noted as an exception because the verified baseline shows the current password interval to be 35 days.

CKREPROD Change Tracking - SC76 Changes		Row 1 of 21
_____ Change Tracking details _____		
-		
Addition on sysplex PLEX76, system SC76 created by NEWLIST TYPE=SYSTEM		
Addition of SETROPTS Password change interval		
PWDINTERVAL	30	

Figure 14-20 Details of the SETROPTS Password change interval addition

Following an investigation into the reason behind this change, the security officer decides to accept this exception using the A line command. It is important to understand that this change to the TCB has already occurred. All the security officer is doing by accepting this change is updating the verified base, so that when change tracking is next run, this change will no longer be considered an exception but a normal and accepted part of the TCB.

On accepting the change, we are prompted for a change record or some other descriptive text that will be stored in the CT database should any later review be required, as shown in Figure 14-21.

CKREPROD Change Tracking - SC76 Changes

Row 1 of 21

Change Tracking confirmation processing

Processing : SETROPTS Password change interval
Please enter change request number or comment text.

C1076539

Figure 14-21 The addition exception has been accepted using a valid change management record number

As a follow-up step to accepting the SETROPTS password interval change, the security officer must also accept the deletion of our previous password interval setting. Figure 14-22 shows the action performed by the security officer to do this task.

CKREPROD Change Tracking - SC76 Changes		Row 1 of 20
Command ==>		Scroll==> CSR
Enter A(ccept),D(efer),R(eject) or S(how), or p 19 APR 2008		
Msg	Description	Detail
__ C210001	Addition, System special	ITS0TSA
__ C210002	Addition, System operations	ITS0TS2
__ C210003	Addition, System auditor	ITS0TS3
<u>A</u> C251020	Deletion, SETROPTS Password change interval	35

Figure 14-22 The deletion exception is accepted and cleared from the verified baseline

Note that we use the same change management record number as shown in Figure 14-23.

CKREPROD Change Tracking - SC76 Changes

Row 1 of 20

Change Tracking confirmation processing

Processing : SETROPTS Password change interval

Please enter change request number or comment text.

C1076539

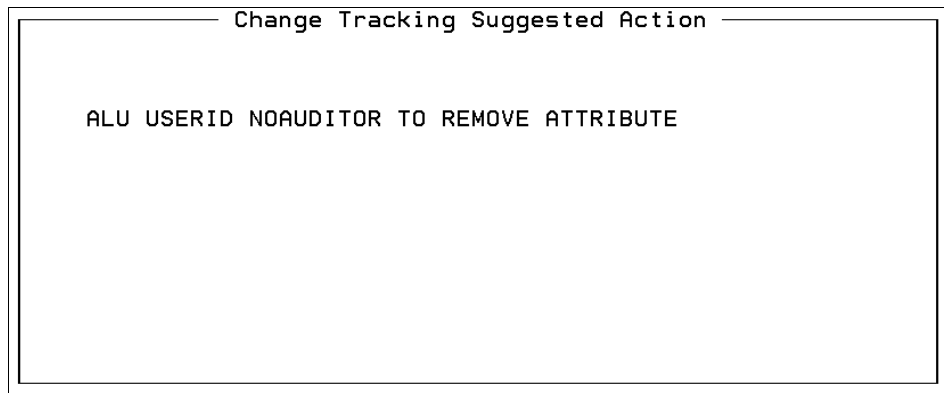
Figure 14-23 The deletion exception has been accepted using a valid change management record number

The next exception we investigate is the addition of a user with a system wide auditor attribute. The security officer has discovered that the attribute was added in error and therefore should be rejected. We enter the R line command, as shown in Figure 14-24.

CKREPROD Change Tracking - SC76 Changes		Row 1 of 19
Command ==>		Scroll==> CSR
Enter A(ccept),D(efer),R(eject) or S(how), or p 19 APR 2008		
Msg	Description	Detail
__ C210001	Addition, System special	ITS0TSA
__ C210002	Addition, System operations	ITS0TS2
R_ C210003	Addition, System auditor	ITS0TS3

Figure 14-24 Rejecting the Addition of System auditor exception

Having used the Reject action, a prompt is displayed to the security officer, as shown in Figure 14-25. This prompt suggests the action required to reverse the change. It should be noted that this is only a suggestion; no action is performed by zSecure Audit to back out of this change.

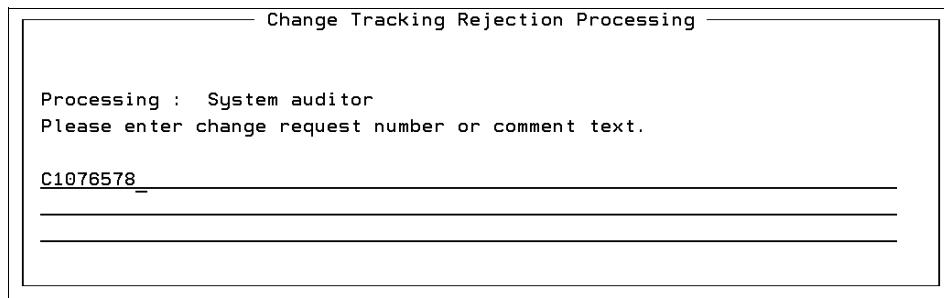


```
Change Tracking Suggested Action

ALU USERID NOAUDITOR TO REMOVE ATTRIBUTE
```

Figure 14-25 Suggested action following Reject of System auditor addition exception

A change management record may also be entered for Rejected changes, as shown in Figure 14-26.



```
Change Tracking Rejection Processing

Processing : System auditor
Please enter change request number or comment text.

C1076578

```

Figure 14-26 The Addition exception is rejected using a valid change management record number

After we have acted on these exceptions, they are no longer displayed, as shown in Figure 14-27.

The security officer now needs to continue investigation into the remaining exceptions. If it is decided to take no action on these exceptions for any period, they will remain in the CT report.

CKREPROD Change Tracking - SC76 Changes		Row 1 of 18
Command ==> _____		Scroll==> <u>CSR</u>
Enter A(ccept),D(efer),R(eject) or S(how), or p 19 APR 2008		
Msg	Description	Detail
___ C210001	Addition, System special	ITS0TSA
___ C210002	Addition, System operations	ITS0TS2

Figure 14-27 There are still some exceptions remaining to be investigated

Tracking changes on your own specific resources

Additionally, you can track your own IT specific sensitive resources for changes. Whether it is groups or data sets, zSecure Audit's change tracking feature can be customized to your needs.

As an example, TRA wants to keep tabs on changes to group @ZSEC001 and to data sets with the high-level qualifier of CKRU.

To accomplish the group change tracking, we go into zSecure ISPF option SE.C.M, as shown in Figure 14-28.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - Setup Change Track				
Option ==> <u>m</u>				
D	Data sets	Maintain list of sensitive data sets		
M	Site msgs	Site defined message table		
C	zSecure msgs	zSecure defined message table		

Figure 14-28 Select zSecure ISPF option SE.C.M to create customized resource change tracking

We then create a new message ID, U000001, with the title “Changes to group @ZSEC001”. On the change track definitions menu for U000001, we define what we want tracked, the messages that should be displayed, and the changes to be tracked. This is shown in Figure 14-29.

Menu	Options	Info	Commands
zSecure Admin+Audit for RACF - Setup - Change Track Me			
Command ==> _____			
Message id U000001			
Message description . . Changes to group @ZSEC001			
Action on addition . . REMOVE userid GROUP(@ZSEC001) to remove group			
Action on delete . . . CONNECT userid GROUP(@ZSEC001) to re-instate group			
Flags:			
Deletes reported YES Additions reported YES			
Confirm allowed YES Reject allowed YES			
Newlist type and CARLa define/select statements			
Newlist type RACF			
Define . . .			
Select . . . CLASS=GROUP SEGMENT=BASE MASK=@ZSEC001			

Figure 14-29 Customized Change Tracking U000001

Notice that CARLa coding and definitions are used to specify the group to be tracked. Pressing Enter at this menu provides the list of fields to be reported on, as shown in Figure 14-30.

Menu	Options	Info	Commands
zSecure Admin+Audit for RACF - Setu Row 1 to 17 of 31			
Command ==> _____			
Fields, field lengths and descriptions to report			
Fieldname	Length	Description	
KEY	8	GROUP	
USERID	8	USERID	

Figure 14-30 Reported fields for customized Change Tracking U000001

To see if any changes are reported, we select zSecure ISPF option AU.C and choose the Exceptions report against the System ID. Several changes were indeed tracked following the addition of U000001, as shown in Figure 14-31.

CKREPROD Change Tracking - SC76 Changes		Row 33 of 41
Command ==>		Scroll==> CSR
Enter A(ccept),D(efer),R(eject) or S(how), or p 23 APR 2008		
Msg	Description	Detail
<u>S</u>	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
—	U000001 Addition, Changes to group @ZSEC001	@ZSEC00
***** Bottom of data *****		

Figure 14-31 Change Tracking Exception report showing U000001 hits

Selecting the first hit displays the user that made the change, as shown in Figure 14-32.

CKREPROD Change Tracking - SC76 Changes		Row 33 of 41
Change Tracking details		
—		
Addition on sysplex PLEX76, system SC76 created by NEWLIST TYPE=RACF		
Addition of Changes to group @ZSEC001		
GROUP	@ZSEC001	
USERID	AMONSAL	

Figure 14-32 Selecting U000001 shows the user ID that instigated the group change

To specify change tracking for the CRKU data sets, we go to option SE.C.D, as shown in Figure 14-28 on page 290 (this time we select option D instead of M).

On this panel, we select I to insert a new change tracking data set definition. On the next panel, the specifics of the definition are entered, as shown in Figure 14-33.

Menu	Options	Info	Commands
zSecure Admin+Audit for RACF - Change Track Sensitive			
Command ==> _____			
Data set mask	Access	System	
CKRU.**	READ	SC76	
Reason	PRODUCTION ZSECURE DATA SETS		

Figure 14-33 Defining sensitive data set change tracking

Going back to zSecure ISPF option AU.C and selecting the exceptions report, we now see change tracking on the sensitive CKRU.** data sets (Figure 14-34).

CKREPROD Change Tracking - SC76 Changes		Row 1 of 42
Command ==> _____		Scroll==> <u>CSR</u>
Enter A(ccept),D(efer),R(eject) or S(how), or p 23 APR 2008		
Msg	Description	Detail
___ C210001	Addition, System special	SP005
___ C210002	Addition, System operations	MCAIRNS
___ C210003	Addition, System auditor	SP005
___ C210004	Addition, User with class authorisation	VISUAL3
___ C210005	Addition, Group special	VISUAL2
___ C210005	Addition, Group special	VISUAL3
___ C210005	Addition, Group special	VISUAL5
___ C251020	Addition, SETROPTS Password change interval	35
___ C251020	Deletion, SETROPTS Password change interval	30
___ C260002	Addition, Sensitive profile audit indicator	CKR.SCK
___ C260002	Addition, Sensitive profile audit indicator	LILIXIE
___ C260002	Deletion, Sensitive profile audit indicator	CKR.**
<u>s</u> C260003	Addition, Sensitive profile access list	*

Figure 14-34 Change tracking exception report showing customized sensitive data set hit

Selecting the change presents a panel with the needed information to investigate, based on the definition that we created for CKRU.**, as shown in Figure 14-35.

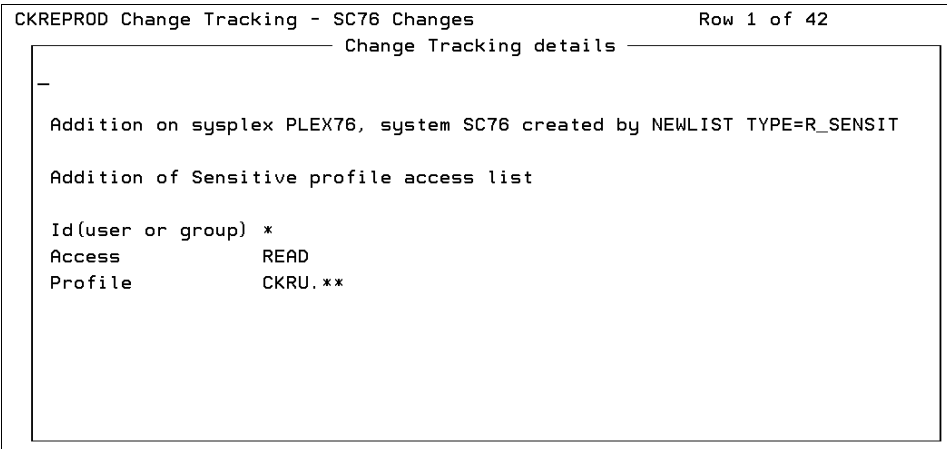


Figure 14-35 Display of customized sensitive data set change tracking

In conclusion, using the change tracking feature enables us to quickly and easily verify the status of the components that make up the trusted computing base. Without this ongoing validation, we are risking the integrity of all the applications and processes that rely on our base z/OS and security definitions being at a known level of integrity. Change tracking gives us confidence that no undesired modifications have made their way into our system without our knowledge.

CT can be used from a central system to verify the baselines of all systems in our sysplex. This feature strongly complements our existing alerting and audit reporting efforts.

14.2.2 Delta reporting: Comparing profiles and databases

A large volume of RACF changes are being implemented on the TRA RACF database as part of the security improvement program. There are occasions where RACF administrators who are implementing these changes need to compare two versions of the same RACF database.

You can easily perform such comparisons with the CARLa language. We show an example below, comparing the current RACF database with one that is a few days old. We want to report on user profiles that have been added and deleted. In this case, we can also highlight changes to any user profiles that still exist in both databases.

This is a powerful reporting capability if you need to identify and possibly back out of any damaging changes. The CARLa program shown in Example 14-5 compares two versions of the same RACF database and reports on:

- ▶ User profiles that exist in the current database only
- ▶ User profiles that exist in the old database only

Example 14-5 CARLa program to compare two versions of the same database

```
newlist type=RACF
  define del boolean where complex=OLD
  define add boolean where complex=NEW
  select complex=(OLD,NEW) class=USER segment=BASE
  summary key(8,'Userid') name add(noprop,hdr$blank,3),
  del(noprop,hdr$blank,3) count(<2,nondisplay)
```

The output of this CARLa program is shown in Figure 14-36. The column ADD shows profiles that exist in the new database only. The column DEL shows profiles that exist in the old database only.

```

BROWSE - JPEASE.C2R234D.REPORT ----- LINE 0000 0.0 s CPU, RC=0
COMMAND ==> _                                SCROLL ==> PAGE
***** Top of Data *****
P R O F I L E   L I S T I N G   16 Apr 2008 20:44

Userid      Name                      ADD DEL
CSV001      CUST SERV REP 01                ADD
CSV002      CUST SERV REP 02                ADD
CSV003      CUST SERV REP 03                ADD
CSV004      CUST SERV REP 04                ADD
CSV005      CUST SERV REP 01                ADD
CSV006      CUST SERV REP 01                ADD
CSV007      CUST SERV REP 01                ADD
CSV008      CUST SERV REP 01                ADD
CSV009      CUST SERV REP 02                ADD
CSV010      CUST SERV REP 02                ADD
CSV011      CUST SERV REP 02                ADD
CSV012      CUST SERV REP 02                ADD

```

Figure 14-36 Comparison report between two versions of the same database

The next CARLa example, shown in Example 14-6, compares the access control list (ACL) for data set profiles.

Example 14-6 CARLa program to compare access control lists for data set profiles

```
newlist type=RACF title='Changes to dataset profile ACLs'
  define del(hdr$blank,4) boolean where complex=OLD
  define add(hdr$blank,4) boolean where complex=NEW
  define aclchnge sumcount
  select complex=(old,new) c=dataset s=base
  summary key aclchnge(nondisplay,>0) count(nondisplay,>1),
  * userid count(noprop,nondisplay,<2) add(noprop),
  del(noprop)
```

An example of the output from the CARLa program is shown in Figure 14-37:

- ▶ The first column shows the data set profiles in question.
- ▶ The second column shows the access control list entries that only exist in one of the complexes.
- ▶ The third column shows entries that exist on the access control list in the new database only and therefore have been added.
- ▶ The fourth column shows entries that exist on the access list in the old database only and therefore are deleted from the new complex.

BROWSE - JPEASE.C2R134D.REPORT	-----	LINE	00000024	COL	001	080
COMMAND ==>						SCROLL ==> CSR
CKR.DATA.CT.*.CKAVERIF		@CTAUD	ADD			
		@CTRUN	ADD			
		C2RCTAUD	DEL			
		C2RCTRUN	DEL			
CKR.DATA.CT.**		@CTAUD	ADD			
		@CTRUN	ADD			
		C2RCTAUD	DEL			
		C2RCTRUN	DEL			
CKR.DATA.CT.CKACDATE		@CTAUD	ADD			
		@CTRUN	ADD			
		C2RCTAUD	DEL			
		C2RCTRUN	DEL			
CKR.DATA.CT.CKADEFER		@CTAUD	ADD			
		@CTRUN	ADD			
		C2RCTAUD	DEL			
		C2RCTRUN	DEL			
CKR.DATA.CT.CKAEXCEP		@CTAUD	ADD			

Figure 14-37 Report of changes to access control lists for data set profiles

Using reports such as this, it is a simple matter to identify changes in RACF profiles from one day to the next. It is even possible to implement an entire report set of these changes as part of your daily batch suite of security reports, and make this report one of your daily review items as the security administrator.

It is far better to know about the delta between one day and the next than be caught out by a production problem introduced by an unintended change. Reports such as the above can help you avoid being caught by ongoing RACF changes.

Note: You can run these CARLa comparison queries in batch. To generate the appropriate JCL for the CARLa, type SUBMIT on the command line in zSecure option CO.C. Make sure you have the appropriate input files selected as input for your CARLa query. Use the SETUP FILES application to do this task.

14.2.3 SYSLOG trapping in zSecure Alert

Some critical system and security events are only seen as they occur in the z/OS SYSLOG. Some of these critical events may be dynamic addition of APF data sets, addition/removal of exits, or simply the buffering of SMF data when the SMF SYS1.MANxx data sets get full.

To trap these events, the Delft Air tiger team has suggested to TRA that they use zSecure Alert's predefined alerts and create some of their own customized alerts. Notifications should then be sent to the security officers through email or to a cell phone indicating that a monitored event has occurred, allowing them to deal immediately with a possible security breach or exposure. This in turn will improve their security posture and assist with their audits.

Compliance monitoring of privileged and authorized users is a critical factor in intrusion detecting and alerting, and getting this information immediately from zSecure Alert through the SYSLOG or SMF can greatly increase operational effectiveness associated with incident response activities and decrease the chance of security breaches by getting immediate notifications.

zSecure Alert can also react to alerts by sending WTO messages to be captured by automated operations and having it issue roll back or risk mitigating commands autonomously.

The implementation and results of TRA's SYSLOG and SMF trapping efforts are covered in 14.2.6, "Monitoring for critical system events" on page 311.

14.2.4 Sending SNMP data

To more closely integrate with enterprise wide compliance management systems, zSecure Alert can be configured to send SNMP data. The panels below show the configuration of zSecure Alert to send this SNMP data.

Usually not all alerts in any one alert configuration or category need to be sent to the SNMP listener, so we use the W line command against specific alerts to specify the destination details separately for each alert, rather than use a global destination.

In this example, we use the alert for High UACC set on a data set profile, as seen in Figure 14-38.

```
zSecure Admin+Audit for RACF - Setup Row 1 to 6 of 6
Command ==> _____ Scroll ==> CSR

Data set alerts
Select the alert you want to work with.
The following line commands are available: A(Preview), C(opy), D(elete),
E(dit), I(nsert), W(Who/Where), S(elect), U(nselect), B(rowse)
-----
Alert                                     Id   Sel  gECSW  C
-
WARNING mode access on data set          1201 Yes  gE      W
W Setting UACC>=UPDATE on a DATASET profile 1202 Yes  gE      W
Setting UACC>NONE on a DATASET profile 1203 Yes  gE      W
- Update on APF dataset                  1204 Yes  gE      W
Data set added to APF list               1205 Yes  gE      W
- Data set removed from APF list         1206 Yes  gE      W
***** Bottom of data *****
```

Figure 14-38 Sending an alert using SNMP - Step 1

When we type the W line command in front of this alert and press enter, the alert destinations panel is shown, as shown in Figure 14-39.

```
zSecure Admin+Audit for RACF - Setup - Alert
Command ==> _____
You may scroll forward/backward to view all recipient types
From . . . . . &jobname at &system <mbox@domain> More: -
Phone@gateway . . _____
(You may specify : to receive a list of defined recipients :setname.fields)
Reply to . . . . . _____
/  SNMP
-  Write SNMP traps to C2RSNMP DD
Specify SNMP receiver
Address . . . . . 9.8.7.6:161
-  WTO
-  Write WTOs to C2RWTO DD
```

Figure 14-39 Sending an alert using SNMP - Step 2

Select the SNMP field and specify the listener's IP address or DNS name. You can specify more than one listening address, separated by commas. For any address, you can also specify a port name in the standard notation. If you also specify the Write SNMP traps to C2RSNMP DD field, the messages will not be sent; rather, they will be written to this DD in the zSecure Alert address space. This is intended for debugging and testing purposes only.

In the example that follows, we have used the Tivoli Security Information and Event Manager servers SNMP listening port, 161. The Security Information and Event Manager Agent has a built-in listening component and can receive both SNMP and SYSLOG messages. For detailed information about data collection processing by Security Information and Event Manager, please refer to *IBM Tivoli Security Information and Event Manager Version 2.0 Users Guide*, SC23-9689.

When we finish configuring the alert, and press F3 to save our changes, we can then browse this alert definition. Entering B at the alert command line, shown in Figure 14-40, provides the panel shown in Figure 14-41. From here we select ISPF Skeleton C2PS1202 and press Enter.

```

zSecure Admin+Audit for RACF - Setup Row 1 to 6 of 6
Command ==> _____ Scroll ==> CSR

Data set alerts
Select the alert you want to work with.
The following line commands are available: A(Preview), C(opy), D(elete),
E(dit), I(nsert), W(Who/Where), S(elect), U(nselect), B(rowse)
-----
Alert                                     Id   Sel  gECSW  C
- WARNING mode access on data set         1201 Yes  gE     W
B Setting UACC>=UPDATE on a DATASET profile 1202 Yes   S
- Setting UACC>NONE on a DATASET profile 1203 Yes  gE     W
- Update on APF dataset                   1204 Yes  gE     W
- Data set added to APF list               1205 Yes  gE     W
- Data set removed from APF list           1206 Yes  gE     W
***** Bottom of data *****

Alert destination(s) changed

```

Figure 14-40 Selecting an alert with browse

```

zSecure Admin+Audit for RACF - Setup - Alert
Command ==> _____

Description . . Setting UACC>=UPDATE on a DATASET profile
Member prefix C2P
Alert id . . . 1202 Severity . . . I (I, W, E or S)
Data source . . SMF
Parameters . . .

Specify SMF records to be collected for this alert
Type Sub  Type Sub  Type Sub  Type Sub
80         80

Specify WTO filters for this alert
Prefix    Prefix    Prefix    Prefix    Prefix

Select allowable destination types
/ E-mail   / Cellphone / SNMP     / WTO

Enter / to view/edit the skeleton for this alert
/ ISPF Skeleton C2PS1202

```

Figure 14-41 Alert details and skeleton

We are presented with the ISPF skeleton for the alert, which contains, among other things, the layout of the SNMP message. The SNMP message layout was generated automatically within the)CM SNMP ISPF skeleton sortlist section, as shown in Example 14-7. This CARLa code defines the variables and content of the SNMP message that the alert will send. It uses the Security Information and Event Manager W7 standardized format for cross platform audit reporting.

Example 14-7 SNMP skeleton section of alert definition

```
)CM SNMP sortlist
)SEL &C2PERCTP = SNMP
sortlist,
  recno(nd),
  '&c2pemem.' /,
  'eventIntegral',
  'Alert: UACC>=UPDATE set:' profile(0) '- ',
  'High UACC specified when adding or altering a data set profile' /,
  'eventWhen' datetime(datetimezone,0) /,
  'onWhatPROFILE' profile(0,hor) /,
  'whatUACC' uread(0) | uupdt(0) | uctrl(0) | ualtr(0) /,
  'whatDESC' desc(0,explode) /,
  'whatRACFCMD' racfcmd(0,hor) /,
  'whoUSERID' userid(0) /,
  'whoNAME' name(0) /,
  'whatJOBNAME' jobname(0) /,
  'whereSYSTEM' system(0)
)ENDSEL
```

You can define your own SNMP traps as custom zSecure Alert reports. To do this, the LIST or SORTLIST output must comply with the defined format for Security Information and Event Manager recognizable SNMP data and the CARLa code must also specify the SNMP keyword on the NEWLIST. This is done for you by zSecure Alert when you select SNMP as an alert destination, as in the above example. For details about how to define your own SNMP traps and the predefined variables that may be used in SNMP messages for Security Information and Event Manager, please refer to *IBM Security zSecure Alert User Reference Manual Version 1.12, SC27-2776*.

14.2.5 Monitoring specific resources

Our goal here is to establish best practice mainframe security monitoring. We select alerts from those provided by zSecure Alert to match the standards already in place at Delft Transport. After we have established a base set of alerts, we then monitor their use and decide on any necessary enhancements.

As soon as these alerts are in place and an ongoing improvement process established, we are confident that the effort put into improving the TRA security posture will not be accidentally regressed.

Using zSecure Alert supplied alerts

We use the predefined alerts to monitor sensitive users, groups, data set profiles, general resource profiles, and UNIX files. Listed below are the alert conditions we have chosen from the supplied set:

- ▶ User alerts category
 - Logon by unknown user
 - Logon with emergency user ID
 - Highly authorized user revoked for pwd violations
 - System authority granted
 - Group authority granted
 - Non-OPERATIONS user accessed data set with OPERATIONS
 - SPECIAL authority used by non-SPECIAL user
 - Invalid password attempts exceed limit
 - Password history flushed
 - Too many violations
- ▶ Group alerts category
 - Connect to an important group
- ▶ Data set alerts category
 - Setting UACC>=UPDATE on a DATASET profile
 - Data set added to APF list
- ▶ General resource alerts category
 - Catchall profile used for STC
 - Audited program has been executed
- ▶ UNIX alerts category
 - UNIX file access violation
 - Global write specified when altering file access

- RACF control alerts category
 - Global security countermeasure or option changed
- System alerts category
 - SMF data loss started

For more information about how to implement these predefined alerts, refer to 4.3, “Implementation suggestions” on page 78. For detailed descriptions of alerts in the predefined zSecure Alert set, refer to *IBM Security zSecure Alert User Reference Manual Version 1.12, SC27-2776*.

Using custom alerts

We will also implement two new custom alerts in the current system. The first of these, Update on CKR* data sets, was introduced in 4.3.4, “Adding your own alerts” on page 83. The second will be an alert on the setting of a UACC greater than NONE on any RACF general resource profile.

To create this new alert, we copy the predefined alert for Setting UACC>NONE on a DATASET profile from the data set alerts category. The panel shown in Figure 14-42 is presented after we used the line command C to copy this alert. We have to assign the new alert a unique ID number over 4000 (we use 4002 in this case). We also have to specify the data source (SMF and the SMF record number, type 80). The alert description is updated and its severity is changed from I for informational to W for warning.

zSecure Admin+Audit for RACF - Setup - Alert

Command ==> _____

Description . . . Setting UACC>NONE on a general resource profile

Member prefix IN

Alert id 4002 Severity W (I, W, E or S)

Data source . . . SMF

Parameters . . . _____

Specify SMF records to be collected for this alert

Type	Sub	Type	Sub	Type	Sub	Type	Sub	Type	Sub
<u>80</u>	_____	_____	_____	_____	_____	_____	_____	_____	_____

Specify WTO filters for this alert

Prefix	Prefix	Prefix	Prefix	Prefix
_____	_____	_____	_____	_____

Select allowable destination types

/ E-mail _ Cellphone / SNMP / WTO

Enter / to view/edit the skeleton for this alert

_ ISPF Skeleton

Figure 14-42 Ongoing monitoring - Adding a new alert

Three of the four possible destinations are selected and used for this new alert: email, SNMP, and WTO.

Now we have to prepare the CARLa code. To do this, we select the ISPF Skeleton option and press Enter. As discussed in 4.3.4, “Adding your own alerts” on page 83, there are several sections in the skeleton. As we are basing the new alert on a similar predefined one, we should only need to make a few minor modifications to the CARLa code. These are described below:

- ▶ In the code section)CM Alert condition, change the text select event=(addsd(0,2),altdsd(0,2)), to select event=(rdefine(0,2),ralter(0,2)),.

This section indicates under what conditions the alert will be issued. We need to change it from events related to adding and updating the data set profiles, addsd and altdsd, into adding and updating general resource profile related events, rdefine and ralter.

- ▶ In the code section)CM EMAIL sortlist
 - Change the text

```
'High UACC specified when adding or altering a data set profile'
/,
to
'High UACC specified when adding or altering a general resource
profile' /,
```
 - Add the following statement above the line / ' Profile'(18) profile:

```
/ ' Class'(18) class,
```

This section specifies the message text that will be sent when this alert is triggered for the email destination. So we need to update the message to reflect the new alert.

- ▶ In the code section)CM SNMP sortlist
 - Change the text

```
'High UACC specified when adding or altering a data set profile'
/,
to
'High UACC specified when adding or altering a general resource
profile' /,
```
 - Add the following statement above the line 'onWhatPROFILE' profile(0,hor) /,:

```
'onWhatCLASS' class(0,hor) /,
```

This section specifies the message text that will be sent with this alert as part of an SNMP destination, so we need to update the message to reflect the new alert.

After finishing these changes to the skeleton and pressing F3 twice, we are returned to the Dataset alert category. When we navigate again to the Other alerts category, we can see that the new alert is now defined. See Figure 14-43, where you can see the updated description Setting UACC>NONE on a general resource profile. We now enter the W line command against this alert to configure the available destinations.

```
zSecure Admin+Audit for RACF - Setup Row 1 to 3 of 3
Command ==> _____ Scroll ==> CSR

Other alerts
Select the alert you want to work with.
The following line commands are available: A(Preview), C(opy), D(elete),
E(dit), I(nsert), W(Who/Where), S(elect), U(nselect), B(rowse)
-----
Alert                                     Id   Sel  gECSW  C
-   SNMP heartbeat                       1001 No   g--   -
-   Update on CKR* datasets               4001 Yes  gE    W
-   Setting UACC>NONE on a general resource profile 4002 Yes  gE-   W
***** Bottom of data *****
```

Figure 14-43 New alert added

Because we only allowed three of the four possible destinations for this alert, a prompt at the bottom of the panel informs us that Not all destination types are allowed for this alert, as shown in Figure 14-44.

```
zSecure Admin+Audit for RACF - Setup - Alert
Command ==> _____

You may scroll forward/backward to view all recipient types
Select the alert destination
/ E-mail
- Write e-mails to C2RSMTP DD

Specify e-mail recipient(s)
From . . . . . &jobname at &system <zsecure@us.ibm.com>

Mail to . . . . . :ITSOMAIL.ADDRESS
(You may specify : to receive a list of defined recipients :setname.fields)

CC . . . . .
BCC . . . . .
Reply to . . . . . noreply@us.ibm.com
Output
Font s [ Not all destination types are allowed for this alert ]
```

Figure 14-44 Alert destinations

For more information about configuring the destinations, refer to 4.3.3, “Sending alerts to their destination” on page 81 and 14.2.4, “Sending SNMP data” on page 298.

Before this alert is activated, we need to return to the main Managing alert configurations panel. From here we must verify the new CARLa code, then refresh the zSecure Alert started task address space. These new alerts were defined in the alert set SC76AS, which you can see in Figure 14-45. After verifying your alert definitions, you first have to refresh the C2POLICE address space before the new alerts are active. This can be accomplished using line command R (for Refresh).

```

zSecure Admin+Audit for RACF - Setup - A Row 1 from 5
Command ==> _____ Scroll ==> CSR

Managing alert configurations
Line commands are available depending on the configuration stage: C(copy),
D(elete), I(nsert), E(dit), W(Who/Where), S(elect), V(erify), F(Refresh),
B(rowse)

----- Configuration steps -----
  Name      Description      Set Des Sel Ver Ref Act
  _  C2PDFL  zSecure Alert default alert configurati  Req Req Req Req Req
-----
V  SC76AS  ITS0 alert configuration      0k  0k  0k  0k  0k  Y
-----
  _  TIVOLI  ITS0 alert configuration      0k  0k  0k  0k  0k
-----
***** Bottom of data *****

```

Figure 14-45 Verify and refresh the alert configuration

Now that our new alerts are active, if someone specifies a UACC greater than NONE on a general resource, we receive the following message in the SYSLOG; in this example, the user making the unwanted change was ITSOTS1 and the general resource changed was CKF.ALERT:

C2P4002W UACC>NONE set: CKF.ALERT
UACC set to READ by ITSOTS1

The security administrators in the email distribution list for this alert will receive the email shown in Example 14-8.

Example 14-8 Email layout

Alert: UACC>NONE set: CKF.ALERT

High UACC specified when adding or altering a general res profile

Alert id	4002
Date and time	15Apr2008 16:26:17.58
Class	XFACILIT
Profile	CKF.ALERT
UACC	READ
Result	Success
RACF command	RALTER XFACILIT (CKF.ALERT) UACC(READ)
User	ITSOTS1 TEST USER 1
Job name	ITSOTS1
System ID	SC76

This alert will also generate an SNMP message, as shown in Example 14-9.

Example 14-9 SNMP layout

```
4002
eventIntegral Alert: UACC>NONE set: CKF.ALERT - High UACC specified
when adding or altering a general res profile
eventWhen 2008-4-15,16:26:17.5,-4:0
onWhatCLASS XFACILIT
onWhatPROFILE CKF.ALERT
whatUACC READ
whatDESC Success
whatRACFCMD RALTER XFACILIT (CKF.ALERT) UACC(READ)
whoUSERID ITSOTS1
whoNAME TEST USER
whatJOBNAME ITSOTS1
whereSYSTEM SC76
```

Periodic CARLa based reporting

Using predefined and customized alerts, we implement real-time monitoring for critical resources. zSecure Alert provides great flexibility and convenience to accomplish this monitoring easily and effectively.

In addition, we can use custom CARLa to implement other monitoring tasks that may not require real-time attention. For example, we have already implemented the Logon with emergency user ID alert, so we know in real time when our emergency user IDs are used. Usually this use is legitimate and related to a specific incident. Often these user IDs are referred to as *firecall* users, and they are used to gain access to resources to fix a production problem.

The staff fixing the problem would not normally have access to these production resources when using their own personal user IDs. The firecall user has this access, and is issued to the production support staff, who then use it to correct the problem.

The firecall user ID is handed back to the security administrator after the problem is resolved, and its password is changed to prevent re-use until the next time it is required.

In this scenario, it is desirable to report on the activity of the firecall user after it has been used. We need to check that the production support staff have only accessed the resources needed to correct the problem, and have not abused their temporary access in using the firecall user ID. We can do this easily with a CARLa program to monitor the activities of the emergency user IDs, as shown in Example 14-10.

Example 14-10 CARLa program to monitor the activities of emergency user IDs

```
SUPPRESS CKFREEZE
SUPPRESS RACF
newlist type=smf name=smfsel outlim=0 ,

select USER=(ITSOTS1,ITSOTS2,ITSOTS3)
  list type

mergelist

newlist type=smf,
  TITLE='Usage of Emergency Id ITSOTS*'
  select listlike=smfsel type=(80 83) event=access class=dataset
  sortlist date(nondispl) time(nondispl) system(nondispl),
    userid(nondispl),
    date(9) time userid system recorddesc(0,ww),
  / " Jobname + id:"(ne) jobname jobid,
  / " Terminal      :"(ne) terminal,
    " Application : " appl,
  / " Dsname        :"(ne) dataset, class(hor),
  / " Access used  :"(ne) intent(retain),
    " Profile:" profile(0,wrap,retain)
```

```

newlist type=smf
select listlike=smf sel type=(80 83) event=access class<>dataset
  sortlist date(nondispl) time(nondispl) system(nondispl),
    userid(nondispl),
    date(9) time userid system recorddesc(0,ww),
    / " Jobname + id:"(ne) jobname jobid,
    / " Terminal      :"(ne) terminal,
    " Application : " appl,
    / " Class         :"(ne) class, "Resource:" resource(0,wrap),
    / " Access used :"(ne) intent(retain),
    " Profile:" profile(0,wrap,retain)

endmerge

```

Using this CARLa program, we monitor the activities of three emergency user IDs: ITSOTS1, ITSOTS2, and ITSOTS3. The report provides details about their access to data sets, and general resources providing this access was logged to SMF. It is important to realize that unless an SMF record was logged for their access, this report will not provide any record of it.

You could also use SMF type record 14 and 15 to provide additional details in the report for any data set access (in the absence of an SMF type 80 record); however, this would not provide details of un-logged access to general resources. We recommend that you specify the RACF attribute UAUDIT on emergency user IDs to ensure that all their access is logged to SMF.

The report uses only the live SMF data, so we code the commands SUPPRESS CKFREEZE and SUPPRESS RACF at the top of the program. The first NEWLIST statement marks the beginning of the report description. The CARLa scans the SMF data and selects *any* SMF records where the emergency user IDs ITSOTS1, ITSOTS2, and ITSOTS3 are present. The subsequent NEWLIST statements are surrounded by a MERGELIST statement. MERGELIST is used to combine the output of multiple NEWLISTs into one report, up until an ENDMERGE statement is coded. Records from NEWLISTs within a MERGELIST are merged according to their SORTLIST sort order.

We use a pair of NEWLIST reports within our MERGELIST/ENDMERGE construct. The first of these NEWLISTs selects only SMF type 80 and 83 from the initial NEWLIST selection of all SMF record types. It further refines this selection by including only records for class data set. The second NEWLIST of the pair also uses only SMF type 80 and 83. It then selects any class other than data set, thus processing all records related to general resources.

Both NEWLIST statements use the same initial variables for their SORTLIST selection. This allows the merging of these two different SMF data selections to occur seamlessly in the final report. The report will show a list of all access by these users, regardless of whether this was for a data set or general resource, sorted in the actual order in which the access occurred. For investigative purposes this is essential, as you need to review the events that this user performed in the order that they were performed, regardless of the type of RACF definition involved.

For more information about writing your own CARLa code, refer to Appendix B, “An introduction to CARLa” on page 427 and especially *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773.

When we finish editing the CARLa program using the zSecure panel option CO, we can either type GO or RUN to test the program immediately, or use the commands SUB or SUBMIT to generate JCL for batch submission. You can save this JCL into a job library for later submission by a batch scheduler, such as IBM Tivoli Workload Scheduler for z/OS.

Part of the output from a batch run of this report is shown in Example 14-11.

Example 14-11 Emergency user access report section

```

ISM F  R E C O R D  L I S T I N G   15Apr08 15:31 to 15Apr08 18:45
Usage of Emergency Id ITSOTS*

Date      Time User      Sys Description
15Apr2008 15:31 ITSOTS1 SC76 RACF ACCESS success for ITSOTS1:
(UPDATE,ALTER) on DATASET CKRU.DATA.SSC76.C2POLICE.C2PCUST
  Jobname + id: ITSOTS1 TSU05955
  Terminal   : SC38TC27 Application:
  Dsname      : CKRU.DATA.SSC76.C2POLICE.C2PCUST      DATASET
  Access used : UPDATE   Profile: CKRU.**
15Apr2008 15:54 ITSOTS2 SC76 RACF ACCESS success for ITSOTS2:
(UPDATE,UPDATE)
  Jobname + id: ITSOTS2 TSU05955
  Class      : OPERCMDS Resource: MVS.MODIFY.STC.C2POLICE.C2POLICE
  Access used : UPDATE   Profile: MVS.MODIFY.STC.C2POLICE.*
.....

```

We can also have this report sent through email. Security administrators, auditors, and managers can receive information about events we are monitoring in a convenient manner, regardless of whether these events are reported by zSecure Alert or using custom CARLa. We will describe the process of establishing an automated email reporting framework in Chapter 15, “Implementation phase III” on page 321.

14.2.6 Monitoring for critical system events

As discussed in 14.2.3, “SYSLOG trapping in zSecure Alert” on page 297, TRA is charged with trapping SYSLOG information and reacting to alerts on system and security events. (zSecure Alert is fully covered in Chapter 4, “IBM Security zSecure Alert” on page 73.)

In addition to the canned alerts already predefined to address common system and security events, such as SMF full and SETROPTS deactivation, we decide to create our own alert to identify the use of SETPROG EXIT changes done from the MVS console. We go to zSecure Alert through zSecure ISPF option SE.A.A and select alert configuration SC76AS, as shown in Figure 14-46.

Menu Options Info Commands Setup

zSecure Admin+Audit for RACF - Setup - A Row 1 from 6

Command ==> Scroll ==> CSR

Managing alert configurations

Line commands are available depending on the configuration stage: C(opy), D(elete), I(nsert), E(dit), W(Who/Where), S(elect), V(erify), F(Refresh), B(rowse)

----- Configuration steps -----						
Name	Description	Set	Des	Sel	Ver	Ref Act
— C2PDFL	zSecure Alert default alert configurati	Req	Req	Req	Req	Req
␣ _ SC76AS	ITSO alert configuration	Ok	Ok	Ok	Ok	Ok Y
— TIVOLI	ITSO alert configuration	Ok	Ok	Ok	Req	Req

***** Bottom of data *****

Figure 14-46 List of available alert configurations

Using the predefined alert SMF data loss started as a model, we create a SETPROG EXIT command issued alert. The resulting newly defined alert is displayed in Figure 14-47, Figure 14-48 on page 313, and Figure 14-49 on page 313.

Menu	Options	Info	Commands	Setup
VIEW	CKRU.DATA.SSC76.C2POLICE.C2PCUST(INS4004) - 01. Columns 00001 00072			
Command ==>				Scroll ==> <u>CSR</u>
000018)SEL &C2PEPASS = Y			
000019)ENDSEL			
000020)CM Alert condition			
000021)SEL &C2PEPASS = N			
000022)IM C2PSGNEW			
000023	select likelist=WTOrec msgid=(CSV4201)			
000024)CM EMAIL sortlist			
000025)SEL &C2PERCTP = MAIL			
000026	sortlist,			
000027	recno(nd),			
000028	'Alert: SETPROG EXIT command used'(t),			
000029	'Alert: SETPROG EXIT command used' /,			
000030	'System messages report the SETPROG EXIT command has been issued' /,			
000031	/ ' Alert id &c2pemem.',			
000032	/ ' Date and time'(18) date(9) time(11),			
000033	/ ' WTO message'(18,noretain) MsgTxt1(wordwrap,0),			
000034	/ ' Console ID'(18) console(8),			
000035	/ ' System ID'(18) system,			
000036	/ /			
000037)ENDSEL			

Figure 14-47 Newly defined SETPROG alert - Part 1, email view

Menu	Options	Info	Commands	Setup
VIEW		CKRU.DATA.SSC76.C2POLICE.C2PCUST(INS4004) - 01. Columns 00001 00072		
Command ==>				Scroll ==> <u>CSR</u>
000038)CM	Cellphone sortlist		
000039)SEL	&C2PERCTP = CELL		
000040	sortlist,			
000041	recno(nd),			
000042	'Alert &c2pemem.: SETPROG EXIT command used. WTO msgid:'(t),			
000043	console(t) ':'(t) MsgTxt1(t),			
000044	'Alert &c2pemem.: SETPROG EXIT command used. WTO msgid:',			
000045	console(0) ':' MsgTxt1(0)			
000046)ENDSEL			
000047)CM	SNMP sortlist		
000048)SEL	&C2PERCTP = SNMP		
000049	sortlist,			
000050	recno(nd),			
000051	'&c2pemem.' /,			
000052	'eventIntegral',			
000053	'Alert: EXIT dynamically added -',			
000054	'System messages report the SETPROG EXIT command has been issued' /,			
000055	'eventWhen' datetime(datetimezone,0) /,			
000056	'whatWTO-MESSAGE' MsgTxt1(0) /,			
000057	'fromWhereCONSOLE' console(0) /,			

Figure 14-48 Newly defined SETPROG alert - Part 2, cellphone and SNMP view

Menu	Options	Info	Commands	Setup
VIEW		CKRU.DATA.SSC76.C2POLICE.C2PCUST(INS4004) - 01. Columns 00001 00072		
Command ==>				Scroll ==> <u>CSR</u>
000058	'whereSYSTEM'	system(0)		
000059)ENDSEL			
000060)CM	WTO sortlist		
000061)SEL	&C2PERCTP = WTO		
000062	sortlist,			
000063	recno(nd),			
000064	'C2P&c2pemem.&c2peflag',			
000065	'SETPROG EXIT command used at console' console(0) ':' /,			
000066	MsgTxt1(0)			
000067)ENDSEL			
000068)ENDSEL			
***** Bottom of Data *****				

Figure 14-49 Newly defined SETPROG alert - Part 3, WTO view

Notice that the alert specifically targets the CSV420I message and that the message title is generic to cover all four instances of this message, that is, ADDED TO, MODIFIED FOR, DELETED FROM, and REPLACED FOR. We could also have created four separate alerts for each instance and parsed out the CSV420I message if we had wanted to make the message title more specific.

Within the code, the email, cellphone, SNMP, and WTO specifications are defined. These indicate destination types that can be used for alerting, which are subsequently selected as the alerting method using the “W” line command from the ISPF interface. The body of the alert notification is to contain the alert title, the CSV420I message text, and the console ID of the SETPROG issuer.

A test is conducted to verify that the new alert is working as designed. The security officer issues a SETPROG command to add an exit, as shown in Figure 14-50.

Display	Filter	View	Print	Options	Help

SDSF SYSLOG	6095.101	SC76	SC76	04/21/2008 0W	10396 COLUMNS 39 118
COMMAND INPUT ==>					SCROLL ==> CSR
JPEASE	00000290	SETPROG EXIT,ADD,EXITNAME(IRREVVX01) MODNAME(C4RMAIN)			
		DSNAME(C4R.SC4RLNK) STATE(ACTIVE)			
	00000290	IEF196I IEF237I D16C ALLOCATED TO SYS00020			
	00000290	IEF196I IEF285I C4R.SC4RLNK			
	00000290	IEF196I IEF285I VOL SER NOS= Z19RE2.			
JPEASE	00000090	CSV420I MODULE C4RMAIN HAS BEEN ADDED TO EXIT IRREVVX01			

Figure 14-50 SETPROG command issued at the MVS console

A few seconds later, zSecure Alert traps the CSV420I message in the SYSLOG and issues the WTO message defined in the alert, as shown in Figure 14-51.

Display	Filter	View	Print	Options	Help

SDSF SYSLOG	6095.101	SC76	SC76	04/21/2008 0W	10402 COLUMNS 39 118
COMMAND INPUT ==>					SCROLL ==> CSR
STC06247	00000090	C2P4004I SETPROG EXIT command used at console JPEASE : 891			
891	00000090	CSV420I MODULE C4RMAIN HAS BEEN ADDED TO EXIT IRREVVX01			

Figure 14-51 zSecure Alert captures the CSV420I message in the SYSLOG and puts out a WTO message

SNMP sends the alert out to the defined destination. Because it has a defined DD statement in the C2POLICE started task JCL, we can have a look at what was sent out through the C2POLICE sysprint. This is shown in Figure 14-52 and Figure 14-53.

Display Filter View Print Options Help									
SDSF JOB DATA SET DISPLAY - JOB C2POLICE (STC06247)					DATA SET DISPLAYED				
COMMAND INPUT ==>					SCROLL ==> CSR				
NP	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-Cnt	Page
	JESMSGLG	JES2		2	C2PSUSER	S		1,258	
	JESJCL	JES2		3	C2PSUSER	S		155	
	JESYSMSG	JES2		4	C2PSUSER	S		3	
	C2XPRINT	C2POLICE		102	C2PSUSER	S		20	
	SYSTSPRT	C2POLICE		105	C2PSUSER	A		0	
	C2RSMTP	C2POLICE		107	C2PSUSER	A		13,102	
s_	C2RSNMP	C2POLICE		108	C2PSUSER	A		0	

Figure 14-52 Selecting the SNMP output from the C2POLICE sysprint

Display Filter View Print Options Help									

SDSF OUTPUT DISPLAY C2POLICE STC06247 DSID 108 LINE 12					COLUMNS 02- 81				
COMMAND INPUT ==> _					SCROLL ==> CSR				
4004									
eventIntegral Alert: EXIT dynamically added - System messages report the SETPROG									
eventWhen 2008-4-21,15:35:6.0,-4:0									
whatWTO-MESSAGE CSV420I MODULE C4RMAIN HAS BEEN ADDED TO EXIT IRREXV01									
fromWhereCONSOLE JPEASE									
whereSYSTEM SC76									
***** BOTTOM OF DATA *****									

Figure 14-53 Alert sent out through SNMP, as seen in C2POLICE sysprint

Based on Figure 14-52, we could select the SMTP output to see what was sent through email. Instead, Figure 14-54 shows the actual email.

 C2POLICE at MVS1 <zSecure@uk.ibm.com> 21/04/2008 22:03 Please respond to noreply@uk.ibm.com Default custom expiration date of 21/04/2009		To	Jamie PearceUKJIBM@IBMGB
		cc	
		bcc	
		Subject	Alert: SETPROG EXIT command used
<p>Alert: SETPROG EXIT command used System messages report the SETPROG EXIT command has been issued</p> <p>Alert id 4003 Date and time 21Apr2008 21:02:53.16 WTO message CSV420I MODULE C4RMAIN HAS BEEN ADDED TO EXIT IRREXV01</p> <p>Console ID PEASE7 System ID TESTHVS</p>			

Figure 14-54 SETPROG alert email

SYSLOG message trapping by zSecure Alert is a powerful tool for the security officer, allowing possible security breaches and data loss situations due to system events initiated by authorized and privileged users to be automatically detected and instantly acted upon. The combination of predefined alerts for common system events that are included with the product and the flexibility of defining your own customized alerts add up to a powerful tool in your monitoring and alerting toolkit.

14.2.7 Monitoring RACF OPERATIONS attribute use

To bring TRA into accordance with Delft Transport security policy, we are required to monitor the activity of users with the RACF system attribute OPERATIONS.

In this section, we describe a method for reporting on the use of the RACF system attribute OPERATIONS. *Many organizations underestimate the power of their operations authorized users.* This attribute allows RACF alter level access to all data unless an access list explicitly defines an alternative access level. The attribute can also apply in general resource classes, such as DASDVOL. To be certain whether OPERATIONS applies in any general resource class, you must review your class descriptor table definitions.

Users with the OPERATIONS attribute represent a risk to the organization's data security by being able to effectively read or change all data where their access level is not explicitly defined. In respect to the DASDVOL class, these users may scratch entire DASD volumes of data unless additional controls are in place.

Most organizations fail to define their operation's user's access, and thus the risk generally extends to such things as employee, customer, or other sensitive data.

We do not recommend the use of the OPERATIONS attribute in production z/OS systems. Instead, we recommend using the facilities provided by your storage management products. All IBM z/OS storage related subsystems use resources in the class FACILITY to allow storage management staff to manage data without having the ability to also read or update this data. Alternative products generally also implement similar functionality. Using these facilities properly completely mitigates the risk to an organizations data that the storage management function might imply.

If, however, your organization is meeting with resistance or other issues with the use of FACILITY profiles in place of operations access by users, the strategy described below may be of assistance. In the case of TRA, we had to monitor and report on users with the OPERATIONS attribute, and we now describe this process.

For users that have been assigned the OPERATIONS attribute, we recommend you monitor their activity using zSecure Audit. The CARLa program shown in Example 14-12 enables you do this.

Example 14-12 CARLa program to report on the use of the OPERATIONS attribute

```
newlist type=smf title='Access granted due to OPERATIONS'
  define highest_access max(intent)
  select racfauth=operations event=access exists(intent)
  summary user * class * profile highest_access count
```

An example of the output is shown in Figure 14-55.

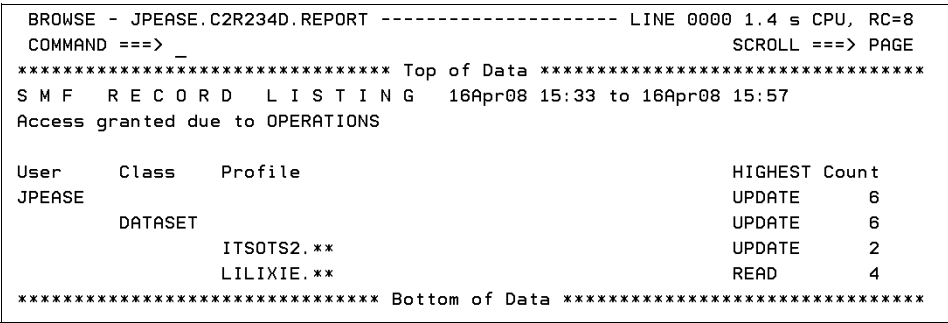


Figure 14-55 Reporting on use of OPERATIONS

Additionally, we prevented users with the OPERATIONS attribute from accessing sensitive data. We used the steps listed below to achieve this:

1. Create a RACF group (in our system OPSATTR).
2. Connect all users with OPERATIONS to the new group.
3. Add the new group to the access list of sensitive profiles with an access level of NONE.
4. We used a CARLa report with TYPE=SENSDSN to generate an initial set of target profiles. We also used the BESTMATCH= function to tie in the sensitive data set names found (in the CKFREEZE file) to the RACF data set profile currently protecting the involved data set(s). Additionally, we included our own set of sensitive data.

To assist you with the implementation of this control, we have provided a CARLa program, as shown in Example 14-13. This program performs the following tasks:

1. Searches for all user IDs with the OPERATIONS attribute that are not already connected to group OPSATTR.
2. Generates RACF commands to connect these user IDs to group OPSATTR.
3. Searches for all user IDs without the OPERATIONS attribute that are still connected to group OPSATTR.
4. Generates RACF commands to remove these user IDs from group OPSATTR.

Example 14-13 CARLa to automate connects and removes for group OPSATTR

```
mergelist
  newlist type=RACF f=CKRCMD nopage
  select class=user segment=base not(cggrpnm=OPSATTR) operations
  sortlist " connect " key(8) " group(OPSATTR) owner(OPSATTR) "

  newlist type=RACF f=CKRCMD nopage
  select class=user segment=base cggrpnm=OPSATTR not(operations)
  sortlist " remove " key(8) " group(OPSATTR) owner(OPSATTR) "
endmerge
```

You could schedule this CARLa program to execute as part of your daily batch suite.

We also provide a CARLa program that can be used to review the activity of operations users in an interactive ISPF display, as shown in Example 14-14.

Example 14-14 CARLa program to display operations activity

```
N TYPE=SMF detail retain
S type=(80,81,83) (racfauth=(operations))
display user(more) user:name user:dfltgrp user:cggrpnm,
time profile,
profile:uacc profile:ac1
dsummary user recorddesc date(9)
n type=racf,
  subtitle='Users with system-wide OPERATIONS attribute'
select class=user operations
display key(8) pgmrname dfltgrp instdata cggrpnm
```

14.3 Conclusion

In this chapter, we have demonstrated the initial stages of how to implement an advanced set of audit reporting and real-time monitoring, all facilitated by elements of the IBM Security zSecure Suite. Processes such as this are a significant part of what helps an organization achieve compliance with internal or regulatory requirements, and to be able to demonstrate this compliance as and when required.

Without this auditing and monitoring, an organization has no information with which to judge its own security posture. However, organizations with well designed and implemented security regimes, such as we have described here, are measurably less likely to experience security failures and the inevitable consequences.

More so, such organizations are better placed than their competitors to accept business and operational risks. Safe in the knowledge that their own infrastructures are well secured and monitored, these organizations can more readily take advantage of new business initiatives and opportunities that might open their systems for closer integration with partners, suppliers, and customers.

Comprehensive auditing and monitoring are a prerequisite to successful business in IT integration in the 21st century, and the IBM Security zSecure Suite is there to prepare your business for this task.

In the next and final phase of the Delft Transport integration of TRA facilities, we will describe the use of the zSecure suite to easily implement what is traditionally considered one of the most difficult tasks in IT Security Management: fully delegated security management to users.



Implementation phase III

In this chapter, we cover the final phase of the Delft Transport security improvement transformation of the TRA systems. We describe the processes and structures that help to establish a delegated security management framework in RACF, using the IBM Security zSecure Suite.

Traditionally, the concepts of delegated administration within RACF have been complex and not well understood, leading many organizations to adopt a purely centralized administration approach for the z/OS platform. Poorly implemented delegation of RACF authority has also lead to data exposures and audit issues. We will show you how zSecure can alleviate this complexity, and how to safely delegate limited RACF administrative privileges to your users.

First, an explanation of what we mean when we say delegated security administration. Delegation of security administration privileges is basically just pushing the security administration responsibilities and authorities down into the organization, that is, away from some centralized business function whose primary responsibility is IT security, and towards business users of IT systems whose primary responsibility is usually not related to IT security. These business users have roles and responsibilities with the organization to actually do business related activities, be this customer facing, back room, operational, or other functions. Typically these business users have no specific IT skills or interest, but they are usually well placed to understand the real data security needs of themselves and their colleagues.

The advantages of using delegated administration include simple things such as the ability to have a business user's password reset directly by their manager and thus avoid using a central help desk, avoid the communication of a password across email or phone, and ensure the validity of the password request person to person. A manager usually knows and recognizes their employees, even if the reset request is done by telephone.

Another advantage of delegation with respect to data access is that a suitable local data owner will have intimate knowledge of the importance or sensitivity of data they work with, while a centralized security function can never hope to have personal knowledge of every piece of data they may grant access to. The theory is that a local data owner will act more responsibly to guard access to their sensitive data.

This approach can also free up time for central security administrators that they can spend on more challenging and important security tasks.

These are just a few of the benefits of delegating security management. However, there are also risks, and these must be mitigated should you choose to go down this security administration path.

This rest of this chapter discusses these risks and mitigations, while explaining how zSecure can be used to safely implement delegated administration.

This chapter covers the following topics:

- ▶ Delegated RACF administration
- ▶ Ensuring system integrity
- ▶ Processes for managing authorization
- ▶ Reporting processes
- ▶ Joiners, leavers, and movers processing
- ▶ Segregation of duties

15.1 Delegated RACF administration

Using either zSecure Admin or zSecure Visual is the easiest way to delegate RACF privileges in a controlled and safe manner. In Chapter 6, "IBM Security zSecure Visual" on page 111, we describe the six user roles that zSecure Visual implements by default. Here we will demonstrate further how these roles are controlled and limited within RACF, using examples from the RACF group tree that we defined in 13.5.3, "Implementing an improved RACF group tree structure" on page 241.

CKGRACF acts as a front end for RACF commands. It can be used to allow administrators access to issue certain RACF commands, without the administrator needing to have native RACF authorizations to perform these actions. The intention allows the simpler, task oriented levels of typical RACF delegated user administration to be implemented without the risk of using RACF native group special. Usually these are functions such as:

- ▶ Password management and user ID revoke/resume (the CKGRACF USER command).
- ▶ Connect/remove users from group (CKGRACF CMD REQ CONNECT/REMOVE).
- ▶ Permit users to access resources (CKGRACF CMD REQ PERMIT).

For higher level RACF administration, such as creating new user IDs or groups, the administrator must still possess the correct RACF attributes to do this task. However, the administrator can still be fully controlled using zSecure Command Verifier and a well designed group tree, within which the necessary *group special* attributes are granted.

To demonstrate establishing a delegated administration framework, we will use the six user IDs defined in Chapter 6, “IBM Security zSecure Visual” on page 111, VISUAL1 through VISUAL6. These were defined as having the following roles or administrative responsibilities:

- ▶ VISUAL1 Full: All administrator functions.
- ▶ VISUAL2 Group: Add, delete, and change groups.
- ▶ VISUAL3 Access List: Permit and delete access to profiles.
- ▶ VISUAL4 User: Create, delete, and manage user IDs.
- ▶ VISUAL5 Connect: Manage user connection and removal from groups.
- ▶ VISUAL6 Helpdesk: Reset passwords, enable and disable user IDs.

We defined a RACF group for each administrator role, with the basic authorities that the role required:

- ▶ @CKG1FLL: Full Admin, VISUAL1 connected
- ▶ @CKG2GRP: Group admin, VISUAL2 connected
- ▶ @CKG3USR: User admin, VISUAL3 connected
- ▶ @CKG4ACL: Access list admin, VISUAL4 connected
- ▶ @CKG5CON: Group connect admin, VISUAL5 connected
- ▶ @CKG6HLP: Help desk password admin, VISUAL6 connected

You can find a reference listing the RACF commands to create the initial authority profiles and permit relevant groups in Appendix C, “User roles for IBM Security zSecure Visual” on page 455. These roles are defined and documented in *IBM Tivoli zSecure Visual Server Manual Version 1.11*, SC23-6549.

In addition, each of the user IDs VISUAL2 through VISUAL6 were further restricted in the RACF group tree using CKGRACF scoping profiles. The function of scoping profiles was briefly discussed in Chapter 6, “IBM Security zSecure Visual” on page 111. We did not restrict the scoping of VISUAL1, as this user ID has full access privileges within zSecure Visual and would only be used by one of the central administrators possessing the system special attribute. Therefore, we will not discuss this administrator level further and will focus only on the scoped user IDs in this chapter.

To easily define and manage the scope of these user IDs, we defined some RACF groups for access to CKG scope profiles, which are:

- ▶ @A#STAFF: Admin scope over all users under the #STAFF group, with update access to:
 - CKG.SCP.G.#FUNCACC.#BRNCACC.**
 - CKG.SCP.G.#FUNCACC.#NOACCES.**
 - CKG.SCP.G.#FUNCACC.#USERACC.**
 - CKG.SCP.G.#USERS.#STAFF.**
 - CKG.SCP.G.#USERS.#STAFF.#BRANCH.**
 - CKG.SCP.ID.*.#B001.*
- ▶ @A#BRNCH: Admin scope over all users under the #BRANCH group, with update access to:
 - CKG.SCP.G.#FUNCACC.#BRNCACC.**
 - CKG.SCP.G.#FUNCACC.#NOACCES.**
 - CKG.SCP.G.#USERS.#STAFF.#BRANCH.**
 - CKG.SCP.ID.*.#B001.*
- ▶ @A#B001: Admin scope over all users under the branch #B001 group, update access to CKG.SCP.ID.*.#B001.*.
- ▶ @A#STECH: Admin scope over resources owned by the #SYSTECH group, update access to:
 - CKG.SCP.G.#FUNCACC.#BRNCACC.**
 - CKG.SCP.G.#FUNCACC.#USERACC.**
 - CKG.SCP.G.#RACFRES.#SYSTECH

- @A#APPS: Admin scope over groups under the #APPS group, with update access to CKG.SCP.G.#APPS.**.

Now we can start to control our administrators with a combination of groups. They will receive a role type group from the six @CKG prefixed set of groups, plus another scoping type group from the five @A# prefixed set of groups. The highest level administrator will receive a connection to all scope related groups.

The matrix shown in Table 15-1 will help understand these users' ultimate access.

Table 15-1 Access matrix for VISUAL1 through VISUAL6 administrators

	Delegated scope groups				
Functional role groups	@A#STAFF	@A#BRNCH	@A#B001	@A#STECH	@A#APPS
@CKG1FLL	VISUAL1	VISUAL1	VISUAL1	VISUAL1	VISUAL1
@CKG2GRP					VISUAL2
@CKG3USR	VISUAL3				
@CKG4ACL				VISUAL4	
@CKG5CON		VISUAL5			
@CKG6HLP			VISUAL6		

The extensions shown in Example 15-1 were added under the #FUNCACC section of the RACF group tree.

Example 15-1 RACF group tree changes for control of delegated administrators

#FUNCACC	Access Granting Structural/Administrative
#SECACC	Security Management Access Administrative
@A#STAFF	CKG scope over group #STAFF and #BRNCACC
@A#BRNCH	CKG scope over group #BRANCH and #BRNCACC
@A#B001	CKG scope over group #B001 only
@A#STECH	CKG scope over group #SYSTECH, #USERACC and #BRNCACC
@A#APPS	CKG scope over group #APPS
@CKG1FLL	Visual full admin
@CKG2GRP	Visual group admin
@CKG3USR	Visual user admin
@CKG4ACL	Visual access list admin
@CKG5CON	Visual connect admin
@CKG6HLP	Visual help desk admin

The intended use of each administrator now starts to become apparent, as described in the following paragraphs.

VISUAL2 is a group level administrator for application related groups. They can create/delete RACF groups under the #APPS part of the RACF tree. They can also create data set profiles for these groups, and permit groups from the #FUNCACC part of the tree to the data set profiles under their scope. With these authorities the administrator will be a suitably restricted, but totally effective, application data set and group administrator. The authority to actually add a new RACF group is still controlled by a group special connection. So we will require extra controls to fully ensure that VISUAL2 cannot step outside his administrative responsibilities.

VISUAL3 is a user ID administrator for the entire #STAFF section of the group tree. They should be able to create and delete users within this section of the tree, and connect these users to the basic groups required for systems access. They should also be able to perform password management for their staff. Once again, group special, and in this case class authorization of user, will be required for the addition and deletion of user IDs. We will suitably limit the effectiveness of the group special authority, though.

VISUAL4 is an access list administrator for the resources owned by the group #SYSTECH. This group owns some sensitive resources and this administrator is scoped so they can work with the access lists of some profiles, but not others. No other RACF attributes will be required to perform this role; CKGRACF can provide this function independently and in a controlled manner through scoping profiles.

VISUAL5 is a connect administrator for the users under #BRANCH. They can connect and remove users from any of the groups required for normal branch access. While group connections can be managed entirely using only controlled CKGRACF authorities, VISUAL5 will also be required to change the users' default groups due to the design of our group tree and will require group special to do this. We will explain how to appropriately control this group special authority.

VISUAL6 is a local password administrator for the users owned by branch office #B001. All this user can do is reset passwords for the local team in this branch. It is assumed that these authorities would be given out to someone actually working in the branch. No authorities other than those provided by CKGRACF will be required to perform this role.

As you can see, some of our delegated administrators will continue to make use of RACF native group special, while others will no longer require such authority.

We will now go on to describe precisely how each delegated administrator was finally set up in RACF and zSecure to perform these roles using the zSecure Admin ISPF interface.

15.1.1 Implementing zSecure Admin scoping

In this section, we walk through the functionality of each of our delegated administrators, and define any additional access they require to perform the functional roles we outlined for them in 15.1, “Delegated RACF administration” on page 322.

Local password administrator

We start with our lowest level admin, VISUAL6, the password administrator for the branch office represented by the RACF owning group #B001.

Using zSecure Admin ISPF option RA.U and default prompting, that is, no user ID mask selected, we see in Figure 15-1 that they can only view user IDs within their defined scope.

zSecure Admin+Audit for RACF USER overview										Line 1 of 13	
Command ==>										Scroll==> CSR	
All users										22 Apr 2008 18:18	
User	Complex	Name		DfltGrp	Owner	RIRP	SOR	gC	LCX	Grp	
— CSV001	SC76	CUST SERV REP 01		@CSV001	#B001				X	1	
— CSV002	SC76	CUST SERV REP 02		@CSV001	#B001				X	1	
— CSV003	SC76	CUST SERV REP 03		@CSV001	#B001				X	1	
— CSV005	SC76	CUST SERV REP 05		@CSV002	#B001				X	1	
— CSV006	SC76	CUST SERV REP 06		@CSV002	#B001				X	1	
— CSV007	SC76	CUST SERV REP 07		@CSV002	#B001				X	1	
— CSV008	SC76	CUST SERV REP 08		@CSV002	#B001				X	1	
— CSV009	SC76	CUST SERV REP 09		@CSV002	#B001				X	1	
— CSV010	SC76	CUST SERV REP 10		@CSV002	#B001				X	1	
— CSV011	SC76	CUST SERV REP 11		@CSV002	#B001				X	1	
— CSV012	SC76	CUST SERV REP 12		@CSV002	#B001				X	1	
— CSV025	SC76	CUST SERV REP 25		@CSV001	#B001				X	1	
— VISUAL6	SC76	VISUAL ADMIN6		@CKG6HLP	#SECADM						3
***** Bottom of Data *****											

Figure 15-1 User view of the local branch password administrator

This is a list of user IDs owned by the #B001 RACF group. Access to see this list is scoped through the XFACILIT profile CKG.SCP.ID.*.#B001.*. Note that #B001 is the RACF owning group of all user IDs in the list.

We can now try to change a users password using the P line command against the user ID CSV001, as shown in Figure 15-2.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - RACF - User password				
Command ==> _____				
Userid : CSV001 Name : CUST SERV REP 01 Instdata : S/N 000001 - SENIOR CUST SERV REP Last use date . . . : 21Apr2008 Password changed . : Last use time . . . : 00:00 Revoked : No Revoke date : Resume date : Protected : No Revoke inactive . . . : No				
Select action:				
1	1. New password	==>	==>	Options
	2. DEFAULT password		-	/ Password expired
	3. PREVIOUS password			/ Resume userid
	4. RANDOM password			- Ignore pw history
	5. Resume only			- Bypass pw rules
	6. Current password			- Bypass pw exits
	7. Make Protected			
	8. New password phrase			

Figure 15-2 Resetting a branch user's password

After entering the new password twice, we can see the CKGRACF command that will be used, as shown in Figure 15-3.

```
zSecure Admin+Audit for RACF - Confirm CKGRACF command
Command ==> _____

Confirm or edit the following CKGRACF command
user CSV001 pwset password(test) expired resume _____
_____
_____

Command execution . 4 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 CKGRACF command
```

Figure 15-3 CKGRACF command to reset a password

We use option 4, REQUEST CKGRACF command, in this case. After pressing Enter to process the CKGRACF command, our administrator is informed that the action was successful, and the user's password is reset and ready for immediate use.

We had previously established the default options for VISUAL6 using SE.4, and limited the panel options available in Figure 15-3 on page 329. Although this was not strictly necessary, it provides for less potential for error should the administrator select options for which they are not authorized. The setup options we chose are shown in Figure 15-4.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - Setup - Confirm				
Command ==> _____				
Action on command	. . <u>2</u>	1. Queue	2. Execute	3. Not allowed
Confirmation <u>4</u>	1. None	2. Deletes	3. Passwords 4. All
Command generation				
Enter "/" to select option(s)				
_ Overtyp e fields in panels				
_ Change generated commands				
/ Specify start/end date				
_ Generate SETROPTS REFRESH commands				
_ Issue prompt before generating SETROPTS REFRESH commands				
Commands to generate				
_ RACF commands				
/ CKGRACF commands				
_ CKGRACF ASK for later execution				
_ CKGRACF REQUEST for later execution				
_ CKGRACF WITHDRAW queued commands				
_ CKGRACF RDELETE commands				

Figure 15-4 Default setup options for a local password administrator

Our administrator can also select a detailed view of user information, as shown in Figure 15-5.

zSecure Admin+Audit for RACF USER overview									
Command ==> _____									
All users									
22 Apr 2008 18:37									
Line 1 of 55									
Scroll==> CSR									
_ Identification of CSV001									
SC76									
_ User name CUST SERV REP 01									
Installation data S/N 000001 - SENIOR CUST SERV REP									
_ Owner #B001									
_ User's default group @CSV001									
Group Auth R SOA AG Uacc Revokedt Resumedt InstData									
_ @CSV001 USE NONE									
System access									
Revoked (may be by date) No									
Inactive, revoked or pending No									
Days of week user can logon SMTWTFS									
Time of day user can logon									
Date user will be revoked									
Date user will be resumed									
Statistics									
Creation date 7Apr08									
Last RACINIT current connects 22Apr08									
User's last use date 22Apr08									
User's last use time 18:32									
(ddmmmyyyy or NOREVOKE)									
(ddmmmyyyy or NORESUME)									
Password									
Has a password Yes									
Password phrase									
Has a password phrase No									

Figure 15-5 User detail view

In addition to the XFACILIT profiles mentioned for scoping, and those defined for the basic @CKG6HLP role, we granted update access to the following two profiles to allow the administrator to perform the type of password reset we required:

```
CKG.CMD.USER.REQ.PWSET.EXPIRED
CKG.CMD.USER.REQ.PWSET.PASSWORD
```

We also created and granted access to a SCPASK profile for the same scope the administrator already had. This would allow them to ask higher level administrators to run commands for them. The profile used was CKG.SCPASK.ID.*.#B001.*. This topic is discussed further in 15.3.1, “Timed (queued) commands” on page 361.

Branch group connections administrator

We now look at the access of VISUAL5, our branch group connections administrator. Once again we have already set the SE.4 confirmation defaults for this user.

Using the default prompting in RA.U, this connect administrator can see all the user IDs involved across all the branches. You can see that some user IDs are owned by #B001, and others by #B002, #B003, #B004, and #B005, as shown in Figure 15-6.

zSecure Admin+Audit for RACF USER overview										Line 8 of 33	
Command ==>										Scroll==> CSR	
All users										22 Apr 2008 18:49	
User	Complex	Name	DfltGrp	Owner	RIRP	SOR	gC	LCX	Grp		
— CSV008	SC76	CUST SERV REP 08	@CSV002	#B001				X	1		
— CSV009	SC76	CUST SERV REP 09	@CSV002	#B001				X	1		
— CSV010	SC76	CUST SERV REP 10	@CSV002	#B001				X	1		
— CSV011	SC76	CUST SERV REP 11	@CSV002	#B001				X	1		
— CSV012	SC76	CUST SERV REP 12	@CSV002	#B001				X	1		
— CSV013	SC76	CUST SERV REP 13	@CSV001	#B003				X	1		
— CSV014	SC76	CUST SERV REP 14	@CSV001	#B003				X	1		
— CSV015	SC76	CUST SERV REP 15	@CSV001	#B003				X	1		
— CSV016	SC76	CUST SERV REP 16	@CSV001	#B003				X	1		
— CSV017	SC76	CUST SERV REP 17	@CSV001	#B004				X	1		
— CSV018	SC76	CUST SERV REP 18	@CSV001	#B004				X	1		
— CSV019	SC76	CUST SERV REP 19	@CSV001	#B004				X	1		
— CSV020	SC76	CUST SERV REP 20	@CSV001	#B004				X	1		
— CSV021	SC76	CUST SERV REP 21	@CSV001	#B005				X	1		
— CSV022	SC76	CUST SERV REP 22	@CSV001	#B005				X	1		
— CSV023	SC76	CUST SERV REP 23	@CSV001	#B005				X	1		
— CSV024	SC76	CUST SERV REP 24	@CSV001	#B005				X	1		
— CSV025	SC76	CUST SERV REP 25	@CSV001	#B001				X	1		
— MAN001	SC76	BRANCH MANAGER 01	@MAN001	#B003				X	1		
— REG001	SC76	REGIONAL MANAGER 01	@REG001	#B004				X	1		

Figure 15-6 Branch wide group connections administrator

Again, using default prompting and looking at groups rather than user IDs through option RA.G, we can see the list of groups that VISUAL5 can view in Figure 15-7.

zSecure Admin+Audit for RACF GROUP Overview						0 s elapsed, 0.1 s CPU			
Command ==>						Scroll==> CSR			
All profiles						22 Apr 2008 18:50			
Group	Complex	SupGroup	X Owner	Grps	Users	Conn	U nTU	Created	
— #B001	SC76	#BRANCH	#BRANCH					07Apr2008	
— #B002	SC76	#BRANCH	#BRANCH					07Apr2008	
— #B003	SC76	#BRANCH	#BRANCH					07Apr2008	
— #B004	SC76	#BRANCH	#BRANCH					07Apr2008	
— #B005	SC76	#BRANCH	#BRANCH					16Apr2008	
— @CSV001	SC76	#USERACC	#USERACC		17	17		07Apr2008	
— @CSV002	SC76	#USERACC	#USERACC		13	13		07Apr2008	
— @CSV003	SC76	#USERACC	#USERACC					21Apr2008	
— @HR001	SC76	#USERACC	#USERACC					21Apr2008	
— @HR002	SC76	#USERACC	#USERACC					21Apr2008	
— @MAN001	SC76	#USERACC	#USERACC		1	1		07Apr2008	
— @OP001	SC76	#USERACC	#USERACC		4	4		14Apr2008	
— @REG001	SC76	#USERACC	#USERACC		1	1		07Apr2008	
— @SP001	SC76	#USERACC	#USERACC		2	2		07Apr2008	
— @SP002	SC76	#USERACC	#USERACC		85	85		07Apr2008	
— @TSTACC	SC76	#USERACC	#USERACC		12	12		09Apr2008	
***** Bottom of Data *****									

Figure 15-7 Groups available to branch wide connections administrator

Going back into the list of user IDs from option RA.U, we want to promote CSV006 from a junior customer services to senior. To accomplish this task, a few steps are required:

1. Connect them to the @CSV001 ‘Senior customer services’ access group.
2. Change their current RACF default group of @CSV002 to @CSV001.
3. Remove them from their old group of @CSV002.

They will then have the access of a senior customer services officer. The panels below demonstrate these steps. First, we simply select the CSV006 user ID detail display, as shown in Figure 15-8. You can see that we have used the line command C to copy this group connection.

zSecure Admin+Audit for RACF USER overview										Line 1 of 54
Command ==> _____										Scroll==> <u>CSR</u>
All users										23 Apr 2008 17:27
_ Identification of CSV006										SC76
User name										<u>CUST SERV REP 06</u>
Installation data										<u>S/N 000006 - JUNIOR CUST SERV REP</u>
_ Owner										<u>#B001</u> PISA CUST SERV CE
_ User's default group										<u>@CSV002</u> JUNIOR CUSTOMER S
Group	Auth	R	SOA	AG	Uacc	Revokedt	Resumedt	InstData		
c @CSV002	USE	-	-	-	NONE			JUNIOR CUSTOMER SER		
System access					Statistics					
Revoked (may be by date)					<u>No</u>	Creation date				16Apr08
Inactive, revoked or pending					No	Last RACINIT current connects				
Days of week user can logon					<u>SMTWTFS</u>	User's last use date				16Apr08
Time of day user can logon					_____	User's last use time				
Date user will be revoked					_____	(ddmmmyyyy or NOREVOKE)				
Date user will be resumed					_____	(ddmmmyyyy or NORESUME)				
Password					Password phrase					
Has a password					Yes	Has a password phrase				No

Figure 15-8 User detail display about to change RACF role groups

We enter the required new group connection, @CSV001, in the panel shown in Figure 15-9 on page 335. We will be prompted with the resulting command, as shown in Figure 15-10 on page 335.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - RACF - Copy connect				
Command ==> _____				
Create connect like existing connect of user CSV006				
Change at least one field				
Userid <u>CSV006</u> (was CSV006)				
Group <u>@CSV001</u> (was @CSV002)				
Optional connect attributes				
Authority <u>USE</u>				
Default URAC <u>NONE</u>				
Connect owner _____				
Future revoke date . _____ (YYYY-MM-DD or DD MMM YYYY)				
Future resume date . _____ (YYYY-MM-DD or DD MMM YYYY)				
_ Revoke				
_ Special _ Operations _ Auditor				
Optional processing mode				
_ Modify existing connect				

Figure 15-9 Entering the new group connection details

zSecure Admin+Audit for RACF - Confirm command	
Command ==> _____	
Confirm the following command	
connect CSV006 group(@CSV001)	
Command execution . <u>4</u>	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 15-10 Connect to new group using CKGRACF REQ

We chose option 4, 'REQUEST CKGRACF command for later execution' in Figure 15-10 on page 335, even though we want the command to occur immediately. Our CKGRACF authorized scoping will allow us to change this connection immediately. We do not need to specify a start or end date.

If we tried to use either option 1, 'EXECUTE RACF command' or 2, 'EXECUTE CKGRACF command', the command would fail, as we have no native RACF authority to issue the proper RACF CONNECT command for this group.

We now need to change the users' RACF default group. This is the part where we do actually require native RACF authority of some kind. In this case, we have group special over the owning groups for the branch users, #BRANCH.

Unfortunately, we must refresh our current view of the users' state from the RACF database to see these changes before we can update the default group. To do this, we need to exit out to at least the RA.U primary panel, and reselect our target users to work with. In this case we continue to use just default prompting.

Upon selecting our target user, we can see that they now have two connect groups. We need to overtype the default group, and delete the unwanted connect group. We can do both these actions on the same panel, as shown in Figure 15-11.

```

zSecure Admin+Audit for RACF USER overview                                     Line 1 of 55
Command ==> _____ Scroll==> CSR
All users                                                                    23 Apr 2008 18:00

_ Identification of CSV006                                                    SC76
  User name                        CUST SERV REP 06
  Installation data                S/N 000006 - JUNIOR CUST SERV REP
  Owner                           #B001                                     PISA CUST SERV CE
  User's default group             @csv001                                JUNIOR CUSTOMER S

  Group   Auth   R SOA AG Uacc   Revokedt   Resumedt   InstData
  _ @CSV001 USE   -   -   - NONE   -         -         SENIOR CUSTOMER SER
d @CSV002 USE   -   -   - NONE   -         -         JUNIOR CUSTOMER SER

  System access                      Statistics
  Revoked (may be by date)           No           Creation date                16Apr08
  Inactive, revoked or pending        No           Last RACINIT current connects
  Days of week user can logon         SMTWTFS    User's last use date                16Apr08
  Time of day user can logon          -           User's last use time
  Date user will be revoked           -           (ddmmmyyyy or NOREVOKE)
  Date user will be resumed           -           (ddmmmyyyy or NORESUME)

  Password                          Password phrase

```

Figure 15-11 Change default and remove old group at once

After pressing Enter, we will be prompted with a sequence of two command panels. The first of these will change the default group, and we must use our native RACF authority for this task, so we select option 1, as shown in Figure 15-12.

zSecure Admin+Audit for RACF - Confirm command

Command ==> _____

Confirm the following command
altuser CSV006 dfltgrp(@CSV001)

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 15-12 The altuser command to change default group

The second command panel will remove the old group, as shown in Figure 15-13. We use option 4 to remove this group connection by using our CKGRACF scoping.

zSecure Admin+Audit for RACF - Confirm command	
Command ==> _____	
Confirm the following delete command remove CSV006 group(@CSV002)	
Command execution . <u>4</u>	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 15-13 The remove command to remove the old group

These commands will run and we can again refresh our list of users to review the changes.

We did not need to establish any extra authority in the class XFACILIT for this administrator to perform this role. However, we did require a group special connection over the group #BRANCH in our group tree.

Resource access list administrator

Now we look at our access list administrator, VISUAL4. He has no authority to list users, but by using the default prompting in RA.G, he can see the following list of user groups. He is scoped by CKGRACF such that these are the only groups that he may grant access to, as shown in Figure 15-14.

zSecure Admin+Audit for RACF GROUP Overview										Line 1 of 7
Command ==>										Scroll==> CSR
All profiles										22 Apr 2008 19:42
Group	Complex	SupGroup	X Owner	Grps	Users	Conn	U nTU	Created		
— #SYSTECH	SC76	#RACFRES	#RACFRES					09Apr2008		
— @HR001	SC76	#USERACC	#USERACC					21Apr2008		
— @HR002	SC76	#USERACC	#USERACC					21Apr2008		
— @OP001	SC76	#USERACC	#USERACC		4	4		14Apr2008		
— @SP001	SC76	#USERACC	#USERACC		2	2		07Apr2008		
— @SP002	SC76	#USERACC	#USERACC		85	85		07Apr2008		
— @TSTACC	SC76	#USERACC	#USERACC		12	12		09Apr2008		
***** Bottom of Data *****										

Figure 15-14 Group scope of the resource administrator

He can also review a list of RACF resources using RA.R, as shown in Figure 15-15.

zSecure Admin+Audit for RACF General resource overview										Line 21 of 57
Command ==>										Scroll==> CSR
Class XFACI*										22 Apr 2008 19:48
Class	Profile key	T	UACC	Owner	S/F	W				
— XFACILIT	CKG.CMD.USER.REQ.PWSET.DEFAULT		NONE	SYS1	R					
— XFACILIT	CKG.CMD.USER.REQ.PWSET.EXPIRED		NONE	SYS1	R					
— XFACILIT	CKG.CMD.USER.REQ.PWSET.NONEXP		NONE	SYS1	R					
— XFACILIT	CKG.CMD.USER.REQ.PWSET.PASSWORD		NONE	SYS1	R					
— XFACILIT	CKG.CMD.USER.REQ.PWSET.PREVIOUS		NONE	SYS1	R					
— XFACILIT	CKG.CMD.USER.REQ.RESUME		NONE	SYS1	R					
— XFACILIT	CKG.CMD.USER.REQ.SCHEDULE		NONE	SYS1	R					
— XFACILIT	CKG.RAC.SCP.CONNECT.BASE.AUTH.USE		NONE	SYS1	R					
— XFACILIT	CKG.RAC.SCP.XFACILIT.BASE.ACCESS.ALTER		NONE	#SYSTECH	R					
— XFACILIT	CKG.RAC.SCP.XFACILIT.BASE.ACCESS.NONE		NONE	#SYSTECH	R					
— XFACILIT	CKG.RAC.SCP.XFACILIT.BASE.ACCESS.*		G NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#APPS.**		G NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#FUNCACC.#USERACC.**		G NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#RACFRES.#CV		NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#RACFRES.#SYSTECH		NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#RACFRES.**		G NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#USERS.#STAFF.#BRANCH.**		G NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#USERS.#STAFF.#SECADM		NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.#USERS.#STAFF.**		G NONE	#SYSTECH	R					
— XFACILIT	CKG.SCP.G.**		G NONE	#SYSTECH	R					

Figure 15-15 Resources that the resource administrator can see

However, if he selects certain resources, he cannot see their access list entries. These resources are outside his CKGRACF scope, and he cannot grant access to these groups, as shown in Figure 15-16.

```

zSecure Admin+Audit for RACF General resource overview
Command ==> _____
Class XFACI*
22 Apr 2008 19:48
Line 1 of 33
Scroll==> CSR

```

Identification					SC76
Class		XFACILIT			
Profile name		CKG.CMD.USER.REQ.PWSET.DEFAULT			
Type					
Volume serial list					
- Owner		SYS1			
Installation data					
Application data					

User	Access	ACL id	When	Name	InstData
Safeguards					
User to notify of violation			Other permissions		
Audit access success/failures			R	Allow all accesses	WARNING No
Global audit success/failures				Universal access authority	NONE
				Resource level	0

Figure 15-16 A resource that is not within the resource administrator's scope

The resources within his scope are those owned by the RACF group #SYSTECH. He can see the access list entries for this group. In Figure 15-17, he has selected to copy an access list entry using the C line command.

zSecure Admin+Audit for RACF General resource overview						Line 1 of 34
Command ==> _____						Scroll==> <u>CSR</u>
Class XFACI*						22 Apr 2008 19:48
Identification						SC76
Class						XFACILIT
Profile name						CKG.SCP.G.#APPS.**
Type						GENERIC
Volume serial list						
- Owner						#SYSTECH RACF RESOURCE OWN
Installation data						
Application data						
User	Access	ACL id	When	Name	InstData	
<u>C</u> -group-	<u>UPDATE</u>	<u>@A#APPS</u>	_____	_____		
-group-	<u>UPDATE</u>	<u>@ZSEC001</u>	_____	_____		
Safeguards						Other permissions
User to notify of violation						Allow all accesses WARNING No
Audit access success/failures R						Universal access authority NONE
Global audit success/failures						Resource level 0

Figure 15-17 A resource within the resource administrator's scope

After pressing Enter, he can fill in the required group name and access level for the new permit, as shown in Figure 15-18.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - RACF - Copy permit				
Command ==> _____				
Change at least one field				
User or group	. . .	<u>@sp001</u>		(was group @A#APPS)
Access level	. . .	<u>UPDATE</u>		(was UPDATE)
Class	<u>XFACILIT</u>		(was XFACILIT)
Profile name	<u>CKG.SCP.G.#APPS.**</u>		
_____ (was CKG.SCP.G.#APPS.**)				
Optional conditions for the permit				
When class	_____		(was empty)
When resource/profile	_____			
(was empty)				

Figure 15-18 Resource administrator issuing a permit command

This administrator has no additional RACF authorities and only has CKGRACF scope over the group that owns the resources he administers, and the user groups that he needs to make permits for.

We did have to add the following extra profiles to have full permission to issue any permit for these resources:

CKG.RAC.SCP.XFACILIT.BASE
CKG.RAC.SCP.XFACILIT.BASE.ACCESS.*

User administrator

VISUAL3 is a user administrator with RACF group special over #BRNCACC, #BRANCH, #OP, and #SP, that is, group special over the owner groups for all user staff, with the exception of the security administrators owned by group #SECADM.

RA.U shows the list of users he can work with, as shown in Figure 15-19.

zSecure Admin+Audit for RACF USER overview										Line 11 of 150	
Command ==> _____										Scroll==> CSR	
All users										23 Apr 2008 20:59	
User	Complex	Name	DfltGrp	Owner	RIRP	SOR	gC	LCX	Grp		
— BWILSON	SC76	BRUCE WILSON	@SP002	#SP	I			X	1		
— CAG00D	SC76	CLARK GOODRICH	@SP002	#SP	I			X	1		
— CCL01	SC76	BILL WHITE	@SP002	#SP	I			X	1		
— CCL02	SC76	BILL WHITE	@SP002	#SP	I			X	1		
— CCL03	SC76	BILL WHITE	@SP002	#SP	I			X	1		
— CCL04	SC76	BILL WHITE	@SP002	#SP	I			X	1		
— CCL05	SC76	BILL WHITE	@SP002	#SP	I			X	1		
— CONWAYM	SC76	MONIQUE CONWAY	@ZSEC001	#SECADM		S A			3		
— CSV001	SC76	CUST SERV REP 01	@CSV002	#B001				X	1		
— CSV002	SC76	CUST SERV REP 02	@CSV001	#B001				X	1		
— CSV003	SC76	CUST SERV REP 03	@CSV001	#B001				X	1		
— CSV004	SC76	CUST SERV REP 04	@CSV001	#B002				X	1		
— CSV005	SC76	CUST SERV REP 05	@CSV002	#B001				X	2		
— CSV006	SC76	CUST SERV REP 06	@CSV001	#B001				X	1		
— CSV007	SC76	CUST SERV REP 07	@CSV002	#B001				X	1		
— CSV008	SC76	CUST SERV REP 08	@CSV002	#B001				X	1		
— CSV009	SC76	CUST SERV REP 09	@CSV002	#B001				X	1		
— CSV010	SC76	CUST SERV REP 10	@CSV002	#B001				X	1		
— CSV011	SC76	CUST SERV REP 11	@CSV002	#B001				X	1		
— CSV012	SC76	CUST SERV REP 12	@CSV002	#B001				X	1		

Figure 15-19 Users the user administrator can see

He can also see a full list of RACF groups, but can only work with those he has scope over, as shown in Figure 15-20.

zSecure Admin+Audit for RACF GROUP Overview										0 s elapsed, 0.1 s CPU	
Command ==> _____										Scroll==> CSR	
All profiles										23 Apr 2008 21:02	
Group	Complex	SupGroup	X Owner	Grps	Users	Conn	U	nTU	Created		
___ #BRANCH	SC76	#STAFF	#STAFF	5	2	2			07Apr2008		
___ #B001	SC76	#BRANCH	#BRANCH						07Apr2008		
___ #B002	SC76	#BRANCH	#BRANCH						07Apr2008		
___ #B003	SC76	#BRANCH	#BRANCH						07Apr2008		
___ #B004	SC76	#BRANCH	#BRANCH						07Apr2008		
___ #B005	SC76	#BRANCH	#BRANCH						16Apr2008		
___ #DELETE	SC76	#SECADM	#SECADM		3	3			22Apr2008		
___ #IT	SC76	#STAFF	#STAFF	3					07Apr2008		
___ #MOVER	SC76	#SECADM	#SECADM						23Apr2008		
___ #OP	SC76	#IT	#IT		1	1			14Apr2008		
___ #SECADM	SC76	#IT	#IT	2					07Apr2008		
___ #SP	SC76	#IT	#IT		1	1			13Apr2008		
___ #TSTUSR	SC76	#STAFF	#STAFF						09Apr2008		
___ @CSV001	SC76	#BRNCACC	#BRNCACC		18	18			07Apr2008		
___ @CSV002	SC76	#BRNCACC	#BRNCACC		13	13			07Apr2008		
___ @CSV003	SC76	#BRNCACC	#BRNCACC						21Apr2008		
___ @HR001	SC76	#USERACC	#USERACC						21Apr2008		
___ @HR002	SC76	#USERACC	#USERACC						21Apr2008		
___ @MAN001	SC76	#BRNCACC	#BRNCACC		1	1			07Apr2008		
___ @NOACCES	SC76	#NOACCES	#NOACCES						23Apr2008		

Figure 15-20 Groups within and without the group administrator's scope

He can copy a user ID simply using the line command C and filling in the required fields. You can see the fields completed for the new user definition in Figure 15-21.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - RACF - User copy				
Command ==> _____				
From userid : CSV025				
To id : <u>CSV026</u>				
Password : <u>today1</u>				
Name : <u>CUST SERV REP 26</u>				
Owner : <u>#B005</u> Default group : <u>@CSV002</u>				
Installation data . . . : <u>S/N 000026 - junior_CUST SERV REP</u>				
<ul style="list-style-type: none"> _ Copy permits only (target id may be a group or a user) _ Generate RACF commands even when the target user exists _ Copy USERDATA _ Specify unique segment data _ Revoke new userid _ Protected _ Copy catalog aliases (only if CKFREEZE is present) _ Issue ADDSD/RDEF for dataset and resource profiles related to the user _ Copy RACFVARS profiles/members too 				

Figure 15-21 Creating a new user ID

RACF commands are generated to define the new user ID. These all execute using the user administrators native RACF authorities, as shown in Figure 15-22.

File	Edit	Edit_Settings	Menu	Utilities	Compilers	Test	Help
EDIT VISUAL3.C2R195E.CKRCMD						Columns 00009 00080	
Command ==> _____						Scroll ==> <u>CSR</u>	
***** Top of Data *****							
000001 /* CKRCMD file CKR1CMD complex SC76 generated 23 Apr 2008 21:05 */							
000002 /* CKRCMD file CKR1CMD complex SC76 generated 23 Apr 2008 21:08 */							
000003 /* WARNING: Create requested for existing user CSV025 */							
000004 /* CKRCMD file CKR1CMD complex SC76 generated 23 Apr 2008 21:10 */							
000005 /* Commands generated by COPY USER/GROUP */							
000006 adduser CSV026 password(today1) +							
000007 name('CUST SERV REP 26 ') +							
000008 data('S/N 000026 - JUNIOR CUST SERV REP') +							
000009 owner(#B005) +							
000010 dfltgrp(@CSV002)							
000011 connect CSV026 group(@CSV001) owner(@CSV001) auth(USE) uacc(NONE)							
000012 connect CSV026 group(@CSV002) owner(@CSV002)							
***** Bottom of Data *****							

Figure 15-22 RACF commands to create a new user ID

Group and data set profile administrator

VISUAL2 is a group and data set administrator for applications, with a RACF group scope of special over the #APPS branch of the tree.

The administrator cannot see the user IDs using RA.U, but can see this list of groups under RA.G, as shown in Figure 15-23.

zSecure Admin+Audit for RACF GROUP Overview									
Command ==> _____									
All profiles									
22 Apr 2008 20:38									
Group	Complex	SupGroup	X	Owner	Grps	Users	Conn	U	nTU Created
— #APPS	SC76	SYS1		SYS1	4	1	1		07Apr2008
— #DEV	SC76	#APPS		#APPS					07Apr2008
— #MNT	SC76	#APPS		#APPS					07Apr2008
— #PRD	SC76	#APPS		#APPS	3				07Apr2008
— #TST	SC76	#APPS		#APPS					07Apr2008
— FINANCE	SC76	#PRD		#PRD					21Apr2008
— HR	SC76	#PRD		#PRD					21Apr2008
— STOCK	SC76	#PRD		#PRD					21Apr2008
***** Bottom of Data *****									

Figure 15-23 Groups within the group administrator's scope

The administrator can copy a group, using line command C, and the commands are generated. In Figure 15-24, you can see that a data set profile is also created for group data sets. These commands run using the administrator's native RACF authorities.

File Edit Edit_Settings Menu Utilities Compilers Test Help									
EDIT		VISUAL2.C2R11F1.CKRCMD					Columns 00009 00080		
Command ==>							Scroll ==> CSR		
***** Top of Data *****									
000001	/* CKRCMD file CKR1CMD complex SC76 generated 22 Apr 2008 20:45 */								
000002	/* CKRCMD file CKR1CMD complex SC76 generated 22 Apr 2008 20:45 */								
000003	/* Commands generated by COPY USER/GROUP */								
000004	addgroup TESTDATA +								
000005	owner(#PRD) supgroup(#PRD)								
000006	/* Commands generated by COPY PERMIT */								
000007	addsd 'TESTDATA.***' generic from('FINANCE.***') fgeneric								
000008	altdsd 'TESTDATA.***' generic owner(TESTDATA)								
000009	SETROPTS REFRESH GENERIC(DATASET)								
***** Bottom of Data *****									

Figure 15-24 RACF commands to create a new group

The group and data set administrator can also manage the access list of data set profiles within his scope. In Figure 15-25, you can see the permit command about to be issued for the FINANCE.** data set profile. Note that this is the only profile on the panel for which the administrator can issue any commands, so it contains the overtappable fields for UACC and Owner.

zSecure Admin+Audit for RACF DATASET Overview				Line 115 of 547
Command ==>				Scroll==> CSR
All profiles		22 Apr 2008 20:50		
Profile key	Type	UACC	Owner	S/F W
— EUV.SEUVLPA	GENERIC READ	EUV		U R
— EUV.**	GENERIC READ	EUV		R
— EUVF.SEUVFLNK	GENERIC READ	EUVF		U R
— EUVF.**	GENERIC READ	EUVF		R
— FFST.V120ESA.SEPWMOD1	GENERIC READ	FFST		U R
— FFST.V120ESA.SEPWMOD2	GENERIC READ	FFST		U R
— FFST.V120ESA.SEPWMOD4	GENERIC READ	FFST		U R
— FFST.**	GENERIC READ	FFST		R
<u>pe</u> FINANCE.**	GENERIC <u>NONE</u>	<u>FINANCE</u>		<u>R</u> —
— FMN.SFMNMOD1	GENERIC READ	FMN		U R
— FMN.**	GENERIC READ	FMN		R
— GDDM.SADMMOD	GENERIC READ	GDDM		U R
— GDDM.**	GENERIC READ	GDDM		R
— GIM.SGIMLMD0	GENERIC READ	GIM		U R
— GIM.**	GENERIC READ	GIM		R
— GLD.SGLDLNK	GENERIC READ	GLD		U R
— GLD.**	GENERIC READ	GLD		R
— GSK.SGSKLOAD	GENERIC READ	GSK		U R
— GSK.**	GENERIC READ	GSK		R
— GXM.**	GENERIC READ	GXM		R

Figure 15-25 Granting access to a data set profile

The permit commands brings up the panel shown in Figure 15-26, where the new group to be granted access and the access level may be specified.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - RACF - Add or delete permit				
Command ==> _____				
Add or delete permit				
Id	@op001	(user or group profile key or filter)	
Access level	read	(N/E/R/U/C/R/D)	
Class	DATASET	(class name)	
Profile key	'FINANCE.**'		
Profile type				
Optional conditions for the permit				
When class	_____		
When resource/profile	_____			

Figure 15-26 Details for a permit

Conclusion

You can see from the above descriptions that we have implemented strong segregation between the various roles involved in security administration. We have also been able to delegate these roles to staff throughout the business user community in a safe and controlled manner. All this has been achieved quite simply with the use of a few RACF profiles. These are used by the CKGRACF component of zSecure Admin to provide controls above and beyond those available in native RACF.

In 15.2.7, “Additional controls required for group special users” on page 358, we describe how to prevent the users who have been granted RACF group special authorities from using them in any way beyond their intended administrative roles.

15.2 Ensuring system integrity

It can be difficult for RACF administrators to prevent noncompliant administrative commands. However, this is critical task in maintaining a clean environment. Mistakes made when granting authorities or ignoring procedures, such as naming conventions for defining users, will result in a polluted security database.

In the long term, if unchecked, this behavior could require countless hours and a major project to rectify. Potentially worse, in the short term this can leave your infrastructure open to vulnerabilities and serious audit concerns.

So after finishing Phase I, the establishment of a trusted computing base in the TRA mainframe system, and then Phase II, the implementation of enhanced auditing and monitoring, we plan to use zSecure Command Verifier to enforce and maintain policy compliance within RACF.

We introduced the zSecure Command Verifier in Chapter 7, “IBM Security zSecure Command Verifier” on page 131. The zSecure Command Verifier acts as a filter for RACF commands as they are entered. Command Verifier intercepts commands, verifies them against security policies, and determines whether or not they should be cancelled, executed, or modified. In this section, we describe more detailed use of the policy rules, which are defined using normal RACF profiles with the prefix ‘C4R’. These profiles are held in the XFACILIT class, unless the installation has chosen to use a different RACF class. We use zSecure Admin to easily define and manage these profiles.

There are a myriad of controls available, described in detail in *IBM Security zSecure Command Verifier User Reference Manual Version 1.12*, SC27-2779. We will briefly outline a few of the controls we implemented here.

15.2.1 Enforcing standards

In this section, we describe the use of C4R prefixed RACF profiles to implement controls on data set profiles, user ID naming conventions, and passwords. This is of particular importance when using delegated administration, as we need to ensure that all administrators follow the correct procedures.

Enforcing generic profiles for all data sets

RACF allows users to protect data sets with discrete or generic profiles. The use of discrete profiles however is mostly historic, and generally not recommended in modern RACF installations.

Using generic profiles, your installation can reduce both the number of profiles required to protect data sets and improve the performance of your RACF database. Additionally, generic profiles are much easier to administer in an SMS managed environment where you cannot always guarantee that a specific data set will exist on a specific DASD volume. Also, generic profiles are loaded into storage when first needed and are not deleted when the data set they protect is deleted.

Generic profiles often contain a RACF generic character, either the % or the * (percent or asterisk) character or some combination of these. However, they may also be defined without an generic characters, in which case they are referred to as “fully qualified generics”, that is, containing no wildcard characters.

To ensure that all new data set profiles are generic, even if they are fully qualified, we implement the following command policy profile:

```
C4R.DATASET.TYPE.DISCRETE.**
```

Throughout this section, we define all command policy profiles in the class XFACILIT with an owner of #CV and UACC of NONE. We also use RACF installation data to provide a meaningful description of the function being implemented. We recommend that you use a similar practice with respect to profile installation data, as this greatly assists problem diagnoses and identifying the correct profile to use for any required control.

The use of a single owning group, #CV in our case, for all Command Verifier profiles will help us segregate administrative authorities at a later point, as described in 15.6, “Segregation of duties” on page 401.

Enforcing user ID naming conventions

Without proper naming conventions, it becomes difficult to navigate and understand the meaning of definitions in your RACF database. The same can be said of z/OS in general. The best run z/OS installations enforce strict naming conventions as much as possible, and document them to ease the task of understanding the overall system setup. Strong RACF naming conventions in general are of great help in this regard.

RACF commands that do not comply with your naming conventions can be issued by virtually any authorized RACF administrator. In this section, we show some methods that may be used to prevent administrators from stepping outside the bounds of your naming convention for user IDs. These techniques can be applied equally to any RACF definitions, including groups, data sets, and general resources.

For this example, we assume that all new user IDs will have a length of five characters, will start with either an S or A or O, and be followed with the character P. Thus, all user IDs must be prefixed with either SP, AP, or OP.

We then require these new user IDs to be suffixed with a number three digits in length. Thus, valid user IDs would be SP001, AP002, and OP003.

To do this, we first need to implement some profiles in the class RACFVARS. RACFVARS stands for *RACF Variables*. Profiles defined in the RACFVARS class can be used to make up the profile name in other general resource profiles. When used this way, they will be substituted with the values that have been defined as members in the specific RACFVARS profile when determining the ultimate profile name. This can be considered an advanced use of RACF profiles. Note that RACFVARS resources may not be used in the class data set.

Profiles in the RACFVARS class must be prefixed with an & (ampersand) character. Although these profiles may be up to eight characters in length, we recommend using short, symbolic profile names for these to keep the resulting C4R prefixed profile's intent recognizable.

We use these RACFVARS profiles later as part of the profile name in an appropriate XFACILIT class profile for the C4R resources.

The following example will help clarify this complex RACF topic.

We define three RACFVARS profiles, &UF, &US, and &UN, using the following RACF commands:

```
RDEF RACFVARS &UF UACC(N) OWN(SYS1)
RDEF RACFVARS &US UACC(N) OWN(SYS1)
RDEF RACFVARS &UN UACC(N) OWN(SYS1)
```

These will be used as the variable content for the potential new user IDs defined on our system. Specifically, &UF will stand for the first character of any new user ID, &US for the second, and &UN for the valid set of numerals to be used as the three character numeric suffix. We define these with UACC read so that they may be used by any user ID (there is no security implication in having access read to these profiles).

To make these profiles useful, we need to add RACF member resources. The RACFVARS class is a RACF grouping class, and uses its member profiles to stand for any use of the main RACFVARS profile name. This will be made clearer below.

We add members to each of the RACFVARS profiles we have defined as follows:

- ▶ RALT RACFVARS &UF ADDMEM(S, A, 0): Recall that the desired prefixes for new user IDs are either S, A, or O.
- ▶ RALT RACFVARS &US ADDMEM(P): Recall that the only allowable character for the second position in a user ID is P.

The naming convention now starts to make some sense. &UF stands for the 'F'irst valid character in a user ID (S, A, or O). &US stands for the 'S'econd valid character (P).

You might now appreciate the last RACF RALT command we need to issue:

```
RALT RACFVARS &UN ADDMEM(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

This &UN profile can now be used as a substitution in any other RACF resource profile definition for the set of single digits from 0 through 9.

Of course, creating and manipulating these profiles is extremely easy using zSecure Admin. You do not have to issue the RACF commands we show above; you can merely create a new profile in the RACFVARS class, then add the members documented above.

To finally implement the zSecure Command Verifier profiles that will use these RACFVARS definitions, and enforce our new naming convention, you can now define the following normal zSecure Command Verifier profiles:

- ▶ C4R.USER.ID.*: This profile prevents any administrator creating a user ID outside the naming convention.
- ▶ C4R.USER.ID.&UF&US&UN&UN&UN: This profile defines the acceptable naming convention for new user IDs.

We issue a permit command allowing members of the @ZSEC001 and @ZSEC002 groups update level access to the second profile. This will allow them to create user IDs that follow our defined naming convention. You may want to consider granting UACC(UPDATE) to describe an acceptable user ID naming convention that applies to any RACF administrator,

You can see that the profile C4R.USER.ID.&UF&US&UN&UN&UN is actually made up of a set of profiles from the definitions we made in the class RACFVARS. It consists of the &UF, which stands for the characters in the first position in the user ID, and &US for the second position. These are followed by three repetitions of the &UN RACFVARS variable profile that stand for the numerics (0 through 9) that are available for the final three character positions in an acceptable user ID profile definition.

This is how the RACFVARS class is designed to be used. With judicious use of the RACFVARS class, you can save yourself from having to make many similar definitions in another class. Use a RACFVARS profile to take the place of any variable part of the desired profile name in whatever general resource class definition you need.

We use the profile C4R.USER.ID.* to cover any user ID and grant no access. This will prevent administrators from creating a user ID that does not comply with our naming conventions.

Access to the second profile, C4R.USER.ID.&UF&US&UN&UN&UN, is granted to all our system administrators who have the authority to add new user IDs. With only access to this profile, zSecure Command Verifier will prevent them from adding any new user ID that does not comply with our declared naming conventions.

Additional requirements: You must issue a SETROPTS REFRESH RACLIST for both the RACFVARS and the XFACILIT class before you use any new or updated profiles defined to these classes. Additionally, the RACFVARS class must be RACLISTed to function.

Preventing default passwords

By default, when a system administrator creates a new RACF user ID and does not specify a password, the new user ID will receive a password value that is equal to their default group. Additionally, should the administrator also forget to specify a default group, then the new user ID will receive the same default group as the administrator.

If this practice is permitted, and these user IDs do not subsequently change their passwords, you will soon have a situation in RACF where many user IDs have passwords that are easily guessed. Their password will either be the system administrators default group, or their own default group. This is easily guessed by anyone who can access your system and has knowledge of your user ID and group names.

To prevent this scenario from developing in the first place, we can use the following zSecure Command Verifier profile:

```
XFACILIT C4R.USER.PASSWORD.=DFLTGRP
```

Additionally, the following profile also prevents an administrator from specifying a password value that is the same as the actual user ID:

```
XFACILIT C4R.USER.PASSWORD.=USERID
```

We implement these profiles with a UACC of NONE and with no one on their access lists. This prevents any administrator, including system wide specials, from breaching these password standards.

15.2.2 Preventing unwanted SETROPTS changes

Although the general design of zSecure Command Verifier policy profiles is to control the result of the command on a specific profile, some RACF commands do not explicitly manage profiles. The SETROPTS system options command is one of these commands. The RACF SETROPTS command dynamically sets system wide RACF options related to resource protection and auditing.

SETROPTS is typically only used by the system wide auditors or administrators. An undesired SETROPTS option change can have disastrous effects on both system performance and resource protection.

zSecure Command Verifier provides controls to prevent this possibility. We use the following policy profiles in class XFACILIT to prevent unwanted changes in SETROPTS settings:

```
C4R.RACF.*.GENERIC
C4R.RACF.*.RACLIST
C4R.RACF.**
```

The first two profiles control the use of SETROPTS REFRESH for all RACF classes. Only users with update access to these profiles will be able to issue the SETROPTS REFRESH command for these classes.

The third profile controls any other SETROPTS command, and can be further qualified to control specific options and system settings. Refer to the *IBM Security zSecure Command Verifier User Reference Manual Version 1.12*, SC27-2779 for information about the full range of options that may be controlled.

15.2.3 No profiles in WARNING mode

RACF profiles may be set in WARNING mode. This is primarily intended for testing purposes. Resources covered by these profiles may be accessed by any system user at alter level, whether the user is defined to RACF or not.

The zSecure Command Verifier profiles below can be used to control the use of warning mode:

```
C4R.DATASET.ATTR.WARNING.**
C4R.*.ATTR.WARNING.**
```

The first profile is used to prevent warning mode from being set on any data set profiles. Unless the administrator attempting to use warning mode has update access to this profile, the RACF command will be rejected.

The second profile covers all non-data set class resources, that is, general resources. Again, update access to the profile is required before warning mode can be specified on a RACF general resource profile.

In our system, we already have another profile defined, C4R.DATASET.**. Due to the process RACF uses to select the best matching profile, known as generic pattern matching, we must also define the C4R.DATASET.ATTR.WARNING.** profile mentioned above. If we fail to do this, then the catchall profile covering all classes, C4R.*.ATTR.WARNING.**, will never be checked for class data set, as the already existing C4R.DATASET.** profile is a better match. RACF generic pattern matching can seem confusing to new RACF administrators; for a complete discussion of this topic, refer to the *IBM Security Server RACF Security Administrator's Guide*, SA22-7683.

Tip: When implementing and testing controls such as these, it is extremely useful to use the zSecure Admin EV SMF reporting functions to verify which profiles are actually being checked and help validate your profile definitions. This is true for any implementation of new RACF profiles.

15.2.4 No high UACC

Perhaps the most important part of a resource profile is its access specification. In RACF profiles, access to the resources covered by a profile is controlled in two ways: by the Universal Access (UACC) of the profile and by the profile's Access List (ACL).

Additionally, there is a special case for an access list entry of the form ID(*). This is a wildcard that is used to mean any RACF defined user ID. This has a subtly different application than the traditional UACC specification, in that it is possible for users not defined to RACF to access a resource through the profile UACC, but not through the ID(*) ACL entry.

To protect resources from unnecessary access, the setting of high UACC or ID(*) in ACL entries should be avoided except in specific cases. Often these cases must be documented and accepted as part of an audit baseline. For example, the use of UACC READ is acceptable for many data set profiles covering resources that are accessed by any system user simply in the process of logging on.

However, in general, we recommend the use of UACC NONE for general resource profiles. The zSecure Command Verifier policy profiles below can be used to prevent high UACC and high level of access from being allocated to the ID(*) in an access list entry on data set and general resource profiles:

► C4R.*.UACC.NONE.**

This profile controls the specification of UACC(NONE) across any RACF resource class, data set or general. Since UACC(NONE) is desirable, you would normally permit any administrator creating or modifying profiles to have access to this (preferably through UACC UPDATE) to allow them to make the correct specification.

► C4R.*.UACC.READ.**

This profile controls the use of UACC(READ) on any RACF resource class. By denying your administrators access to this profile, you can prevent them from specifying UACC(READ).

► XFACILIT C4R.*.UACC.**

This profile controls the specification of any other UACC, provided the more fully qualified examples first introduced have also been defined.

Once again, due to the fact that we already had profile C4R.DATASET.** defined, we had to qualify the profiles controlling UACC for the data set class more fully than those for all other classes as follows:

```
C4R.DATASET.UACC.NONE.**
C4R.DATASET.UACC.READ.**
C4R.DATASET.UACC.**
```

Apart from applying them only to the data set class, these profiles behave exactly the same as those described above for general resources. Again, the profile C4R.DATASET.UACC.NONE.** should have a UACC of UPDATE to allow any RACF administrator to specify a UACC of NONE for data set profiles.

The following three profiles apply in the exact same manner as those introduced above, with the variation that they are designed to control the use of ACL entries for the ID(*) construct:

```
C4R.DATASET.ACL.=STAR.NONE.**
C4R.DATASET.ACL.=STAR.READ.**
C4R.DATASET.ACL.=STAR.**
```

You will note in these profile names that we cannot use an asterisk to stand for the ID(*). This is because the existence of an asterisk in a RACF profile has special meaning to RACF, that is, it is a generic or wildcard character. Therefore, some alternative is required. zSecure Command Verifier implements fixed strings in certain profiles to circumvent this issue. In these particular profiles, the string is '=STAR'. This string cannot be masked or "wildcarded" in any way and must appear in the profile exactly as shown here for the control to be effective.

Once again, you can use a generic character in the second qualifier to stand for any RACF resource class:

```
C4R.*.ACL.=STAR.READ.**  
C4R.*.ACL.=STAR.NONE.**  
C4R.*.ACL.=STAR.**
```

15.2.5 Preventing or allowing elevation of authority

There are some situations in which we might want a system user to be able to bypass the zSecure Command Verifier implemented controls. This might be of particular importance to provide a resolution for an issue affecting production services, or due to a badly defined profile locking users out of RACF administration.

As mentioned in 7.2, "Controlling RACF commands" on page 133, we can use the zSecure Command Verifier profile below to help avoid a situation where the zSecure Command Verifier itself prevents us from fixing the problem:

```
C4R.EXEMPT
```

We grant an emergency group, perhaps @EMERG, UPDATE level access to this profile. We have some user IDs reserved for emergency use connected to this group. The passwords of any such emergency user IDs should be kept secret and set with the NOEXPIRE attribute. A CARLa report can easily be created to verify that these user IDs are not used, and also ready for use should an emergency occur.

A related scenario is the prevention of administrators granting access to themselves. This is known as a "self-permit" and zSecure Command Verifier controls this situation with three types of profiles:

```
XFACILIT C4R.CONNECT.ID.*.=RACUID.**  
XFACILIT C4R.DATASET.ACL.=RACGPID.**  
XFACILIT C4R.DATASET.ACL.=RACUID.**
```

For more information about the use of these profiles, refer to "Preventing the granting of access to self" on page 135.

15.2.6 Lockdown profiles for segregation of responsibilities

RACF security administrators possessing the system special attribute can do literally anything related to the security controls over a z/OS system. Often this attribute is granted to a wider group of staff than desirable, usually just to allow simple administration tasks. There is an inherent risk that these lower level administrators will accidentally disable an essential data protection mechanism, or perhaps even damage the operating system itself.

Using zSecure Command Verifier, however, we can limit and control what these administrators can do in a granular manner. The policy profiles of zSecure Command Verifier are designed to align with your declared company policies for access to data, naming conventions, and segregation responsibilities. These are flexible enough to accommodate virtually any security protocols that a company might chose to establish.

All the zSecure Command Verifier related profiles we have described throughout this security implementation have been defined with a specific RACF owning group of #CV. We can now use the functions of a resource administrator, as described in “Resource access list administrator” on page 339, to allow an administrator to manage the access lists of the zSecure Command Verifier profiles. This administrator will have no other significant RACF authorities, and will be known as the Command Verifier administrator.

Using the profile’s lockdown facilities of zSecure Command Verifier described in 7.3.1, “Profile locking” on page 137, we can prevent any system special user from modifying the access list of the Command Verifier policy profiles.

In 15.6.1, “Separating administrators by specialized function” on page 401, we describe more fully how this segregation of administrative roles is implemented.

15.2.7 Additional controls required for group special users

We have made some limited use of RACF group special scope for our delegated administrators, specifically:

- ▶ User administrator (VISUAL3)
- ▶ Application group and data set profile administrator (VISUAL2)
- ▶ Branch group connections administrator (VISUAL5)

We now need to establish controls to prevent these administrators from abusing the capabilities provided by RACF group scope. Perhaps the most dangerous of these capabilities is for a group special administrator to pass on their group special authority to other users.

Preventing the passing on of authorities

zSecure Command Verifier provides policy profiles to control these users. To prevent a group special from passing on their attributes, you can use a profile of the following format:

```
C4R.CONNECT.ATTR.SPECIAL.group.userid
```

In our case, we defined C4R.CONNECT.ATTR.SPECIAL.** and denied access to all users.

Another risk associated with the user administrator VISUAL3 is that of passing on their RACF Class Authority, CLAUTH(USER). Again, a simple definition of a policy profile easily eliminates this risk:

```
C4R.USER.CLAUTH.class.owner.userid
```

In our case, we define C4R.USER.CLAUTH.** and deny all access.

Ensuring correct placement of new user IDs

We defined the following profiles to ensure that our administrators could only define new user IDs in the correct default group, within their RACF group special scope.

- ▶ C4R.USER.DFLTGRP./SCOPE.**: This profile ensures that all new user IDs are defined only within the administrators scope.
- ▶ C4R.USER.DFLTGRP./SCOPE.@*.*: This profile ensures that all new user IDs are given a default group that begins with the @ character.
- ▶ C4R.USER.DFLTGRP./SCOPE.@CSV*.CSV*: This profile ensures that all new user IDs beginning with the characters CSV are also given a default group that begins with @CSV.

No RACF access to these profiles is required; their existence enforces control.

For the use of these controls to be logged in SMF, the following profile must also be defined:

- ▶ C4R.USESCOPE.group: We define one of these for each of our main user default group patterns and grant the relevant administrators update access to generate SMF data for these controls.

Ensuring correct RACF ownership is assigned to new user IDs

From the many available styles of controls, we chose to use the owner style profiles to enforce our user ownership standards. Profiles of the following form may be defined:

```
C4R.USER.DFLTGRP./OWNER.group.userid
```

We define a generic profile `C4R.USER.DFLTGRP./OWNER.**`, which ensures all user IDs must be owned by their default group. However, in our installation this is not the convention we have chosen, so we deny all users access to this profile, thus preventing the creation of a user ID that is owned by its default group. We then define exception rules to allow for the specific types of user ID and owner relationships we are using:

```
C4R.USER.DFLTGRP./OWNER.#B001.*  
C4R.USER.DFLTGRP./OWNER.#SP.*
```

We also used the basic owner naming convention profiles of the format `C4R.USER.OWNER.owner.userid` as follows:

```
C4R.USER.OWNER.#B%%.CSV*  
C4R.USER.OWNER.#B%%.MAN*  
C4R.USER.OWNER.#B%%.REG*  
C4R.USER.OWNER.#SP.SP*
```

Administrators are granted access to the profiles that cover only the types of users they are responsible for. Fallback profiles such as `C4R.USER.OWNER.**` are defined and all access is denied. Thus, administrators are prevented from creating user ID or groups outside the standards and naming conventions we have established.

Preventing unusual attributes from being granted

Sometimes a system special grants an attribute that is either not understood, not required or both, to a user ID. Often this happens by accident when using RACF commands. The use of zSecure Admin should minimize this; however, the possibility still exists that the administrator might go outside the product to perform RACF administration.

To control additional RACF user attributes, we define the following profiles:

```
C4R.USER.ATTR.REVOKE.**  
C4R.USER.ATTR.REVOKEDT.**  
C4R.USER.ATTR.RESUME.**  
C4R.USER.ATTR.RESUMEDT.**
```

We allow our user administrators access to these profiles so that they may revoke and resume users under their control. We then define the fallback profile of `C4R.USER.ATTR.**`.

We deny all administrators access to this profile. This will cover such attributes as ADSP, GRPACC, and other undesirable settings, preventing their propagation through our RACF database.

15.2.8 Assigning mandatory values

You can set up policy profiles to enforce mandatory values when creating or modifying RACF profiles. This can help significantly reduce your clean-up effort.

A common clean-up activity is to ensure that the CONNECT-OWNER value for a user's group connect matches that of the group name. Normally, when you issue a CONNECT command and you do not specify OWNER(value), the owner will default to the user ID of the RACF administrator who issued the CONNECT command. The following policy profile will ensure that the CONNECT-OWNER will always default to the actual group name. This profile should have a UACC of READ so that the policy applies to all RACF administrators:

```
C4R.CONNECT.OWNER.=GROUP.**
```

Another example of assigning a mandatory value is for the creation of new RACF groups. Many installations have a standard where the owner of a RACF group must match the group's superior group. Implementing the following policy profile will enforce this standard. The policy profile will require a UACC of READ and the value =SUPGRP must be stored within the APPLDATA field of the profile. This ensures that the owner for groups will always default to that of the group's superior group. For example:

```
C4R.GROUP./OWNER.*
```

15.3 Processes for managing authorization

In this section, we describe some other commonly implemented procedural controls that may be needed to properly control and administer a delegated security framework.

15.3.1 Timed (queued) commands

Delft Transport Authority has always insisted that when security requests are submitted to the security administration team, the requestor must specify whether the access is required on a permanent or temporary basis. When temporary access is required, the requestor must specify a start and end date for the access. If permanent access is specified, the requestor is not only required to justify why access is required, they are also asked if access is required due to one of the following situations:

- ▶ Projects (a project always has an end date)
- ▶ Secondments, temporary assignments, or providing cover for another employee

- ▶ Fixing production problems
- ▶ Scheduled system or application upgrades and changes
- ▶ Planned disaster recovery testing

If one of these situations apply, the requester is prompted to apply for temporary access.

TRA is now required to adopt this process as part of the security improvement program.

zSecure Admin has a feature for assigning temporary access through queued (timed) commands. Queued commands are CKGRACF commands that are either subject to multiple authority, timed, or temporary. In this section, we will discuss both timed and temporary queued commands. We discuss multiple authority in 15.3.3, “Workflow for RACF commands” on page 376.

Examples of queued commands include:

- ▶ Set a date when a user ID should be resumed or a date when it should be revoked. This is a useful feature for managing departing employees, temporary employees, employees going on a long vacation, or employees going on maternity leave or long term sick leave.
- ▶ Timed connect and remove to/from a RACF group. This can be an efficient method of granting temporary access to mainframe resources.
- ▶ Timed permits to data set and general resource profiles. This is a feature not available using only native RACF.
- ▶ Creation of temporary data set and general resources profiles.

The use of the queued commands feature can help to significantly reduce audit concerns regarding excessive access. Security Administrators often permit access to resources on a permanent basis and do not challenge whether access is required for a temporary period. If the administrator does permit temporary access, they may forget to remove it on the date when access should terminate.

Queued commands are managed by CKGRACF and can be used to implement both permanent and temporary access. For permanent access, a security administrator can request for access to become effective on a date in the future. For temporary access, the administrator can specify a start and end date, or number of days the access is required for. This will execute the appropriate RACF command on the required start date to bring access into effect. On the specified end date, a RACF command will be issued to remove access.

You can use queued commands using zSecure Admin options RA.U, RA.G, RA.D, and RA.R or through the line command MS in option RA.U. zSecure Visual also utilizes queued commands.

Before you can use CKGRACF to queue commands, the CKGRACF program must be APF authorized and defined to the IKJTSONn section of your PARMLIB library as an authorized TSO command, as discussed in 13.1, “Post systems programmer installation setup” on page 202. You will also need to regularly schedule the CKGRACF refresh job, which can be found in your SCKRSAMP library member C2RJXRFR. This job issues the REFRESH command to update a user profiles revoke/resume settings and to execute or expire queued commands. You should schedule this to run daily through IBM Tivoli TWS for z/OS or your equivalent your job scheduler. The user ID under which this job executes will require update access to the XFACILIT profile CKG.CMD.REFRESH to enable the REFRESH to take place.

Entering a queued or timed command

In this particular scenario, we are going to demonstrate how you can permit access to a resource for a temporary period. We have a security administrator with RACF system special who needs to implement some changes to RACF system settings using SETROPTS RACF commands. However, change management only allows these changes to take place out of hours and only within the data center change window on weekends. To prevent these changes taking place at any other time, we use a zSecure Command Verifier policy profile C4R.RACF.** in class XFACILIT. This profile prevents RACF system special users from issuing SETROPTS commands, with the exception of class REFRESH commands for which more specific profiles have been defined. Refer to 15.2.2, “Preventing unwanted SETROPTS changes” on page 354 for information about these SETROPTS controls.

```

Menu      Options      Info      Commands      Setup
-----
                                zSecure Admin+Audit for RACF - 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 . . . . XFACILIT (class or filter)
Resource profile . C4R.RACF.**
-----
Owned by . . . . . _____ (group or userid, or filter)
Installation data . _____ (substring or *)
Additional selection criteria
_ Profile fields      _ Access list      _ Segment presence  _ Absence
1 1 EGN mask
2 Exact
3 Match
4 Any match

Output/run options
_ Show segments      _ All      _ Enable full ACL  _ Specify scope
_ 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

```

In Figure 15-28, we select the profile that we need to grant the security administrator access to.

Figure 15-28 Select the profile for which to issue the permit

Figure 15-29 shows the use of the I line command to insert a new entry to the access control list.

```

zSecure Admin+Audit for RACF General resource overview          Line 1 of 33
Command ==> _____ Scroll==> CSR
Class XFACILIT, like C4R.RACF.**                               20 Apr 2008 16:51

  Identification                                             SC76
  Class              XFACILIT
  Profile name       C4R.RACF.**
  Type              GENERIC
  Volume serial list
  Owner             SYS1
  Installation data
  Application data

  User   Access  ACL id  When          Name          InstData
  I _group- UPDATE @ZSEC001 _____ _____ ZSECURE SU

  Safeguards                                Other permissions
  User to notify of violation _____ Allow all accesses  WARNING No
  Audit access success/failures R R      Universal access authority NONE
  Global audit success/failures _____ Resource level    0

```

Figure 15-29 Insert a new entry to the access control list

In Figure 15-30, we enter the required information to generate a new permit.

```

Menu  Options  Info  Commands  Setup
-----
                                zSecure Admin+Audit for RACF - RACF - New permit
Command ==> _____

Profile to be changed
Class . . . . . XFACILIT
Profile name . . . . C4R.RACF.**

Permit to be added
User or group . . . @ZSEC002
Access level . . . . UPDATE

Optional conditions for the permit
When class . . . . . _____
When resource/profile
_____
_____

```

Figure 15-30 Add a new permit

In Figure 15-31, we have entered the following information:

- ▶ We selected command execution option 4 to request the RACF command to be executed later.
- ▶ We entered a start date and until date (end date) for access.
- ▶ The reason field is optional and can be used to log additional information with the RACF command. We have chosen to enter the change management record number that relates to the overall SETROPTS change. The reason will be logged in SMF with the RACF command event.

When all the required data has been entered, press Enter to continue.

zSecure Admin+Audit for RACF - Confirm command

Command ==> _____

Confirm or edit the following command

permit C4R.RACF.** id(@ZSEC002) access(UPDATE) class(XFACILIT)

Command execution . 4

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 26APR2008 (ddmmmyyyy, yyyy-mm-dd or TODAY)

Until/for 27APR2008 (ddmmmyyyy, yyyy-mm-dd or number of days)

Reason C7107832

Press ENTER to continue or END to cancel the command

Figure 15-31 Enter the required information for the temporary permit

Figure 15-32 shows confirmation of the command about to be issued. Notice that the original RACF command has been prefixed by CKGRACF and extra parameters have been added that are specific to the CKGRACF command syntax. These parameters are what the CKGRACF daily refresh job uses to perform its extra processing. They contain all the necessary information that ensures the native RACF commands will be issued on the required dates.

```

zSecure Admin+Audit for RACF - Press ENTER to process

Command ==> _____

Confirm or edit the following command
ckgracf cmd at 26APR2008 until 27APR2008 request reason(`C7107832`) permit C4R
.RACF.** id(@ZSEC002) access(UPDATE) class(XFACILIT)
_____

Command execution . 4 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 . . . . . 26APR2008 (ddmmmyyyy, yyyy-mm-dd or TODAY)
Until/for . . . . . 27APR2008 (ddmmmyyyy, yyyy-mm-dd or number of days)
Reason . . . . . C7107832

Press ENTER to continue or END to cancel the command

```

Figure 15-32 Confirmation of the CKGRACF command to be issued

Having pressed Enter to execute the CKGRACF command, you will be returned to the profile overview panel where we originally inserted the new access control list entry, as shown in Figure 15-33. Note that the ACL does not contain the new group. The permit will not become effective until the CKGRACF refresh job has run on the date specified. The CKGRACF job will execute the RACF command on our behalf.

zSecure Admin+Audit for RACF General resource overview

Change successful

Command ==> _____ Scroll==> CSR

Class XFACILIT, like C4R.RACF.**20 Apr 2008 16:51

Identification

SC76

Class

XFACILIT

Profile name

C4R.RACF.**

Type

GENERIC

Volume serial list

Owner

SYS1

Installation data

Application data

User

Access

ACL id

When

Name

InstData

= -group-

UPDATE

@ZSEC001

ZSECURE SU

Safeguards

Other permissions

User to notify of violation

Allow all accesses

WARNING

No

Audit access success/failures

R R

Universal access authority

NONE

Global audit success/failures

Resource level

0

Figure 15-33 The current access control list

Viewing queued or timed commands

There are two places where we can view queued commands that are pending execution.

The first method is to display the profile through option RA.U, RA.G, RA.D, or RA.R, depending on the RACF class. In this case, we need to use option RA.R to display our XFACILIT profile C4R.RACF.**, as shown in Figure 15-34. In the profile overview panel, scroll down to see the heading “Timed commands waiting for execution”.

```
zSecure Admin+Audit for RACF General resource overview          Line 21 of 34
Command ==> _____ Scroll==> CSR
Class XFACILIT, like C4R.RACF.**                                20 Apr 2008 19:10

Mandatory Access Control          Statistics
Security label                    _____ Creation date          20Apr08
Security level
Categories list

Timed commands waiting for execution
_ Queued command (P): CMD AT 26Apr2008 UNTIL 27Apr2008 REASON('C7107832') PERMI

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

Figure 15-34 Display timed commands waiting for execution in option RA.R

The second method is to use option RA.2 to search the RACF database for queued commands. In the RA.2 panel, shown in Figure 15-35, we have entered the profile name we are specifically interested in. If required, you can leave all fields blank and select the default option to search for all profiles for which queued commands are pending.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - RACF - Queued				
Command ==> _____				
Show only profiles that fit all of the following criteria:				
Class name <u>XFACILIT</u> (class or filter)				
Profile pattern . . <u>C4R.RACF.**</u> (EGN mask)				
Complex _____ (complex name or filter)				
Show only profiles with				
2 1. Queued commands requiring action				
2. Commands in the execution queue				
3. Withdrawn, denied or expired commands				
4. Any queued commands				
Enter "/" to select option(s)				
_ Show all queued commands within selected profiles				

Figure 15-35 Search for queued commands

Figure 15-36 shows the queued command awaiting execution. Use the S line command to show additional information.

IBM Tivoli zSecure RACF display				Line 1 of 1	
Command ==> _____				Scroll==> <u>CSR</u>	
20 Apr 2008 19:25					
Class	Profile key	Expires	LastReq	LastChg	
<u>S</u> XFACILIT	C4R.RACF.**	20Apr08	20Apr08		
***** Bottom of Data *****					

Figure 15-36 Queued command displayed

Figure 15-37 shows additional information for the queued command. You can use the / line command to list the available actions for this queued command. One of these actions includes D to delete the queued command. This will stop the queued command from being executed on the original date specified.

```

IBM Tivoli zSecure RACF display                               Line 1 of 3
Command ==> =                                                Scroll==> CSR
                                                                20 Apr 2008 19:25
      Class      Profile key                                Expires LastReq LastChg
      XFACILIT C4R.RACF.**                                  20Apr08 20Apr08
      CKGRACF authority requirement

      Timed commands waiting for execution
_ Queued command (P): CMD AT 26Apr2008 UNTIL 27Apr2008 REASON('C7107832') PERMI
***** Bottom of Data *****

```

Figure 15-37 Display additional information for the queued command

In this case, our queued command was generated by a user with the necessary RACF authorities to issue the command. This command will execute as though the original requestor had entered it manually on the day that it is scheduled for.

If, however, the original requestor does not have the RACF authorities needed to issue a command they have requested, other options on the queued commands ISPF menu allow an administrator to run the command on their behalf. This topic will be covered in 15.3.3, “Workflow for RACF commands” on page 376.

15.3.2 Temporary (queued) commands

In this section, we discuss temporary commands. A temporary command is a command that is executed and reversed at a later date. For example, you can create a temporary profile to protect a resource and have that profile automatically deleted at a later date. This is extremely useful for when users require access to a resource for a number of days and a more specific profile is needed to prevent them from accessing other resources.

In the following scenario, an application developer in TRA needs to be able to read some data from a data set as part of a project to consolidate HR data between Delft Transportation Authority and TRA.

The data set is called HR.PR.PAYROLL.TAXCODE.APR08; the application developer does not normally have access to any HR.PR.PAYROLL prefixed data sets. The profile currently protecting this data set is HR.PR.PAYROLL.** and due to the sensitivity of other data falling under this profile, we do not want to permit the developer to have read access to the existing profile.

To address this short term access requirement, we create a temporary fully qualified generic data set profile and permit the application developer read access to it.

In Figure 15-38, the best matching profile for data set HR.PR.PAYROLL.TAXCODE.APR08 is displayed in option RA.D. We use the C line command to copy the profile.

```
zSecure Admin+Audit for RACF DATASET Overview          0 s elapsed, 0.0 s CPU
Command ==> _____ Scroll==> CSR
like HR.PR.PAYROLL.**                                21 Apr 2008 17:13
  Profile key                                         Type      UACC      Owner      S/F W
C HR.PR.PAYROLL.**                                GENERIC  NONE      HR          R  R  _
***** Bottom of Data *****
```

Figure 15-38 Copying an existing data set profile

In Figure 15-39, we have:

- ▶ Entered the fully qualified generic profile we need to define to RACF
- ▶ Selected option 2 to create the profile as a temporary profile
- ▶ Specified that the profile is to be deleted two days from now using the date of removal field
- ▶ Entered the change management record number for this change into the reason field

```
Menu  Options  Info  Commands  Setup
-----
zSecure Admin+Audit for RACF - RACF - Data set Copy
Command ==> _____

From profile . . . . . HR.PR.PAYROLL.**

To profile . . . . . HR.PR.PAYROLL.TAXCODE.APR08

2 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)

  2. Create temporary profile
    Date of removal . . . . . +2 (ddmmmyyyy or yyyy-mm-dd or +days)
    Reason . . . . . C7859488
```

Figure 15-39 Specify the date when the profile should be deleted

We press Enter to proceed to the next panel.

Figure 15-40 shows the RACF command generated to create the temporary fully qualified generic profile, based on the existing HR.PR.PAYROLL.** profile. A CKGRACF command is also generated to flag the temporary profile for deletion in two days.

```

                                zSecure Admin+Audit for RACF - Confirm commands
Command ==> _____

Confirm or edit the following commands
addsd 'HR.PR.PAYROLL.TAXCODE.APR08' generic from('HR.PR.PAYROLL.**') fgeneric
_____

ckgracf cmd req after 2 reason('C7859488') deldsd 'HR.PR.PAYROLL.TAXCODE.APR08'
generic
_____

```

Figure 15-40 Commands to create the data set profile and flag it for deletion

We use the I line command to add our application developer's RACF group to the access list of this data set profile, as shown in Figure 15-41.

Press Enter to create the new profile.

```

IBM Tivoli zSecure RACF display                                0 s elapsed, 0.0 s CPU
Command ==> _____ Scroll==> CSR
                                21 Apr 2008 17:37

Identification                                                    SC76
Profile name              HR.PR.PAYROLL.TAXCODE.APR08
Type                      GENERIC
Volume serial list
_ Effective first qualifier  HR
_ Owner                    HR
Installation data
_____

User   Access  ACL id  When      Name      InstData
I _group-  READ    @HR001  _____  _____  HR SUPPORT

Safeguards                Other permissions
Erase on scratch          No      Allow all accesses  WARNING No
Audit access success/failures R R      Universal access authority  NONE
Global audit success/failures _____ Resource level      0
User to notify of violation _____
Days protection provided # _____

```

Figure 15-41 Insert a new entry to the access control list

We enter the application developer's RACF group @HR002 with read access for our new profile, as shown in Figure 15-42.

Press Enter to proceed to the next panel.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF - RACF - New permit				
Command ==> _____				
Profile to be changed				
Class DATASET				
Profile name 'HR.PR.PAYROLL.TAXCODE.APR08'				
Profile modifier . . generic				
Permit to be added				
User or group . . . @HR002				
Access level READ				
Optional conditions for the permit				
When class _____				
When resource/profile _____				

Figure 15-42 Specify the new access control list entry and access level

The PERMIT command is generated, as shown in Figure 15-43. We select option 1 to execute the RACF command, followed by pressing Enter.

zSecure Admin+Audit for RACF - Confirm command

Command ==> _____

Confirm or edit the following command

permit 'HR.PR.PAYROLL.TAXCODE.APR08' generic id(@HR002) access(READ)

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 _____ (ddmmyyyy, yyyy-mm-dd or TODAY)

Until/for _____ (ddmmyyyy, yyyy-mm-dd or number of days)

Reason _____

Press ENTER to continue or END to cancel the command

Figure 15-43 Confirm the permit command

The access is now permitted, as shown in Figure 15-44.

zSecure Admin+Audit for RACF DATASET Overview
Line 1 of 34

Command ==>
Scroll==> CSR

like HR.PR.PAYROLL.**
21 Apr 2008 17:56

Identification
SC76

Profile name
HR.PR.PAYROLL.TAXCODE.APR08

Type
GENERIC

Volume serial list

Effective first qualifier
HR

Owner
HR

Installation data

User	Access	ACL id	When	Name	InstData
-group-	READ	@HR001			HR SUPPORT
-group-	READ	@HR002			HR DEVELOP

Safeguards

Other permissions

Erase on scratch
No

Allow all accesses
WARNING No

Audit access success/failures
R R

Universal access authority
NONE

Global audit success/failures

Resource level
0

User to notify of violation

Days protection provided #

Figure 15-44 Display of the temporary data set profile with access control list

The profile will be automatically deleted in two days by the CKGRACF refresh job C2RJXRFR. The application developer will lose access to the data set at this time. You can also create temporary profiles in general resource classes.

Creating temporary profiles is a good method of enforcing and automating temporary access. Using this technique can also help reduce the number of data set and general resource profiles defined in your RACF database, which is recommended for best RACF performance.

For further information about queued commands, refer to Chapter 2, “RA.2 QUEUED”, in the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773.

15.3.3 Workflow for RACF commands

zSecure Admin and zSecure Visual have built-in workflow capability to allow a user to ASK or REQUEST for a CKGRACF command to be executed. This only applies to CONNECT, REMOVE, PERMIT, DELDSD, and RDELETE commands.

The ASK function enables a user (typically with no authorizations) to ask a security administrator to execute a CKGRACF command, with timed features if required. ASK requests will always queue the command for approval, while REQUEST can be used to queue the command for execution at a later date, as described in 15.3.1, “Timed (queued) commands” on page 361.

The REQUEST function can also be used to generate workflow for a CKGRACF command, similar to the ASK function. A key feature of the REQUEST function is that you have a decentralized administration interface to allow decentralized administrators to issue the CONNECT, REMOVE, PERMIT, DELDSD, and RDELETE commands without them needing RACF system SPECIAL or group SPECIAL. The access required to issue these commands is controlled by CKG profiles in the XFACILIT class, which we discuss in 15.1, “Delegated RACF administration” on page 322.

An ASK or REQUEST action can be subject to:

- ▶ Single authority, where one person is required to review the request.
- ▶ Dual authority, where two people are required to review the request.
- ▶ Triple authority, where three people are required to review the request.

This feature is referred to as *multiple authority* and the requirements are set by CKGRACF administrators.

The implementation of multiple authority enables you to enforce segregation of duties by implementing a requestor and checker environment. For example, a local security administrator with less RACF authority can request for a RACF command to be issued, such as a connect command for a sensitive RACF group that is out of the local administrator’s normal scope. The command is then subject to review and authorization by one, two, or three security administrators.

The default multiple authority level is set at system level. This level is set in your site module CKRSITE and the default multiple authority requirement is SINGLE, unless you specify differently. To display the CKRSITE module, use the CKGRACF SHOW CKRSITE command. A CKGRACF administrator can also specify the multiple authority requirements at the profile level for user, group, data set, and general resource profiles, where an additional level of security is required. To do this, you need to use the line command MR when displaying a profile in option RA.U, RA.G, RA.D, or RA.R.

An approval queue is available in zSecure Admin option RA.2 to manage the workflow for requests that have been entered. This queue contains requests that are awaiting authorization. There is also an execution queue for timed commands.

You can submit requests for approval using either the ASK or REQUEST options in the Confirm command panel, as shown in Figure 15-45. This panel is normally invoked when you generate RACF commands from the RA options.

In Figure 15-45, we have chosen to use option 3 to ASK an administrator to execute a CKGRACF command. This generates a CKGRACF command with a status of ASK.

```
zSecure Admin+Audit for RACF - Confirm command
Command ==> _____

Confirm or edit the following command
connect CSV001_group(@CSV001)
_____
_____

Command execution . 3  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 . . . . . New joiner in the call centre - requires update access

Press ENTER to continue or END to cancel the command
```

Figure 15-45 ASK an administrator to execute a CKGRACF command

In Figure 15-46, we confirm the ASK request by pressing Enter. The CKGRACF command is then executed and the request is now in the approval queue awaiting authorization.

```

zSecure Admin+Audit for RACF - Press ENTER to process

Command ==> _____

Confirm or edit the following command
ckgracf cmd ask reason(^New joiner in the call centre - requires update access) connect CSV001 group(@CSV001)
_____

Command execution . 3 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 . . . . . New joiner in the call centre - requires update access

Press ENTER to continue or END to cancel the command

```

Figure 15-46 Confirm the command execution

An authorized CKGRACF security administrator will then select option RA.2 to search for requests that require action, as shown in Figure 15-47.

```

zSecure Admin+Audit for RACF - RACF - Queued

Command ==> _____

Show only profiles that fit all of the following criteria:
Class name . . . . . _____ (class or filter)
Profile pattern . . _____ (EGN mask)
Complex . . . . . _____ (complex name or filter)

Show only profiles with
1 1. Queued commands requiring action
2 2. Commands in the execution queue
3 3. Withdrawn, denied or expired commands
4 4. Any queued commands

Enter "/" to select option(s)
_ Show all queued commands within selected profiles

```

Figure 15-47 Search for queued commands

The security administrator selects the appropriate request, as shown in Figure 15-48.

```

IBM Tivoli zSecure RACF display                                0 s elapsed, 0.1 s CPU
Command ==> _____ Scroll==> CSR
                                     22 Apr 2008 12:43
Class      Profile key      Expires LastReq LastChg
s GROUP    @CSV001 SENIOR CUSTOMER SERVICE REP ACCESS 28Apr08 21Apr08 21Apr08
— GROUP    @CSV002 JUNIOR CUSTOMER SERVICES REP ACCE 28Apr08 21Apr08 21Apr08
— GROUP    @CSV003 BAD CUSTOMER SERVICES REP ACCESS 28Apr08 21Apr08 21Apr08
— OPERCMDS MVS.MCSOPER.**      10Apr08 3Apr08 3Apr08
— USER    CSV001 CUST SERV REP 01      28Apr08 21Apr08 21Apr08
— XFACILIT CKG.SCP.G.#FUNCACC.#USERACC.** 28Apr08 21Apr08 21Apr08
***** Bottom of Data *****

```

Figure 15-48 Display of queued commands in the approval queue

The administrator selects the request to display further information. Notice that the request has an expiry date. If no action is taken, the request will automatically expire on the specified date (seven days from the original request date) and will be removed from the approval queue. An audit trail is generated for this action. The expiry processing is performed by the daily CKGRACF refresh job. To keep a request in the approval queue for a longer period, the line command H (HOLD) should be used. It can also be useful to use the HOLD command to prevent other authorized CKGRACF security administrators from approving a CKGRACF REQUEST command while you are investigating whether the command is appropriate and legitimate.

```

IBM Tivoli zSecure RACF display                                Line 1 of 3
Command ==> _____ Scroll==> CSR
                                     22 Apr 2008 12:43
Class      Profile key      Expires LastReq LastChg
GROUP      @CSV001 SENIOR CUSTOMER SERVICE REP ACCESS 28Apr08 21Apr08 21Apr08
CKGRACF authority requirement

Commands requiring administrator action
s Queued command (A): CMD AT 21Apr2008 REASON('NEW JOINER IN THE CALL CENTRE -
***** Bottom of Data *****

```

Figure 15-49 Selecting a request to display further information

Using the line command S enables the security administrator to display additional information about the request, as shown in Figure 15-50.

```

zSecure Admin+Audit for RACF - Queued command
Command ==> _____
ask by JPEASE at 21 Apr 2008 21:09

Status . . . . . : ask

Command:
AT 21Apr2008 REASON('NEW JOINER IN THE CALL CENTRE - REQUIRES UPDATE ACCESS')
CONNECT CSV001 GRO(@CSV001)

```

Figure 15-50 Details of queued command

The administrator selects the appropriate action. In this case, they select A to approve the request, as shown in Figure 15-51.

```

IBM Tivoli zSecure RACF display
C      zSecure Admin+Audit for RACF - Actions
      Select one of 7 actions
      A Approve
      C Copy
      D Delete
      H Hold
      I Insert
      R Repeat
      S Select

/
*

Scroll==> CSR
r 2008 12:43
Expires LastReq LastChg
SS 28Apr08 21Apr08 21Apr08

INNER IN THE CALL CENTRE -
*****

```

Figure 15-51 Selecting the appropriate action for the queued command

A CKGRACF command is generated to authorize and complete the request, as shown in Figure 15-52. Having pressed Enter, the user ID CSV001 will be connected to RACF group @CSV001.

```

zSecure Admin+Audit for RACF - Confirm CKGRACF command

Command ==> _____

Confirm or edit the following CKGRACF command
ckgracf cmd complete approve AT 21Apr2008 REASON('NEW JOINER IN THE CALL CENTR
E - REQUIRES UPDATE ACCESS') CONNECT CSV001 GRO(@CSV001)
_____
_____

Press ENTER to continue or END to cancel the CKGRACF command

```

Figure 15-52 Approving the CKGRACF command for execution

If dual or triple authority is required for CKGRACF commands, the second or third authorizer (security administrator) will follow the same process to review and authorize the command, using option RA.2.

RACF security can be set up to control queued commands by implementing CKG profiles in the XFACILIT class. CKGRACF scoping controls may also apply to administrator's processing of multiple authority commands. We recommend you refer to 15.1, "Delegated RACF administration" on page 322 for further information about scoping.

Additional capability: Until now, we have discussed the ASK and REQUEST functions of CKGRACF. There is an additional function, the EXECUTE CKGRACF function (option 2 from the “Confirm Command” panel), which is basically executing a normal standard RACF command with the option to add a “Reason” to the command. The “Reason” is written to SMF along with the RACF command event. The person issuing the command still requires regular standard RACF administrative authorities (such as system special, group special, class authorization, and field access). In addition, when a CKGRACF EXECUTE command is issued, the CKGRACF scope profiles (CKG.SCP.***) are checked to verify whether the target profile lies within the scope of the involved RACF administrator.

A second step is to check CKGRACF command profiles (CKG.CMD.CMD.EX.***) to determine whether the RACF administrator is allowed to issue this RACF command. Thus, a system-wide RACF special user could be disallowed to execute a CKGRACF EXECUTE ADDUSER command, when they lack authorization to profile CKG.CMD.CMD.EX.ADDUSER in class XFACILIT. A regular RACF ADDUSER command would still work as normal.

15.3.4 Access re-validation reporting

As part of the security improvement program, TRA is required to introduce *periodic access re-validation reporting*. This is a process where owners or custodians of systems, applications, and data are required to periodically review users’ access to resources. They assess whether access is still appropriate and the overall objective is to ensure that access is appropriate for the user’s job function or role. This process is sometimes referred to as re-certification rather than the term we use, that is, re-validation.

Owners and custodians within TRA must provide evidence they have conducted a review. This can be achieved by returning the updated re-validation report to security administration through email. This process is subject to an annual audit.

Re-validation reporting should occur on a regular basis, depending on the type of resource(s), and re-validation should be more frequent for sensitive resources. For example, you may want to issue a report about users with RACF system wide attributes (SPECIAL, OPERATIONS, and AUDITOR) on a monthly basis. This can be sent to the security manager, storage manager, and IT audit manager to re-validate users who have these attributes, while re-validation reporting for group attributes (SPECIAL, OPERATIONS, AUDITOR) may occur every three months. Re-validation for access to normal business data might occur once or twice a year.

In 14.1.3, “XML format audit reporting using CARLa” on page 275, we show an example of a report containing a list of users who are connected to a RACF group. This report is sent to the owner of the group who is required to re-validate the users’ connection requirements. The owner can export the XML report and edit this to re-validate or provide other comments back to a security administrator. After updating the report, the group owner returns the report to the security administrator with a list of changes, and signs off that access is still appropriate where no changes are necessary.

In the absence of access re-validation, a user’s access to systems, applications, and data may become excessive over a period of time. Many organizations have been subject to fraud because an employee has gained sufficient access to carry out a fraudulent transaction.

zSecure Admin can help meet your access re-validation reporting requirements by producing user friendly security reports that can be automatically distributed to owners and custodians using Tivoli Workload Scheduler (TWS) or your equivalent job scheduler. In Figure 15-53, we show a re-validation report that is sent to the IT security manager on a quarterly basis, so they can re-validate RACF class authorizations.

List of users with RACF class authorisations - please re-validate

Notes: Edit this table with Excel; add your comments in the field provided; save the table; e-mail it to SECADM

User ID	Name	Class Auth	LastConDate	Revoked	Unused	Reviewers	Remarks
AUTNV6KM	AUTNV6KM	APPL			Unused		
AUTNV6K1	AUTNV6K1	APPL			Unused		
AUTNV6K2	AUTNV6K1	APPL			Unused		
AUTTCP	AUTTCP	APPL			Unused		
AUTTCPTS	AUTTCPTS	APPL			Unused		
AUTT390M	AUTT390M	APPL			Unused		
PLS	PETER SOPER	CONSOLE	20 Sep 2006		Unused		
		OPERCMDS					
		TSOAUTH					

Figure 15-53 Re-validation report for RACF class authorizations

The CARLa program that produced this report is shown in Example 15-2.

Example 15-2 CARLa to report in XML on users with RACF Class Authorizations

```
fileoption dd=c2remai fileformat=xml xml_datadict,  
  xml_stylesheet=imbed(m=c2rxsl01) encoding=utf-8  
option dd=c2remai mailto=jamie_pease@uk.ibm.com,  
  from=Access_revalidation@uk.ibm.com outputformat=attach  
  
newlist dd=C2REMAIL type=RACF name=CLAUTH,  
tt='List of users with RACF class authorisations - please re-validate',  
subtitle='Notes: Edit this table with Excel; add your comments in the  
  field provided; save the table; email it to SECADM'  
  
select class=USER segment=BASE cname=*  
define comment('') boolean  
define unused('Unused',8,hb) boolean where last_connect_date<today-300  
sortlist key('User ID',8) name(22) cname('Class Auth',10),  
  last_connect_date revoked('Revoked',7,hb) unused,  
  comment('Reviewers Remarks',20)
```

15.4 Reporting processes

An essential component in delegated administrative frameworks is reporting. Two main types of reports are generally required:

- ▶ Reports for the centralized administrators, used to ensure that their delegated administrators have not managed to step outside the authorities we want them to have.
- ▶ Reports for the delegated administrators, listing information about the resources under their control and who has accessed them or has access to them.

We demonstrate a technique for establishing reporting to delegated administrators or data owners in the following section.

15.4.1 Advanced use of CARLa for email bundle reporting

We can use the CARLa emailing capability to automate reporting of data access or other security related events. In this example, we will show how to take nominated data owners, and automatically notify them through email of who has been accessing the data for which they are responsible.

This report could just as easily be a list of users with access, and a re-validation requirement, as described in 15.3.4, “Access re-validation reporting” on page 383.

Referencing external data

We define a data set and populate it with a list of data set high-level qualifiers, the user ID and email address of the person responsible for that high-level qualifier, and a department name. You could chose any data relevant to your organization in this external file; an example layout is shown in Example 15-3.

Example 15-3 External file listing data owners by high-level qualifier

----	1-----	2-----	3-----	4-----	5-----	6-----	7-----
SYS1	LILIXIE	LILI.XIE@CN.IBM.COM				SYSPROGS	
SYS2	JPEASE	JAMIE.PEASE@UK.IBM.COM				PRODUCTS	
DATA	MCAIRNS	MIKE.CAIRNS@AU.IBM.COM				APPS	

The CARLa program in Example 15-4 will produce two simple reports, listing the responsible owners, then the data they own and who has accessed it. See the first report titled MAIL in the output shown in Example 15-5 on page 387 for the CARLa report on the contents of the external file.

Example 15-4 CARLa to report on data owners and access to their data

```
deftype type=mail
alloc   type=mail dsn='mcairns.test.cntl(email)'
define type=mail hlq(8,'HLQ')           as substr(record,1,8)
define type=mail userid(8,'Owner')       as substr(record,10,8)
define type=mail email(43,'Email Address') as substr(record,18:50)
define type=mail dept(10,'Department')   as substr(record,60,10)
newlist type=mail
  sortlist recno hlq userid email dept
suppress ckfreeze
newlist type=smf
  define qual as word(dsname,1,'.')
  select dsname=(sys1.**,**,testdata.**) intent>=update,
    not(qual==user)
  x type=80
  x dsname=sys1.brodcast
  sortlist date time dsname(22) user intent,
    qual:mail.hlq.userid(8) qual:mail.hlq.email(43)
/*
```

Example 15-5 Output reports merging data owners with data access records

M A I L				25 Apr 2008 16:43		
Recno	HLQ	Owner	Email Address	Department		
1	SYS1	LILIXIE	LILI.XIE@CN.IBM.COM	SYSPROGS		
2	SYS2	JPEASE	JAMIE.PEASE@UK.IBM.COM	PRODUCTS		
3	TESTDATA	MCAIRNS	MIKE.CAIRNS@AU.IBM.COM	APPS		
S M F	R E C O R D		L I S T I N G	25Apr08 14:10 to 25Apr08 16:31		
Date	Time	Dataset	User	Intent	USERID	EMAIL
25 Apr 2008	14:10	SYS1.SC76.MAN1.DATA	SMF	CONTROL	LILIXIE	LILI.XIE@CN.I
25 Apr 2008	14:10	SYS1.SC76.MAN1	IBMUSER	CONTROL	LILIXIE	LILI.XIE@CN.I
25 Apr 2008	16:28	SYS2.TEST	MCAIRNS	ALTER	JPEASE	JAMIE.PEASE@U
25 Apr 2008	16:28	SYS2.TEST	MCAIRNS	UPDATE	JPEASE	JAMIE.PEASE@U
25 Apr 2008	16:30	TESTDATA.TEST	MCAIRNS	ALTER	MCAIRNS	MIKE.CAIRNS@A
25 Apr 2008	16:30	TESTDATA.TEST	MCAIRNS	UPDATE	MCAIRNS	MIKE.CAIRNS@A

Using CARLa BUNDLE

Now we would like to be able to email our data owners to notify them of access to their data using a scheduled daily report.

The CARLa program in Example 15-6 shows where we have added additional CARLa BUNDLE statements. These will control the emailing of reports based on the value of the BUNDLEBY variable, which we have set to use the **mail:hlq:email** CARLa lookup construct, that is, the data owner's email address from the external file.

We now have an automated emailing system based on the HLQ and data owners, which we submit as part of our nightly batch reporting suite. You can see a part of the SMTP dialog in Example 15-7 on page 388 that shows the email being sent.

Example 15-6 CARLa to email reports out based on data owner

```
alloc type=racf backup active complex=sc76
  alloc smf active complex=sc76
deftype type=mail
  alloc type=mail dsn='mcairns.test.cntl(email)'
define type=mail hlq(8,'HLQ') as substr(record,1,8)
define type=mail userid(8,'Owner') as substr(record,10,8)
define type=mail email(43,'Email Address') as substr(record,18,50)
define type=mail dept(10,'Department') as substr(record,60,10)
newlist type=mail
  sortlist recno hlq userid email dept
```

```

bundle bundleby=qual bundlemailto=bundleby:mail.hlq.email,
      from=itso@us.ibm.com

newlist type=smf
  define qual as word(dsname,1,'.')
  select dsname=(sys1.**,sys2.**,testdata.**) intent>=update,
    not(qual==user)
  x type=80
  x dsname=sys1.broadcast
  sortlist date time dsname(22) user intent,
    qual:mail.hlq.userid(8) qual:mail.hlq.email(43)

endbundle
\

```

Example 15-7 Excerpt from the automatic email to data owners SMTP dialog

```

MAIL FROM:<itso@us.ibm.com>
RCPT TO:<JAMIE.PEASE@UK.IBM.COM>
DATA
From: itso@us.ibm.com
Reply-To: DontReply@AutoGenerated
To: JAMIE.PEASE@UK.IBM.COM
Subject: S M F   R E C O R D   L I S T I N G   25Apr08 14:10 to 25Apr08 16:31  S
Mime-version: 1.0
Content-Type: text/html; charset="iso-8859-1"

```

```

<HTML><BODY><PRE><FONT SIZE="1">
S M F   R E C O R D   L I S T I N G   25Apr08 14:10 to 25Apr08 16:31  SYS2

```

Date	Time	Dataset	User	Intent	USERID	EMAIL
25 Apr 2008	16:28	SYS2.TEST	MCAIRNS	ALTER	JPEASE	JAMIE.PEASE&#
25 Apr 2008	16:28	SYS2.TEST	MCAIRNS	UPDATE	JPEASE	JAMIE.PEASE&#
25 Apr 2008	16:28	SYS2.TEST	MCAIRNS	ALTER	JPEASE	JAMIE.PEASE&#

```

</FONT></PRE></BODY></HTML>

```

15.5 Joiners, leavers, and movers processing

In all organizations, people move from job to job during their career. New recruits arrive, and other employees leave. Managing these changes is what this section is about. This process is often referred to as *user provisioning*. With the documented access structure we have established at TRA, we are ideally positioned to take advantage of provisioning systems available for z/OS, such as IBM Tivoli Identity Manager. However, without such automated provisioning software, we must implement alternative processing for the kinds of changes. Using zSecure, we can easily establish basic provisioning processes for the TRA System, which we will now describe.

15.5.1 Flagging users for revocation, revoking them, and changing ownership of those users

The CARLa program shown in Example 15-8 identifies users who have not accessed the system in the last 100 days, revokes these users, and places them under the control of a central administration group named #DELETE.

Example 15-8 CARLa to revoke unused user IDs

```
newlist type=racf nopage f=ckrcmd
s c=user s=base last_connect_date<TODAY-100

/* Exceptions to be excluded */

x dfltgrp=' ' /* This excludes the RACF digital certificate anchor
useridids irrcerta, irrmulti, and irrsitec */
x key=IBMUSER
x protected /* This excludes Started Task and Batch useridids */

/* Generate commands */

sortlist "altuser " key(0) " revoke data('Revoked due to non-use')"
sortlist "connect " key(0) " group(#delete) owner(#delete)"
sortlist "altuser " key(0) " dfltgrp(#delete) owner(#delete)"
```

The first line of this CARLa program contains the parameter f=CKRCMD, which indicates an output DD name, where the command generated is placed for later execution. The CARLa generates three commands for each user ID selected:

- ▶ ALTUSER user ID REVOKE DATA(REVOKED DUE TO NON-USE), which revokes the user ID and updates the installation data.

- ▶ CONNECT user ID GROUP (#DELETE) OWNER(#DELETE), which connects the user ID to the #DELETE group.
- ▶ ALTUSER user ID DFLTGRP(#DELETE) OWNER(#DELETE), which sets the default group of the user ID to #DELETE.

This CARLa may be executed in batch mode by an appropriately authorized user ID, or in the foreground by a RACF administrator. Note that this CARLa excludes user IDs IBMUSER, user IDs with the PROTECTED attribute, and users without a default group (this will prevent processing user IDs such as the RACF digital certificate anchor user IDs irrcerta, irmulti, and irrsitec).

As a subsequent procedure, we can use the CARLa program shown in Example 15-9 to generate commands to delete users that have not accessed the system for a period we specify. In this code, we select users who have not accessed the system for a period of 120 days, and who are already selected in our first stage processing described above, that is, they are already owned by the group #DELETE.

Example 15-9 CARLa to delete revoked users

```
newlist type=racf nopage f=CKR2PASS
s c=user s=base last_connect_date<TODAY-150 dfltgrp=#DELETE

sortlist "REMOVE USER=" | key(0)
```

Commands of the form REMOVE USER=userid are generated by this CARLa code. An example of them are shown in Figure 15-54.

```

EDIT          LILIXIE.C2R1B9A.CKR2PASS                      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 REMOVE USER=ITSOTSA
000002 REMOVE USER=ITSOTSB
000003 REMOVE USER=ITSOTSC
***** ***** Bottom of Data *****

Press PF3, enter R at the cursor location, press ENTER to run these commands

```

Figure 15-54 Unused user ID owned by #DELETE about to be deleted

The use of the f=CKR2PASS parameter in the first line of the CARLa program shown in Example 15-9 on page 390 generates the CARLa output in the CKR2PASS DD for later execution. If you run this CARLa program in the foreground, the REMOVE USER commands are generated in the CKR2PASS DD. You can then select the CKR2PASS data set with the line command R to run the generated commands. This generates the actual RACF commands needed to cleanly delete these user IDs, as shown in Figure 15-55.

```

                                zSecure Admin+Audit for RACF Enter R to generate comm
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                      W Write file into seq. or partitioned data set
  M E-mail report

Enter a selection in front of a highlighted line below:
_ SYSPRINT  messages
_ REPORT    printable reports
_ CKRTSPRT  output from the last TS0 command(s)
_ CKRCMD    queued TS0 commands
R _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 15-55 Use CKR2PASS to generate the RACF commands

The RACF commands are generated in the CKRCMD data set. They will completely remove all references to these user IDs. You should review these commands, as default values are supplied to replace occurrences of the deleted user IDs in certain RACF profiles fields, such as the NOTIFY field. These defaults might need to be changed in your environment.

Figure 15-56 shows the actual RACF commands generated using this two pass CARLa approach. If the commands are acceptable, the security administrators may execute these either online (use line command R for Run) or through a batch job (use line command J for Submit Job). After thorough testing of this process, you probably want to run this process fully automated as a scheduled batch job in a scheduler application, such as Tivoli Workload Scheduler, rather than requiring the manual activities in the foreground.

```

EDIT          LILIXIE.C2R1B9A.CKRCMD                      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 CKR1CMD complex SC76 NJE WTSC76 generated 22 Apr
000002          /* Commands generated by DELETEDDATASETS */
000003          DELETE 'ITSOTSA' ALIAS
000004          DELETE 'ITSOTSB' ALIAS
000005          /* Commands generated by REMOVE PERMIT */
000006          deldsd 'ITSOTSA.**' generic
000007          deldsd 'ITSOTSB.**' generic
000008          /* Commands generated by (RE)MOVE USER/GROUP */
000009          remove ITSOTSA group(@TSTACC )
000010          remove ITSOTSA group(@ZSEC002)
000011          remove ITSOTSB group(@TSTACC )
000012          remove ITSOTSC group(@TSTACC )
000013          deluser ITSOTSA /* dfltgrp=#DELETE */

Press PF3, enter R at the cursor location, press ENTER to run these commands

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

```

Figure 15-56 The result of the second pass CARLa program

15.5.2 Reporting on deleted user IDs

To ensure that any scheduled deletion of mainframe user IDs has no undesired impact, and that deletions are also noted for any non-mainframe systems action, we establish a report to the security administrators that lists all recently deleted user IDs. This is run as a part of the daily batch processing for security.

This job generates a report that is sent to the security administrators by email in XML format. This report provides a summary of all deletion activities in RACF and serves to remind the administrators to delete related used IDs from other systems.

Example 15-10 contains three program sections. The first section defines the output file options and the email distribution address. The second section defines reporting options, such as the data source and title. The last section performs the real query, and generates the output report that is then sent using the output and other options.

Example 15-10 CARLa program to report on deleted user IDs

```
fileoption dd=c2remai fileformat=xml xml_datadict,
  xml_stylesheet=imbed(m=c2rxsl01) encoding=utf-8
option dd=c2remai mailto=othergroup_admin@us.ibm.com,
  from=racf_admin@us.ibm.com outputformat=attach

newlist dd=C2REMAIL type=SMF name=DELETES,
  tt="Daily report of user IDs deleted from RACF",
  subtitle="Note: Please delete these user IDs from non RACF systems"

select type=80 event=(DELUSER)
sortlist date(7) time(8) userid(10,'Cmd-issuer'),
  userid:name(20,'Cmd issuer name') racfcmd_user(8)
```

Figure 15-57 shows the XML formatted report after the administrator has received the email with the XML attached, and opened the attachment.

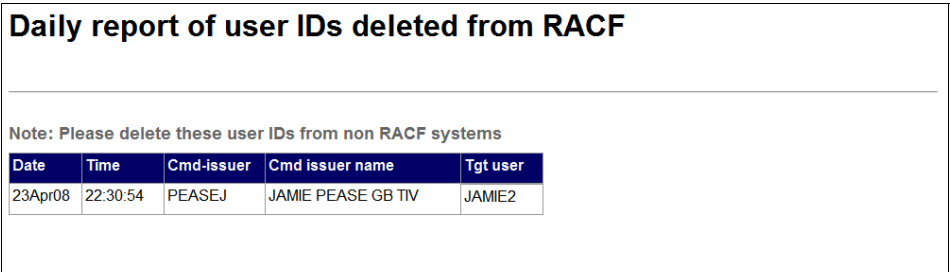


Figure 15-57 Report of users deleted

15.5.3 Leavers processing

When an employee leaves a company, it is common practice that the security administrators will first revoke their user ID, then delete it after some period of time. At this time, any permits, owned profiles, data sets, and related TSO alias entries should also be removed or updated. We call this process *cleanly deleting* the user ID. The zSecure Admin ISPF option RA.U cleanly deletes user IDs from the RACF database by addressing the attendant issues by providing optional parameters on the panel.

Figure 15-58 shows the D line command issued against a user ID from the RA.U panel.

zSecure Admin+Audit for RACF - RACF - User Delete	
Command ==>	
Userid	ITS0TS1
Specify action to perform	
<u>1</u>	1. Delete userid
	2. Move userid to holding group _____
	3. Remove userid from resource profiles (remove permit)
	4. Remove userid from NOTIFY fields
Specify resources to delete (actions 1, 2 and 3 only)	
<u>/</u>	Data set and id-specific profiles
	Only if previous option selected:
<u>/</u>	RACFVARS profiles and members
<u>/</u>	Data sets and their catalog entries
<u>/</u>	Incl. catalog entries without data sets
<u>/</u>	Incl. uncataloged data sets
Change USERID in Notify fields to _____	(default is NONOTIFY)
New Owner for non-dataset profiles _____	(default is SYS1)

Figure 15-58 RA.U, D - Delete user cleanly

We use the first option, 1 Delete user ID, to delete the user ID. We also select the options to delete RACF entries related to the user and any residual data sets, including their TSO alias. To delete data sets and the user's alias, we must have a current CKFREEZE file allocated to our session through SE.1 or the SETUP FILES command.

After pressing Enter, then F3, we see the RACF commands generated, which are divided into several sections:

- ▶ Commands generated by DELETE DATASETS

This section contains the commands to delete user data sets and the TSO alias.
- ▶ Commands generated by REMOVE PERMIT

This section contains the commands to delete any permits to resource profiles for the user ID.
- ▶ Commands generated by (RE)MOVE USER/GROUP

This section contains the commands to remove the user ID from any groups to which it is connected.

► Remaining sections

These contain the RACF commands to actually delete the user ID, and to issue a SETROPTS REFRESH for the classes where the user was previously present in access list entries.

We can execute these commands by pressing F3 and using the R line command against the CKRCMD data set. If you select the COMMANDS data set, you can see the CARLa program that generated all the TSO RACF commands necessary to perform the deletion of this user ID. It is a simple one line CARLa statement:

```
REMOVE USER=ITS0TS1
```

You can also use ISPF EDIT function to copy this CARLa code, repeating and modifying lines if you have a requirement to process many user IDs in one pass. Generate the TSO RACF commands using the line command GO or RUN, review these in the CKRCMD data set, and then execute them in the normal manner.

Note: It is not necessary to have a wide level of RACF data set access to delete user data sets and catalog entries. zSecure Admin generates IDCAMS DELETE statements that will honor storage administration profiles in the FACILITY class. Specifically, to use the REMOVE USER command effectively, you will need READ level access to these profiles:

- STGADMIN.IGG.DELETE.NOSCRCH
- STGADMIN.IGG.DEFDEL.UALIAS

15.5.4 Joiners processing

zSecure Admin provides easy solutions to create user IDs for new staff. In Chapter 3, “IBM Security zSecure Admin” on page 29, we showed how to use the Mass update feature to add many similar or dissimilar user IDs using a few simple steps. If you only need to create a single user ID, the simplest method is to use the C line command from the RA.U ISPF menu, and copy the new user from an existing definition. Using either method, the RACF commands are generated automatically, and the TSO alias can also be automatically defined at the same time by selecting this option on the panel. For detailed information, please refer to 3.2, “Automating and simplifying routine administration tasks” on page 42.

15.5.5 Movers processing

If an employee's role within the organization changes, such as a promotion or move to another department, the security administrators need to reflect this change by updating the access granted to their user ID. zSecure Admin provides the M line command from the ISPF RA.U panel for this purpose. We call this *mover processing*. Figure 15-59 shows the line commands available for user IDs and the M line command about to be issued.

zSecure Admin+Audit for RACF USER overview									
C	zSecure Admin+Audit for RACF - Actions					Scroll==> CSR			
U	Select one of 22 actions					r 2008 21:30			
/	M_	A	Authorization (permits and scope)	001		ner	RIRP	SOA	gC LCX Grp
		AC	Access Check for userid on one profile	001					1
		C	Copy userid	001				X	1
		CO	Add connect for this userid	001				X	1
		D	(Prepare actions for) delete userid	002				X	1
		E	Display event logging	001				X	2
		L	RACF listuser all command	001				X	1
		M	Move user from group (to another)	001				X	1
		MI	Manage userid-information	001				X	1
		ML	Manage logon-information	001				X	1
		MR	Manage CKGRACF authority requirements	001				X	1
		MS	Manage CKGRACF revoke/resume schedules	001				X	1
		MT	Manage TSO-information	001				X	1
		MU	Manage installation-defined USERDATA	003				X	1
		P	Change password and resume	003				X	1
		PE	Add or delete permit for this userid	003				X	1
		R	Recreate userid	003				X	1
				004				X	1
				004				X	1
				004				X	1
				004				X	1
CSV019		SC76		CUST SERV REP 19		@CSV001		#B004	
								X	

Figure 15-59 Automatic processing of movers - The M command

After pressing Enter, you will be presented with the next panel where you must specify the group or groups to which the user ID will be connected after the move, as shown in Figure 15-60.

zSecure Admin+Audit for RACF - RACF - User Move					
Command ==> _____					
Userid : CSV001					
Move userid from or between groups					
Group(s) from which user is to be moved:					
@CSV002	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
Group(s) to which user is to be moved:					
@CSV001	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

Figure 15-60 New groups for the user ID being moved

In this example, we want to move the user ID CSV001 from group @CSV002 to @CSV001. We enter this information into the panel and then press Enter. In Figure 15-61, you can see the RACF commands that are generated. After pressing F3, we can then use the R line command to run these commands.

To ensure the user's authorities are correct after moving from one group to another, we recommend you check for any RACF permit entries using the A line command from RA.U. You should check for direct permits using option 1 Direct permit to the Id (Id on access list) and remove any permissions not required in the user's new role.

EDIT		LILIXIE.C2R1B9A.CKRCMD		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 CKR1CMD complex SC76 generated 23 Apr 2008 15:52			
000002	/*	CKRCMD file CKR1CMD complex SC76 generated 23 Apr 2008 15:52			
000003	/*	Commands generated by (RE)MOVE USER/GROUP */			
000004	connect	CSV001 group(@CSV001) owner(@CSV001)			
000005	altuser	CSV001 dfltgrp(@CSV001)			
000006	remove	CSV001 group(@CSV002)			
***** Bottom of Data *****					
Press PF3, enter R at the cursor location, press ENTER to run these commands					

Figure 15-61 RACF commands to move a user ID

Many security administrators implement an alternative method for processing movers. This alternative process uses a temporary or staging group, in which the user ID being moved is temporarily *parked* during the move.

In this example, we use a group named #MOVER. You can use a group such as this one to connect every user who is about to change their job role during a transition period while you remove their old authorities. Again, we use the example of moving user ID CSV001 from the group @CSV002 to @CSV001.

Using this transitional group approach, we first move the user ID to the group #MOVER before finally moving it to the target group. Again, we use D line command on the zSecure Admin RA.U panel shown in Figure 15-59 on page 396. In this case, rather than just delete a user ID, we use options on this panel to prepare the ID for deletion, specifically, moving the user ID to a group and removing permits (options 2 and 3). First, we select option 2 Move user ID to holding group and enter the #MOVER group as the target, as shown in Figure 15-62. We also deselect the resource deletion options under Specify resources to delete (actions 1, 2 and 3 only).

zSecure Admin+Audit for RACF - RACF - User Delete	
Command ==>	_____
Userid	CSV001
Specify action to perform	
2 1. Delete userid	
2. Move userid to holding group <u>#mover</u>	
3. Remove userid from resource profiles (remove permit)	
4. Remove userid from NOTIFY fields	
Specify resources to delete (actions 1, 2 and 3 only)	
- Data set and id-specific profiles	
Only if previous option selected:	
- RACFVARS profiles and members	
- Data sets and their catalog entries	
- Incl. catalog entries without data sets	
- Incl. uncataloged data sets	
Change USERID in Notify fields to _____	(default is NONOTIFY)
New Owner for non-dataset profiles _____	(default is SYS1)

Figure 15-62 Moving a user ID to a holding group

After pressing Enter and then F3, we see that the commands that have been generated, as shown in Figure 15-63.

```

EDIT          LILIXIE.C2R1B9A.CKRCMD                      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 CKR1CMD complex SC76 generated 23 Apr 2008 16:56
000002          /* CKRCMD file CKR1CMD complex SC76 generated 23 Apr 2008 16:56
000003          /* Commands generated by (RE)MOVE USER/GROUP */
000004          altuser CSV001      revoke
000005          connect CSV001      group(#MOVER) owner(#MOVER)
000006          altuser CSV001      dfltgrp(#MOVER )
000007          remove CSV001      group(@CSV002 )
***** ***** Bottom of Data *****

Press PF3, enter R at the cursor location, press ENTER to run these commands

```

Figure 15-63 Commands to move a user ID to the holding group

These commands can then be executed in the normal manner.

For various reasons, RACF best practices normally recommend that all RACF access is granted through groups. However, RACF itself will allow the granting of access directly to user IDs. If direct permits to user IDs have been used in your installation and you want to remove them as part of moving this user ID, you can edit the CARLa commands generated by this process and remove the command that suppresses deletion of these direct permits.

In Figure 15-64, we show the CARLa commands generated. If required, remove the first command SUPPRESS DELDSD to generate commands to delete the user's own data sets. Add the optional parameter ALLPERMITS at the end of the MOVE USER command to generate RACF commands to remove any direct permits to this user ID from all RACF profiles.

```

EDIT          CKR.SCKRCARL(@LILIXIE) - 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 SUPPRESS  DELDSD
000002 MOVE USER=CSV001 TOGROUP=#MOVER,REVOKE
000003
***** ***** Bottom of Data *****

```

Figure 15-64 CARLa to move a user ID - Deleting the DELDSD suppression

Executing this CARLa using the GO or RUN command produces the RACF commands shown in Figure 15-65. You can see the RACF commands generated to delete a direct permit to the JCL profile in the class TSOAUTH.

File Edit Edit_Settings Menu Utilities Compilers Test Help	
EDIT	MCAIRNS.C2R1408.CKRCMD Columns 00001 00072
Command ==>	Scroll ==> CSR
***** Top of Data *****	
000001	/* CKRCMD file CKR1CMD complex SC76 NJE WTSC76 generated 25 Apr
000002	/* Commands generated by REMOVE PERMIT */
000003	deldsd 'CSV001.***' generic
000004	permit JCL cl(TSOAUTH) delete id(CSV001)
000005	permit 'CSV002.***' generic delete id(CSV001)
000006	/* Commands generated by (RE)MOVE USER/GROUP */
000007	altuser CSV001 revoke
000008	connect CSV001 group(#MOVER) owner(#MOVER)
000009	altuser CSV001 dfltgrp(#MOVER)
000010	remove CSV001 group(@CSV002)
000011	SETOPTS REFRESH RACLIST(TSOAUTH) /* POSIT 124 */
000012	SETOPTS REFRESH GENERIC(DATASET)
***** Bottom of Data *****	
Press PF3, enter R at the cursor location, press ENTER to run these commands	

Figure 15-65 User ID being moved with direct permits deleted

These commands can be executed in the usual manner.

After executing these commands, the user CSV001 has no access to the system, and is connected to the group #MOVER for the duration of the transition. The next step is to use the M line command against user CSV001 from the RA.U user panel. We want to move the user from the #MOVERS group to their new target group, @CSV002. We also need to resume the user, and add any additional group access or permits required. This is easily done using the zSecure Admin functions.

In general, the second approach to moving user IDs described here is used to have a clear transition between job roles. Both approaches work well though when applied using the steps we described.

If many user IDs need to be moved, you can edit the CARLa program, as shown in Figure 15-64 on page 399, repeat the MOVE USER line with the appropriate changes, and generate all the required RACF commands in one simple action.

15.6 Segregation of duties

Segregation of duties is an essential part of well designed security mechanisms. The basic tenet of segregation, in a security sense, is that one user should not have all the required access to complete a sensitive business process. In other words, the business process must be at least in part subject to review or input from at least one other staff member. Hence the term segregation, as the business process end to end requires segregation of some parts into the authorities granted to two or more staff members.

An implication of properly established segregation is that should a staff member want to commit a fraud or other undesirable action, they are forced into collusion with at least a second staff member. Thus, for the undesired action to be completed, not just one staff member has to be in violation of your governing rules and procedures. There must be at least two perpetrators, and the theory is that the more people you have to involve to commit a fraud, the more likely that the incident will be prevented or revealed.

Segregation is especially required within application systems, and made more difficult to ensure when an application embeds security within the application code or tables. For this reason, we always recommend externalizing such security logic and using RACF to control application security. Apart from the benefits of better security assurance and management, you can also then use the reports and processes we will describe here to help ensure that your required segregation principles have not broken down over time.

15.6.1 Separating administrators by specialized function

As part of the controls required to meet Delft Transport security management standards, we need to establish some segregation in the roles performed by our security administrators.

We want at least two different roles in the security administration team. One set of administrators is responsible for managing system-wide RACF options, and the other for routine RACF administration.

Using zSecure Command Verifier policy profiles such as C4R.RACF**, we control the access of all users with the system special attribute to issue RACF SETROPTS commands. Detailed examples of the use of these controls is described in 7.2, “Controlling RACF commands” on page 133 and 15.2, “Ensuring system integrity” on page 348.

We also use many other zSecure Command Verifier policy profiles to control delegated administrators access to manage users, groups, group connections, and passwords. See 15.1, “Delegated RACF administration” on page 322 for more detail about these controls.

In 15.2.6, “Lockdown profiles for segregation of responsibilities” on page 358 and 7.3.1, “Profile locking” on page 137, we described how to use zSecure Command Verifier lockdown profiles to prevent even system special users from changing these CV profiles to give themselves access and bypass these controls.

Given all these controls, we now need to authorize an appropriate administrator to manage them, and grant access as necessary on occasion for other administrators to temporarily bypass them. This administrator will be known as the Command Verifier administrator, and we will use the user ID CVADM1 for this purpose.

The CVADM1 user does not possess the system special attribute, and as such can only perform limited system administration under the control of zSecure. However, by having access to the relevant zSecure Admin CKGRACF scoping rules, this administrator is able to manage the access lists of the zSecure Command Verifier related profiles.

Herein lies our segregation of security administration roles. The CVADM1 user cannot change system wide settings or other sensitive profile access rules. The system special users likewise are limited in their ability to change system wide settings by the Command Verifier profiles in place. To bypass any controls established by the organization, the system special users must request the CVADM1 user to grant them the necessary access. Additionally, even if the CVADM1 user somehow granted access to themselves in the Command Verifier profiles, they lack the necessary native RACF authorities to do any damage even given the authority to bypass the Command Verifier controls and defaults.

Both the CVADM1 user, and the system special user must work together to make any nonstandard changes to the RACF processing rules the organization has established.

The process of segregation for security administrators where zSecure Command Verifier has been exploited can be summarized as:

1. Define CV controls using C4R prefixed profiles.
2. Nominate a CV administrator with limited native RACF authority and grant this user the necessary CKGRACF scope over the CV related profile definitions.
3. Configure these CV profiles using the CV *lockdown* configuration feature.

4. Allow the CV administrator user access to bypass the *lockdown* profiles for the CV related profiles.

Depending on precisely how you implement these profiles and their access lists, you may need to use the C4R.EXEMPT profile to establish some of these controls. We recommend this profile be only used temporarily, and that only your emergency user IDs are granted permanent access to this function. You may want to consider using zSecure Alert to monitor any changes to this profile.

In our scenario, we used the following Command Verifier RACF profile to allow our CV administrator to use zSecure Admin through CKGRACF to manage access to the CV related profiles:

```
CKG.SCP.G.#RACFRES.#CV
```

All CV profiles are defined with a RACF owner of #CV. You could use zSecure Alert to ensure that no system special attempted to violate this control.

15.6.2 Conflict detection in permits/roles

As part of the security improvement program at TRA, there is a requirement to implement daily monitoring to detect where a user is connected to RACF groups that may cause inadequate segregation of duties.

In the following scenario, a manager has the authority to submit a request in the online HR system to increase an employee's salary. This request is then subject to review and approval by the senior manager of the department. There are two RACF groups to control this process, the first being @HR003, which allows managers to submit requests to update an employee's record. The second group, @HR004, allows senior managers to give the final approval for certain changes, such as a salary increase or bonus payment.

In Figure 15-66, we show a report where a user is connected to group @HR003 and @HR004. This demonstrates how zSecure Admin can help identify conflicting access or inadequate segregation of duties between roles.

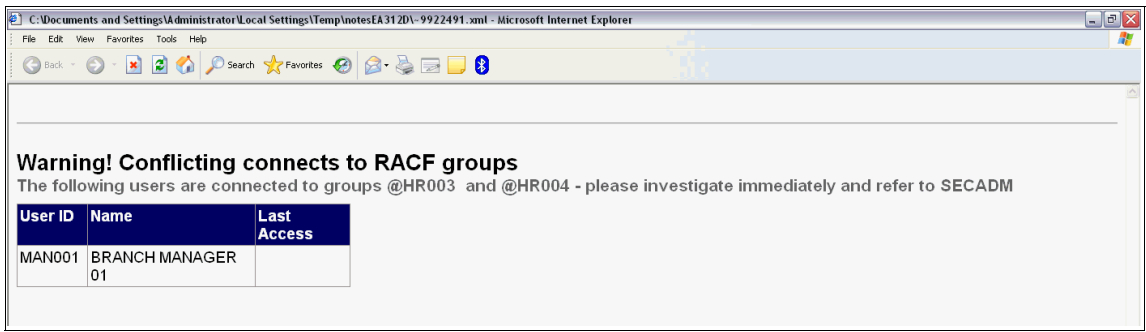


Figure 15-66 Report on conflicting connects to RACF groups

The CARLa program used to generate this report is show in Example 15-11. You could run this CARLa program on a daily basis using Tivoli Workload Scheduler (TWS) for z/OS or your equivalent job scheduler.

Example 15-11 CARLa program to generate the report shown in Figure 15-66

```
fileoption dd=c2remail fileformat=xml xml_datadict,  
    xml_stylesheet=imbed(m=c2rxsl01) encoding=utf-8  
option dd=c2remail mailto=hrsrvcs@uk.ibm.com,  
    from=secadm@uk.ibm.com outputformat=attach  
  
newlist dd=C2REMAIL type=RACF name=CONFLICT,  
    title="Warning! Conflicting connects to RACF groups",  
    subtitle="The following users are connected to groups @HR003  
    and @HR004 - please investigate immediately and refer to SECADM"  
  
select class=USER segment=BASE (cggrpnm=@HR003 and cggrpnm=@HR004)  
sortlist key("User ID",8) name last_connect_date("Last Access")
```

15.6.3 Mutually exclusive access reporting

An important part of segregation of duties is to ensure that users do not gain access to mutually exclusive privileges. A typical example is business transactions where access to these could introduce the risk of easy fraud, with only one user involved in the entire sequence of transactions used to commit the fraud.

Example 15-12 is a 2-pass CARLa report, that, when given two mutually exclusive transactions, reports on all users who have access to both of them. We assume in this example that the first transaction, ABC1, allows the generation of a financial request of some kind, perhaps a new loan, and the second transaction, ABC2, allows for completion and approval of this financial request. Thus, you would want it to be the case that no single user should be able to both generate and approve the financial transaction from end to end.

Example 15-12 Mutually exclusive CICS transaction access

```
//STEP1 EXEC PGM=CKRCARLA,REGION=0M
//STEPLIB DD DISP=SHR,DSN=CKR.SCKRLOAD
//SYSPRINT DD SYSOUT=*
//CKR2PASS DD DISP=(NEW,PASS),DSN=&&PASSFILE,
//      UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN DD *
newlist type=racf nopage retain dd=ckr2pass
  select class=tcicstrn segment=base,
    key=(ABC1,ABC2)
  sortlist key acl(effective,universal)
/*
//STEP2 EXEC PGM=CKRCARLA,REGION=0M
//STEPLIB DD DISP=SHR,DSN=*.STEP1.STEPLIB
//CKR2IN DD DISP=OLD,DSN=*.STEP1.CKR2PASS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
alloc type=racf backup
deftype type=accesses
alloc type=accesses dd=ckr2in
define type=accesses #prof('profile',44) as substr(record,1,44)
define type=accesses #user('userid',8) as substr(record,46,8)
define type=accesses #access('acc-lvl',7) as substr(record,55,7)
define type=accesses #aclid('acl-id',8) as substr(record,63,8)
newlist type=accesses,
  t='users with access to both ABC1 and ABC2      ',
  empty='no users with access to ABC1 and ABC2    ',
  sortlist #access #aclid #prof(7) #user:name #aclid:instdata
  summary #user count(>1, nd)
/*
```

This code in Example 15-12 on page 405 takes a simple CARLa report of resources in the class TCICSTRN that produces a list of all users with access, using the ACL(EFFECTIVE) output variable and modifier. What makes this code complex is that the output report from this initial program is then post processed by the second step of the batch job. The report is passed as input to the second step just as though it were an external file. A record format is defined for it using the DEFTYPE TYPE=ACCESSES statement, and then the external data file is processed with simple SORTLIST and SUMMARY CARLa statements.

The final SUMMARY by #USER where the COUNT is greater than 1 is what produces the real analysis. We were looking for instances where a user had access to two transactions and should not have. Therefore, listing the access of both transactions, and sorting by user ID, if the user appears more than once; by definition they have access to both transactions. The COUNT greater than 1 filter on the SUMMARY command captures only these user IDs who appear more than once in the first pass report and thus have access to both transactions.

Example 15-13 shows the output of this report.

Example 15-13 Report of users with access to mutually exclusive CICS transactions

users with access to both ABC1 and ABC2

userid	acc-lvl	acl-id	profile	Name	InstData
REG001					
	READ	@REG001	ABC1	REGIONAL MANAGER 01	REGIONAL MANAGER ACCESS
	READ	@REG001	ABC2	REGIONAL MANAGER 01	REGIONAL MANAGER ACCESS

Extend your reporting: TRA only uses the TCICSTRN class. Should you want to report on other CICS classes or all CICS classes, consider using `select class=%cicstrn` in your CARLa query. If you do this, you will also need to add an additional DEFINE statement and update the sortlist to include the class field.

2-pass CARLa is a complex topic, beyond the scope of this book, and best learned through your own practice. Start with simple CARLa reports and then move onto more complex techniques. There are other examples of 2-pass CARLa referenced in this book for your study.

To fully master CARLa, you might want to plan on attending the following education class:

IBM Security zSecure Auditing and Reporting Language (CARLa) - 3 day class.

15.7 Conclusion

This chapter has been an overview of some of the many sophisticated features of the zSecure suite of products that may be useful in managing a delegated administration framework. Many of the features described here are equally applicable to a centralized management structure.

We encourage you to explore the programs and structures detailed here by testing them on your own systems.

We now turn to the appendixes of this book, where you can find other detailed information referenced throughout the book.



Part 3

Appendixes



Troubleshooting

This appendix describes some common and frustrating challenges that zSecure users have come across. This list is by no means comprehensive, but includes issues that have appeared more than once to zSecure support or that have been experienced by IBM staff working in the field.

This appendix covers the following topics:

- ▶ Installation challenges
- ▶ How to get help from zSecure Support

Installation challenges

The challenges at and around installation are summarized in the following sub sections:

- ▶ “Ordering the software” on page 412
- ▶ “Obtaining the licensed documentation” on page 412
- ▶ “Receiving multiple zSecure products through SMP/E” on page 415
- ▶ “Receiving zSecure Command Verifier” on page 417
- ▶ “CKRINST” on page 418

Ordering the software

For the new zSecure customer, simply ordering the software can be the most frustrating part. This is mostly due to the fact that many times a new and separate IBM Customer Number (ICN) may be generated for the zSecure contract. When placing the zSecure order through ShopzSeries, you must ensure that your IBM ID is entitled to zSecure. Alternatively, you could request that your IBM representative place the order for you.

Problems with entitlement, that is, placing a zSecure order through ShopzSeries¹, should be routed to your IBM representative, or you can open a problem report with IBM, as shown in “How to get help from zSecure Support” on page 418.

Obtaining the licensed documentation

As mentioned previously in this book, non-licensed zSecure documentation may be obtained from the zSecure Information Center, which can be found through the zSecure Support pages at <http://www.ibm.com>.

There are, however, only two ways to obtain the licensed documentation:

- ▶ Through a download at the time the order is being received
- ▶ Through the IBM Publications website

¹ ShopzSeries can be found at this link (you must sign into this application):
<https://www14.software.ibm.com/webapp/ShopzSeries/ShopzSeries.jsp>

Through a download at the time the order is being received

After the zSecure order has been entitled and fulfilled, an email will be sent to you with a link to the order on a secure IBM FTP server and a user ID and password to use. The link will take you to a window similar to Figure A-1.

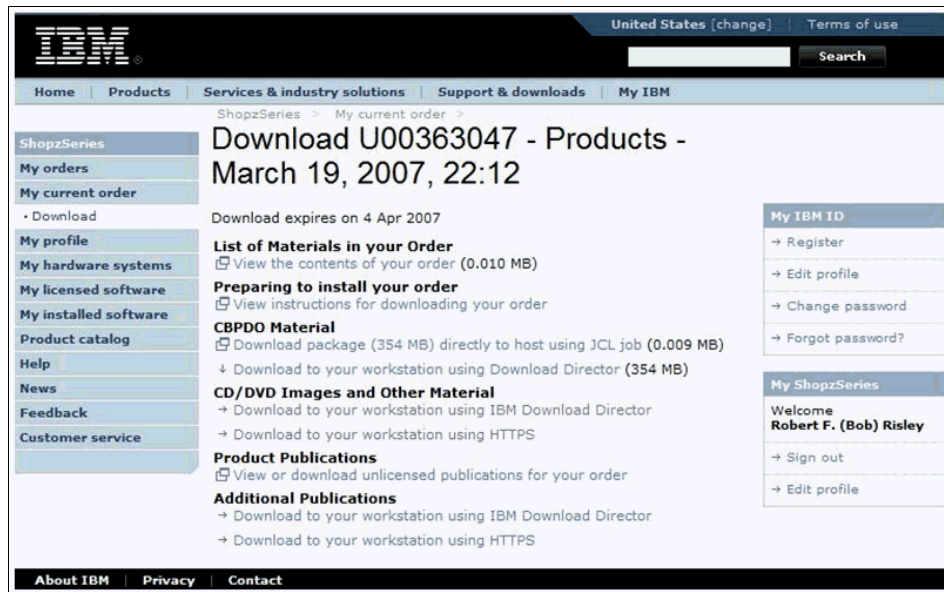


Figure A-1 Screen capture of the download window for a ShopzSeries order

Click the **CD/DVD Images and Other Material** link to get to a window that will provide you with the options to either create a CD or DVD image or to download the documentation (licensed and non-licensed) to your workstation.

Important: ShopzSeries orders are only valid for 14 days. If you have not downloaded the documentation before the order expires, you will need to either place another zSecure order through ShopzSeries (at no extra cost) or use the other method, "Through the IBM Publications website" on page 413.

Through the IBM Publications website

If you did not get the licensed documentation downloaded with your ShopzSeries order, you can get it from the IBM Publications website.

You may already have that URL bookmarked or you may be able to find it by other searches, but the easiest way to start is by going to the zSecure Information Center page, which can be found on the zSecure Support pages. Look for the Licensed Documentation link, as shown in Figure A-2.

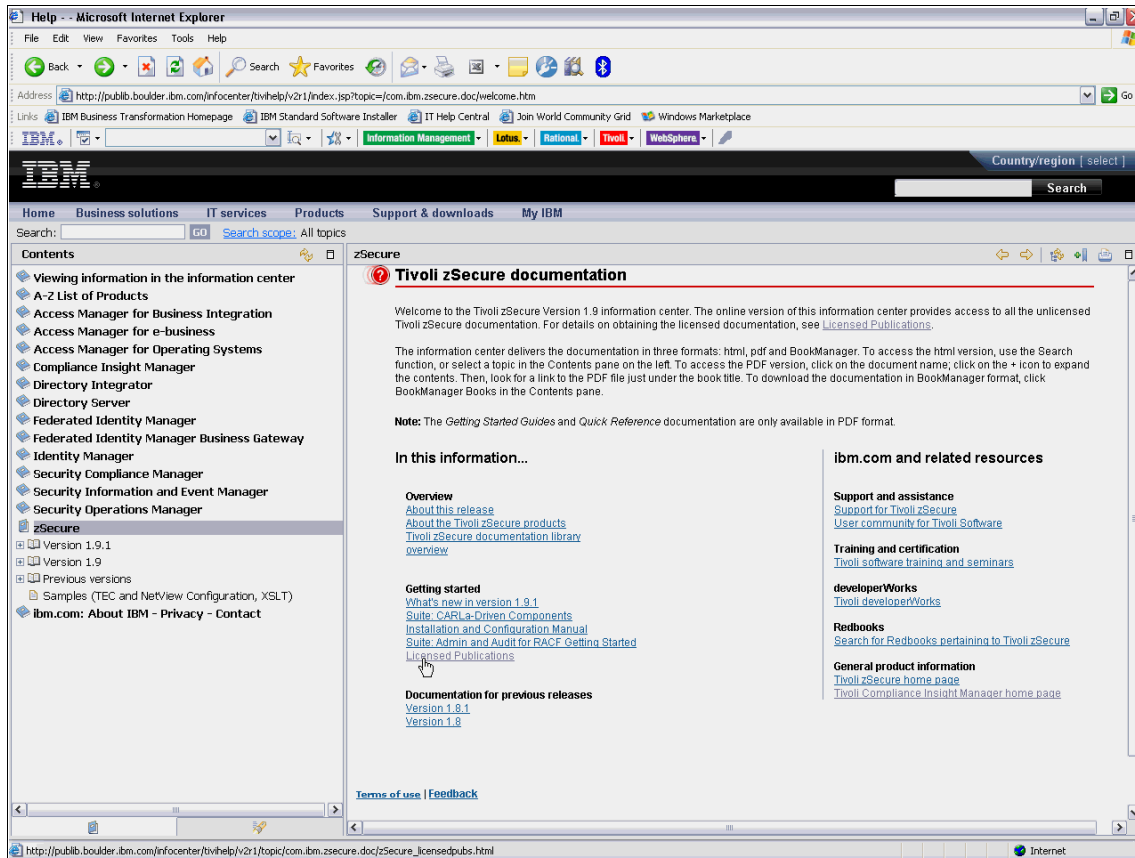


Figure A-2 Screen capture of the zSecure Information Center web page

Clicking this link will take you to another page that lists what licensed documentation is available, a link to the IBM Publications website, and the ordering instructions for either hardcopies of individual licensed manuals or the entire zSecure documentation CD. These materials are only available to entitled customers and are available at no additional cost.

Receiving multiple zSecure products through SMP/E

Obtaining multiple products out of the zSecure suite requires placing one order for each through ShopzSeries. This results in multiple web downloads, one for each product. This is fine, but care must be taken at the point where you are ready to do the SMP/E RECEIVE, APPLY, and ACCEPT of the products.

The following products from the zSecure suite may be received at the same time:

- ▶ zSecure Admin
- ▶ zSecure Audit (for RACF, CA-ACF2, or CA-Top Secret)
- ▶ zSecure Alert (for RACF or ACF2)
- ▶ zSecure Visual
- ▶ Compliance Insight Manager Enabler for z/OS (for RACF, CA-ACF2, CA-Top Secret, or DB2)

The following zSecure products *must* be installed separately from the above zSecure products; to ensure product separation, create separate product libraries:

- ▶ zSecure Command Verifier (see “Receiving zSecure Command Verifier” on page 417 for cautionary information regarding the Command Verifier install)
- ▶ zSecure CICS Toolkit

To save time and to ensure that all products are installed into the same common libraries:

1. Use any of the tapes (download files) for install up to and including the SMP/E RECEIVE. You can use Formal install up to job CKRZREC, or Fast install and terminate the job after the RECEIVE is done.
2. RECEIVE all the other tapes (download files) into that same zone. If any “already received” messages arise, ignore them.
3. After everything is received, run the remainder of install through one APPLY and one ACCEPT job for all received FMIDs. Verify the APPLY and ACCEPT steps; all required FMIDs should be un-commented.

A sample of the APPLY step with combined FMIDs is shown in Figure A-3 and in Figure A-4.

```

Menu Utilities Compilers Help
BROWSE    CONSUL.CKR190S.RCLEVL1.SCKRSAMP (CKRZAPP)    Line 00000042 Col 001 080
Command ==>            Scroll ==> CSR
SET BDY(#tzone).                /* Set to TARGET zone          */
APPLY PTFS FUNCTIONS
GROUPEXTEND                      /* Also all requisite PTFS    */
CHECK                          /* Do not update libraries    */
FORFMID (HCKR190,              /* zSecure Base               */
        HCKR19B,              /* zSecure Bookshelf          */
        HC4R190,              /* Command Verifier Base      */
/* JCKA19R, */                /* zSecure Audit for RACF     */
/* JCKR19R, */                /* zSecure Admin              */
/* JCKT19T, */                /* zSecure Audit for Top Secret */
/* JC2A19A, */                /* zSecure Audit for ACF2     */
/* JC2A190, */                /* zSecure Audit for ACF2 Base */
/* JC2E19A, */                /* Insight Manager Enabler ACF2 */
/* JC2E19D, */                /* Insight Manager Enabler DB2 */
/* JC2E19R, */                /* Insight Manager Enabler RACF */
/* JC2E19T, */                /* Insight Manager Enabler TSS */
/* JC2P19A, */                /* zSecure Alert for ACF2     */
/* JC2P19R, */                /* zSecure Alert for RACF     */
/* JC2R19R, */                /* zSecure Visual             */
)

```

Figure A-3 Sample zSecure APPLY step - Part 1

```

Menu Utilities Compilers Help
BROWSE    CONSUL.CKR190S.RCLEVL1.SCKRSAMP (CKRZAPP)    Line 00000062 Col 001 080
Command ==>            Scroll ==> CSR
SELECT (HCKR190,              /* zSecure Base               */
        HCKR19B,              /* zSecure Bookshelf          */
        HC4R190,              /* Command Verifier Base      */
/* JCKA19R, */                /* zSecure Audit for RACF     */
/* JCKR19R, */                /* zSecure Admin              */
/* JCKT19T, */                /* zSecure Audit for Top Secret */
/* JC2A19A, */                /* zSecure Audit for ACF2     */
/* JC2A190, */                /* zSecure Audit for ACF2 Base */
/* JC2E19A, */                /* Insight Manager Enabler ACF2 */
/* JC2E19D, */                /* Insight Manager Enabler DB2 */
/* JC2E19R, */                /* Insight Manager Enabler RACF */
/* JC2E19T, */                /* Insight Manager Enabler TSS */
/* JC2P19A, */                /* zSecure Alert for ACF2     */
/* JC2P19R, */                /* zSecure Alert for RACF     */
/* JC2R19R, */                /* zSecure Visual             */
)
BYPASS (HOLDSYS (DEP,DELETE)). /* Bypass options            */
/*

```

Figure A-4 Sample zSecure APPLY step - Part 2

Note that this JCL, provided at installation time, has all FMIDs except for Base zSecure, Command Verifier, and the bookshelf commented out. If, for example, you have received Admin, Audit RACF, and the Compliance Insight Manager Enabler for RACF, in this APPLY JCL you would un-comment the lines for JCKR19R, JCKA19R, and JC2E19R in both the FORFMID and the SELECT statements. FMIDs HCKR190, HCKR19B, and HC4R190 must remain un-commented.

APAR OA23787 has been created to address this issue and has been incorporated in the Program Directory for zSecure Suite: CARLa-Driven Components Release V1.9.1.

Receiving zSecure Command Verifier

The scenario: You have already received, applied, and accepted zSecure Admin and zSecure Audit. You are now ready to install zSecure Command Verifier. But you notice in the RECEIVE, APPLY, and ACCEPT JCL that it includes FMID HC4R190, which was already done with the Admin and Audit install. So you decide to remove HC4R190 from the Command Verifier SMP JCL and only work with JC4R190.

Unfortunately, doing this will cause C4RMAIN to be omitted from the SC4RLNK data set that is created as part of the Command Verifier install, because JC4R190 does not contain the code for C4RMAIN, the primary program for Command Verifier. You can fix this by re-running the APPLY and ACCEPT with a REDO and including HC4R190.

Better yet, do not remove it and run the JCL as is, with both FMIDs.

```
Menu  Utilities  Compilers  Help
-----
BROWSE      C4R.SC4RINST(C4RJREC) - 01.00                      Line 00000044 Col 001 080
Command ==> _____ Scroll ==> CSR
//SMPCTL DD *
SET         BOUNDARY (GLOBAL) .                                /* Set to GLOBAL zone      */
RECEIVE     SOURCEID (C4R)                                     /* Assign sourceid         */
            SELECT (HC4R190,JC4R190)                          /* The Command Verifier FMIDs*/
            SYSMODS                                           /* Any type of sysmod      */
            FORFMID (HC4R190,JC4R190) .                       /* The Command Verifier FMIDs*/
***** Bottom of Data *****
```

Figure A-5 Sample Command Verifier RECEIVE step

CKRINST

You have now completed the SMP/E work to install zSecure and you are doing the customization as described in *IBM Security zSecure Suite: CARLa-Driven Components Installation and Deployment Guide Version 1.12, SC27-2772*. At some point, you need to run a JCL that has a variable called &instlib that is pointing to a data set called #hlq.CKRINST. But you do not have the variable and cannot find a reference to it anywhere. What is the JCL looking for?

In essence, CKRINST is your customized copy of the sample JCL. The Program Directory makes a mention of this in the Sample Jobs section, by stating: “You can access the sample installation jobs directly from tape or perform an SMP/E RECEIVE and then copying the jobs from IBM.HCKR190.F1 to a work data set for editing and submission.” The sample JCL that follows shows an OUT DD statement with a DSN= jcl-library-name. *This* is the data set that is being referred to as #hlq.CKRINST.

If you cannot find your IBM.HCKR190.F1 file, the same sample installation JCL can be found in IBM.HCKR190.SCKRSAMP.

How to get help from zSecure Support

All IBM Support pages have been revamped to have the same look and feel. Figure A-6 on page 419 and Figure A-7 on page 420 show a view of the zSecure Admin for RACF Support page.

United States [change]

Home
Solutions
Services
Products
Support & downloads
My IBM

Welcome [IBM Sign in] [Register]

Tivoli

Products

IBM Tivoli zSecure Audit for RACF

Solutions

Software Demos

Technical resources

News

Events

Success stories

Training

Services

Community

How to buy

Support

Software > Tivoli > Products >

IBM Tivoli zSecure Audit for RACF Support

Software Support

Overview

Download

Troubleshoot

Notice anything new?

We're pleased to introduce our new Web pages designed to help you locate resources for important support tasks.

[Learn more](#)

Search IBM Tivoli zSecure Audit for RACF support

Your focused search in IBM Tivoli zSecure Audit for RACF support can be refined on the results page. You can also access advanced options by choosing Search in the support task navigator.

Enter search terms

Flash 6 Mar, 2008:

Changes to Daylight Saving Time will affect IBM Tivoli zSecure Audit

Flash 12 Feb, 2008:

DD and File Usage Considerations when Migrating to zSecure 1.8.1+ products

Flash 18 Dec, 2007:

IBM Tivoli zSecure Audit V1.9 is Available

[View all Flashes](#)

IBM Tivoli zSecure Audit for RACF support

Overview

Download

Troubleshoot

Search

Documentation

Forums & Communities

Plan

Install

Use

Open service request

Assistance

Personalized support

Visit [My support](#) for fast access to your favorite features.

System availability

→ Last updated

Sunday, March 09, 2008 7:00:00 AM

Support feedback

Help us improve online software support

Translate my page

Select a language

→ Translate

Other support sites

→ Software

Figure A-6 zSecure Audit for RACF Support page - Part 1

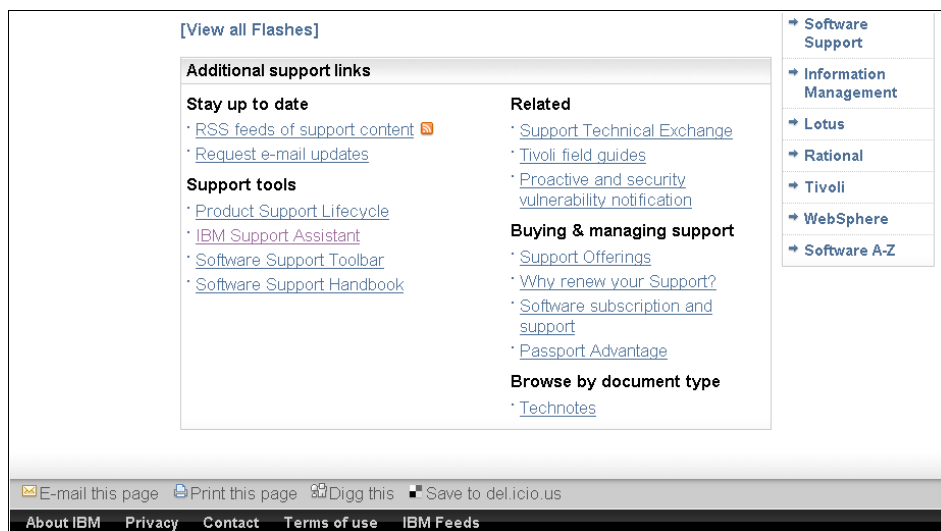


Figure A-7 zSecure Audit for RACF Support page - Part 2

The main Support page for each zSecure will contain a Search function to enable you to do a focused keyword search pertaining to the product. Flashes for the product are also prominently displayed for easy reference.

The bottom of the page also provides links to EOS dates for the product (Product Support Life cycle under Support Tools) and Technotes for the product (under Browse by document type).

From this and the other zSecure product Support pages, you can get to the following information:

- ▶ How to contact zSecure Support
- ▶ What information to send to zSecure Support
- ▶ Researching zSecure product maintenance
- ▶ Finding zSecure technical documentation

How to contact zSecure Support

Looking at the blue IBM Security zSecure Audit for RACF support box on the right hand side of Figure A-6 on page 419, you will notice a link for Open service request. Clicking this link will take you to the Electronic Service Request (ESR) page where you can link to a login page.

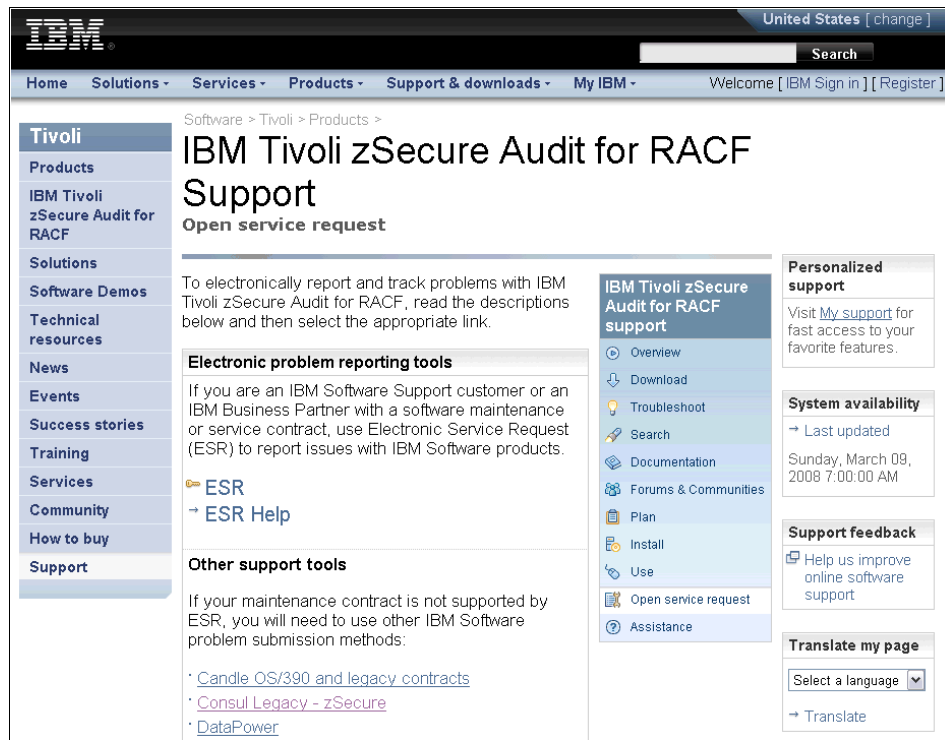


Figure A-8 Open service request page for zSecure Audit for RACF - Part 1

With your zSecure IBM Customer Number (ICN) in hand, you will be able to open problem tickets that will put you in touch with zSecure Level2 Support to address your questions and problems. In order to use ESR, you must have a valid IBM ID and password, which would be the same one you would use if you were logging on to IBMLINK. Help is available on the ESR login page if you either need to request an IBM ID or have forgotten the password.

If your IBM ID has been entitled against your zSecure IBM Customer Number, you can also open problem tickets at IBMLINK. Follow the link to Electronic Technical Response (ETR) and Report a defect (Problem). zSecure products will be listed under Tivoli, as shown in Figure A-9.



Figure A-9 View of zSecure products listed under Tivoli in IBMLINK ETR

What information to send to zSecure Support

At the bottom of the Support page (the top being shown in Figure A-8 on page 421) is a link for the Collect troubleshooting data page.

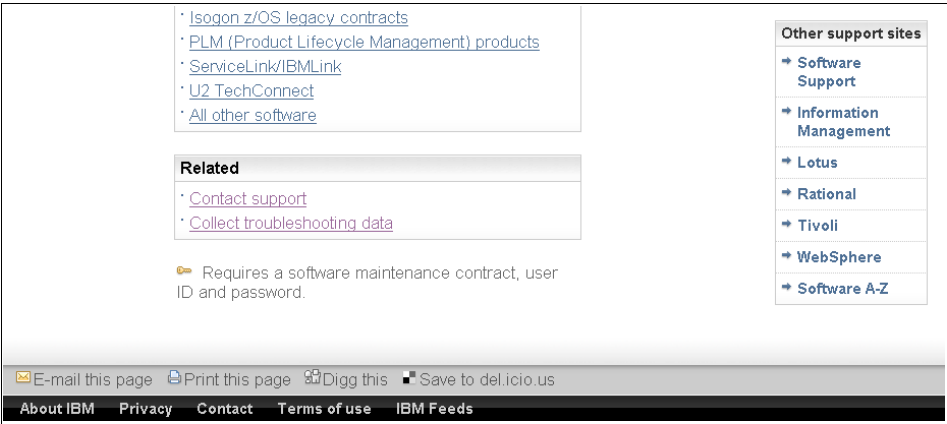


Figure A-10 Open service request page for zSecure Audit for RACF - Part 2

This link will take you to the product's MustGather page, where you will find a wealth of information, from what information to send and how to gather it, to online resources and other related information, as shown in Figure A-11.

IBM

Home Solutions ▾ Services ▾ Products ▾ Support & downloads ▾ My IBM ▾

Software > Tivoli > Products >

Support & downloads

IBM Tivoli zSecure Audit for RACF

Tivoli

Products

Solutions

Software Demos

Technical resources

News

Events

Success stories

Training

Services

Community

How to buy

Support

MustGather: IBM Tivoli zSecure Audit for RACF problem reporting

Technote (FAQ)

Question

What basic data should be collected to document a problem with IBM Tivoli zSecure products when reporting a problem to IBM?

Answer

Collecting MustGather data early, even before opening the PMR, helps IBM® Support quickly determine if:

1. Symptoms match known problems (rediscovery).
2. There is a non-defect problem that can be identified and resolved.
3. There is a defect that identifies a workaround to reduce severity.
4. Locating root cause can speed development of a code fix.

You can find other product's MustGather documents by searching on the word **MustGather** on the eSupport Web page:
<http://www.ibm.com/software/sysmgmt/products/support/>.

MustGather: Readme first table of contents:

↓ [Gathering General Information](#)

↓ [Gathering Problem Specific Information](#)

↓ [Submitting Information to IBM Support](#)

↓ [Online Self-Help Resources](#)

↓ [Related Information](#)

Gathering General Information

Gather general information about your environment, provide a good problem description for the PMR and send relevant basic diagnostic data as follows.

1. IBM Tivoli zSecure release and maintenance level
2. Operating system: z/OS or z/VM release and maintenance level
3. DB2® level if z/OS actuator problem
4. CICS level if zSecure CICS Toolkit problem

Document information

Product categories:

Software

Security

Security

Compliance and Vulnerability Management

[IBM Tivoli zSecure Audit for RACF](#)

zAudit RACF

Operating system(s):

z/OS

Software version:

All Versions

Software edition:

All Editions

Reference #:

1288559

IBM Group:

Software Group

Modified date:

2007-11-16

Translate My Page

Select language ▾

Figure A-11 MustGather page for zSecure Audit for RACF

Researching zSecure product maintenance

Looking back at the blue Support box at the top of the zSecure product Support page (Figure A-6 on page 419), clicking the **Download** link will bring you to the page where you can go to the Recommended Fixes page, where the latest fixes are posted and where you can do a keyword search for fixes to problems you have encountered.

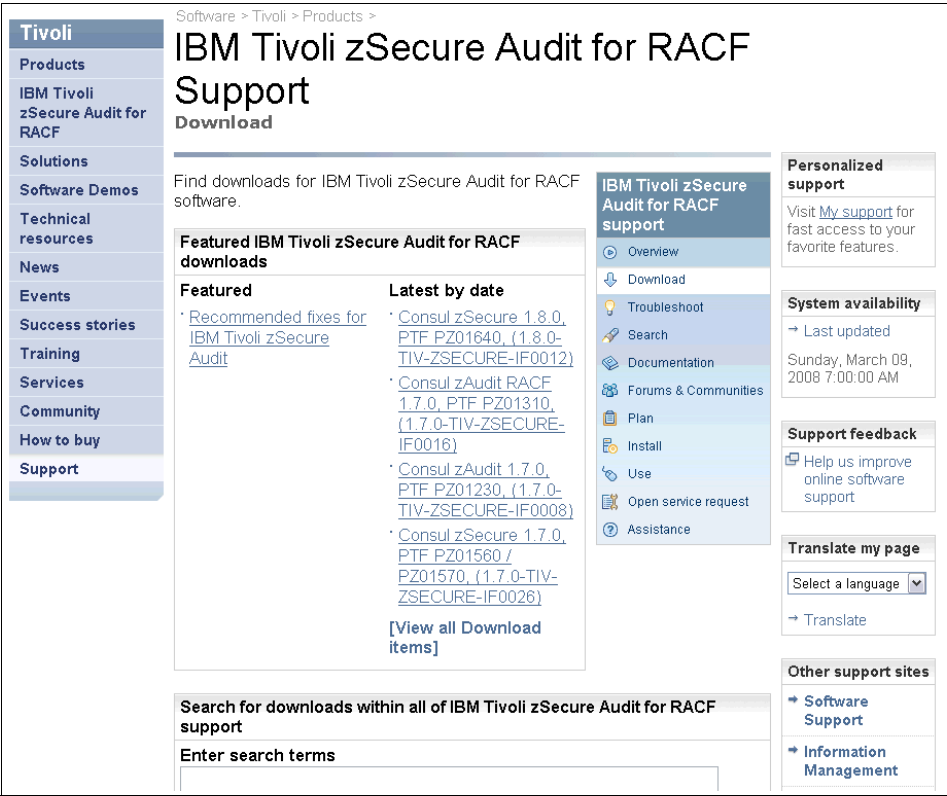


Figure A-12 zSecure Audit for RACF fixes download page

Finding zSecure technical documentation

Before you contact Support, it is always a good idea to see if any help can be found by reviewing existing documentation. In the blue support box, there is a link to Troubleshooting. This page lists the latest information about the product, from Technotes, to product releases, and to fix information. The top of this page also provides a Search bar for keyword searches within the product's web information pages, as shown in Figure A-13.

Tivoli

Products

IBM Tivoli zSecure Audit for RACF

Solutions

Software Demos

Technical resources

News

Events

Success stories

Training

Services

Community

How to buy

Support

Software > Tivoli > Products >

IBM Tivoli zSecure Audit for RACF

Support

Troubleshoot

Is your IBM Tivoli zSecure Audit for RACF software reporting errors? Are you getting unexpected results? These resources can help diagnose and resolve these issues.

Search IBM Tivoli zSecure Audit for RACF support for Troubleshoot material

Your focused search in IBM Tivoli zSecure Audit for RACF support can be refined on the results page. You can also access advanced options by choosing Search in the support task navigator.

Enter search terms

Latest Troubleshooting information

8 Jan, 2008: Is it possible to only report RACF changes that have occurred since the last audit?

16 Nov, 2007: MustGather: IBM Tivoli zSecure Audit for RACF problem reporting

3 Oct, 2007: IBM Tivoli zSecure Mainframe Products COMPID 5655T0100, 5655T0700, 5655T0900,

IBM Tivoli zSecure Audit for RACF support

Overview

Download

Troubleshoot

Search

Documentation

Forums & Communities

Plan

Install

Use

Open service request

Assistance

Personalized support

Visit [My support](#) for fast access to your favorite features.

System availability

→ Last updated

Sunday, March 09, 2008 7:00:00 AM

Support feedback

Help us improve online software support

Translate my page

Select a language

→ Translate

Other support sites

Figure A-13 View of the zSecure Audit for RACF Troubleshooting page - Part 1

At the bottom of the page are links to a more complete listing of APARs, Flashes, and Technotes, as shown in Figure A-14.



Figure A-14 View of the zSecure Audit for RACF Troubleshooting page - Part 2

Conclusion

There is a wealth of information and help available from the zSecure product Support pages, but if you prefer to speak to someone, you may call 1-800-IBMSERV.



B

An introduction to CARLa

This appendix provides an introduction to the CARLa reporting language. We describe some of the features in the language, together with useful aids to help you build CARLa programs. We also provide you with examples of CARLa programs, which you can use to produce custom reports and build commands to automate administration and auditing tasks.

This appendix covers the following topics:

- ▶ About CARLa
- ▶ Data sources
- ▶ Writing a CARLa program
- ▶ Basic CARLa to get you started
- ▶ Where to store your CARLa programs for reuse
- ▶ Useful primary commands
- ▶ Additional examples of CARLa
- ▶ Conclusion

About CARLa

CARLa stands for Consul Auditing and Reporting Language. It is the main reporting engine used within zSecure Admin, zSecure Audit, zSecure Alert, zSecure Visual, and zSecure Manager for RACF z/VM. CARLa is extremely sophisticated and can be used to generate your own reports as well as actions, SNMP data, emails, and WTO. An example of actions includes RACF, TSO, or UNIX commands.

One of the major advantages of CARLa is that you do not need to write programs such as REXX or CLISTs to interrogate and correlate data sources. What can normally be achieved with just a few lines of CARLa might take several lines of REXX. CARLa can process large volumes of data in an efficient manner using minimal CPU resources. CARLa has many of the features you would expect to find in a data processing language, including statistical and mathematical features.

Standard reports available in the ISPF interfaces of IBM zSecure are produced using CARLa. For any of the reports generated, a user can edit the CARLa that was used to generate the report and customize it to meet their reporting requirements. Many customers use this method to learn CARLa, which also saves time in coding new CARLa programs. We discuss this further in “Basic CARLa to get you started” on page 432.

CARLa is structured, repeatable, and scalable. For example, you can:

1. Build a suite of CARLa programs to assist with security administration and auditing tasks. These programs can be run in foreground or batch.
2. Use the same CARLa program(s) on any system where IBM zSecure is installed with little or no changes to the CARLa program.
3. CARLa programs you have written for SMF event reporting as part of zSecure Audit can be migrated to zSecure Alert for real-time event reporting.

CARLa programs can generate output in various formats, including ISPF, print, or XML.

Data sources

CARLa can process multiple types of data sources (input files). For example, you can correlate information from the RACF database, SMF data, and CKFREEZE file (system configuration) using one CARLa program. Unlike other programming languages, you do not need to define the record format for supported data sources, which includes RACF databases (active, unloaded, or copied), SMF (active and dumped), and CKFREEZE files. This is important because it can significantly reduce the time spent on coding efforts. CARLa can interpret the layout and content of supported data sources through the sophisticated intelligence built into the language. CARLa can also provide in-depth data mining through granular access to the supported data sources.

CARLa allows you to work with external files. Examples of an external file include:

- ▶ Data extracts from a Human Resources system
- ▶ Unloaded data from DB2 tables
- ▶ Unloaded tables or databases containing a list of user IDs
- ▶ Unloaded data from a directory
- ▶ Unloaded log files

One advantage of being able to work with external files is for reconciliation purposes. If you have an application that uses an internal security database for authorization checking rather than RACF, you may want to periodically reconcile this database with RACF so they are kept in sync. Note that you are required to define a data structure and variables in your CARLa program when working with external files. In general, we would always discourage using internal security on the z/OS platform if this can be avoided. We recommend using the ESM (whether it is RACF, ACF2, or Top Secret) for all security decisions.

In Figure B-1, we show the input sources you can use with CARLa programs.

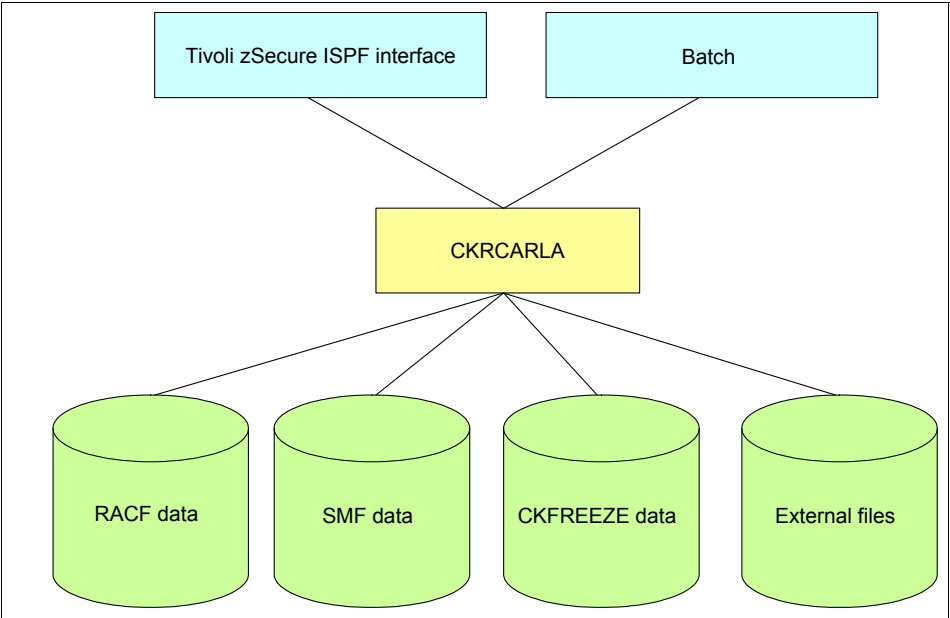


Figure B-1 Data sources supported by the CARLa program

Table B-1 gives a description of these data sources.

Table B-1 Description of data sources

Data source	Description
RACF data	<div>RACF data includes:</div> <ul style="list-style-type: none">▶ The active primary or active backup RACF database.▶ An unloaded RACF database, unloaded by the zSecure Admin UNLOAD utility.▶ A copy of the RACF database, copied by the IBM RACF utility IRRUT200.
SMF data	<div>SMF data includes:</div> <ul style="list-style-type: none">▶ The active SMF data set.▶ SMF data dumped by the IBM utility IFASMFDP or zSecure Audit utility C2RJFUNL.

Data source	Description
CKFREEZE	CKFREEZE data includes: <ul style="list-style-type: none"> ► Live settings, based on information in the control blocks of the current live system. Note that this information is limited. ► A file(s) containing resource information gathered by a zSecure Collect Job.
External files	These are Installation defined data sets, which can be referenced in CARLa programs. You must define variables in your CARLa program to enable the program to work with external data.

Input files are allocated through the SETUP FILES command or option SE.1 in your zSecure Admin or zSecure Audit ISPF session. If you are running a CARLa program in batch, you can allocate input files in your JCL using the ALLOC statement (the ALLOC statement can also be used in foreground to allocate external files). Refer to “Examples of basic CARLa programs for RACF reporting” on page 436 for details of how to submit CARLa in batch.

Writing a CARLa program

When you would need to write a CARLa program depends on the complexity of your organization, including the security and audit reporting requirements. There are numerous ready for use reports that normally satisfy reporting requirements for many organizations. However, some organizations have specialized reporting and automation needs and require a highly customizable, repeatable, and efficient reporting language to meet the demands driven by policies and regulations.

Security administrators, auditors, and system programmers often use CARLa to build *ad-hoc* or continuous reporting and automation.

Basic CARLa to get you started

CARLa is a simple language to learn and IBM zSecure users often gain experience through copying and customizing CARLa from:

- ▶ The SCKRSAMP library (invoke through option CO.1 to allocate the library, followed by option CO.2 to access the members).
- ▶ CARLa generated by running reports in the ISPF interface. This is a popular method of generating a CARLa program that is similar to your actual requirements. Use the RESULTS command as soon as you have run a report using the ISPF interface and select the COMMANDS data set to obtain the CARLa.
- ▶ Option RA.C panel, by tagging one of the report templates displayed at the bottom of the panel to show common fields.
- ▶ Referring to Chapter 11, “Command Language”, in *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773.

IBM provides training in CARLa. The name of the class is *IBM zSecure CARLa Auditing and Reporting Language*, and full details of this and other IBM zSecure training offerings can be found on the Tivoli software training website at the following address:

<http://www.ibm.com/software/tivoli/education/>

Where to write and run your CARLa program

CARLa can be stored into (almost) any data set or file and passed to the CKRCARLa program in a batch job. The way to set up this batch job is described in *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773. For example, the Compliance Insight manager Agent runs CARLa programs, the zSecure Visual client generates CARLa that the Visual server runs, the zSecure Alert address space can run CARLa, and batch jobs like Daily Refresh use CARLa.

CARLa can also be written and run from option CO.C (as shown in Figure B-2) or you can issue the command CARLA or COMM in your zSecure Admin or zSecure Audit ISPF session. Alternatively, you can use option RA.C in zSecure Admin; however, you can only produce RACF reports from this option and reports will be displayed in an ISPF panel.

Menu Options Info Commands Setup		
zSecure Admin+Audit for RACF - Main menu		
Option ==> <u>CO.C</u>		
		More: +
SE	Setup	Options and input data sets
RA	RACF	RACF Administration
AU	Audit	Audit security and system resources
EV	Events	Event reporting from SMF and other logs
CO	Commands	Run commands from library
IN	Information	Information and documentation
LO	Local	Locally defined options
X	Exit	Exit this panel
Input complex: Active backup RACF data base and live SMF data sets		

Figure B-2 Selecting option CO.C

Alternatively, you can select option CO to see the menu for Commands, as shown in Figure B-3.

Menu	Options	Info	Commands	Setup	StartPanel
zSecure Admin+Audit for RACF - Commands					
Option ==> <u>C</u>					
1	Libraries	Select and maintain command library			
2	Members	Work with members from current command library			
3	Edit	Edit member from current command library			
4	Run	Run member from current command library			
5	Submit	Run member from current command library in background			
C	Command	Type in any CARLa command			
Member name _____ (if 3, 4 or 5 selected)					
Two pass query <u>N</u> (Y/N, option 4 only)					
Current library . . . : DD:CKRCARLA					
Input complex : Active backup RACF data base and live SMF data sets					
Current mask type . . : EGN					

Figure B-3 The Commands main menu

In Figure B-4, the CO.C panel is displayed. This is where you can type in any CARLa command.

Menu	Options	Info	Commands	Setup
EDIT	CKR.SCKRCARL (#JPEASE) - 01.00	Columns 00001 00072		
Command ==> _____ Scroll ==> <u>CSR</u>				
***** 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 B-4 Option CO.C (allows you to type in any CARLa command)

A prerequisite to run any CARLa program is that you must have the appropriate input file(s) selected. You can do this through the SETUP FILES command (as shown in Figure B-5) or by selecting option SE.1. For the purpose of the CARLa program shown in Figure B-7 on page 436, we will have the active primary RACF data base selected as input (as shown in Figure B-6).

```
Menu  Options  Info  Commands  Setup
-----
EDIT      CKR.SCKRCARL (#JPEASE) - 01.00                      Columns 00001 00072
Command ==> SETUP FILES                                     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 B-5 Using the SETUP FILES command to set up input files

```
Menu  Options  Info  Commands
-----
zSecure Admin+Audit for RACF - Setup - I Row 1 from 8
Command ==> _____ Scroll ==> CSR

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

Description                                Complex
- Active primary RACF data base              SC76      selected
- Active backup RACF data base and live SMF data sets SC76
- test daily signature gdg input file        SC76
- test daily ckfreeze gdg input file         SC76
- test daily unload gdg input file           SC76
- Active backup RACF data base               SC76
- CT CKFREEZE                               SC76
- CT UNLOAD                                 SC76
***** Bottom of data *****
```

Figure B-6 Setup input files panel

Select the desired input file or insert your own if required on the setup input file panel.

Examples of basic CARLa programs for RACF reporting

Having established the data source (input file) to work with, we have written a basic CARLa program, as shown in Figure B-7. This particular program will extract all user IDs with a MASK of ITSO and produces a printed report listing all user IDs matching this mask. The report also includes the user's name, default group, and owner group.

```
Menu  Options  Info  Commands  Setup

EDIT      CKR.SCKRCARL (#JPEASE)  - 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
000002 newlist type=RACF
000003   select class=USER segment=BASE mask=ITSO*
000004   sortlist key(8) name dflgrp owner
***** ***** Bottom of Data *****
```

Figure B-7 CARLa program written in option CO.C

In the example shown in Figure B-7, we have used three of the most commonly used statements:

- NEWLIST
- SELECT
- SORTLIST

Refer to Table B-2 where we describe these and other statements used in this example.

Table B-2 Description of CARLa statements used in Figure B-7

CARLa statement	Description
NEWLIST	Marks the beginning of a new report.
TYPE=	<p>Select the newlist type of information to be used as an input for the NEWLIST. We use TYPE=RACF in our example to extract RACF information. This is the default should you omit the statement.</p> <p>There are many other newlist types of information you can use as input for the NEWLIST. Use the FIELDS command and select the BUILTIN field to display the supported newlist types.</p>

CARLa statement	Description
SELECT	This command is used to select records that match the conditions specified in the select statement.
CLASS=	Select the RACF class to report on.
SEGMENT=	Select the segment to report on from the RACF profile (for example, BASE, TSO, or OMVS).
MASK=	Select profile keys using a mask.
SORTLIST	Specify the fields to be shown in your report, based on the select criteria and specify the sort order of the output. LIST can be used if sorting is not desired. You may want to refer to the “List family of commands”.
KEY	The profile name(s) found from your select criteria.
DFLTGRP	The default group of the profile.
OWNER	The owner of the profile.

In Figure B-7 on page 436, there is a “=NOTE=” above the CARLa program itself. This note advises you how to execute your CARLa program and how to save it for future use. You have three options:

1. For foreground execution, issue the GO or RUN command.
2. To submit in batch, enter the SUB or SUBMIT command. This will generate a JCL for you with all the necessary JCL statements to execute your CARLa program. You should supply a job card when prompted by the Submit menu.
3. To save your CARLa program, enter SAVE. If you press F3 to exit from option CO.C, your CARLa program will be automatically saved in your ISPF profile. Should you want to use this program in the future, we recommend you save it to a partitioned data set. We discuss this further in “Where to store your CARLa programs for reuse” on page 450.

On this occasion, we have chosen to execute the CARLa program (as shown in Figure B-7 on page 436) in foreground using the RUN command. The report produced from our CARLa program is shown in Figure B-8.

```

BROWSE - JPEASE.C2R234D.REPORT ----- LINE 00000000 COL 001 080
COMMAND ==> _                                SCROLL ==> CSR
***** Top of Data *****
P R O F I L E   L I S T I N G      8 Apr 2008 19:54

Profile  Name                DfltGrp  Owner
ITS0TSA  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TSB  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TSC  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS1  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS2  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS3  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS4  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS5  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS6  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS7  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS8  TEST FOR ITS0        @ZSEC002 #SECADM
ITS0TS9  TEST FOR ITS0        @ZSEC002 #SECADM
***** Bottom of Data *****

```

Figure B-8 Report produced from CARLa program

In Example B-1, we show you how to generate and display a simple RACF report in an ISPF panel to demonstrate how flexible CARLa is for displaying output. We have used a CARLa program similar to the one shown in Figure B-7 on page 436; however, we have introduced the following statements:

- ▶ **DISPLAY:** To display the report in an ISPF panel. Note that we previously used SORTLIST to print the appropriate fields in our report.
- ▶ **CONNECTS:** To show the groups to which the users are connected.

Example: B-1 CARLa program displays output in an ISPF panel

```

newlist type=RACF
  select class=USER segment=BASE mask=ITS0*
  display key(8) name dfltgrp owner connects

```

```

IBM Tivoli zSecure RACF display                                0 s elapsed, 0.0 s CPU
Command ==> _____ Scroll==> CSR
                                     8 Apr 2008 21:32

  Profile  Name                DfltGrp  Owner
__  ITS0TSA  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TSB  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TSC  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS1  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS2  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS3  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS4  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS5  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS6  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS7  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS8  TEST FOR ITS0      @ZSEC002 #SECADM
__  ITS0TS9  TEST FOR ITS0      @ZSEC002 #SECADM
***** Bottom of Data *****

```

Figure B-9 Output from CARLa program displayed in an ISPF panel

To show the groups the user is connected to, we can place the “S” action character against the user ID, as shown in Figure B-10. The panel shown in Figure B-11 will appear. Alternatively, if you want to perform a different action, type “/” against the appropriate user ID to display a list of actions.

```

IBM Tivoli zSecure RACF display                                     Line 1 of 12
Command ==> =_____ Scroll==> CSR
                                     9 Apr 2008 12:55

  Profile  Name                DfltGrp  Owner
S ITS0TSA  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TSB  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TSC  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS1  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS2  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS3  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS4  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS5  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS6  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS7  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS8  TEST FOR ITS0      @TSTACC  #TSTUSR
_ ITS0TS9  TEST FOR ITS0      @TSTACC  #TSTUSR
***** Bottom of Data *****

```

Figure B-10 Using the “S” action character to display additional information

```

IBM Tivoli zSecure RACF display                                     Line 1 of 1
Command ==> =_____ Scroll==> CSR
                                     9 Apr 2008 12:55

  Profile  Name                DfltGrp  Owner
ITS0TSA   TEST FOR ITS0      @TSTACC  #TSTUSR
User/Grp  Auth    R SOA AG Uacc  Revokedt  Resumedt
_ @TSTACC  USE      _      _  NONE
***** Bottom of Data *****

```

Figure B-11 A list of groups to which the user is connected

Notice that some fields are underlined. This allows you to overtypе fields so you can make changes to a user ID in the same way you can overtypе certain fields in other ISPF panels within zSecure Admin and zSecure Audit.

In Example B-2, we generate some RACF commands using a CARLa program. This demonstrates how you can perform mass updates to the RACF database, based on a selection criteria. For the purpose of this exercise, we are going to assign a TSO segment to some users. We will need to introduce some additional fields to enable this segment:

1. FILE: Specifies the output file name for the report. In this case, we have used a zSecure DD name of CKRCMD, which is used to write RACF commands to a file so we can run them.
2. NOPAGE: Suppresses the headers and titles from the output. You should always use this statement when you specify FILE=CKRCMD or DD=CKRCMD.
3. In our SORTLIST statement, you will notice that we have specified the RACF command we need to issue to assign a TSO segment to the affected users. We wrap quotation marks around the command syntax to distinguish constants from variable(s), in this case, key(8). This is the variable that will contain the user ID(s) we found in our SELECT criteria.

Example: B-2 CARLa program to generate RACF commands

```
newlist type=RACF file=CKRCMD nopage
  select class=USER segment=BASE mask=ITS0*
  sortlist "alu" key(8) "TSO(SIZE(4096) ACCTNUM(1234) PROC(TSOPROC))"
```

In Figure B-12, we show the generated RACF commands from the CARLa program. There are two points we need to bring to your attention in the CKRCMD output file:

1. The first eight columns of each line are blank because the program IKJEFT01 in batch ignores the contents of these columns, irrespective of content.
2. A long ISPF message is displayed to advise you about how to execute the generated commands. Press F3 to invoke the RESULTS panel, where you can execute the RACF commands.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      JPEASE.C2R134D.CKRCMD                      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 CKR1CMD complex SC76 generated 10 Apr 2008 11:44
000002      ALU  ITSOTSA  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000003      ALU  ITSOTSB  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000004      ALU  ITSOTSC  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000005      ALU  ITSOTS1  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000006      ALU  ITSOTS2  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000007      ALU  ITSOTS3  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000008      ALU  ITSOTS4  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000009      ALU  ITSOTS5  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000010      ALU  ITSOTS6  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000011      ALU  ITSOTS7  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000012      ALU  ITSOTS8  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))
000013      ALU  ITSOTS9  TSO (SIZE (4096) ACCTNUM (1234) PROC (TSOPROC))

Press PF3, enter R at the cursor location, press ENTER to run these commands

```

Figure B-12 RACF commands generated from our CARLa program

In Figure B-13 on page 443, the RESULTS panel is displayed. The cursor is automatically placed onto the CKRCMD field, which is the file name containing the RACF commands. Notice that, at the top right hand corner of the RESULTS panel, a short message is displayed to prompt us to enter the R action character to execute the commands.

When generating any report from the zSecure ISPF interface, you can issue the RESULTS command after the report has been generated. This is extremely useful should you need to:

- Extract the CARLa that was used to generate the report (select the COMMANDS field for this).
- Review the SYSPRINT.

- Write your generated report or commands to a sequential or partitioned data set.
- Email your report to one or more email addresses.

If a command from CKRCMD has failed during execution, this is indicated by the message `command failed` appearing in the top right corner of your panel. You can scroll to the bottom, using the primary ISPF command `BOTTOM` or alternatively entering `M` (for maximum) followed by pressing `F8`, of your output data set. This will show a panel that contains a reference to all record numbers that contained commands that failed during execution.

Menu	Options	Info	Commands	Setup
zSecure Admin+Audit for RACF 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	W	Write file into seq. or partitioned data set	
M	E-mail report			
Enter a selection in front of a highlighted line below:				
—	SYSPRINT	messages		
—	REPORT	printable reports		
—	CKRTSPRT	output from the last TSO command(s)		
<u>R</u>	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 B-13 RESULTS panel to execute the RACF commands

Having executed the RACF commands, an overview of the command output is displayed. The purpose of this overview (as shown in Figure B-14) is to advise you whether or not the commands have been successfully executed.

```

Menu Utilities Compilers Help

BROWSE      SYS08101.T121649.RA000.JPEASE.R0139372      Line 00000000 Col 001 080
Command ==> _____ Scroll ==> PAGE
***** Top of Data *****
=====
=== Multiple TSO command output file - scroll max down for overview ===
=== Input data set JPEASE.C2R134D.CKRCMD                               ===
=====
/* CKRCMD file CKR1CMD complex SC76 generated 10 Apr 2008 12:26 */

===== 10Apr08 12:26:20.08595 start record 2 =====
ALU  ITSOTSA  TSO(SIZE(4096) ACCTNUM(1234) PROC(TSOPROC))

===== 10Apr08 12:26:20.11725 start record 3 =====
ALU  ITSOTSB  TSO(SIZE(4096) ACCTNUM(1234) PROC(TSOPROC))

===== 10Apr08 12:26:20.14687 start record 4 =====
ALU  ITSOTSC  TSO(SIZE(4096) ACCTNUM(1234) PROC(TSOPROC))

===== 10Apr08 12:26:20.17777 start record 5 =====
ALU  ITSOTS1  TSO(SIZE(4096) ACCTNUM(1234) PROC(TSOPROC))

===== 10Apr08 12:26:20.21156 start record 6 =====

```

Figure B-14 Overview of RACF commands issued

Examples of basic CARLa programs for SMF reporting

So far, we have produced RACF related reports using TYPE=RACF in our CARLa programs. In the next example, we are going to produce an SMF related report using TYPE=SMF.

Before you can generate SMF related reports, you must ensure that you have allocated an appropriate SMF file (active or dumped SMF records) as input, using the SETUP FILES command or option SE.1. For the purpose of this example, we have allocated the live SMF data sets, as shown in Figure B-15. This is so that we can generate a report using the most recent SMF records.

```
Menu  Options  Info  Commands
-----
                                zSecure Admin+Audit for RACF - Setup - I Row 1 from 8
Command ==> _____ Scroll ==> CSR

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

Description                                Complex
- Active backup RACF data base and live SMF data sets    SC76      selected
- Active primary RACF data base                          SC76
- test daily signature gdg input file                    SC76
- test daily ckfreeze gdg input file                     SC76
- test daily unload gdg input file                       SC76
- Active backup RACF data base                           SC76
- CT CKFREEZE                                             SC76
- CT UNLOAD                                              SC76
***** Bottom of data *****
```

Figure B-15 Selecting SMF data sets as input

Having allocated an SMF file, we have written a CARLa program (as shown in Example B-3) to produce a simple SMF report to list RACF commands issued by users with RACF System SPECIAL.

Example: B-3 CARLa program reports on RACF commands issued

```
newlist type=SMF title="RACF commands issued by System Special or Group
Special users"
  select racfauth=SPECIAL event=ALLCOMMANDS
  sortlist date time user racfcmd(hor,wrap,0)
```

Before reading on, refer to Table B-3, where we describe the statements used in this program:

Table B-3 Description of CARLa statements used in Figure B-16 on page 447

CARLa statement	Description
NEWLIST	Marks the beginning of a new report.
TYPE=	<p>Select the type of information to be used as an input for the NEWLIST. We use TYPE=SMF in our example to extract SMF records.</p> <p>There are many other types of information you can use as input for the NEWLIST. Use the FIELDS command and select the BUILTIN field to display types.</p>
TITLE=	Allows you to insert a title heading for your report.
SELECT	This command is used to select records that match the conditions specified in the select statement.
RACFAUTH=	This command is used to select records by the RACF authority used for executing commands or accessing resources, for example, RACF SPECIAL or OPERATIONS.
EVENT=	The RACF event code. For example ACCESS, ADDUSER, ALTUSER, DELUSER, PERMIT, RACINIT, or ALLCOMMANDS.
DATE	The date the command was executed.
TIME	The time the command was executed.
USER	The user ID that issued the command.
RACFCMD	The RACF command that was executed.
HOR, WRAP, and 0	HOR, which is an output modifier, lists a repeat group on a single line, which can be truncated when the end of a line is crossed. Use the WRAP output modifier to continue the RACF command on the next line, which prevents truncation. The value 0 is a length value to strip off all trailing blanks.

In Figure B-16, we show the output from the CARLa program we have just run.

```

BROWSE - JPEASE.C2R134D.REPORT ----- LINE 00000000 COL 001 080
COMMAND ==> _                                SCROLL ==> CSR
***** Top of Data *****
S M F   R E C O R D   L I S T I N G   10Apr08 12:26 to 10Apr08 12:26
RACF commands issued by System Special users

User      Date      Time  RACF cmd
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTSA TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTSB TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTSC TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS1 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS2 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS3 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS4 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS5 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS6 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS7 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS8 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
JPEASE    10 Apr 2008 12:26 ALTUSER ITSOTS9 TSO (ACCTNUM (1234)) TSO (PROC (TSOPROC))
***** Bottom of Data *****

```

Figure B-16 Report of RACF commands issued

To conclude this section on basic CARLa reporting, we show you how to extract SMF records based on SMF record types. zSecure Audit supports many SMF record types, which are documented in Chapter 12, “Supported record types”, in *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773. Some of the SELECT statements you specify within your CARLa program will default to the appropriate SMF record type, such as the program listed in Example B-3 on page 445. The default SMF record type selected in this example is 80.

In the next example, we are going to produce a report on failed access to RACF resources, by selecting SMF record types 80, 81 and 83, as shown in Example B-4.

Example: B-4 CARLa program reports on failed access

```

suppress CKFREEZE

newlist type=SMF title="RACF events - Failures"
  select type=(80,81,83) event=ACCESS(FAILURES)
  sortlist date(5) time user class resource intent access

```

Before reading on, refer to Table B-4, where we describe the additional statements used in this program.

Table B-4 Description of CARLa statements used in Figure B-17 on page 449

CARLa statement	Description
SUPPRESS	Allows you to suppress verification failure messages for specific volumes or catalogs, suppress generation of commands, or suppress use of data sources, such as the CKFREEZE file. You should refer to the <i>IBM zSecure Admin and Audit User Reference Manual</i> , LC23-6555 for further guidance on the SUPPRESS statement.
TYPE= (applies to use within the SELECT statement)	SMF record type, for example 80, 81 or 83 for RACF related records.
EVENT=	The RACF event code. For example ACCESS, ADDUSER, ALTUSER, DELUSER, and PERMIT.
CLASS	The affected RACF class.
RESOURCE	The affected resource protected by RACF. If you want to see the profile that was used during access checking, specify the PROFILE field in your SORTLIST statement.
INTENT	The access intent of the user.
ACCESS	The access allowed.

Figure B-17 shows the output that was generated by the CARLa program.

```

BROWSE - JPEASE.C2R334D.REPORT ----- LINE 0000 0.1 s CPU, RC=4
COMMAND ==> _                                SCROLL ==> PAGE
***** Top of Data *****
S M F   R E C O R D   L I S T I N G   19Apr08 17:43 to 19Apr08 17:49
RACF events - Failures

Date Time User      Class      Resource                                Intent  Allowed
19Apr 17:43 JPEASE   DATASET  LILIXIE.TEST.CARLA                    READ    NONE
19Apr 17:43 JPEASE   DATASET  LILIXIE.TEST.CNTL                    READ    NONE
19Apr 17:43 JPEASE   XFACILIT C4R.USER.ID.JAMIE              UPDATE  NONE
19Apr 17:46 JPEASE   OPERCMDS MVS.MODIFY.STC.C2POLICE.C2POLICE  UPDATE  NONE
19Apr 17:46 JPEASE   OPERCMDS MVS.MODIFY.STC.C2POLICE.C2POLICE  UPDATE  NONE
19Apr 17:47 JPEASE   OPERCMDS MVS.STOP.STC.C2POLICE.C2POLICE  UPDATE  NONE
19Apr 17:47 JPEASE   OPERCMDS MVS.STOP.STC.C2POLICE.C2POLICE  UPDATE  NONE
19Apr 17:48 JPEASE   OPERCMDS MVS.STOP.STC.C2POLICE.C2POLICE  UPDATE  NONE
19Apr 17:49 JPEASE   OPERCMDS MVS.STOP.STC.C2POLICE.C2POLICE  UPDATE  NONE
***** Bottom of Data *****

```

Figure B-17 Report of failed accesses

By default, CARLa will add titles and headings to your report unless you either suppress them with the NOPAGE statement or add output modifiers. Output modifiers override defaults, such as the length of field and column header. The person who receives your report may not understand some of the field names, so you can change column headings to make them more meaningful. For example, the report shown in Figure B-17 contains a field called “Resource”. You could change the column heading to “RACF Resource”, as shown in Example B-5.

We have also added a statement called sum (or summary). This statement enables you to create summary reports of the selected data. In this case, we have chosen to summarize the field “user”. This will result in events being summarized by user, as shown in Figure B-18 on page 450.

Example: B-5 CARLa program with output modifiers and summary statement

```
suppress CKFREEZE
```

```

newlist type=SMF title="RACF events - Failures"
  select type=(80,81,83) event=ACCESS(FAILURES)
  sortlist date(5) time user class resource("RACF Resource") ,
    intent access
  sum user

```

Figure B-18 shows the output from the CARLa program, which is summarized by user and includes an output modifier for the resource field.

```

BROWSE - JPEASE.C2R334D.REPORT ----- LINE 0000 0.2 s CPU, RC=4
COMMAND ==> _ SCROLL ==> PAGE
***** Top of Data *****
S M F R E C O R D L I S T I N G 18Apr08 14:51 to 18Apr08 16:34
RACF events - Failures

User      Count      Date   Time   User      Class      RACF Resource
JPEASE          4
    18Apr 16:03 JPEASE    XFACILIT C4R.USER.ID.JAMIE
    18Apr 16:03 JPEASE    XFACILIT C4R.USER.ID.SP005
    18Apr 16:05 JPEASE    XFACILIT C4R.USER.ID.SP100
    18Apr 16:06 JPEASE    XFACILIT C4R.USER.ID.XP100

LILIXIE        11
    18Apr 14:51 LILIXIE   XFACILIT C4R.USER.ID.SP002
    18Apr 14:55 LILIXIE   XFACILIT C4R.USER.ID.SP002
    18Apr 14:59 LILIXIE   XFACILIT C4R.USER.ID.XP001
    18Apr 15:25 LILIXIE   XFACILIT C4R.USER.ID.XP001
    18Apr 15:25 LILIXIE   XFACILIT C4R.USER.ID.SP003
    18Apr 15:27 LILIXIE   XFACILIT C4R.USER.ID.SP003
    18Apr 15:42 LILIXIE   DATASET  JPEASE.CNTL
    18Apr 15:55 LILIXIE   XFACILIT C4R.USER.ID.XP003
    18Apr 16:09 LILIXIE   XFACILIT C4R.USER.ID.XP003
    18Apr 16:34 LILIXIE   XFACILIT C4R.USER.PASSWORD.=USERID
    18Apr 16:34 LILIXIE   XFACILIT C4R.USER.PASSWORD.=DFLTGRP

```

Figure B-18 Report of failed accesses summarized by user

Where to store your CARLa programs for reuse

Both you and your colleagues may collect a number of CARLa programs over a period of time, which you will want to share and reuse again and again. You can store CARLa programs centrally and make them easily accessible from IBM zSecure ISPF option CO, so that any zSecure Admin or zSecure Audit user can run them.

To do this, you need to:

1. Allocate a partitioned data set. The suggested record format is FB with a record length of 80.
2. Select option SE.D.8 from the IBM zSecure main ISPF menu.
3. Insert your new partitioned data set using the “I” action character to insert an empty line.

4. At this stage you need to decide whether or not to set this command file as everyone's default for use in option CO. In either case, they can set their own default in option CO.1 or in option SE.8, where they can also specify their own command files.
5. Having selected a default command file, press F3 to save the change. You will then be prompted to specify whether the change you just made will be copied to all users as soon as they re-enter zSecure Admin or zSecure Audit. In most cases, you would reply Y (Yes).
6. Select option CO.1. The new command library should appear in your list as a selectable library. When you use option CO.2, you can select a CARLa program from your command library, which you can edit if you need to do so, and then run it. You can also run or submit CARLa programs from options CO.4 and CO.5. When writing new CARLa in option CO.C or through the CARLa command, you can save your CARLa in the current allocated command library.

You now have a central repository for your CARLa programs. We recommend you add a \$INDEX member to your new command library so that you have a documented central index of programs and their purpose.

Useful primary commands

We have documented in Table B-5 some useful primary commands to use when working with CARLa. These commands are available anywhere in your zSecure Admin and zSecure Audit ISPF session. Press F1 on the main menu, followed by option P, to see additional commands.

Table B-5 Useful primary commands

Command	Description
CARLa or COMM	Type of CARLa commands.
FIELDS	List fields for all newlist types (for example, NEWLIST TYPE=RACF). Tag the BUILTIN option to display fields for each TYPE. Issue the FIND command to find the TYPE you are interested in, for example, FIND RACF. You then need to select the type to display the fields.
MESSAGE or MSG	Show the zSecure Admin or zSecure Audit message if one is supplied when running your CARLa program, for example, MSG 0664.
RESULT(S)	Work with the result(s) of your last query.

Command	Description
SETUP	Go to the SETUP menu. For example, use SETUP FILES to go directly to the setup input file panel.
SYSPRINT	See the query output, although we recommend you use the RESULTS panel.

Additional examples of CARLa

We have provided some additional CARLa programs in this section to give you further insight into the capabilities of the language. These particular examples are RACF related CARLa programs.

The CARLa program shown in Example B-6 selects user IDs with a mask of TRNG and generates RACF commands to connect the selected user IDs to RACF group TRN009. Notice that we have abbreviated some of the statements, such as s for select and c for class. For a full list of abbreviations, refer to Chapter 11, “Command Language”, in the *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773.

Example: B-6 CARLa program connects multiple user IDs to a group

```
newlist type=RACF f=CKRCMD nopage
  s c=USER mask=TRNG* s=BASE
  sortlist "connect" key(8) "group(TRN009) owner(TRN009)"
```

Example B-7 shows the same CARLa program; however, we have used the exclude statement to prevent connect commands for those users that are already connected to the group.

Example: B-7 Use of exclude statement to exclude users already connected

```
newlist type=RACF f=CKRCMD nopage
  s c=USER mask=TRNG* s=BASE
  exclude cggrpnm=TRN009
  sortlist "connect" key(8) "group(TRN009) owner(TRN009)"
```

The CARLa program shown in Example B-8 on page 453 generates a report to highlight where a user’s default group and owner do not match. Note that the duplicated operators (<<>>) check contents against contents, while normal operators compare contents against a literal value.

We could optionally fix a mismatch in the default group and owner with a RACF command in our sortlist.

Example: B-8 CARLa program to report on default group and owner mismatch

```
newlist type=RACF title="Default group and owner mismatch report",
          subtitle="Default group and owner must match!"
  select class=USER segment=BASE dfltgrp<<>>owner
  exclude dfltgrp=" "
  exclude key=IBMUSER
  sortlist key("User ID",8) name(20) dfltgrp owner
```

The CARLa program shown in Example B-9 will generate RACF commands to REVOKE user IDs that have not been used in the last 100 days. The program will also connect these user IDs to a group called DELETE and will alter their default group and owner to DELETE. We have excluded some user IDs from the process using the exclude statement.

Example: B-9 CARLa program to generate commands to REVOKE unused user IDs

```
newlist type=RACF f=CKRCMD nopage
  s c=USER s=BASE mask=* last_connect_date<TODAY-100

/* Exceptions to be excluded */

  x dfltgrp=' '
  x key=IBMUSER
  x protected

/* Generate commands */

  sortlist "altuser" key(0) "revoke"
  sortlist "altuser" key(0) "data('Revoked due to non-use')"
  sortlist "connect" key(0) "group(delete) owner(delete)"
  sortlist "altuser" key(0) "dfltgrp(delete) owner(delete)"
```

In our final example, we demonstrate how you can verify your current security configuration against a security baseline using CARLa. The CARLa program shown in Example B-10 on page 454 will check that only a known set of users have RACF system SPECIAL. If a new security administrator joins the RACF team and we have been authorized to assign RACF system SPECIAL to them, we update our baseline by adding an additional exclude statement for the user in the CARLa program, for example, exclude key=MCAIRNS.

Example: B-10 Check that only a known set of users have RACF system special

```
newlist type=RACF title="Unexpected administrators",  
empty="All Special attributes are compliant"  
  select class=USER special  
  exclude key=IBMUSER  
  exclude key=JPEASE  
  sortlist key(8) name last_connect_date
```

If no additional users with SPECIAL are found when we run this CARLa program, the report will simply show “All Special attributes are compliant”.

Conclusion

Many of the reports you require are already shipped in zSecure Admin and zSecure Audit. The SCKRSAMP library is also packed with samples of CARLa programs. You can invoke this library by using option CO.1 to allocate the SCKRSAMP library, followed by option CO.2 to view the members. The member names of those samples and their purpose are documented in Chapter 10, “IBM zSecure CARLa library”, in *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773.

There may be occasions when you need to produce a report that requires additional information, requires reformatting, or maybe you want to do bulk updates. You will rarely find a need to write a CARLa program from scratch. If you follow one of the four steps at the beginning of “Basic CARLa to get you started” on page 432, you should be able to find a CARLa program to get you started.



User roles for IBM Security zSecure Visual

This appendix contains RACF commands that you can use to create appropriate user roles, as documented in Chapter 3, “Configuring client authorities”, in the *IBM Tivoli zSecure Visual Server Manual Version 1.11*, SC23-6549.

These commands are not intended to be executed as provided, and serve merely as a guideline and quick start to getting your users set up in zSecure Visual. Modify these commands to suit your installation standards and intended use of the product.

RACF commands to generate zSecure Visual user roles

Refer to Chapter 6, “IBM Security zSecure Visual” on page 111 for descriptions of the roles and authorities implemented by these commands. These were the initial commands we used to create user roles on our test system. These were subsequently enhanced by adding scoping profiles.

RACF commands sample

```
/* ROLE      USER      GROUP      */
/* ===== */
/* FULL  = TEST1 = CKG1FLL */
/* GROUP = TEST2 = CKG2GRP */
/* USER  = TEST3 = CKG3USR */
/* ACL    = TEST4 = CKG4ACL */
/* CONN   = TEST5 = CKG5CON */
/* HELP   = TEST6 = CKG6HLP */

/* only 'super' admins should have access to these backstop profiles */

RDEF XFACI CKG.CMD.CMD.**                OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.*            OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.**                    OWNER(SYS1) UACC(N)
RDEF XFACI CKG.RAC.**                    OWNER(SYS1) UACC(N)
RDEF XFACI CKG.SCP.ID.*.SYS1.*           OWNER(SYS1) UACC(N)
RDEF XFACI CKG.**                        OWNER(SYS1) UACC(N)

RALT XFACI CKG.CMD.CMD.**                OWNER(SYS1) UACC(N)
RALT XFACI CKG.CMD.USER.REQ.*            OWNER(SYS1) UACC(N)
RALT XFACI CKG.CMD.**                    OWNER(SYS1) UACC(N)
RALT XFACI CKG.RAC.**                    OWNER(SYS1) UACC(N)
RALT XFACI CKG.SCP.ID.*.SYS1.*           OWNER(SYS1) UACC(N)
RALT XFACI CKG.**                        OWNER(SYS1) UACC(N)

RDEF XFACI CKG.CMD.CMD.EX.ADDGROUP        OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWSET.DEFAULT OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWSET.EXPIRED OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWSET.NONEXP  OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWSET.PASSWORD OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWSET.PREVIOUS OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.RESUME        OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.SCHEDULE      OWNER(SYS1) UACC(N)
RDEF XFACI CKG.RAC.SCP.CONNECT.BASE.AUTH.USE OWNER(SYS1) UACC(N)
```

RDEF XFACI CKG.CMD.CMD.EX.ADDSD	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.ADDUSER	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.ALTDSD	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.ALTDGROUP	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.ALTUSER	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.DELDSD	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.DELGROUP	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.PERMIT	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.RALTER	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.RDEFINE	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.RDELETE	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.EX.SETROPTS	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.REQ.CONNECT	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.REQ.PERMIT	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.CMD.REQ.REMOVE	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.COMMENT	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.LIST	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.SHOW.MYACCESS	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWDEFAULT	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWNOHIST	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWNORULE	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWRESET	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.CMD.USER.REQ.PWSET	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.RAC.SCP.CONNECT.BASE	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.RAC.SCP.*.BASE.*	OWNER(SYS1) UACC(N)
RDEF XFACI CKG.SCP.ID.**	OWNER(SYS1) UACC(N)
PE CKG.CMD.CMD.EX.ADDGROUP	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.ADDSD	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.ADDUSER	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.ALTDSD	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.ALTDGROUP	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.ALTUSER	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.DELDSD	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.DELGROUP	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.PERMIT	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.RALTER	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.RDEFINE	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.RDELETE	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.EX.SETROPTS	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.REQ.CONNECT	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.REQ.PERMIT	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.CMD.REQ.REMOVE	CLASS(XFACI) ID(CKG1FLL) AC(U)
PE CKG.CMD.COMMENT	CLASS(XFACI) ID(CKG1FLL) AC(R)
PE CKG.CMD.LIST	CLASS(XFACI) ID(CKG1FLL) AC(R)

PE	CKG.CMD.SHOW.MYACCESS	CLASS(XFACI)	ID(CKG1FLL)	AC(R)
PE	CKG.CMD.USER.REQ.PWDEFAULT	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.PWNOHIST	CLASS(XFACI)	ID(CKG1FLL)	AC(R)
PE	CKG.CMD.USER.REQ.PWNORULE	CLASS(XFACI)	ID(CKG1FLL)	AC(R)
PE	CKG.CMD.USER.REQ.PWRESET	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET	CLASS(XFACI)	ID(CKG1FLL)	AC(R)
PE	CKG.CMD.USER.REQ.PWSET.DEFAULT	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.EXPIRED	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.NONEXP	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PASSWORD	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PREVIOUS	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.RESUME	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.USER.REQ.SCHEDULE	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.USE	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.*	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.RAC.SCP.*.BASE.*	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.SCP.ID.**	CLASS(XFACI)	ID(CKG1FLL)	AC(U)
PE	CKG.CMD.CMD.EX.ADDGROUP	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.ADDSD	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.ADDUSER	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.ALTDSD	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.ALTRGROUP	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.ALTUSER	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.DELDSD	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.DELGROUP	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.PERMIT	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.RALTER	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.RDEFINE	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.RDELETE	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.SETROPTS	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.REQ.CONNECT	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.REQ.PERMIT	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.REQ.REMOVE	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.COMMENT	CLASS(XFACI)	ID(CKG2GRP)	AC(R)
PE	CKG.CMD.LIST	CLASS(XFACI)	ID(CKG2GRP)	AC(R)
PE	CKG.CMD.SHOW.MYACCESS	CLASS(XFACI)	ID(CKG2GRP)	AC(R)
PE	CKG.CMD.USER.REQ.PWDEFAULT	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.USER.REQ.PWRESET	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET	CLASS(XFACI)	ID(CKG2GRP)	AC(R)
PE	CKG.CMD.USER.REQ.PWSET.DEFAULT	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.EXPIRED	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.NONEXP	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PASSWORD	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PREVIOUS	CLASS(XFACI)	ID(CKG2GRP)	AC(U)

PE	CKG.CMD.USER.REQ.RESUME	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.USER.REQ.SCHEDULE	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.USE	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.*	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.RAC.SCP.*.BASE.*	CLASS(XFACI)	ID(CKG2GRP)	AC(U)
PE	CKG.CMD.CMD.EX.ADDSD	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.ADDUSER	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.ALTDSD	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.ALTUSER	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.PERMIT	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.RALTER	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.RDEFINE	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.SETROPTS	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.REQ.CONNECT	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.REQ.PERMIT	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.REQ.REMOVE	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.COMMENT	CLASS(XFACI)	ID(CKG3USR)	AC(R)
PE	CKG.CMD.LIST	CLASS(XFACI)	ID(CKG3USR)	AC(R)
PE	CKG.CMD.SHOW.MYACCESS	CLASS(XFACI)	ID(CKG3USR)	AC(R)
PE	CKG.CMD.USER.REQ.PWDEFAULT	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.PWRESET	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET	CLASS(XFACI)	ID(CKG3USR)	AC(R)
PE	CKG.CMD.USER.REQ.PWSET.DEFAULT	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.EXPIRED	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.NONEXP	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PASSWORD	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PREVIOUS	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.RESUME	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.USER.REQ.SCHEDULE	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.USE	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.*	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.RAC.SCP.*.BASE.*	CLASS(XFACI)	ID(CKG3USR)	AC(U)
PE	CKG.CMD.CMD.EX.ADDSD	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.EX.ADDUSER	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.EX.ALTDSD	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.EX.ALTUSER	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.EX.PERMIT	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.EX.RALTER	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.EX.RDEFINE	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.EX.SETROPTS	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.REQ.CONNECT	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.REQ.REMOVE	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.COMMENT	CLASS(XFACI)	ID(CKG4ACL)	AC(R)

PE	CKG.CMD.LIST	CLASS(XFACI)	ID(CKG4ACL)	AC(R)
PE	CKG.CMD.SHOW.MYACCESS	CLASS(XFACI)	ID(CKG4ACL)	AC(R)
PE	CKG.CMD.USER.REQ.PWDEFAULT	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.PWRESET	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET	CLASS(XFACI)	ID(CKG4ACL)	AC(R)
PE	CKG.CMD.USER.REQ.PWSET.DEFAULT	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.EXPIRED	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.NONEXP	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PASSWORD	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET.PREVIOUS	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.RESUME	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.USER.REQ.SCHEDULE	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.USE	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.*	CLASS(XFACI)	ID(CKG4ACL)	AC(U)
PE	CKG.CMD.CMD.REQ.CONNECT	CLASS(XFACI)	ID(CKG5CON)	AC(U)
PE	CKG.CMD.CMD.REQ.REMOVE	CLASS(XFACI)	ID(CKG5CON)	AC(U)
PE	CKG.CMD.COMMENT	CLASS(XFACI)	ID(CKG5CON)	AC(R)
PE	CKG.CMD.LIST	CLASS(XFACI)	ID(CKG5CON)	AC(R)
PE	CKG.CMD.SHOW.MYACCESS	CLASS(XFACI)	ID(CKG5CON)	AC(R)
PE	CKG.CMD.USER.REQ.PWRESET	CLASS(XFACI)	ID(CKG5CON)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET	CLASS(XFACI)	ID(CKG5CON)	AC(R)
PE	CKG.CMD.USER.REQ.RESUME	CLASS(XFACI)	ID(CKG5CON)	AC(U)
PE	CKG.CMD.USER.REQ.SCHEDULE	CLASS(XFACI)	ID(CKG5CON)	AC(U)
PE	CKG.RAC.SCP.CONNECT.BASE.AUTH.USE	CLASS(XFACI)	ID(CKG5CON)	AC(U)
PE	CKG.CMD.COMMENT	CLASS(XFACI)	ID(CKG6HLP)	AC(R)
PE	CKG.CMD.LIST	CLASS(XFACI)	ID(CKG6HLP)	AC(R)
PE	CKG.CMD.SHOW.MYACCESS	CLASS(XFACI)	ID(CKG6HLP)	AC(R)
PE	CKG.CMD.USER.REQ.PWRESET	CLASS(XFACI)	ID(CKG6HLP)	AC(U)
PE	CKG.CMD.USER.REQ.PWSET	CLASS(XFACI)	ID(CKG6HLP)	AC(R)
PE	CKG.CMD.USER.REQ.RESUME	CLASS(XFACI)	ID(CKG6HLP)	AC(R)
PE	CKG.CMD.USER.REQ.SCHEDULE	CLASS(XFACI)	ID(CKG6HLP)	AC(U)



D

A look at the Consul to IBM Tivoli transformation

This appendix contains information for those users of the zSecure software who are running versions prior to the acquisition of Consul by IBM. This information is provided to assist you in the process of migrating to a current, IBM supported, version of the equivalent products.

Also in this appendix, we wanted to provide some historical perspective detailing some of the history of the Consul company and the transformation that has occurred with the acquisition by IBM. To this effect, this appendix contains some personal reflections by two of the authors who have had a long standing association with the software as users and consultants, prior to the IBM acquisition of Consul in 2007.

Information for users migrating from previous releases

This section discusses some issues you must address when migrating from older Consul versions of the zSecure products to the recent IBM releases. IBM engineered releases began with Version 1.8.1. Those first IBM versions were branded as IBM Tivoli zSecure, and with Version 1.12 of the solution, the branding was changed to IBM Security zSecure.

Program name changes

At Version 1.6 of zSecure, several program modules were combined into one program, C2RCARLA. In order to ease this migration and avoid the need to update any batch jobs that used the old program names, program aliases could be defined. These were known as “backward compatible aliases”. Most users took the approach of using these aliases rather than updating any previously generated JCL.

From Version 1.8.0 to the first IBM supported version, Version 1.8.1, the C2RCARLA program name changed again, to CKRCARLA. For this conversion, the sample job CKRZALIA has been provided should you want to avoid or minimize JCL changes.

For any users migrating from a version earlier than V1.6, we recommend that you update all batch invocations of zSecure, as significant changes in zSecure JCL have occurred. We suggest that rather than attempting to manually change any JCL you may have generated previously, it may be simpler to regenerate the appropriate batch jobs using the zSecure ISPF interface, then use this JCL to replace any you already have.

In general, basic CARLa processing has remained backward compatible with each release. If you encounter problems migrating any CARLa programs, check the “Summary of amendments” section near the beginning of each product manual, as you may be using an option or keyword whose operation has changed slightly.

RACF resource class changes

From Version 1.8.0 to Version 1.8.1, the general resource class used for all zSecure related RACF profiles has changed from FACILITY to XFACILIT. Additionally, profile names changed, dropping the dollar '\$' prefix recommended for non IBM defined profiles and in some cases changing the three character profile prefix used.

A sample batch job CKRJCONV has been provided to assist in migrating any definitions you made for previous releases into the new naming standards.

It is also worth noting the default return code for the XFACILIT class is 8, while FACILITY is 4. Unless you have backstop or catchall profiles for the resources CKF.**, CKG.**, C2R.**, and C2X.**, you may experience access failures that would not have occurred when using class FACILITY.

Consul and zSecure history

The first product of the zSecure family was built by Consul Risk Management and started shipping in 1990. It was Consul/RACF, the RACF administration and reporting product that is today known as zSecure Admin. In 1993, it was followed by an SMF reporting and MVS auditing tool called Consul/Audit. Consul/RACF and Consul/Audit shared their programming language, so a new name was found: Consul Auditing and Reporting Language, or CARLa.

The software quickly became popular, easily gaining support from local Netherlands based mainframe users. There was simply nothing else around at this time to assist the RACF security administrator in performing their difficult and complex role. Some companies had even invested in home grown utilities at great expense to perform similar functions. Back then, this was not uncommon, but the software landscape was changing. Staff costs were starting to become a more significant portion of total IT expenditure, and the effort and expense of developing in-house software was starting to become prohibitive.

Realizing this, the now growing Consul development team turned their attention to developing an auditing solution for the Novell and OS/2 servers that were quickly requiring attention from data center managers. The resulting product was subsequently called Consul/Enterprise Audit, eAudit, InSight, and now IBM Tivoli Security Information and Event Manager.

Early in the development of what we now know as zSecure, a challenging architectural decision was taken. Rather than develop a product that directly addressed each particular issue typically encountered across a range of RACF installations, a solution was needed that was flexible enough to extend to provide a viable solution to potentially any possible RACF infrastructure.

To address this, the programming language we now know as CARLa was created. The architectural decision to write a specific programming language, and all associated compilation and parsing processes required for such, was simply visionary in its scope. This language had to be designed to process the kinds of information contained in the RACF database, and extensible to process other sources of security information available in the mainframe environment. This became the core of the entire zSecure product suite.

CARLa became the engine for the rapid development of further products to enhance z/OS security overall, such as zSecure Alert, Visual, and the Security Information and Event Manager.

These software products have a long history of fully satisfying diverse customer needs. The design of the basic solutions is now well over 20 years old, a long time in the life of any software product. The software has been continuously enhanced to match new RACF or z/OS features and the changing landscape of IT security in general. They have been proven again and again in real world use cases to be best of breed solutions for customer problems. This is a set of well proven products that can satisfy the requirements of even the most unusual customer configurations and needs. We hope you enjoy using them as much as we have enjoyed sharing our knowledge of them during the process of writing this book.

Note: It is not necessary to learn the CARLa programming language to fully exploit all provided product features and functions. CARLa is available if a user requires it to provide specialized processing to meet unique needs found in their organization. This built-in extensibility is a major strength of the zSecure suite.

A personal note from Jamie Pease

I have worked with the zSecure software since 1997, mainly as a customer in Financial Services. I have used the software for RACF management, monitoring, and auditing for both RACF and CA ACF2 security systems. My personal opinion is that the software is a fantastic suite of tools that brings a jigsaw puzzle together.

I have worked on a number of mainframe security initiatives and have managed multiple and complex RACF databases over the last few years. I was fortunate enough to have access to zSecure software throughout this time and I personally believe that the software gave us a big return on investment in addition to improving controls within the various mainframe environments.

From personal experience, zSecure has helped increase my knowledge of z/OS, RACF, and CA ACF2 controls, mainly due to the wealth of information available in the user interfaces and documentation.

As an IBM employee, I visit zSecure clients and they always tell me how excellent the software is and how it makes their job easier. One client who recently acquired zSecure quoted the software as being “fabulous”. For clients that I visit who do not have zSecure, you can see how the software would significantly improve security posture and working practices. I personally could not imagine running mainframe security without zSecure.

What has always impressed me about zSecure is the intelligence built into the software. Some incredible engineering has gone into the software over the years to help you manage mainframe security. The software always seems to be one step ahead and keeps you one step ahead in mainframe security.

A personal note from Mike Cairns

I was delighted to be invited to participate in creating this book. I had been an admirer of the Consul software since the early 1990s when I first encountered their RACF admin product, and I looked forward to the opportunity to introduce this amazing software to a wider audience of RACF professionals whom I hope will read and enjoy this book.

As a systems programmer with a passion for RACF and security, what has always impressed me about this software is the high quality of the engineering behind it. The built in help of zSecure Admin, the knowledge base of zSecure Audit, the core CARLa programming language, and the design thought that has gone into this thoroughly integrated suite of products is simply incredible.

As the Asia Pacific technical person responsible for zSecure from 2005 until 2007, I was thrilled when IBM announced their intention to acquire the Consul company and integrate the zSecure suite more closely with z/OS. This meant that the zSecure developers now had the backing of IBM resources in their R&D efforts to best support RACF and future developments, which could only be good for future product features.

Also, becoming an IBM employee meant that I now had significant resources behind me to help my zSecure users obtain the best value possible from their software investment, both in terms of zSecure exploitation, and more importantly, consequent improvement in their security posture. This IBM Redbooks publication is a practical example of using these IBM resources to help our customers improve security and ease the security management burden of their System z systems.

In my career I have seen many RACF installations that are not taking full advantage of the world class security management system at their disposal, RACF. I think I speak for all the authors in saying that we hope this deliverable can help you improve your exploitation of z/OS RACF, whether you are a current user of zSecure or not, and that your organization benefits from any improvements in z/OS security inspired by our work here.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 468. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Enterprise Multiplatform Auditing*, SG24-7472
- ▶ *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014
- ▶ *IBM Tivoli Security and System z*, REDP-4355
- ▶ *IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager*, SG24-7530

Other publications

The following product publications are relevant as further information sources. They can be obtained at the IBM Tivoli Information Center location:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp>

All zSecure documents are located under the product category *zSecure for z/OS* and *zSecure for z/VM* in the left hand pane.

- ▶ *IBM Security zSecure Admin and Audit for RACF User Reference Manual Version 1.12*, LC27-2773
- ▶ *IBM Security zSecure Alert User Reference Manual Version 1.12*, SC27-2776
- ▶ *IBM Security zSecure Audit for ACF2 User Reference Manual Version 1.12*, LC27-2774
- ▶ *IBM Security zSecure Audit for RACF Getting Started Guide Version 1.12*, GI11-9355
- ▶ *IBM Security zSecure Audit for Top Secret Version 1.12*, LC27-2775

- ▶ *IBM Security zSecure CICS Toolkit User Guide Version 1.12, SC27-2780*
- ▶ *IBM Security zSecure Command Verifier User Reference Manual Version 1.12, SC27-2779*
- ▶ *IBM Security zSecure Messages Guide Version 1.12, SC27-2783*
- ▶ *IBM Security zSecure Release Information Version 1.12*
- ▶ *IBM Security zSecure Suite: CARLa-Driven Components Installation and Deployment Guide Version 1.12, SC27-2772*
- ▶ *IBM Tivoli zSecure Visual Server Manual Version 1.11, SC23-6549*
- ▶ *IBM zSecure Audit for ACF2 Getting Started Guide Version 1.12, GI11-9356*
- ▶ *IBM zSecure Visual Client Manual Version 1.12, SC27-2778*

Besides the zSecure product information, you should also look into the IBM Tivoli Security Information and Event Manager documentation that can be found under the *Security Information and Event Manager* category on the left hand pane.

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



Redbooks

IBM z/OS Mainframe Security and Audit Management Using the IBM Security zSecure Suite

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



IBM z/OS Mainframe Security and Audit Management Using the IBM Security zSecure Suite



Increase the efficiency of your RACF security management

Address mainframe audit and compliance

Understand all zSecure components

Every organization has a core set of mission-critical data that must be protected. Security lapses and failures are not simply disruptions—they can be catastrophic events, and the consequences can be felt across the entire organization. As a result, security administrators face serious challenges in protecting the company's sensitive data. IT staff are challenged to provide detailed audit and controls documentation at a time when they are already facing increasing demands on their time, due to events such as mergers, reorganizations, and other changes. Many organizations do not have enough experienced mainframe security administrators to meet these objectives, and expanding employee skillsets with low-level mainframe security technologies can be time-consuming.

The IBM Security zSecure suite consists of multiple components designed to help you administer your mainframe security server, monitor for threats, audit usage and configurations, and enforce policy compliance. Administration, provisioning, and management components can significantly reduce administration, contributing to improved productivity, faster response time, and reduced training time needed for new administrators.

This IBM Redbooks publication is a valuable resource for security officers, administrators, and architects who wish to better understand their mainframe security solutions.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks