Tivoli software

IBM

Empowering Security and Compliance Management for the z/OS RACF Environment

Using IBM Tivoli Security Management for z/OS

- Administering your mainframe security while helping to reduce administration time, effort, and costs
- Leveraging seamless integration of audit and compliance efforts
- Increasing mainframe security while decreasing complexity

Axel Buecker
Michael Cairns

Redpaper

ibm.com/redbooks

International Technical Support Organization

Empowering Security and Compliance Management for the z/OS RACF Environment: Using IBM Tivoli Security Management for z/OS

August 2010

REDP-4549-00

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

First Edition (August 2010)

This edition applies to the IBM Tivoli Security Management for z/OS V1.11 offering (product number 5698-B43).

This document created or updated on August 12, 2010.

Contents

Notices v
Trademarks vi

Preface vii
The team who wrote this paper vii
Now you can become a published author, too! viii
Comments welcome. viii
Stay connected to IBM Redbooks viii

Chapter 1. IBM Tivoli Security Management for z/OS 1
1.1 Overview of the solution 2
1.1.1 Audit and security activity reporting 3
1.1.2 Security event alerting 3
1.1.3 Fine grained command control 4
1.1.4 Efficient security administration. 4
1.1.5 Security and audit baseline establishment 5
1.1.6 Automated cleanup of redundant security definitions 5
1.1.7 Segregation of sensitive privileges and authorities. 5
1.1.8 Identification of trusted users 6
1.2 IBM Tivoli Security Management for z/OS components 7
1.2.1 IBM Tivoli zSecure Admin. 7
1.2.2 IBM Tivoli zSecure Audit. 8
1.2.3 IBM Tivoli zSecure Command Verifier 8
1.2.4 IBM Tivoli Security Information and Event Manager. 9
1.3 Tangible benefits and ROI 10
1.3.1 Impact on business drivers 11
1.3.2 Impact on IT operations 12
1.4 Conclusion 13

Chapter 2. Customer scenarios 15
2.1 Satisfy internal and external auditors that the z/OS security environment is being appropriately managed and secured 16
2.1.1 Phase 1 – Deploy Tivoli zSecure Admin and Audit. 16
2.1.2 Phase 2 – Implement zSecure Audit recommended baseline improvements. . . . 19
2.1.3 Phase 3 – Baseline tracking 23
2.1.4 Scenario 1 conclusions. 24
2.2 Provide protection for critical RACF resources from abuse by privileged insiders 24
2.2.1 Phase 1 – Design the new structure, roles, and workflow 25
2.2.2 Phase 2 – Implement and test the segregation capabilities 28
2.2.3 Scenario 2 conclusions. 28
2.3 Demonstrate audit readiness and policy-based management of security access rights 29
2.3.1 Phase 1 – Information discovery. 29
2.3.2 Phase 2 – Installation and configuration. 30
2.3.3 Phase 3 – Closed loop auditing with RACF 33
2.3.4 Senario 3 conclusions. 34
2.4 Conclusion 35

Related publications 37
IBM Redbooks 37

Online resources 37
How to get Redbooks. 37
Help from IBM 38

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

Preface

Every organization has a core set of mission-critical data that requires protection. Security lapses and failures are not simply disruptions, they can be catastrophic events with consequences felt across the enterprise. The inadvertent mistakes of privileged users alone can result in millions of dollars in damages through unintentional configuration errors and careless security commands. Malicious users with authorized access can cause even greater damage. As a result, security management faces a serious challenge to adequately protect a company's sensitive data. Likewise, IT staff is challenged to provide detailed audit and controls documentation in the face of increasing demands on their time.

Automation and simplification of security and compliance processes can help you meet these challenges and establish effective, sustainable user administration and audit solutions. This includes security database cleanup, repeatable audit of configurations and settings, and active monitoring of changes and events. IBM Tivoli Security Management for z/OS V1.11 provides these solutions to help enhance the security of mainframe systems through automated audit and administration.

In this IBM® Redpaper™ document we discuss how Tivoli® Security Management for z/OS® allows you to submit mainframe security information from z/OS, RACF®, and DB2® into an enterprise audit and compliance solution and how to combine mainframe data from z/OS, RACF, and DB2 with that from other operating systems, applications, and databases in order to provide the ability to capture comprehensive log data, interpret that data through sophisticated log analysis, and communicate results in an efficient, streamlined manner for full enterprise-wide audit and compliance reporting.

The team who wrote this paper

This paper 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 23 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 z/OS security, particularly the 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.

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

Alison Chandler
International Technical Support Organization, Poughkeepsie Center

Glinda Cummings, Rob Weiss
IBM

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 papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks® in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@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

1

IBM Tivoli Security Management for z/OS

In this chapter we present an overview of the IBM Tivoli Security Management for z/OS solution and briefly introduce the individual components that are included in this offering. Next we take a look at some of the tangible benefits and ROI statements that this solution can help you achieve.

In order to demonstrate the cohesiveness of the individual technical solutions contained in this offering we use three distinct customer scenarios in Chapter 2, “Customer scenarios” on page 15.

1.1 Overview of the solution

IBM Tivoli Security Management for z/OS V1.11 presents the new face of the mainframe, using a browser-based auditing interface, automated reporting and alerting in XML, and intuitive user interfaces. It can help organizations meet the increasing challenges of z/OS-based IT security with greater efficiency, allowing more resources to be directed to actual security improvement projects. At the same time, it provides a work plan for z/OS administrators to follow in order to improve their systems IT security profile.

For organizations that use an IBM System z®, IT security is usually a high priority overall, deserving significant budget allocation from the IT department's total costs. However, too often, the mainframe is neglected in this security budget; this is a typical case of *the squeaky wheel getting the grease*. As security vulnerabilities for distributed systems are revealed on a daily, weekly, and monthly basis, these always tend to soak up the greatest portion of the IT security budget. There is also a general perception that the mainframe is *secure by design*, which might be true to some extent. Without a doubt, the z/OS environment combined with current System z hardware can provide the most "secure-able" commercial computing system available on the planet. However, in many commercial installations it has been shown that significant security exposures do exist, often unknown and un-mitigated. This is the reason for the existence of the Tivoli Security Management for z/OS solution bundle—to enable less experienced mainframe administrators to leverage the skills and knowledge of their worldwide peer group leaders and properly secure their System z environments with a minimum of fuss, invested time, and risk.

Strip away the modern browser interfaces, and underneath, driving Tivoli Security Management for z/OS is a 20 year plus database of mainframe security configuration best practice, combined with a custom query engine (the *CARLa* programming language) specifically designed to process every kind of security-related data available in the z/OS environment. The effectiveness of z/OS security administrators, auditors, managers, or other authorized staff using these tools, is enhanced by the knowledge of worldwide experts on z/OS security, to the benefit of your organization's overall security posture and risk management compliance requirements.

IBM Tivoli Security Management for z/OS V1.11 provides the following capabilities:

- Facilitates compliance with security requirements and policies
- Leverages seamless integration with an enterprise-wide view of audit and compliance management efforts
- Monitors and audits incidents to help detect and prevent security exposures and to minimize risk
- Automates routine administrative tasks to help reduce costs and complexity, and to help improve productivity and efficiency
- Includes centralized server administration integrity, including virtual servers
- Proactively enforces policy compliance on RACF, which can decrease RACF database pollution by helping to prevent noncompliant security commands
- Helps prevent privileged command abuse and errors by allowing selective distribution of RACF command access, verifying RACF security commands before processing, and retrieving security command information with audit trails

The combination of software provided in the Tivoli Security Management for z/OS solution bundle is an integrated suite, working together to provide comprehensive z/OS security. In the following sections we describe some common z/OS security and management topics that are addressed by components in the suite.

1.1.1 Audit and security activity reporting

Traditional reporting tools for z/OS security can be cumbersome to use and difficult to interpret. Tivoli Security Management for z/OS provides an intuitive ISPF[1]-based interface for both RACF administration and audit reporting as well as a web browser-based interface for audit and compliance reporting and near real-time security event alerting. Should true real-time security event alerting be a requirement, additional IBM Tivoli zSecure Suite components can provide this.

The ISPF-based components of the Tivoli Security Management for z/OS solution bundle provide literally hundreds of supplied audit reports, and dozens of administrative tools required to perform common RACF tasks. In addition, these tools provide deep visibility into z/OS configuration data, normally the domain of the systems programmer and beyond the awareness of most security administrators. Easy access to this information is important because errors in the z/OS configuration are the most common *back door* leading to z/OS security compromises.

The tools provide the capability to compare a current configuration against established industry best practice, and thus provide a road map for less experienced administrators to improve their RACF and z/OS configuration. Also, this kind of comparison can be used as a change tracking function, comparing an accepted *security baseline* with the current security settings, thus effectively ensuring that no security relevant change goes unnoticed or is able to bypass the organization's change control functions.

1.1.2 Security event alerting

Most organizations today run intrusion prevention software (IPS) on our critical distributed platforms, but rarely do we see the same standards employed on z/OS.

Why is this so?

In most cases, organizations reply that they are not aware that IPS or other security event driven reporting functionality is available for z/OS and its subsystems. The Tivoli Security Management for z/OS solution bundle provides these capabilities in the form of the IBM Tivoli Security Information and Event Manager.

With this tool it is possible to audit access to many different system resources, to alert on use of sensitive resources, or access by highly authorized staff, and to compare access patterns against industry regulations and other guidelines. The reports generated by Tivoli Security Information and Event Manager are web browser based, and can be exported into several common formats, for example, PDF, Microsoft® Excel, and so on. Best of all, Tivoli Security Information and Event Manager reports can be run and interpreted by someone without specialized knowledge of the z/OS platform.

Tivoli Security Information and Event Manager is a generalized audit tool, available to report on the audit logs generated by over 300 differing kinds of IT platforms, databases, and applications. Tivoli Security Information and Event Manager can also support processing of custom application logs.

A comprehensive discussion about Tivoli Security Information and Event Manager can be found in the IBM Redbooks publication *IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager*, SG24-7530.

[1] The *Interactive System Productivity Facility* is the traditional mainframe system interface.

1.1.3 Fine grained command control

A shared experience of virtually all system administrators, no matter what the platform, is humorously referred to as the *Oh-No! Moment*. A definition of the Oh-No! Moment is that sinking feeling you get in the pit of your stomach a moment after you pressed the *enter* key on a highly sensitive system command, and at the same time realize that you missed a critical step in the process, and may have just done irreparable harm to a live system.

All experienced IT administrators have, at some time in their career, experienced an Oh-No! Moment. They are an almost unavoidable consequence of the rate of change in IT systems, and the pressure of working in these highly sensitive environments. Tivoli Security Management for z/OS provides functionality to help reduce, perhaps even eliminate, outages, damage caused by, or risks introduced by erroneously entered RACF and security related changes.

The IBM Tivoli zSecure Command Verifier is a RACF enhancement that can provide additional segregation of access to highly sensitive RACF commands. Properly deployed, Tivoli zSecure Command Verifier can help prevent the most common RACF command errors that threaten system availability or functionality. Additionally, the product provides for standardization of RACF processes by enforcing site defined custom naming conventions and other basic RACF configuration standards.

Sometimes, working with RACF controls on a live system is like carrying a loaded gun, very useful when you need one, but also potentially quite dangerous. Tivoli zSecure Command Verifier is there to prevent you shooting yourself in the foot, and potentially damaging your organization's critical infrastructure at the same time.

1.1.4 Efficient security administration

Typical industry-reported statistics for RACF userid management, for example, creating or deleting a RACF userid, can range between 30 minutes to an hour or more for what should be a relatively simple task.

Why does it take so long?

It requires several steps to create a userid in RACF. These steps must be completed in the correct sequence, and often, some research into the access requirements of the potential new user are required in advance. It is not uncommon that new users request additional assistance and changes to their userid definition several times until all access is properly defined. This is not due to a lack of skill on the part of the system security administrators, it is more due to the technical complexity of getting all the settings correct in a large z/OS-based environment.

z/OS RACF security administrators have to perform many other tasks besides the provisioning of userids. Many of those tasks are much more complex than is shown in this example, and can involve securing of critical system resources and subsystems. These are tasks which by definition must be done correctly the first time, or severe security exposures might be introduced to the system.

Using IBM Tivoli zSecure Admin for RACF can significantly reduce the time-consuming portions of most RACF administrative activity, as demonstrated across many field installations. For example, customers have reported that complex jobs, which have previously taken one hour to run, can now be completed in often less than 5 minutes when compared to previous business practices. Even if you perform only minimal amounts of RACF work on your system, the saved time can add up very quickly.

But efficient security administration is more than just reducing the time spent on common, repetitive tasks. It is about increasing the time spent on the more difficult and potentially dangerous tasks – things such as cleaning up old definitions in a safe and risk free manner. Once your administrators are educated to use Tivoli zSecure Admin, they will find out that they now have the time to focus on the many tasks that have been filling their wish list inbox for years, tasks that are actually going to enhance your z/OS security posture rather than just have you treading water with no real improvement over time.

1.1.5 Security and audit baseline establishment

One of the unique features of the Tivoli Security Management for z/OS solution bundle, provided by IBM Tivoli zSecure Audit, is the ability to compare your system against industry best practices for z/OS security. Many organizations report that their auditors, often not mainframe experienced, require some documentary evidence that the z/OS system meets some identifiable best practice standards. Tivoli zSecure Audit utilizes a 20 year plus historical database of security miss-configurations and potential vulnerabilities to provide this capability. An actual system under analysis can be compared to some US Department of Defense accepted standards for IT security, B1, C1, and C2, as well as a quasi *zSecure standard* that has been developed over 20 years and that is based on commonly accepted commercial (rather than military) best practices as we have observed in many of our customer deployments.

This can provide you the benefit of knowing that your system is robustly secured, and if not, what changes you need to make to achieve any desired level of security. In addition, you can then use Tivoli zSecure Audit reports on a regular basis to compare your system with your accepted best practice standards to ensure no deviation is introduced over time by normal system changes and maintenance.

1.1.6 Automated cleanup of redundant security definitions

Tivoli Security Management for z/OS provides automated tools to analyze the usage of all RACF definitions and can deliver reports that allow you to generate RACF commands required to remove any definitions found to be redundant (by lack of use in a specified time period). We have reports of customers removing up to 50% of the definitions in their database after analyzing a full business cycle of user and system activity.

It is commonly accepted in IT security that unused definitions in a security database are an avenue for attack. This is especially true for userid definitions, but also the case for other RACF resources and groups. The cleanup of unused resources after de-commissioning of applications, restructuring of data, or other naming convention changes, rarely happens in a z/OS RACF environment. This is due to the inherent risk from any change to the overall system stability and availability, critical features of a z/OS environment. However, using IBM Tivoli zSecure Admin Access Monitor and Cleanup capabilities, you can now safely delete these potential back doors into your system, with the knowledge that no undesirable side effects can occur.

1.1.7 Segregation of sensitive privileges and authorities

Similar to the UNIX® *root* user, the system administrators in a z/OS RACF environment have *the keys to the entire system*. Even though appropriate audit tools are available, these are of little benefit after some event, with a system down or damaged, and competitive or other sensitive business data either in the public domain or the hands of your competitors.

Put simply, the issue is not whether to *trust* your systems administrators, but what level of trust should any individual be assigned. It is never good practice to allow any one user to access all the data available on a z/OS system. IT security principle 101, the *principle of least privilege*, exhorts us to ensure that even system administrators have only the access and privileges required to perform their day to day work. Anything in excess of that is an invitation to internal fraud or worse.

Unfortunately, many existing installations do exactly this. The assignment of the RACF administrative privileges is to a much wider user community than is really required, and the consequent risk to existing IT systems is often grossly underestimated. Using the Tivoli Security Management for z/OS component Tivoli zSecure Command Verifier, the use of highly sensitive RACF commands can be segregated between different sets of administrators. Additionally, you can split administrators into differing functional groups – thus providing a workflow and second or third level authorization required before highly sensitive data can be compromised by any one individual.

This segregation approach can be seen as a mitigation of the inherent risk of assigning security privileges to a wider audience than needed. It can be quite difficult, often for political reasons, to actually withdraw privileges once they have been assigned. People will complain, management will get involved, and the IT security administrator is forced to justify their actions in advance, before anything *bad* has actually happened. Using Tivoli zSecure Command Verifier can provide a way to allow these staff to retain their privileges, while at the same time substantially reducing the possibility that they might damage or otherwise access data, either intentionally or accidentally. Furthermore, monitoring and auditing of the privilege use by this community of users can then be established in order to reduce their privileges to *the least required to perform their job* over time. This is a safe and politically acceptable approach to a common security problem.

1.1.8 Identification of trusted users

Tivoli Security Management for z/OS provides a unique viewpoint on what is typically referred to as *trusted users*. We define a trusted user to be anyone who can, via any means, damage or otherwise corrupt the operations of the z/OS environment. Trusted user reports are critical to ensuring that issues like *segregation* and *least privilege* are thoroughly dealt with. Unless you know who your trusted users are, you cannot begin to address the issue of reducing this trust to the bare minimum.

When a trust analysis report is run using Tivoli zSecure Audit, the results show, in a prioritized order of severity, the users you are trusting, and the RACF resources they are able to access that give them effective trusted status. Additionally, audit concern findings in plain English accompany all trust status findings, giving non-technical auditors better appreciation of the risks each trust vector introduces.

Trust analysis in Tivoli zSecure Audit works from different points of view: for example, *who are my trusted users*, and alternatively, *what resources can be compromised by the base of trusted staff*? Given these two typical questions about essentially the same issue, it becomes relatively clear where the greatest gains in security can be made with the least impact on the smallest number of staff. This capability can give you an automatic 80/20 rule approach to the problem of trust. That is, you can readily achieve an 80% improvement by making changes to perhaps only 20% of the resource definitions or userids on your system. The difficult part has always been figuring out what or who the 20% are. Tivoli zSecure Audit can do this for you now, so you can get on with the important work of actually securing your z/OS environment with minimal impact and effort.

External documentation: For further information on the IBM Tivoli Security Management for z/OS V1.11 suite of products refer to the following documentation.

For the Tivoli zSecure Suite Version 1.11 Information Center go to:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.zsecure.doc/welcome.html

More information about the Tivoli zSecure Suite is also available here:

http://www.ibm.com/software/tivoli/products/zsecure/

For the Tivoli Security Information and Event Manager V2.0 Information Center go to:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tsiem.doc/welcome.html

More information about Tivoli Security Information and Event Manager is also available here:

http://www.ibm.com/software/tivoli/products/security-info-event-mgr/index.html

1.2 IBM Tivoli Security Management for z/OS components

Tivoli Security Management for z/OS V1.11 provides security management and is tightly integrated with RACF, enabling compliance management, security administration, user management, and security monitoring on the mainframe. Tivoli Security Management for z/OS V1.11 consists of the following products:

- IBM Tivoli zSecure Admin
- IBM Tivoli zSecure Audit
- IBM Tivoli zSecure Command Verifier
- IBM Tivoli Security Information and Event Manager
- IBM Tivoli Compliance Insight Manager Enabler for z/OS components:
  - IBM Tivoli Compliance Insight Manager Enabler for z/OS - RACF
  - IBM Tivoli Compliance Insight Manager Enabler for z/OS – DB2
  - IBM Tivoli Compliance Insight Manager Enabler for z/OS - CICS®

Overviews of these products are presented in the following sections.

1.2.1 IBM Tivoli zSecure Admin

Tivoli zSecure Admin is the new face of RACF in the traditional (ISPF-based) user interface to z/OS. It is intuitive and easy to use for both new and experienced RACF administrators, providing a searchable, sortable, scrollable (up/down and left/right) table display of RACF *Userids*, *Groups*, *Datasets,* and *General Resources* via its various main menu selections.

Each main menu selection presents a similar screen that provides optional filters, selection criteria, and more advanced resource-specific selections that allow the administrator to easily drill down to the profiles and definitions they need to work with to accomplish any particular task. Selecting a specific resource (Userid, Group, Dataset, or General Resource) displays a scrollable screen containing all relevant information about the resource, with plain English

descriptions of the various fields, and comprehensive, context-sensitive help screens available to explain the meaning and use of any resource attribute. If Tivoli zSecure Audit is also installed, then specific audit findings for any particular resource are also displayed, and can be *jumped to* using the user interface to obtain details from Tivoli zSecure Audit about why that particular finding is applicable.

Primary and *Line commands* are available for all common RACF administrative tasks, for example, delete or define a user, report on all user access rights, report on user activity from SMF, and so on. All reporting can be generated as either in-line ISPF displays, batch reports, or sent immediately as an email to any concerned party.

As previously mentioned, the increased administrator efficiencies achieved by using Tivoli zSecure Admin can provide not just a better business outcome for your end users, but also a better security outcome for your business. Administrators are no longer purely procedure-following piece workers, but can now be empowered to perform the RACF administrative function at a higher level to actually secure your z/OS like it needs to be.

1.2.2 IBM Tivoli zSecure Audit

How often does your organization pay a significant amount of money to outside consultants and auditors to review your z/OS security? And how satisfied are you with the results of these reviews? While qualified people capable of performing thorough technical reviews of a z/OS and RACF security implementation surely exist, they are relatively rare, and correspondingly expensive. Many agencies that advertise support for z/OS in their audit programs follow outdated and rather simplistic audit guidelines obtained through the Internet or from historical, and no longer current, documentation. Often a *rubber stamp* audit will give your management a good feeling about z/OS security, but the technical staff are aware of significant shortfalls in both the audit and their system configuration. Sometimes, after these audits, security holes not uncovered in the review are actively used to access sensitive data and compromise systems.

Tivoli zSecure Audit addresses these concerns, acting as an automated *auditor in a box*, bringing 20 plus years of deep technical audit experience into your organization, available to be tapped for an expert opinion any time you need one.

Organizations deploying Tivoli zSecure Audit are *audit ready*. They are able to produce the documentation regarding their current audit status, recommendations for improvements, and standard periodic audit reports easily and in an end user friendly manner. In fact, a properly deployed Tivoli zSecure Audit performs the job of an auditor, and does it every day rather than once a year. Users of Tivoli zSecure Audit are leading the global change in audit best practice by moving from periodic auditing to daily or real-time security monitoring.

1.2.3 IBM Tivoli zSecure Command Verifier

As previously mentioned, Tivoli zSecure Command Verifier can stop you from accidentally damaging your system via inappropriate RACF commands. Also, it allows for fine-grained segregation of RACF command privileges, and together with Tivoli zSecure Admin can implement a multi-level authorization process to ensure that no single user can issue sensitive commands without at least some level of peer or management review occurring first. These capabilities enhance your system resiliency and allow you to take acceptable risks with the delegation of RACF privileges in a controlled and safe manner.

Additionally, Tivoli zSecure Command Verifier provides an enhanced audit trail, known as the Command Audit Trail (CAT) feature, which addresses the issue of knowing when and by whom a change was made to a RACF definition. Often, RACF administrators or auditors are

requested to determine when a specific command was issued and by whom for forensic purposes. Depending on how long ago this action may have occurred, it can take weeks of searching through the SMF audit trail to discover the answer to what seems like a simple question. With Command Audit Trail active, the administrator examines the profile in question and Tivoli zSecure Command Verifier displays the last 64 changes made to the profile, including who issued the commands. This allows you to zero in on the specific SMF date range for the suspect command immediately, and report on other relevant activities that may have occurred around the same time, hugely speeding the response to these forensic questions.

Another major capability of Tivoli zSecure Command Verifier is that it can enforce naming conventions and standards to follow the organization's documented guidelines for RACF resource naming. This can prevent bad definitions occurring in your database, and can keep your internal practices in line with you documented standards, thus helping you to achieve your overall policy compliance objectives.

1.2.4 IBM Tivoli Security Information and Event Manager

Tivoli Security Information and Event Manager is a cross-platform log management and analysis, auditing, and reporting tool. It generates reports on collected log data referring to security policies to identify policy violations.

Tivoli Security Information and Event Manager compares real end user behavior as observed by the system log records with the desired behavior that you can configure using the Tivoli Security Information and Event Manager management console. Tivoli Security Information and Event Manager can monitor your users' access and interaction with your organization's data, and it can alert you when a user steps outside the acceptable use definitions.

Tivoli Security Information and Event Manager can achieve this by generating normalized meta-data over the user base and the classification of your data sets. The way this meta-data gets collected and normalized can be individually defined to be relevant to your unique organization. Additionally, you can exploit pre-defined user and data classification models that are derived from several of the industry regulatory frameworks now common in many countries, and increasingly a legal requirement for certain types of business. Tivoli Security Information and Event Manager can make compliance reporting for legislative regulations such as the Sarbanes-Oxley Act (SOX), Health Insurance Portability and Accountability Act (HIPAA), and Payment Card Industry Data Security Standard (PCI DSS) a repeatable and tune-able activity.

Tivoli Security Information and Event Manager closes the loop on the auditing process. By analyzing the actual user behavior (logs), and highlighting deviations from your policy, you can use this information to either refine the policy, or correct the security implementation to match the policy and prevent further deviations from your desired user behavior.

All this can be achieved using a web browser-based reporting interface, understandable to non-technical auditors, and providing a range of commonly requested standard audit reports. While Tivoli Security Information and Event Manager can process the z/OS SMF-based information, it can also collect and manage log information from over 300 different types of applications, platforms, and databases. The ability to bring all these disparate data sources together into one reporting framework means that at last organizations can gain some real benefits from those cumbersome system logs we have been generating and retaining for all those years now.

IBM Tivoli Compliance Insight Manager Enabler for z/OS

The Tivoli Security Management for z/OS solution bundle includes Tivoli Security Information and Event Manager components specifically built to process RACF, DB2, and CICS data. Without any significant effort around the Tivoli Security Information and Event Manager meta-data and configuration, you can get meaningful reports after a very short setup time because Tivoli Security Information and Event Manager is RACF, DB2, and CICS aware. That is, it knows who on your system has high level privileges or access to sensitive data. Tivoli Security Information and Event Manager can immediately perform some basic classification of both your users and your data to start giving you an immediate return on the deployment investment.

Naming mix-up: IBM Tivoli Security Information and Event Manager v2 has recently replaced the IBM Tivoli Compliance Insight Manager product. Some of the existing add-ons for the previous version still carry the Tivoli Compliance Insight Manager name, like in this case, the IBM Tivoli Compliance Insight Manager Enabler for z/OS. But they work fine in conjunction with Tivoli Security Information and Event Manager.

1.3 Tangible benefits and ROI

In the previous sections we mentioned some of the immediate benefits of utilizing Tivoli Security Management for z/OS. These benefits include:

- Reduced time and associated costs to be audit ready.
- Reduced time and increased compliance for standard security activities.
- Reduced security risk by a combination of alerting and baseline security improvements.
- Enhanced change control tracking to reduce availability risks introduced by changes.
- Reduced reliance on highly specialized (expensive) staff to perform basic audit reporting.
- Reduced risk of unintended outage caused by erroneous RACF commands.
- Improved security posture by being able to re-direct efforts of highly skilled staff.
- Improved risk management by ensuring that your system meets recognized international security baseline standards.
- Reduced security exposure risk by automated removal of redundant security definitions.
- Reduced security risk by appropriate segregation of high level privileges.
- Improved user satisfaction with the security process, one that gets them the access they need, in a safe and timely manner.
- Reduced requirement to employ specialists for periodic deep technical audits.
- Improved capability to report on security changes in a timely manner, and additionally prevent unwanted changes occurring in the first instance.
- Centralized log collection and analysis and the attendant benefits achieved by this more efficient approach.

The quantification of these savings in ROI terms remains a difficult and error-prone process. There are always differing ways of looking at the same data, resulting to quite different conclusions. In order to assist you in addressing this dilemma, IBM partners with an independent company, Alinean Inc., which produces well-defined Return on Investment (ROI) analyses as a vendor-agnostic service to the IT community in general.

In the following sections we look at the ROI impact for business drivers as well as IT operations. We reproduce data from a report provided by Alinean Inc. that documents the expected minimum cost reductions that could be achieved using typical industry standard best practice for IT security. Remember that all the documented line items enumerated previously can be achieved by employing the IBM Tivoli Security Management for z/OS solution bundle, and many of these are tangibly quantified in the report excerpt we provide here.

More about our partner: Alinean Inc. (http://www.alinean.com/) is a leading provider of on-demand sales tools and related services. IBM has partnered with Alinean to create the IBM Business Value Analyst to help our customers financially justify IBM solutions by focusing on business value and return on investment. The Business Value Analyst is a tool available to Tivoli sales teams via Extreme Leverage and IBM Business Partners via the Tivoli Knowledge Center.

1.3.1 Impact on business drivers

In this section we examine the impact on the business drivers.

- Insider threat / Data theft

  80% of insider threats are caused by privileged or technical users. Tivoli Security Information and Event Manager adds a *camera lens* to your network by collecting and allowing you to view the audit trail logs as evidence of user behavior. When insiders know you are watching, the chance of data theft is reduced and the ability to understand, avoid, and remedy mistakes improves.

  Also, Tivoli zSecure Audit's checks on configurations and vulnerabilities, comparison to best practices, and remediation capabilities, mean your system will be less susceptible to external and internal attacks and mistakes. Additionally, when an intrusion or mistake occurs, Tivoli zSecure Audit enables you to isolate the situation, understand the cause and remediate it rapidly. Finally, Tivoli zSecure Audit ensures that vulnerable default settings used by technical insiders are disabled so that privileged user breaches do not occur.

  In this respect, Tivoli Command Verifier can help ensure the mainframe configurations and settings are compliant, lowering the likelihood of internal and external breaches, and self-inflicted wounds.

  With Tivoli zSecure Admin, RACF administration will be cleaner and less error prone, not to mention more compliant with your security and regulatory policy. All of this helps to reduce vulnerabilities and the likelihood of internal breaches and costly mistakes.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 10% - 15%.

- User access savings

  RACF is hard to learn for newer and decentralized administrators; Tivoli zSecure Admin provides an easier interface to RACF for administrators and can save them time and effort in performing their tasks.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 5% - 10%.

1.3.2 Impact on IT operations

In this section we examine the impact on IT operations.

- Policy management

  Tivoli Security Information and Event Manager enables you to codify your log management collections in practical rules — Who can do What, When, Where, Where from and Where to — so that *acceptable use and change management policies* can be monitored and enforced automatically.

  On the System z side, Tivoli zSecure Audit helps you to advance manual checking of your policies to automated processes. The output can be used in a consolidated fashion within Tivoli Security Information and Event Manager.

  You can enforce RACF policies *in-line* and automatically with Tivoli Command Verifier, which verifies that commands meet your audit and regulatory policies *before* they are executed.

  You can enforce identity and access policies with Tivoli zSecure Admin's user friendly interface for RACF administration. You can administer the entire user life cycle at lower cost, with greater ease, and according to company policies.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 10% - 15%.

- Compliance management and reporting

  Tivoli Security Information and Event Manager can automate log management by allowing for universal collection, storage, retrieval, and investigation of security log data, and then automatically formats and processes logs for compliance and investigatory reports. Modules for specific regulations, such as SOX, HIPAA, ISO, and GLBA, can save you even more time by automating reporting.

  You can utilize dozens of built-in reports for auditors and the Tivoli zSecure Audit CARLa reporting language for your custom needs.

  You can pass audits more easily because Tivoli zSecure Command Verifier can keep your RACF database clean and compliant by verifying that commands meet your audit and regulatory policies *before* they are executed.

  You can ensure compliance through automated policy compliance security administration on RACF with Tivoli zSecure Admin.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 15% - 25%.

- Log management

  Tivoli Security Information and Event Manager can automate log management by allowing for universal collection, storage, retrieval, and investigation of security log data and then automatically formats and processes logs for compliance and investigatory reports.

  You can also automate your log management with Tivoli zSecure Audit on the mainframe and feed logs to enterprise log management solutions, like Tivoli Security Information and Event Manager.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 20% - 40%.

- Forensics

  Tivoli Security Information and Event Manager's ubiquitous log collection, forensics, and management capability allows you to store, retrieve, and investigate logs for user behavior across any server, application, database, or device.

According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 15%.

- Security tools customization, management, and maintenance

  Avoid the need for custom tools for RACF audit with Tivoli zSecure Audit's depth of audit capabilities.

  Save on in-house tool creation efforts by leveraging Tivoli zSecure Command Verifier as a solution that verifies that commands meet your audit and regulatory policies *before* they are executed.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 30%.

- Average internal and external time spent on audit and pre-audit preparation

  Audits can cost hundreds of thousands of dollars to prepare for. Tivoli Security Information and Event Manager and Tivoli zSecure Audit can help automate the aspects related to gathering log files, generating compliance reports, demonstrating evidence of meeting regulations and standards, enabling audit investigations, and more.

  Tivoli zSecure Audit can automate and streamline your audits by continuously analyzing security compliance of RACF, ACF2, TopSecret, z/OS, DB2, and UNIX on the mainframe. Dozens of reports and analyses are available at your fingertip and can be used when the auditor arrives.

  This can save significant time and work before, during, and after an audit.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 10% - 15%.

- Average internal and external time spent on audits

  While auditors are on site, they can ask for significant volumes of data and reports. For security audits, Tivoli Security Information and Event Manager can automate the collection of log information and reporting against compliance. This means that consultants are less needed and audits are shorter.

  Tivoli zSecure Audit can automate and streamline the preparation for audits by continuously analyzing security compliance of RACF, ACF2, TopSecret, z/OS, DB2, and UNIX on the mainframe. Dozens of reports and analyses are available at your fingertips and can be used when the auditor arrives. This can save significant time and work before, during, and after an audit.

  According to the Business Value Analyst tool from Alinean this can provide an organization with cost savings around 10% - 15%.

If you are looking to create an ROI analysis to help justify to your management the cost returns of any investment in Tivoli zSecure products, these numbers can be used in reference to your current known costs of providing the equivalent functions. Savings can be calculated from this across the range of IT security processes you currently carry out, and it is highly likely you can provide a reasonable business justification for investment in your z/OS and RACF security infrastructure.

1.4 Conclusion

In this chapter we discussed how IBM Tivoli Security Management for z/OS can provide audit and compliance management reporting for your organization. We talked about how to aggregate, analyze, and monitor for threats by auditing security changes that affect security information from z/OS, RACF, CICS, and DB2. As a result, IBM Tivoli Security Management

for z/OS can capture comprehensive log data, interpret that data through sophisticated log analytics, and communicate results in an efficient, streamlined manner for timely follow-up investigation.

IBM Tivoli Security Management for z/OS V1.11 can help you administer your mainframe security while also reducing administration time, effort, compliance overhead, and costs. It addresses the problem of obsolete authorizations with a RACF database clean-up function and provides audit usage of resources while reporting on exceptions. It helps enforce policy compliance and provides automated access monitoring to help ensure an uncontaminated database.

In the next chapter we discuss three business scenarios that illustrate these capabilities.

2

Customer scenarios

In this chapter we describe three common scenarios for deploying various components of the Security Management for z/OS products. Not all products are used in each example, in order to help show that immediate benefits can be achieved by simple deployment of one component, then enhanced later by additional component deployments from the suite.

A specific business objective is the main driver behind each customer scenario. Here is an overview of the challenges these organizations are facing:

- Satisfy internal and external auditors that the z/OS security environment is being appropriately managed and secured.
- Provide protection for critical RACF resources from abuse by privileged insiders.
- Demonstrate audit readiness and policy-based management of security access rights.

These are common business challenges many organizations are faced with. We now show how you can successfully deploy various components from the Tivoli Security Management for z/OS offering to effectively address these concerns, while at the same time reducing costs, improving security, and meeting industry best practice standards.

2.1 Satisfy internal and external auditors that the z/OS security environment is being appropriately managed and secured

In the first scenario, a government department is being audited on a regular basis, and only recently the audit department has obtained greater skill in z/OS-specific auditing. The questions are getting harder, the depth and technical detail being requested during the audit is increasing, and the organization has decided it is time to do a cleanup and modernization of their z/OS security management practices in order to pass these increasingly more stringent audits more easily.

To tackle these requirements the agency has decided on the following three phase approach in order to minimize risk and bring as much of the ROI into the earliest parts of the project as possible:

Phase 1 Deploy Tivoli zSecure Admin and Audit.

Phase 2 Implement the Tivoli zSecure Audit recommended baseline improvements.

Phase 3 Establish baseline tracking to ensure continuous compliance with declared security policies and best practice standards.

2.1.1 Phase 1 – Deploy Tivoli zSecure Admin and Audit

Tivoli zSecure Audit will provide the government agency with reports detailing their current security status and offering recommendations for improvement of this status. Tivoli zSecure Admin provides the administrative tools required to effectively and rapidly implement the improvements recommended by zSecure Audit.

Both products are installed using standard z/OS installation processes via the System Modification Program Enhanced (SMPE).

After introductory training immediate benefits are realized in the efficiency of RACF administration. This allows RACF administration staff to redirect their time to the recommended improvements in baseline security.

The implementation includes the automated generation via scheduled batch jobs of daily, weekly, and monthly archives of three relevant categories of security-related data:

1. An unloaded format of the RACF database, optimized for efficient processing of batch or other periodic reporting, referred to as the zSecure UNLOAD file.
2. A snapshot of relevant system security settings from PARMLIB and other system configuration data, known as the CKFREEZE file.
3. Copies of the matching SMF data for each daily, weekly, and monthly set of archived data.

Of these data sources, 1 and 2 are generated specifically by zSecure Audit to enable its deep inspection of z/OS- and RACF-related security configuration. The third data source is usually generated using existing tools and automation on the z/OS platform. In most deployments you want to retain some archive of this SMF audit trail; however, it rarely contains sufficient information for a full forensic analysis of activity on the system without being specifically configured to do so (an example of this is shown in “Ensure correct audit settings on critical infrastructure resources to generate relevant SMF audit trail for changes to sensitive configuration data” on page 22).

These archived data sources can provide additional functionality, for example, being able to compare a historical point in time versus today in respect of the system security baseline.

This can demonstrate progress in improving security to management and serve as a metric for the success of the security improvement project. zSecure Audit functions are used to compare historical against current definitions.

The installation consists of several z/OS logical partitions (lpars) spread across two physical System z machines in separate data centers for operational reasons, such as disaster recovery. The zSecure Audit and zSecure Admin solutions are installed on all lpars, including systems programming test lpars (often referred to as *sandpits*) where new z/OS releases are installed and tested.

The installation has its DASD shared across lpars, something that can introduce vulnerabilities should the RACF databases differ between the lpars. zSecure Audit takes into account these fairly common configuration issues and provides analysis from both or all sides in the case of data accessible from more than one lpar. This deployment architecture is shown in Figure 2-1.

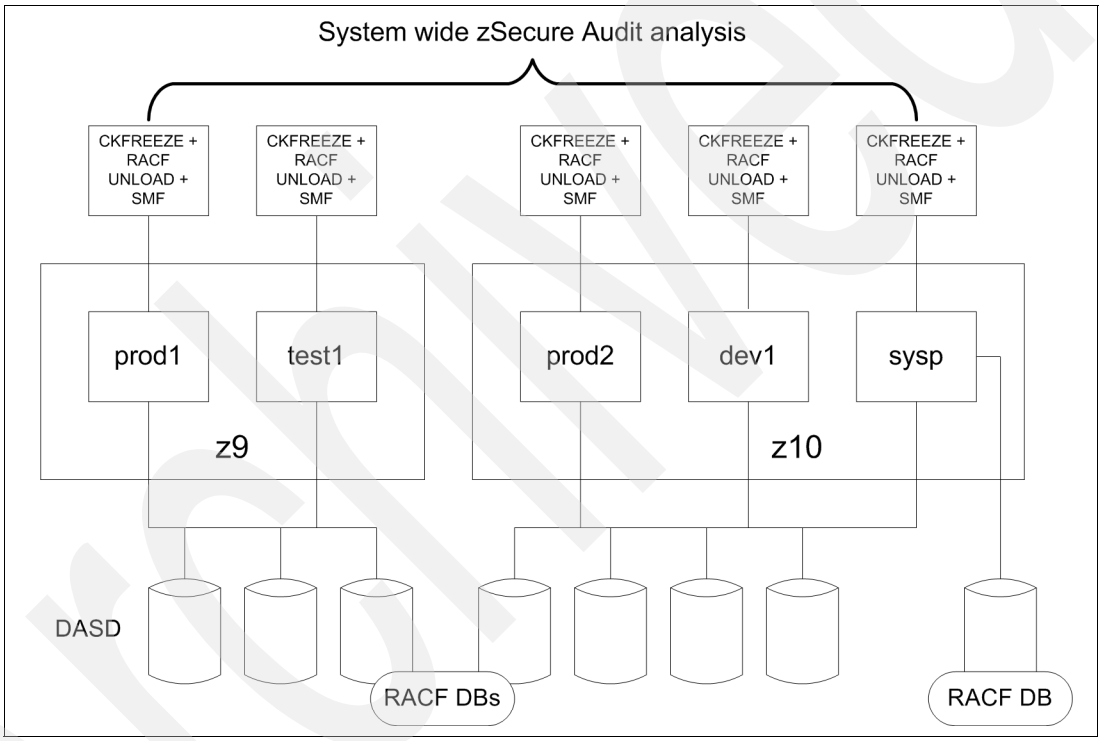

Figure 2-1 Deployment architecture

The security administrators, working with the systems programmers and operations staff, implement a set of scheduled security activity and baseline comparison reports. Generally, these reports employ the data generated previously in the daily suite of data collection jobs, the CKFREEZE file, the RACF UNLOAD file, and the SMF data from the matching day.

This results in the batch data flow shown in Figure 2-2 on page 18. You can see that the data is generated, saved, reported on, then archived for historical use.

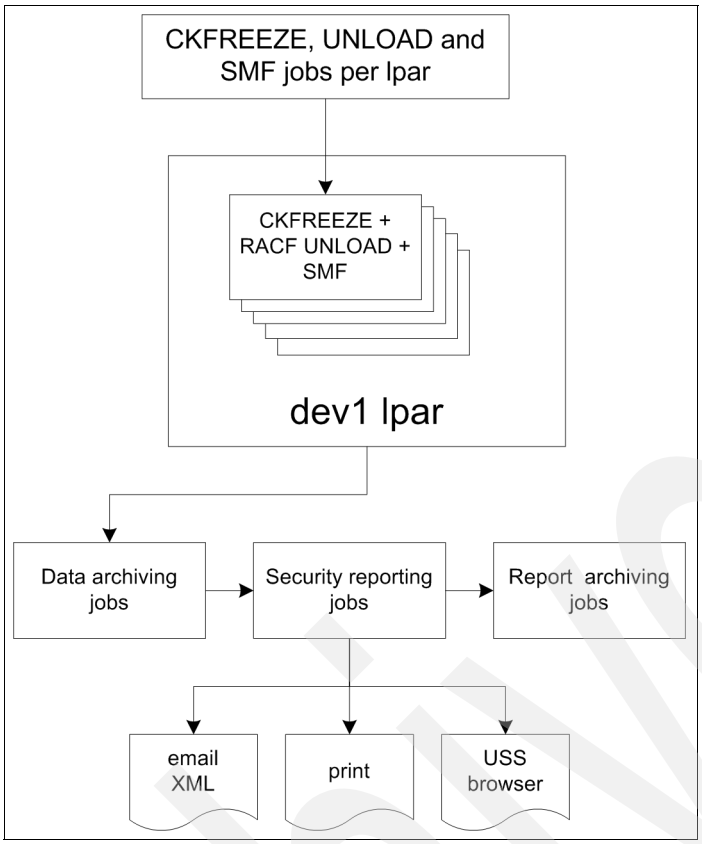

Figure 2-2 Batch data and reporting flow

In this case, the government agency decided to implement several custom CARLa-based reports to generate automated email reports containing XML formatted attachments. These reports are targeted to the non-technical audience because they are accessible via readily understood desktop-based technology rather than the traditional mainframe user interfaces.

The custom reports included:

1. Daily user access violations, delivered to responsible managers using email.
2. Quarterly access re-validation reports, similarly delivered to team leaders and other responsible managers.
3. High level access to sensitive data daily email report to data owning managers.
4. Additions or changes in RACF settings, emailed to security administrators.
5. Additions or changes in z/OS system settings, emailed to systems programmers.
6. Summary of RACF commands issued, emailed to security.
7. Use of high level privilege to access resources, emailed to security.

All data gathered in the collection stage of zSecure operations is archived for future reference, providing a demonstrable and comprehensive historical audit trail. The collected data from all lpars is stored on a specific, non-production, lpar where all reporting is generated, thus relieving the production system of this management workload.

All reports are similarly archived, and hosted via the UNIX System Services file system, made available via the built-in z/OS web server so that users and auditors can view historical reports using their standard web browser. zSecure can automatically export these reports

into the UNIX System Services file system for you. You might consider writing a small z/OS UNIX System Services shell script to cycle and archive the saved reports.

At this point, the government agency considers that the deployment of IBM Tivoli Security Management for z/OS has been a successful project. Already, auditors are seeing the benefits of standardized and readily accessible reports, and administrators are realizing efficiency improvements in their day to day work. Next we start to improve the systems baseline security in order to better pass our audits.

2.1.2 Phase 2 – Implement zSecure Audit recommended baseline improvements

With zSecure Audit implemented, it is a simple task to generate a prioritized list of system audit concerns via a standard, provided report from the ISPF interface. Figure 2-3 shows the typical output of such a baseline report.

```
Session A - [32 x 80]
SETROPTS settings - audit concerns                                  Line 1 of 11
                                                    4 Sep 2007 12:17
   Pri Complex  System   Count
    34 ZT01     ZT01        11
   Pri Parameter                 Value    Audit concern
__  34 PROTECTALL                Warning  Warnings do not prevent unauthorized a
__  30 BATCHALLRACF              No       Allowing unidentified batch work makes
__  30 REVOKE                    No       Too many password violations allowed
__  20 OPERAUDIT                 No       OPERATIONS activity undetectable
__  15 AUDIT_GROUP               No       Profile changes in GROUP class are not
__  15 AUDIT_USER                No       Profile changes in USER class are not
__  15 ERASEONSCRATCH            None     Disk scavenging threat not countered /
__  15 HISTORY                   No       Users can use same passwords over and
__  11 MINCHANGE                 No       Without MINCHANGE users can thwart the
__  10 INACTIVE                  No       Apparently unused userids increase ris
__   2 TAPEDSN                   No       Tape datasets are unprotected unless T
******************************** Bottom of Data ********************************

Command ===> _                                                Scroll===> CSR
MA   a                                                                 31/015
```

Figure 2-3 zSecure Audit baseline report

These reports are executed from the security management (that is, non-production) lpar, and process RACF databases and CKFREEZE system snapshots from all lpars in one pass. This gives the government agency a whole system view, regardless of segregation between the systems. In a multi-lpar environment this view is essential to ensure that you understand the overall security and any implications of security changes.

In this government agency's z/OS environment the reports revealed that some lpars have significantly weaker baseline controls than others. Notably, the system programmers' sandpit has much less stringent controls in place than the production lpars. While this would be expected for a test environment, in this case zSecure Audit has highlighted that, due to shared DASD, certain critical resources belonging to the production systems are vulnerable when accessed from the systems programming lpars. This is due to the systems programming lpars using a separate RACF database, in which several critical controls have

been de-activated. The task now is to re-activate those controls with a minimum of disruption to the legitimate use of these testing lpars.

Several controls need to be reactivated or other access paths removed or reduced. These include:

- Use of RACF SETROPTS NOPROTECTALL
- Wide use of user attribute OPERATIONS for access to data
- Wide use of UNIX System Services superuser (UID 0) privileges
- Incorrect audit settings on critical infrastructure resources, resulting in a lack of SMF audit trail for changes to sensitive configuration data

Eliminate use of NOPROTECTALL

The use of NOPROTECTALL as a system setting allows data to exist on the system with no RACF protection, effectively available to all users of the system. While this is rarely seen in production systems, it does happen, and it can be difficult to remove once this practice has been established for any length of time.

The question that comes up is *What data exists on my system now without RACF protection?* Fortunately, zSecure Audit provides simple reporting that can show you both what data exists without any matching RACF profile, and conversely what, if any, RACF profiles exist for which there is no matching data on DASD available to that lpar.

The next question requiring resolution is *How do I know when all previously unprotected data has been protected, and can de-activate the NOPROTECTALL SETROPTS parameter?* Again, zSecure Audit can help here by providing a report of all access to data via non-normal means.

The final question that must be answered is *How do I know that users have been granted the correct levels of access to the previously unprotected data?* And again, zSecure can help us here.

This leads our government agency to develop a multi-step, somewhat iterative process to gradually reduce and finally eliminate this security exposure:

1. Run zSecure Audit reports to identify non-protected DASD datasets.
2. Define profiles for these datasets, set the profile itself into WARN mode so it will not deny access (use zSecure Admin for this).
3. Repeat steps 1 and 2 until no non-protected datasets remain.
4. Move from monitoring for unprotected data to monitoring dataset profiles in WARN mode – zSecure Audit has a built-in report just for this purpose.
5. Review the zSecure Audit daily WARN mode access report. Sometimes it will be obvious that a user requires a certain level of access.
6. After a period of time has elapsed, sufficient to capture SMF records for most common access to the WARN mode dataset profiles, use a CARLa summary report to show which users accessed the data, and the highest level of access each user actually employed over the entire monitoring period. This provides the necessary data required to grant the appropriate levels of access with a high degree of certainty that this is a legitimate requirement for the users.
7. Using zSecure Admin, remove the WARN mode flag from the profiles, and continue to report; however, now you should report on access violation attempts, again a standard report from zSecure Audit.

8. Using the zSecure access violation reports, together with any likely user access requests generated as a result, fine tune the access lists of the previously unprotected dataset profiles.

Walking through this process, and reviewing daily reports of access via WARN mode and access to unprotected data, can quickly reduce the number of accesses of this type. At some point, the undesirable accesses become so few and far between that it makes sense to activate PROTECTALL and finally close this exposure. Depending on the level of comfort, and the implications of any security errors on the particular system the government agency is working on, they may run with PROTECTALL in WARN mode for some period of time prior to finally activating PROTECTALL in FAIL mode – which should be the ultimate goal.

At each step of the way, the agency can use zSecure Admin to generate any necessary RACF commands or change RACF SETROPTS settings with simple over-typeable fields. This helps ensure that they issue the correct commands without introducing syntax errors or having to search documentation to ensure the commands are correct.

Eliminate use of user attribute OPERATIONS for access to data

The use of the OPERATIONS attribute is still widely practiced today, often on production systems. This attribute has been gradually phased out as a requirement for operational tasks in a z/OS environment over many years; the preferred and recommended methods of granting universal access to data are by using functional controls available to users of Data Facility Data Set Services (DFDSS). The reason for the gradual move away from OPERATIONS was that it encompasses more access than is generally required to perform the job function (usually that of storage administrator). OPERATIONS allows not only access to move and manage the data, it also allows the user to read and change the data. This is obviously not part of the role of a typical storage administrator. Thus, OPERATIONS violates the fundamental IT security principle of *least privilege*.

The alternative to OPERATIONS is a set of RACF profiles defined in the RACF FACILITY class, and referenced by programs within the DFDSS suite of data management tools. Access to these RACF profiles, usually defined as starting with STGADMIN (storage admin), allow the storage administrator to manage the data, but will not allow them to read or alter the data in most circumstances.

In order to use the DFDSS functional profiles, the storage administrator must code a special *admin* keyword on their storage administration batch jobs. So, in order to eliminate this old, redundant, and dangerous OPERATIONS attribute, the government agency implements a multi-step reporting and analysis process that helps minimize the risk of unintended consequences due to the security improvement project.

The agency performs the following tasks to accomplish this:

1. Using zSecure Admin, define profiles in the class FACLILITY covering the STGADMIN functions and allows storage administrators the required level of access.
2. Update all storage administrator jobs to include the ADMIN keyword.
3. Run the supplied zSecure Audit reports for activity where the OPERATIONS attribute was used. In some cases it will be obvious that the user requires a certain level of access; grant this where appropriate.
4. After gathering some period of data detailing OPERATIONS use, run a CARLa summary report showing the highest unique levels of access per user. After appropriate validation of this, grant the access determined to be correct.
5. Repeat steps 3 and 4 until the reports contain no, or very infrequent, accesses of this type.

At this point, it should be possible to remove the OPERATIONS attribute from all users and rely on the new storage administrator functional profiles for day to day systems management. When using zSecure Admin and Audit combined, this kind of migration is not nearly as challenging as it may initially seem. Given the appropriate tools, there is no longer an excuse for continuing with these outdated, and frankly dangerous, security practices in the z/OS environment.

Reduce use of UNIX System Services superuser (UID 0) privileges

Similar to the OPERATIONS attribute discussed in the previous section, UNIX UID 0 grants access in an uncontrolled manner, typically not acceptable in well managed IT security infrastructures. Often, systems programmers are granted UNIX UID 0, as well as a home directory of "/" or `root`, and no other controls on their access within the UNIX System Services environment. Experienced UNIX or Linux® administrators would be aghast at this approach. In their defense, experienced traditional z/OS systems administrators had little or no experience of commonly accepted UNIX security standards when we first started dealing with z/OS UNIX System Services. However, this situation has changed, more z/OS administrators are also UNIX security aware, and the importance of running z/OS UNIX System Services to the same or better standards than other UNIX installations is becoming more readily acknowledged in the z/OS world.

Once again our government agency developed a checklist of steps to follow to reduce the widespread use of UID 0 in the z/OS environment:

1. Identify all users currently assigned UID 0. zSecure Audit has a built-in report for this purpose.
2. Report on home directories assigned to these users. The same zSecure Audit report contains this data.
3. Create and assign unique home directories for these users. zSecure Admin can assign the home directories after they have been created using either the ISHELL utility or native UNIX commands.
4. Assign unique UIDs to staff who previously had UID 0. This is easy using the zSecure Admin interface.
5. Where staff have a documented and legitimate requirement to access superuser services, use zSecure Admin to grant them access to the FACILITY class profile BPX.SUPERUSER. They can now use the UNIX **`su`** command to assume superuser privileges in a controlled and audited manner.
6. Move any private user data from the previous home directory to their unique new home directory, and make appropriate ownership changes to this data to reflect their new unique UID. This is done using the UNIX **`chown`** command.

You can see the steps involved are not overly complex, although they must be completed for *each* target userid. Where a large number of users with this condition exist, it is possible to script these changes and somewhat automate the assignment of UIDs and the movement of users and their data to the new structure. This is outside the scope of zSecure, requires some basic UNIX programming skills, and also is outside the scope of this paper.

Ensure correct audit settings on critical infrastructure resources to generate relevant SMF audit trail for changes to sensitive configuration data

It is common to find that organizations allow the default RACF SMF logging of *violations only* to be generated for most, if not all, RACF resource profile definitions. While this provides some level of audit trail, additional data is required for thorough audit logging and potential

forensic investigation should this ever be required. If the data is not being logged, you will not be able to determine who did what if an inquiry is ever made.

Using standard zSecure Audit system verification reports, the government agency discovered that inadequate auditing was in place for most, if not all, sensitive system resources. This means that should someone with legitimate access (for example, a systems programmer or other highly privileged user) use this access to corrupt or otherwise subvert a system definition dataset, there would be no effective record of this action in the SMF audit trail.

The zSecure Audit verification reports provide a list of RACF commands recommended to correct this situation. For each sensitive dataset, determined via the CKFREEZE system snapshot database, zSecure Audit recommends the correct audit, erasure, and universal access attributes that should be assigned to data with this level of criticality. It is up to the security administrators to review these command recommendations, modify them as necessary to suit local naming conventions or standards, then implement the resulting set of profiles.

This marks the end of the second implementation phase, and we now have a system where zSecure Audit and zSecure Admin are being used to report on and control the environment. We have also made significant improvements in the overall security posture of the system, removing several audit-able findings and reducing business risk overall. In phase 3 we establish processes to ensure things stay that way.

2.1.3 Phase 3 – Baseline tracking

In the final phase, the government agency establishes some baseline tracking to ensure continuous compliance with declared security policies and best practice standards.

Establish a Trusted Computing Base

The first step in monitoring a *security baseline* for changes is to have a baseline that is agreed to be as close to best practice as we can reasonably achieve. The work the agency has done in the previous phases helps to get to this stage. Now it is time to take a snapshot of the system, and declare this to be the acceptable baseline. We call this snapshot the *Trusted Computing Base* (TCB).

zSecure Audit provides this capability in the form of a primary menu option known as *Change Tracking*. This function compares a saved copy of the TCB with the settings found on the system today. Any variations from the accepted TCB are reported for one of three types of action:

1. Accept the change into the TCB.

   This indicates that you were aware of the change, and it is a desirable improvement to incorporate into the existing and accepted TCB.

2. Reject the change.

   This indicates that the change was un-controlled, and should not be incorporated into the TCB. This change will keep appearing in the daily analysis until it is either backed out of the system, or you decide it should be part of the current TCB and subsequently accept the change.

3. Defer the change.

   Deferring a change is the action taken when you are unsure of the desirability of incorporating it into the TCB. You may need to discuss this change with system programmers, administrators and auditors before deciding on the appropriate actions (accept or reject).

Using this built-in zSecure Audit capability provides a functional method to ensure the system stays the way you want it to be with respect to security and operating system configuration.

Using Access Monitor and automated cleanup capability

At the same time they established their TCB baseline, the agency also deployed the zSecure Access Monitor and Cleanup utilities supplied with zSecure Admin. These functions run passively in the background for an entire year before the next steps will be taken. During this time, data is gathered in a highly optimized format specifically for the purposes of determining *who did what* on their system, and what RACF access rights or other privileges were used to perform the actions.

Once this data has been collected for a full business cycle (in this case, the agency decided one entire year of processing would be needed), the Cleanup function is used to compare all access reported against all access rights defined in the RACF database. The difference between the RACF database defined access and the actual observed access gives the administrator a list of RACF definitions that have not been referenced in an entire year. With a high degree of confidence that no negative system impact will occur, the administrator can now start to delete these redundant RACF definitions.

2.1.4 Scenario 1 conclusions

Our government agency has now established the use of Tivoli Security Management for z/OS in its initial stages. There is great potential for further improvement in security and consequent ROI for this agency as they deploy more of the products contained in the total offering. But at this point, already, they are in the comfortable position of being able to demonstrate audit readiness and strong management focus on maintaining a highly secure RACF and z/OS configuration and environment from which to run their business.

Additionally, with the Access Monitor and Cleanup functions deployed, by this time next year the agency will be able to demonstrate to their auditors that a true process of continuous improvement has been established with respect to RACF definitions. The combination of assured TCB and a clean security database position this organization among the best run infrastructures with respect to RACF security in the world. They can expect to receive increasingly positive audit reports on this system.

2.2 Provide protection for critical RACF resources from abuse by privileged insiders

In our second customer scenario, we encounter a financial sector organization with a pressing internal audit requirement to demonstrate proper segregation of security privileges in their RACF environment. A primary audit concern has been the fact that all security administrators use the highest level RACF privilege of System SPECIAL. It has been suggested several times in prior years' audit reviews that the auditors feel it should be possible to allow administrators to perform their job functions without requiring this highest level of RACF administrator rights.

The company has purchased the Tivoli Security Management for z/OS offering and intends to deploy the zSecure Command Verifier component first to address this primary concern. At later stages an implementation of zSecure Audit and Tivoli Security Information and Event Manager are intended to implement a better overall auditing solution for the z/OS platform – one that the internal audit department themselves can access to free up the security and systems programming teams for more appropriate work than preparing reports.

Once again, this work must be approached in phases. In this case, the following two phase approach is used:

Phase 1 Establish a better RACF resource owning and classification structure.

Phase 2 Employ this structure using the zSecure Command Verifier product to segregate various administrative roles.

The following steps will be performed, often somewhat in parallel, across the two phases:

1. Establish different target roles for administrators.
2. Establish a resource ownership structure.
3. Establish resource and profile naming conventions to be enforced.
4. Establish RACF rules used by zSecure Command Verifier to exploit the new structure.
5. Activate zSecure Command Verifier and test the new processes.
6. Refine as necessary.

2.2.1 Phase 1 – Design the new structure, roles, and workflow

This first phase can almost look like a research project, determining what roles are currently in place, what security administration practices are actually used, and designing a better way to segregate these.

We start by noting that all security administrators, from the most senior to the newest junior member of the team, employ the highest level RACF privileges at present. There is no doubt that many of these staff have no requirement to exercise these levels of privilege on a day to day basis, so we are already in violation of the first principle of *least privilege*.

Our first task is to discover exactly what the various administrators are charged with as their daily duties, and determine the minimum privileges required to allow these duties. We find that the administrators fall into some distinct categories:

1. Basic user provisioning and password administrators.
2. Administrators who work with RACF functional resource definitions, mostly administering access lists of these resources.
3. Administrators who work with the RACF group structure and who define new resources or activate new classes as necessary. They may also be called upon to issue occasional changes to overall system settings via the RACF SETROPTS command.
4. A special case, the most senior administrator and overall team leader, who can perform any and all of the previously described roles.

This seems a reasonable set of target roles, satisfying the first point in our task list. We now need to define these roles within RACF structures and control them using zSecure Command Verifier.

The next two steps, establish a resource ownership structure and establish resource and profile naming conventions to be enforced, are closely tied together. From the list of administrator categories, we can readily split out the first category, those who perform basic user provisioning and password management.

A simple approach for this first group of administrators is to define a RACF group as the owner of all userids, or even more refined, the owner of the part of the RACF group tree that maps the organizational chart and contains subgroups that own all STAFF userids. In this part of the RACF group tree there should be no IT infrastructure related userids, such as for batch processing, started tasks, or any system or application userid definitions. Effectively we

are creating a *sectioned off area* within RACF exclusively for the management of actual HR-recognized staff only.

We can now use zSecure Command Verifier to ensure that the *user administrators* cannot apply any RACF commands to users outside this section of the group tree, shown in Figure 2-4. We have our first segregation example already.

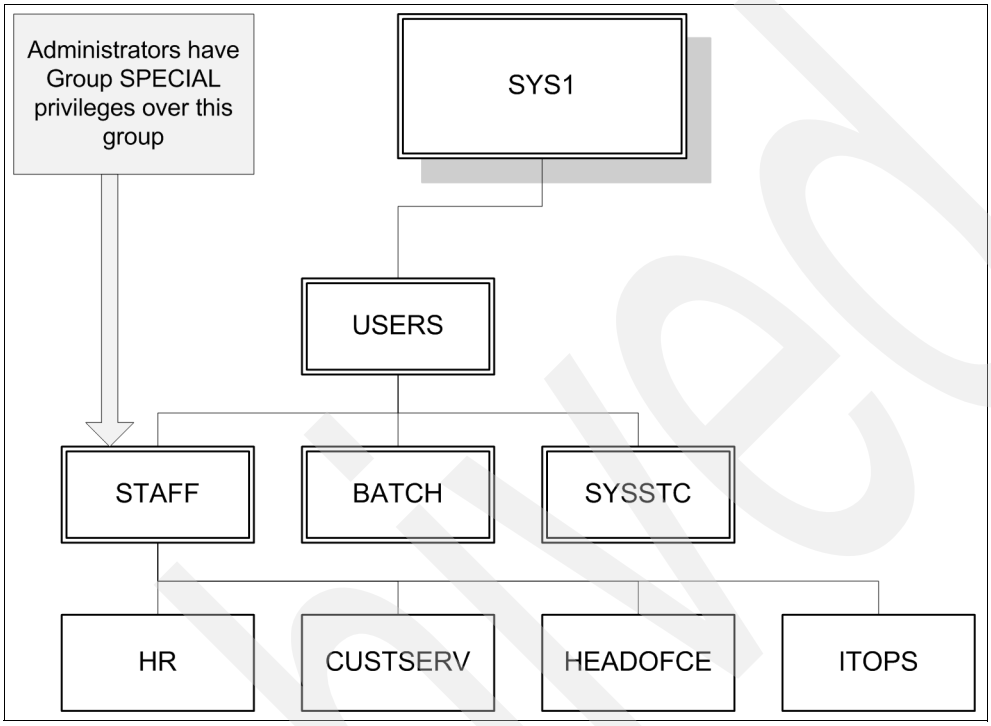

Figure 2-4 STAFF section of RACF group tree

We now need to define a management structure for our second major grouping of administrators, those who manage resource access lists in general. Once again we turn to the RACF concept of resource ownership. We define a new group to RACF, and make this group the *RACF owner* of all functional profiles that this group is expected to manage. We can now use zSecure Command Verifier to restrict commands executed by these administrators from applying to any resource not directly owned by this new group. We have effectively broken this group of administrators away from any user administration capability, and restricted them to work only on functional profiles (for example, FACILITY, OPERCMDS, SDSF, and so on). Segregation is now starting to become real.

We use additional zSecure Command Verifier built-in capabilities to prevent all administrators from defining RACF profiles, userids, or groups that fall outside our accepted naming conventions. This is done via the definition of RACF profiles themselves – getting RACF itself to control or limit what RACF administrators can do. We call these zSecure Command Verifier RACF profiles *control profiles* because they are referenced by zSecure Command Verifier to control what other users can and cannot do with respect to other RACF commands.

Herein lies the real trick to this process. By using actual RACF profiles, referenced by zSecure Command Verifier, we are controlling RACF administrators' actions. But what if an administrator alters one of the control profiles to allow himself access to some function we wanted segregated?

To properly address this concern and risk in the resource administration area we must have additional segregation. We create a group specifically to own the zSecure Command Verifier

control profiles, as shown in Figure 2-5. We use zSecure Command Verifier again to restrict who can issue RACF commands against the control profiles, and we give this authority to one or two of the relatively junior resource or user administrators. We suggest that you pick these staff carefully because they have some control over the system now above and beyond their normal duties. These staff become the “extra eyes” that have to participate in any change of RACF definitions that require to be outside the normal standards, and therefore need to override the zSecure Command Verifier control profiles. They are the watchers of the watchers. Ideally these staff are being groomed to take on higher level responsibilities in IT security, and should have demonstrated the skills and attitudes required to do so already.

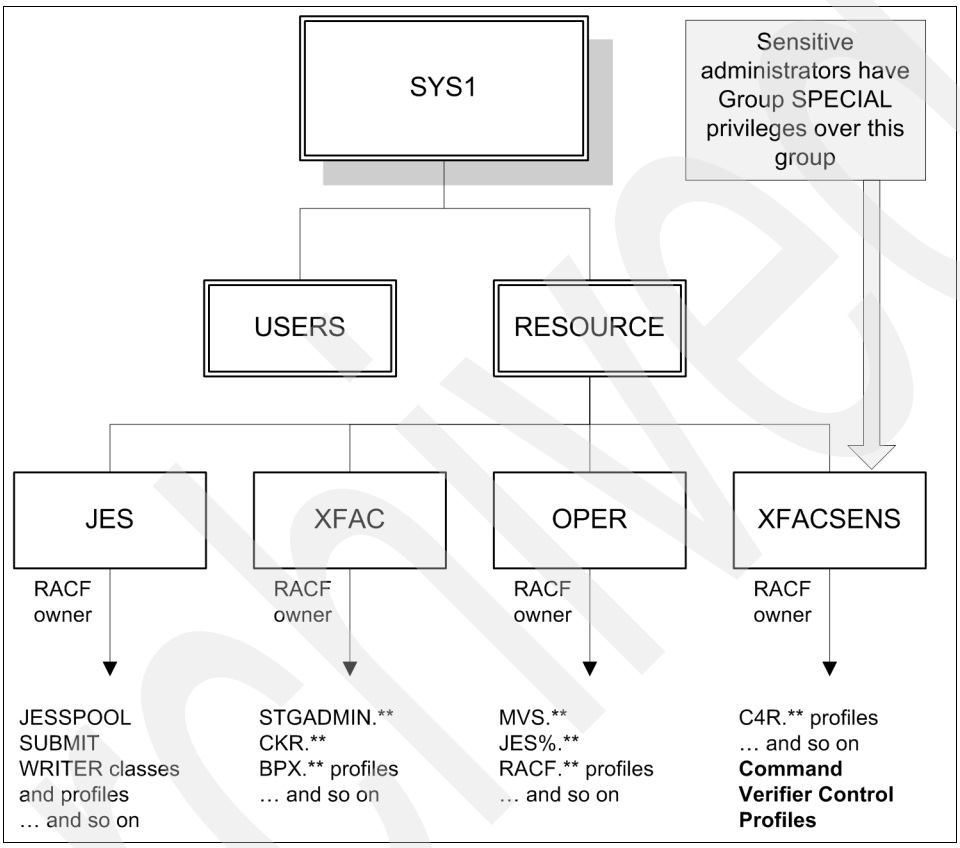

Figure 2-5 Resource owning section of RACF group tree

The beauty of this solution is that the watchers do not require high level RACF privileges; a simple Group SPECIAL, or direct resource ownership, will be enough for them the manage the zSecure Command Verifier control profiles. Should they attempt to use the control profiles to grant themselves higher level access, their native RACF access is not high enough for them to do any real damage themselves. All they can do is allow the most senior groups of administrators (those listed as 3. and 4. on page 25) to bypass the controls, should such an action have the proper management approval. By themselves they can do nothing untoward. True segregation has been achieved. It now takes collusion on behalf of at least one senior and one junior administrator before site accepted RACF standards, and other established segregation, can be subverted.

In a similar manner you could set up differing security protocols using this segregation feature of zSecure Command Verifier. Rather than have a senior and junior administrator required in order to bypass zSecure Command Verifier controls in exceptional situations, you may require two senior administrators, or one senior plus a member of the IT change executive committee.

2.2.2 Phase 2 – Implement and test the segregation capabilities

During the implementation, new RACF groups are created to own the newly defined sections of the RACF database structure using zSecure Admin. The next task, moving all existing user owning or resource owning groups under the new structural groups, is also a trivial task when using zSecure Admin.

It is these *trivial tasks* where the real power and benefits of zSecure Admin are realized. A task such as identifying and moving all resource owning groups to a new structural owning point might take weeks using traditional tools, but is reduced to minutes by employing zSecure Admin. With zSecure Admin for RACF there is no excuse to live with outdated RACF group tree structures that no longer map to your organization chart or serve your business IT security needs.

Establishment of the zSecure Command Verifier control profiles is also implemented using zSecure Admin, as is the extra segregation for the control profiles themselves.

Another feature made possible by zSecure Admin is the ability to *mark* the highly sensitive zSecure Command Verifier control profiles as requiring *multi-level authorization* before changes can be made to them. This is yet another example of the sophisticated segregation capabilities provided by the Tivoli Security for z/OS offering.

After implementing the zSecure Command Verifier control profiles, and segregation of the RACF resources, we can now proceed to removing the System SPECIAL privilege for almost all administrators, replacing this privilege with RACF Group level SPECIAL, and testing the final implementation.

In this implementation we left the senior team leader with full User SPECIAL privileges. Although this might not be required, it was felt a prudent mitigation to any negative impact experienced by the new segregation regime.

2.2.3 Scenario 2 conclusions

The issue of segregation in RACF command authorities within medium to large teams of z/OS RACF security administrators is addressed to a certain extent by the native capabilities of RACF Group level SPECIAL authority. zSecure Command Verifier combined with zSecure Admin can take this native capability to a new level, where administrators can be classified and their actions limited to specific RACF profiles or types of profiles in a safe and RACF-controlled manner.

Traditionally, many organizations eschew the use of Group SPECIAL because it is difficult to administer correctly and hard to ensure that the controls are working as designed. zSecure Admin and Command Verifier sidestep these issues by employing multiple fine-grained levels of identifying which profiles any specific administrator can work with. This is a much more flexible approach than the native RACF group scope concept, and can avoid the organization being forced to rigorously manage their RACF group tree. zSecure can provide both the necessary controls and a great degree of flexibility in the manner in which you choose to employ them.

2.3 Demonstrate audit readiness and policy-based management of security access rights

In our final customer scenario we discuss a manufacturing and distribution company that is required by legislation to protect customer data from misuse, while at the same time requires the use of data in competitive and sales programs. The two issues at heart here are *Who are my authorized users with respect to this sensitive data?* and *How do I show that the data is only being used by these authorized users?* To address these issues the organization has chosen a Security Information and Event Management (SIEM) oriented approach, to be applied across all platforms in the company.

In order to centralize and control all necessary audit and security-related data, a single integrated solution that addresses SIEM across multiple platforms is desired. The IBM Tivoli Security Information and Event Manager offering was chosen to manage audit logs and security information across all platforms. The company chose to extend their Tivoli Security Information and Event Manager purchase with the competitively priced IBM Tivoli Security Management for z/OS offering.

In this scenario we focus on implementing the Tivoli Security Information and Event Manager component of the Security Management for z/OS offering. This deployment demonstrates the closed loop auditing enabled by Tivoli Security Information and Event Manager, where the analysis of actual user activity logs is employed as part of the continuous improvement cycles that should be a part of any IT security management function. The manufacturing and distribution company intends to add extra functions from the Security Management for z/OS offering, including zSecure Admin, Audit, and Command Verifier controls, at a later stage of their security improvement program.

As usual, we use a phased approach to deploy Tivoli Security Information and Event Manager. Phase 1 consists of the analysis of currently defined roles and responsibilities for the user community, as well as a rudimentary classification of the data managed by the z/OS environment. In phase 2 we explore the installation and configuration of the Tivoli Security Information and Event Manager software, and in phase 3 we demonstrate the kinds of reports and analysis that Tivoli Security Information and Event Manager can provide and how to use this to improve the underlying z/OS security configuration.

Phase 1 Information discovery.

Phase 2 Installation and configuration.

Phase 3 Closed loop auditing with RACF.

2.3.1 Phase 1 – Information discovery

The secret of a successful Tivoli Security Information and Event Manager installation lies in the analysis and planning stages that occur long before any software is actually installed. The more thoroughly you perform this step, the easier and more effective the actual deployment of Tivoli Security Information and Event Manager will be. The realization of business benefits will be proportional to the thoroughness with which you engage in this planning process.

In our manufacturing company case study, we are fortunate because the target system for audit log analysis is RACF. RACF employs the concept of groups, these being collections of users with shared access requirements. Tivoli Security Information and Event Manager can use these groups of users as a primary method of user classification. This assumes that the user groups defined to RACF actually do map to common access requirements, and if not, the organization has the capabilities to use zSecure Admin to help redress this RACF configuration issue. If the user groups are not well defined in RACF and you do not wish to

change this, it is still possible to use Tivoli Security Information and Event Manager meta-data instead to properly define the various categories of users found on a RACF system.

User classification template

In our scenario, RACF groups are well defined and managed already, so we can use these to classify our user community into various types. Analysis reveals that users on this z/OS system fall into the following major categories:

1. Finance
2. Administrators
3. Division managers
4. Sales
5. IT support staff
6. HR
7. Marketing
8. Users (all users but administrators)
9. Other, including system, application, started task, or other IT system related functional userids (not used by staff)

Data classification template

We now have a basic user classification that Tivoli Security Information and Event Manager can use. The next step is to similarly classify the data managed by the system. Once again, we are fortunate that z/OS and RACF are employed, because the typically strong naming conventions used in z/OS environments make it much easier to identify and classify this data. We identify the following major types of data on the system:

1. Finance data
2. HR data
3. System data
4. Customer data
5. System test data
6. Other, unclassified data

Given this level of detail, we now have a functional, albeit relatively simple, framework within which to deploy the Tivoli Security Information and Event Manager solution. It is possible to define a simple framework at first, and obtain some basic reporting against this, then later refine the framework as you identify more fine-grained categorization of users and data over time.

2.3.2 Phase 2 – Installation and configuration

Tivoli Security Information and Event Manager is installed on Windows® servers and connected to the z/OS environment using a Tivoli Security Information and Event Manager Agent that is installed on the z/OS lpar that is being audited. The Agent for z/OS is installed using standard z/OS SMPE processes. SMF data is gathered from the z/OS system on a periodic schedule (defined in the Windows Management Console for Tivoli Security Information and Event Manager) using Remote Procedure Call (RPC) to the z/OS UNIX System Services component. z/OS UNIX System Services then calls the CARLa processing engine (installed in a traditional z/OS executable library) to prepare the SMF for encrypted and compressed transmission to the centralized log management repository on a Tivoli Security Information and Event Manager server.

After Tivoli Security Information and Event Manager has been successfully installed and SMF data received correctly from the z/OS lpar, we can define the acceptable data use policy that we wish to have in place for our z/OS users. This is done using a browser interface to the

Tivoli Security Information and Event Manager server known as the Compliance Dashboard. Using the Compliance Dashboard you can create and archive a series of policies, then choose which policy is to be used when mapping the collected data. With more than one policy defined, a useful technique is to compare the compliance level achieved between any two or more policies. Because Tivoli Security Information and Event Manager applies its reporting only after mapping source data against a policy, you can generate reports from historical policies using either historical or current source data. This approach can help to demonstrate both that your policy is improving over time, and your level of compliance with policy is increasing.

We can immediately see from the exercise of user and data classification that there are several rules we have to implement in the policy:

- Finance and HR users (user types 1 and 6 on our list) can access production Finance and HR data.
- Application and task-related users (user type 9) can access all data.
- End users (type 8) can access production HR, Finance, and Customer data only.
- Sales users (type 4) can only access production data relevant to their roles (Customer).
- System administrators (type 2) can only access System, Test, and Other data.
- IT support staff (type 5) can only access some System, Test, and Other data.
- Marketing end users (type 7) access production Finance, Customer, and Other data only.
- Divisional management (type 3) have no access to production data. (Executive access to data is via management reporting; these users have no requirement to directly access z/OS data).

To define acceptable use policies we can draw the matrix shown in Table 2-1, which can help us better understand the user and data classification relationship in defining who should be allowed to access what data.

Table 2-1 User classification versus data classification matrix

	Finance	HR	System	Customer	System test	Other
Finance	OK	OK	OK	x	x	OK
Administrators	x	x	OK	OK	OK	OK
Management	x	x	x	x	x	OK
Sales	x	x	x	OK	x	OK
IT staff	x	x	OK	x	OK	OK
HR	x	OK	x	x	x	OK
Marketing	OK	x	x	OK	x	OK
End users	OK	OK	x	OK	x	OK
Other	x	x	x	x	OK	x

While it seems that we will need many policy rules to cover all relationships in the table, the design of Tivoli Security Information and Event Manager only requires us to specify the allowable access behavior in the policy. Any user access to data that is not specifically mentioned as part of the policy is considered a violation and will be reported. We need to define a rule for each of the OK boxes in the matrix, giving us a total of 24 policy rules at this stage. We can now define these policies to our z/OS audit policy in Tivoli Security Information and Event Manager using the Policy Explorer.

The next time we run the Tivoli Security Information and Event Manager mapping against the collected SMF data, we can see the ad hoc results in the Compliance Dashboard. They depict the areas where the organization has an immediate need to investigate, using the meta-data we entered to map our various types of users and data, as shown in Figure 2-6.

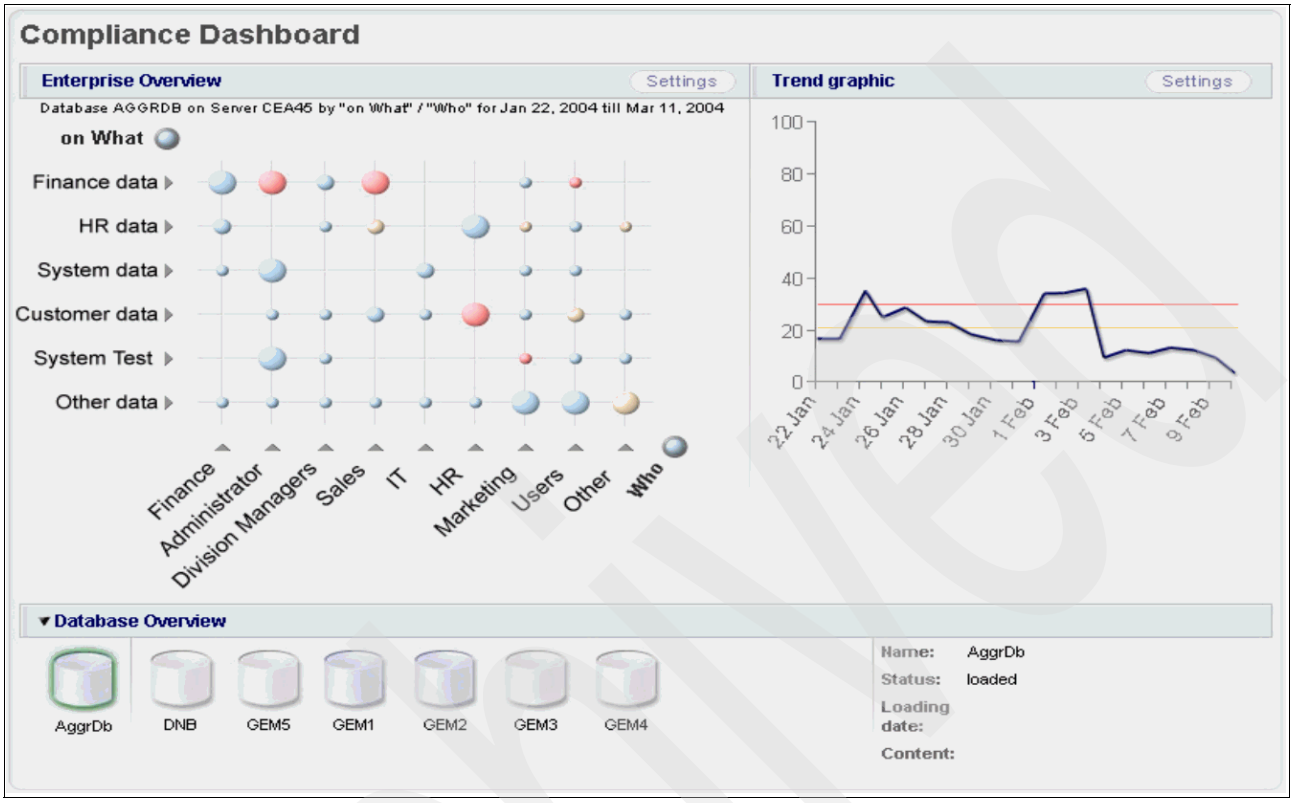

Figure 2-6 Compliance Dashboard Enterprise Overview

Tivoli Security Information and Event Manager highlights where any type of user accesses any type of data in a manner not allowed by the policy. In the report shown in Figure 2-6 we can immediately see the following problem areas:

- Administrators have accessed financial data.
- Users in the Sales role have accessed financial data.
- HR users have accessed customer-related data.

In the next step these exceptions have to be investigated for malicious intent. If there was no malicious intent in accessing this data, the policies may need to be refined. In the investigations we can use Tivoli Security Information and Event Manager to drill down to the level of detailed SMF information in the log records that demonstrate the breach of policy and the RACF profiles involved. We can use this information to help us rectify the situation in the RACF database that allowed this non-compliant use of data. An example event list is shown in Figure 2-7 on page 33.

Event list by Event type Grant : Profile / Success
Database OS390 on Server EPRORADB

Time zone: Event time zone

Severity	When	#	What	Where	Who	From Where	On What	Where To
40	Wed Feb 20 2002 23:16:12 GMT+01:00	1	Grant / Profile : Success	DINO	STRTASK	DINO	PROFILE : - / CRMQARUN.F.**	DINO
40	Wed Feb 20 2002 23:16:12 GMT+01:00	1	Grant / Profile : Success	DINO	STRTASK	DINO	PROFILE : - / CRMQARUN.F.**	DINO
40	Thu Feb 21 2002 08:18:17 GMT+01:00	1	Grant / Profile : Success	DINO	RCCSLIN	DINO	PROFILE : - / CRMBRG%.SUBMIT	DINO
40	Thu Feb 21 2002 09:01:03 GMT+01:00	1	Grant / Profile : Success	DINO	CRMBMR1	DINO	PROFILE : - / $C2R.OPTION.RA.H.2	DINO
40	Thu Feb 21 2002 09:01:03 GMT+01:00	1	Grant / Profile : Success	DINO	CRMBMR1	DINO	PROFILE : - / $C2R.OPTION.RA.H.3	DINO
40	Thu Feb 21 2002 09:01:03 GMT+01:00	1	Grant / Profile : Success	DINO	CRMBMR1	DINO	PROFILE : - / $C2R.OPTION.RA.H.8	DINO
40	Thu Feb 21 2002 10:29:14 GMT+01:00	1	Grant / Profile : Success	DINO	CRMBMR1	DINO	PROFILE : - / $C2R.OPTION.HD.3	DINO

Figure 2-7 Detailed event list

2.3.3 Phase 3 – Closed loop auditing with RACF

Now that Tivoli Security Information and Event Manager is effectively deployed and producing audit reports of z/OS activity, we review these reports and investigate whenever a deviation from the policy is highlighted. Then we need to go back into RACF and remove whatever access permissions actually allowed that user to breach the policy.

At this stage we are using our collected log data to drive security management and improvements in the security system itself. This answers the perennial question of all security administrators, *How do I know that the access rights granted within my security database are actually appropriate to the job functions of my users?* Tivoli Security Information and Event Manager can do this by providing the feedback from actual user activity, against your declared policies, to help improve security definitions.

In addition, Tivoli Security Information and Event Manager provides a reporting engine that uses our collected log data, and is readily accessible to non-technical staff using browser-based user interfaces. From this point, our auditors can perform detailed inquiries they have always wanted to themselves, unaided by and independent of our specialized technical staff. This is a much more ideal situation than relying on technical staff to report on their own activity when questioned by auditors. Figure 2-8 on page 34 shows a browser-based list of ISO 17799 related reports that can be easily selected by non-technical auditors.

ISO 17799 Regulation Reports

▼ ISO 17799

Title	Description
ISO 17799 (10.4.1) Control of Operational Software	Changes that have been made to the operational software
ISO 17799 (10.4.2) System Test Data	Access of the System test data
ISO 17799 (10.4.3) Source Code Access	Source code accessed by whom
ISO 17799 (10.5.4) Covert channels and trojan code	Exceptions found from anti virus software
ISO 17799 (12.1.3) HR Report	Access to HR data by Who, What and When
ISO 17799 (12.1.4) Data Access	Data accessed by Who , What and onWhat
ISO 17799 (12.1.5) Prevention of misuse of information processing facilities	Misuse of information processing facilities for non-business purposes
ISO 17799 (12.1.7.1) Rules for evidence	evidence collected conforms to the rules laid down in the relevant law
ISO 17799 (12.2.2) Technical Compliance Checking	Show that the systems have been checked for compliance with security implement...
ISO 17799 (12.3.2) Protection of system audit tools	shows if the audit software or data has been misused or compromised
ISO 17799 (3.1) Security Policy report	No description given
ISO 17799 (5.1) Accountability for Assets	Report showing who owns what assets
ISO 17799 (5.1, 5.2) Classification report	No description supplied
ISO 17799 (6.3) Security Alert report	The response taken to a security incident and/or malfunction
ISO 17799 (8.1.2) Operation Change Control Report	Operational changes to the production environment
ISO 17799 (8.1.3) Incident managements procedures	procedures that are in place in response to security incidents
ISO 17799 (8.1.6) External Contractors Report	External Contractors have accessed
ISO 17799 (8.3) Malicious Attacks	Malicious software activities
ISO 17799 (8.4) Log Archive report	*** LOG ARCHIVE ***
ISO 17799 (8.4) Log Storage report	No description given
ISO 17799 (8.5) Network Management	Connections to the network by users and other systems

Figure 2-8 Easily accessible list of ISO 17799 reports

2.3.4 Senario 3 conclusions

Tivoli Security Information and Event Manager provides capabilities to automate log collection and storage from a wide variety of platforms. In this particular case we are only using it to audit the z/OS platform, but this could be extended for a wider enterprise deployment at any time.

The steps that were implemented to have Tivoli Security Information and Event Manager provide effective audit reports for us are:

1. Collect the SMF data from z/OS using automated log collection with the Tivoli Security Information and Event Manager Agent for z/OS.
2. Define a Tivoli Security Information and Event Manager policy.
3. Define the user and data groupings that our initial analysis revealed for these policies.
4. Create the policy rules within the new policy, based on the acceptable use of the defined groupings of data, by the defined groupings of users.
5. Create any required attention rules for this policy and associate alerts if desired.
6. Load the collected data. Tivoli Security Information and Event Manager maps the rules contained in the policy against the data, and immediately highlights any deviance from the acceptable use policy.

For more detailed information on deployment and configuration of Tivoli Security Information and Event Manager check “Related publications” on page 37.

2.4 Conclusion

In this chapter we looked at three quite different customer scenarios, all employing various components from the Tivoli Security Management for z/OS solution bundle. Each scenario focused on different business and technical requirements, remembering that each customer has at their disposal the entire capabilities of the other scenarios when they choose to further exploit this suite of integrated products.

As a brief summary, we were able to demonstrate how each customer addressed the following concerns:

- Satisfy internal and external auditors that the z/OS security environment is being appropriately managed and secured.
- Implement baseline improvements and automated tracking of baseline variations using zSecure Admin and Audit to provide protection for critical RACF resources from abuse by privileged insiders.
- Provide segregation within administrative teams, implemented in combination with zSecure Command Verifier and zSecure Admin to ease the effort and risk.
- Demonstrate audit readiness and policy-based management of security access rights by deploying Tivoli Security Information and Event Manager, a simple audit profile for z/OS able to be expanded upon and refined, while providing immediate ROI to auditors and feedback into continuous improvement in security implementation.

The IBM Tivoli Security Management for z/OS v1.11 solution bundle provides a comprehensive and modern approach to the traditional issues encountered in securing a z/OS RACF based system.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 37. Note that some of the documents referenced here may be available in softcopy only.

- *IT Security Compliance Management Design Guide with IBM Tivoli Security Information and Event Manager*, SG24-7530
- *z/OS Mainframe Security and Audit Management using IBM Tivoli zSecure*, SG24-7633
- *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, SG24-6014

Online resources

For further information on the IBM Tivoli Security Management for z/OS V1.11 suite of products refer to the following documentation.

- For the Tivoli zSecure Suite Version 1.11 Information Center go to:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.zsecure.doc/welcome.html

- More information about the Tivoli zSecure Suite is also available here:

  http://www.ibm.com/software/tivoli/products/zsecure/

- For the Tivoli Security Information and Event Manager V2.0 Information Center go to:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tsiem.doc/welcome.html

- More information about Tivoli Security Information and Event Manager is also available here:

  http://www.ibm.com/software/tivoli/products/security-info-event-mgr/index.html

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 publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Empowering Security and Compliance Management for the z/OS RACF Environment

Using IBM Tivoli Security Management for z/OS

Administering your mainframe security while helping to reduce administration time, effort, and costs

Leveraging seamless integration of audit and compliance efforts

Increasing mainframe security while decreasing complexity

Every organization has a core set of mission-critical data that requires protection. Security lapses and failures are not simply disruptions, they can be catastrophic events with consequences felt across the enterprise. The inadvertent mistakes of privileged users alone can result in millions of dollars in damages through unintentional configuration errors and careless security commands. Malicious users with authorized access can cause even greater damage. As a result, security management faces a serious challenge to adequately protect a company's sensitive data. Likewise, IT staff is challenged to provide detailed audit and controls documentation in the face of increasing demands on their time.

Automation and simplification of security and compliance processes can help you meet these challenges and establish effective, sustainable user administration and audit solutions. This includes security database cleanup, repeatable audit of configurations and settings, and active monitoring of changes and events. IBM Tivoli Security Management for z/OS V1.11 provides these solutions to help enhance the security of mainframe systems through automated audit and administration.

In this IBM Redpaper document we discuss how Tivoli Security Management for z/OS allows you to submit mainframe security information from z/OS, RACF, and DB2 into an enterprise audit and compliance solution and how to combine mainframe data from z/OS, RACF, and DB2 with that from other operating systems, applications, and databases in order to provide the ability to capture comprehensive log data, interpret that data through sophisticated log analysis, and communicate results in an efficient, streamlined manner for full enterprise-wide audit and compliance reporting.

REDP-4549-00

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