

Security on z/VM

z/VM in the enterprise security solution
- Sample scenarios

z/VM security features, LDAP,
RACF

Cryptography with Linux
guests on z/VM



Paola Bari
Helio Almeida
Gary Detro
David Druker
Marian Gasparovic
Manfred Gnirss
Jean Francois Jiguet
Michel Raicher

Redbooks



International Technical Support Organization

Security on z/VM

November 2007

Archived

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

First Edition (November 2007)

This edition (SG24-7471-00) applies to the z/VM V5R3.

© Copyright International Business Machines Corporation 2007. All rights reserved.

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

Contents

Notices	vii
Trademarks	viii
Preface	ix
The team that wrote this book	ix
Become a published author	xi
Comments welcome	xi
Chapter 1. z/VM and security	1
1.1 Introduction to z/VM virtualization	2
1.2 z/VM security features	4
1.2.1 The SIE instruction	5
1.2.2 System z cryptographic solution	5
1.2.3 Intrusion detection	7
1.2.4 Accountability	7
1.2.5 Certification	7
1.2.6 Debugging in a virtual environment	9
1.2.7 Virtual networking	9
1.2.8 Compliance to policy	10
1.3 Additional features	16
1.3.1 The Resource Access Control Facility	16
1.3.2 LDAP	17
1.3.3 z/VM VSWITCH networking	18
Chapter 2. RACF feature of z/VM	21
2.1 RACF z/VM concepts	22
2.2 Installing and configuring RACF	22
2.2.1 Post-installation tasks	23
2.2.2 Build the RACF enabled CPLOAD MODULE	33
2.2.3 Update the RACF database and options	36
2.2.4 Place RACF into production	40
2.2.5 Using HCPRWAC	41
2.3 RACF management processes	44
2.3.1 DirMaint changes to work with RACF	44
2.3.2 RACF authorization concepts	48
2.3.3 RACF passwords and password phrases	48
2.3.4 Adding virtual machines and resources to the system and the RACF database ..	54
2.3.5 Implementing LOGONBY with RACF	61
2.3.6 Managing VSWITCH and Guest LANS	64
2.3.7 Managing RSCS nodes	67
2.4 RACF security labels	72
2.4.1 Security labels overview	72
2.4.2 Creating a security label	73
2.4.3 Security label naming restrictions	73
2.4.4 Security label NONE	74
2.5 RACF auditing	74
2.5.1 Enabling auditing	74
2.5.2 RACF Data Security Monitor Utility (RACDSMON)	76
2.5.3 RACF SMF Data Unload Utility (RACFADU)	84

2.5.4 RACF report writer utility (RACFRW)	90
2.6 RACF database backup	96
2.6.1 RACF database verification utility program (IRRUT200)	96
2.6.2 RACF database Split/Merge/Extend utility program (IRRUT400)	97
2.6.3 The RACF database unload utility (IRRDBU00)	104
Chapter 3. z/VM LDAP server	111
3.1 LDAP terminology	112
3.1.1 LDIF files	114
3.2 z/VM LDAP	115
3.2.1 LDAP client	115
3.2.2 z/VM LDAP back end services	116
3.2.3 Native authentication	117
3.2.4 Multiple back end services	118
3.2.5 Multiple servers	118
3.3 Installing z/VM LDAP server	119
3.3.1 Implementing a new TCP/IP stack	120
3.3.2 Creating the LDAP server	123
3.3.3 Creating a BFS file pool VMSEVL	126
3.3.4 Configuring the LDAP server	129
Chapter 4. Implementing Pluggable Authentication Modules LDAP for Linux servers	137
4.1 PAM and Name Service Switch	138
4.1.1 PAM configuration files	138
4.1.2 Linux Name Service Switch	138
4.2 Configuring PAM LDAP and NSS	138
4.2.1 SUSE Linux	138
4.2.2 Red Hat Linux	142
4.3 Changing the password	144
Chapter 5. Enterprise integration	145
5.1 Using a central z/VM LDAP server	146
5.1.1 LDAP architecture choices	146
5.1.2 Configuring z/VM LDAP server	147
5.1.3 Verifying the LDAP server	150
5.1.4 Loading Linux schema and sample data in the LDBM	151
5.1.5 Adding a Linux user in z/VM LDAP	152
5.1.6 Adapting Linux servers to use LDAP service	153
5.1.7 Linux logon test	154
5.1.8 Summary	154
5.2 Sharing RACF database with another z/VM system	155
5.2.1 Preparing SYSTEM1 for RACF database sharing	157
5.2.2 Re-instating IBMUSER	157
5.2.3 Directory entries	158
5.2.4 DES encryption	161
5.2.5 Initializing the RACF database on SYSTEM2	162
5.2.6 Database verification	166
5.2.7 Summary	170
5.3 Sharing a RACF database with z/OS	171
5.3.1 Planning RACF/VM installation	172
5.3.2 Installing RACF on z/VM	172
5.3.3 Summary	177

5.4 Using a central z/OS IBM Tivoli Directory Server	177
5.4.1 LDAP architecture	177
5.4.2 Implementing the IBM Tivoli Directory Server for z/OS	178
5.4.3 Verifying the LDAP server	181
5.4.4 Loading the schema	181
5.4.5 Loading the Linux specific schema	181
5.4.6 Adding a Linux user in IBM Tivoli Directory Server for z/OS	182
5.4.7 Adapting Linux servers to use LDAP service	183
5.4.8 Linux logon test	184
5.4.9 Summary	184
5.5 Synchronizing LDAP/RACF database with IBM Tivoli Directory Integrator	184
5.5.1 Architecture	186
5.5.2 RACF password mirroring	188
5.5.3 LDAP backends	189
5.5.4 z/VM LDAP configuration	189
5.5.5 IBM Tivoli Directory Server for z/OS configuration	191
5.5.6 Verifying the GDBM backend	193
5.5.7 RACF changes	194
5.5.8 IBM Tivoli Directory Integrator configuration	196
5.5.9 Summary	199
Chapter 6. Cryptography on z/VM	201
6.1 Secure communication to the z/VM System using SSL	202
6.2 Preparing System z for the hardware encryption support	204
6.2.1 Planning to use Crypto Express2	206
6.2.2 Customize the partition image profile for cryptographic usage	209
6.2.3 Configuration of the cryptography adapter as coprocessor or accelerator	212
6.3 z/VM definitions	220
6.3.1 Setup for a Linux guest to use cryptography cards	221
6.4 Using cryptography hardware support with Linux	231
6.4.1 Verify availability of cryptography hardware support	240
6.4.2 Using OpenSSL	242
6.4.3 Example: Configuring Apache 2.0 for using HW support	249
6.4.4 Example of OpenSSH	254
6.4.5 Using PKCS#11 API	254
6.4.6 Example: configure IBM HTTP server for using HW support	259
6.4.7 In-kernel cryptography with Linux kernel 2.6	262
6.4.8 Using secure key encryption with Linux: an outlook	264
Chapter 7. IBM Tivoli zSecure for z/VM RACF	271
7.1 Consul InSight Suite benefits	272
7.2 Tivoli zSecure Pro Suite	272
7.3 Introducing Tivoli zSecure	274
7.3.1 CARLa	274
7.3.2 Tivoli zAudit	275
7.3.3 Tivoli zAlert	277
7.3.4 Tivoli zAdmin	277
7.4 Tivoli zSecure installation	278
7.4.1 Creating the load library	281
7.4.2 Place Tivoli zSecure in production	282
7.4.3 Applying maintenance	282
7.4.4 Installation verification	283

7.5	Configuring Consul zSecure	284
7.5.1	Making the software (CKREVM EXEC) available to VM/CMS users	284
7.5.2	Configuring the parm file	288
7.5.3	IOCONFIG file	289
7.5.4	Installing the license file	289
7.5.5	Changing the default setup	290
7.5.6	Checking the zCollect function	292
7.5.7	Fresh CKFREEZE and UNLOAD	293
7.5.8	TCP/IP domain name resolution	293
7.6	Examples of some reports generated by Consul zSecure	293
7.7	Sample UAUDIT list	293
7.8	Personalized reports for RACF users with special and operations authority	296
Appendix A. DirMaint implementation		301
A.1	DirMaint implementation and configuration	302
A.2	DirMaint installation	302
A.2.1	Completing your installation of DirMaint	303
A.2.2	Tailoring the DirMaint installation	304
A.2.3	Placing DirMaint into production	305
A.3	DirMaint tailoring	307
A.4	DirMaint testing and operations	312
A.4.1	Adding virtual machines	312
A.4.2	Updating EXTENT CONTROL	314
A.4.3	Adding MDISK to a Virtual Machine	315
A.5	Conclusion	317
Appendix B. RACF procedural checklist		319
B.1	RACF installation steps	320
Appendix C. Additional material		321
	Locating the Web material	321
	Using the Web material	321
	How to use the Web material	321
Related publications		323
	IBM Redbooks publications	323
	Other publications	323
	How to get IBM Redbooks publications	324
	Help from IBM	324
Index		325

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

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

Redbooks (logo) ®
eServer™
i5/OS®
z/Architecture®
z/OS®
z/VM®
z/VSE™
zSeries®
z9™
CICS®
DirMaint™
DB2®
DS4000™

DS6000™
DS8000™
Enterprise Storage Server®
ESCON®
FICON®
HiperSockets™
IBM®
IBMLink™
Lotus®
OS/390®
Parallel Sysplex®
Print Services Facility™

Processor Resource/Systems
Manager™
Redbooks®
RACF®
REXX™
System z™
System z9™
SQL/DS™
Tivoli®
VM/ESA®
VTAM®
WebSphere®
Workplace™

The following terms are trademarks of other companies:

Java, JavaScript, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

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

Preface

Discussions about server sprawl, rising software costs, going green, or moving data centers to reduce the cost of business are held in many meetings or conference calls in many organizations throughout the world. And many organizations are starting to turn toward System z™ and z/VM® after such discussions. The virtual machine operating system has over 40 years of experience as a hosting platform for servers, from the days of VM/SP, VM/XA, VM/ESA® and especially now with z/VM. With the consolidation of servers and conservative estimates that approximately seventy percent of all critical corporate data reside on System z, we find ourselves needing a highly secure environment for the support of this infrastructure. This document was written to assist z/VM support and security personnel in providing the enterprise with a safe, secure and manageable environment.

This IBM® Redbooks® publication provides an overview of security and integrity provided by z/VM and the processes for the implementation and configuration of z/VM Security Server, z/VM LDAP Server, IBM Tivoli® Directory Server for z/OS®, and Linux® on System z with PAM for LDAP authentication.

Sample scenarios with RACF® database sharing between z/VM and z/OS, or through Tivoli Directory Integrator to synchronize LDAP databases, are also discussed in this book.

This book provides information about configuration and usage of Linux on System z with the System z Cryptographic features documenting their hardware and software configuration.

The Consul zSecure Pro Suite is also part of this document: this product helps to control and audit security not only on one system, but can be used as a single point of enterprise wide security control. This document covers the installation and configuration of this product and detailed information is presented on how z/Consul can be used to collect and analyze z/VM security data and how it can be helpful in the administration of your audit data.

The team that wrote this book

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

Paola Bari is an Advisory Programmer at the ITSO, Poughkeepsie Center. She has 22 years of experience as a systems programmer in OS/390®, z/OS, and Parallel Sysplex®, including several years of experience in WebSphere® MQ and WebSphere Application Server.

Helio Almeida is an IBM Senior IT Specialist in Brazil. He has 23 years of experience in the VM field. He has worked at IBM for 30 years. He holds a degree in Computing Technology from Instituto Tecnológico de Aeronáutica (Aeronautics Technological Institute). His areas of expertise include z/VM and Storage for System z. He has written and delivered education and Customer support extensively on z/VM.

Gary Detro is a Texan, an IBM Senior IT Specialist in the USA. He has 17 years of experience in the VM field. He has worked at IBM for 38 years. He holds a degree in Business Administration from Texas A&M at Commerce. His areas of expertise include z/VM, VM Networking, Storage for System z and Open Systems. He has written and delivered education extensively on z/VM, TCP/IP, VTAM®, RSCS, RACF/VM, DirMaint™, Linux on System z, Enterprise Storage Server®, DS4000™, DS6000™, and DS8000™.

David Druker is a Certified Consulting IT Specialist at IBM. His current role is Tivoli Security Specialist for Americas TechWorks, where he focuses on identity integration using Tivoli Directory Integrator, Tivoli Directory Server, and Tivoli Identity Manager. David is known worldwide as an expert in Tivoli Directory Integrator and works actively with large customers, product support and development. He has over 20 years of broad technology experience that includes programming, scientific computing, systems administration and enterprise architecture. David holds a Ph.D. in Speech and Hearing Science from the University of Iowa.

Marian Gasparovic is an IT Specialist from IBM Slovakia. He worked as an Administrator for z/OS at Business Partner for six years. He joined IBM in 2004 and now works in Field Technical Sales Support for System z in the CEMA region as a member of a new workload team. His areas of expertise include Linux, z/VM and z/OS as well as System z hardware and storage solutions.

Manfred Gnirss is an IT specialist at the IBM Technical Marketing Competence Center (TMCC) and the Linux Center of Competence, Boeblingen, Germany. He holds a PhD in theoretical physics from the University of Tuebingen, Germany. Before joining the TMCC in 2000 he worked in z/VM and z/OS development for more than 12 years. Currently he is involved in several Linux for System z Proof-of-Concept projects and customer projects running at the TMCC.

Jean Francois Jiguet is a certified IBM IT specialist in system management from France. He has 27 years of experience in the computer industry field. His expertise area are VM, Linux, VSE and System z hardware. He has experience in teaching, products support and software implementation services.

Michel Raicher is a Senior Programmer in Tucson, Arizona. He has 20 years of experience in z/VM System Test. He has worked at IBM for 34 years. His areas of expertise include Networking (OSA and CISCO), z/VM Vswitch, TCP/IP, ISFC, MCDS, SFS, BFS, NFS and SCSI for z/VM. He has worked testing early versions of Linux on System z and z/OS Networking under z/VM. He has supported the z/VM Lab raised floor for many years, and worked on early Hardware testing.

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

Richard Conway, Roy Costa, Robert Haimowitz
International Technical Support Organization, Poughkeepsie Center

Jay Leiserson
IBM Austin

Arthur Winterling
IBM Boeblingen

Alan Altmark, Doug Barnhart, Karen Gardner, Michael Wilkins
IBM Endicott

Jack Hoarau, Yann Kindelberger, Patrick Kappeler
IBM Montpellier

Hans Schoone
IBM Netherlands

John Dayka, Peter Spera, Michael Tebolt, Ross Uhlfelder, Bruce Wells
IBM Poughkeepsie

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:
ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- Send your comments in an e-mail to:

redbooks@us.ibm.com

- Mail your comments to:

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

Archived

z/VM and security

This chapter discusses the security aspects of z/VM facilities and introduces how z/VM virtualization can provide a secure isolation between guests on System z. The chapter also discusses Resource Access Control Facility(RACF) and Lightweight Directory Access Protocol (LDAP) on z/VM and how you can use them to allow an enterprise wide point of control.

M has the System z hardware resources virtualization architecture built in the software through the concept of the virtual machine. z/VM Virtual Machine is also referred to as user IDs, guests or hosts.

z/VM can host multiple operating systems as guests of the virtual machines by emulating the System z hardware within that same hardware, while providing extra features and benefits that are implemented in software in a more cost effectively way.

Virtualization of the machine enables you to do the following:

- ▶ Manage many servers using universal management tools
- ▶ Reduce management costs
- ▶ Maximize the utilization of existing hardware

Like all System z operating systems, z/VM can run alone or share the mainframe with others by using the LPAR capabilities of System z. z/VM can either virtualize the same System z server it is running on, or emulate hardware features that are not necessarily installed on that particular model of server, and it can provide a customized environment for each guest.

z/VM provides System z features to the guest operating systems transparently and simultaneously, without the need of having a physical server per guest. At the same time, z/VM can isolate each guest operating system and handle access to real devices as needed.

The Start Interpretive Execution Instruction (SIE) available on the System z can create the environment for Virtual Machines (user IDs). Most of the of instructions are executed by the hardware with little z/VM intervention. The SIE is implemented on the System z by the Interpretive Execution Facility. The z/VM Control Program gets an interruption when the hardware detects conditions such as Timer expiration, unassisted Input Output operation (I/O), instructions that require some privileges and Program Interrupts. SIE also handles the usage of region, segment and page tables previously set up by the z/VM Control Program for the user ID. SIE instruction is available only on System z, and it is exploited by z/VM to decrease Control Program overhead.

In addition to SIE, some of the newer System z provide I/O assist functions to QUEUE DIRECT I/O(QDIO), used by System z adapters such as Networking and Fibre Channel Protocol (FCP) to assist reducing the amount of interrupts passed to the z/VM Control Program.

Resources that can be shared between guests are CPU, real memory, disk volumes, network adapters, printers, and devices specific to System z such as cryptographic cards. It also supports the latest Storage Area Network (SAN) and virtual tape library systems. For example, in the case of disk storage, z/VM is capable of partitioning a disk volume and assigning portions to each guest. Control of read-only and read-write access, or none at all, is at the discretion of the z/VM administrator.

The number of guests that z/VM can operate concurrently is limited only by the amount of resource available to the System z. This is in contrast to LPARs which have a fixed limit to the number of LPARs, dependent on the System z used.

Guests of a z/VM host run within user IDs, or just “ID”, for short. Typically, people logging on to z/VM are called “users”, whereas automated user IDs are known as “service machines” and run in a disconnected state, that means without a console or display terminal attached.

1.1 Introduction to z/VM virtualization

z/VM has the System z hardware resources virtualization architecture built into the software through the concept of the *virtual machine*. z/VM virtual machine is also referred to as *user IDs*, *guests*, or *hosts*. z/VM can host multiple operating systems as guests of the virtual machines by emulating the System z hardware within that same hardware, while providing extra features and benefits that are implemented in software in a more cost effective way.

Virtualization of the machine enables you to:

- ▶ Manage many servers using universal management tools
- ▶ Reduce management costs
- ▶ Maximize the utilization of existing hardware

Like all System z operating systems, z/VM can run alone or share the mainframe with z/VM, Linux for System z or z/OS using the LPAR capabilities of System z. z/VM can either virtualize the same System z server on which it is running or can emulate hardware features that are not necessarily installed on that particular model of server. It can also provide a customized environment for each guest.

z/VM provides System z features to the guest operating systems transparently and simultaneously, without having a physical server per guest. At the same time, z/VM can isolate each guest operating system and handle access to real devices as needed.

The *Start Interpretive Execution Instruction* (SIE) that is available on System z can create the environment for virtual machines (user IDs). Most of the of instructions are executed by the hardware, with little z/VM intervention. The SIE is implemented on the System z by the *Interpretive Execution Facility*. The z/VM Control Program gets an interruption when the hardware detects conditions such as Timer expiration, unassisted Input Output operation (I/O), instructions that require some privileges, and Program Interrupts. SIE also handles the use of region, segment, and page tables that were set up previously by the z/VM Control Program for the user ID. SIE instruction is available only on System z, and it is exploited by z/VM to decrease Control Program overhead.

In addition to SIE, some of the newer System z provide I/O assist functions to QUEUE DIRECT I/O (QDIO) that is used by System z adapters such as Networking and Fibre Channel Protocol (FCP) to reduce the amount of interrupts that are passed to the z/VM Control Program. Resources that can be shared between guests are CPU, real memory, disk volumes, network adapters, printers, and devices specific to System z such as cryptographic cards. It also supports the latest Storage Area Network (SAN) and virtual tape library systems. For example, in the case of disk storage, z/VM is capable of partitioning a disk volume and assigning portions to each guest. Control of read-only and read-write access, or none at all, is at the discretion of the z/VM administrator.

The number of guests that z/VM can operate concurrently is limited only by the amount of resource that is available to the System z. This is in contrast to LPARs which have a fixed limit to the number of LPARs that are dependent on the System z used.

Guests of a z/VM host run within user IDs, or just *ID* for short. Typically, people logging on to z/VM are called *users*, whereas automated user IDs are known as *service machines* and run in a disconnected state (which means without a console or display terminal attached).

z/VM user definition and part of z/VM security is accomplished by the z/VM *directory*. The directory is encrypted on the DASD to avoid hacking its contents. To allow multiple users to change and maintain the directory, your installation can use a directory Control Program such as *Directory Maintenance Facility* (DirMaint). You can configure DirMaint to allow specific levels of change control as needed to specifics administrators and to the users. For more information about DirMaint, go to Appendix A, “DirMaint implementation” on page 301.

z/VM isolation between users and security can be enhanced using RACF. RACF allows a deep and more flexible security implementation than z/VM controls. You can configure RACF to control z/VM commands and resources and to define who can access these resources and at what level of authority.

1.2 z/VM security features

z/VM is today's version of a hypervisor operating system. Virtual machine design began in the early 1960s, when IBM was exploring how to meet customer expectations using virtualization. The development of virtual machine was closely tied to the development of virtual storage, because they needed to operate together. The core of z/VM (that is, the hypervisor), is actually the *Control Program*. The Control Program creates and maintains virtual environments for virtual machines (guests).

Figure 1-1 represents a z/VM system with multiple guests.

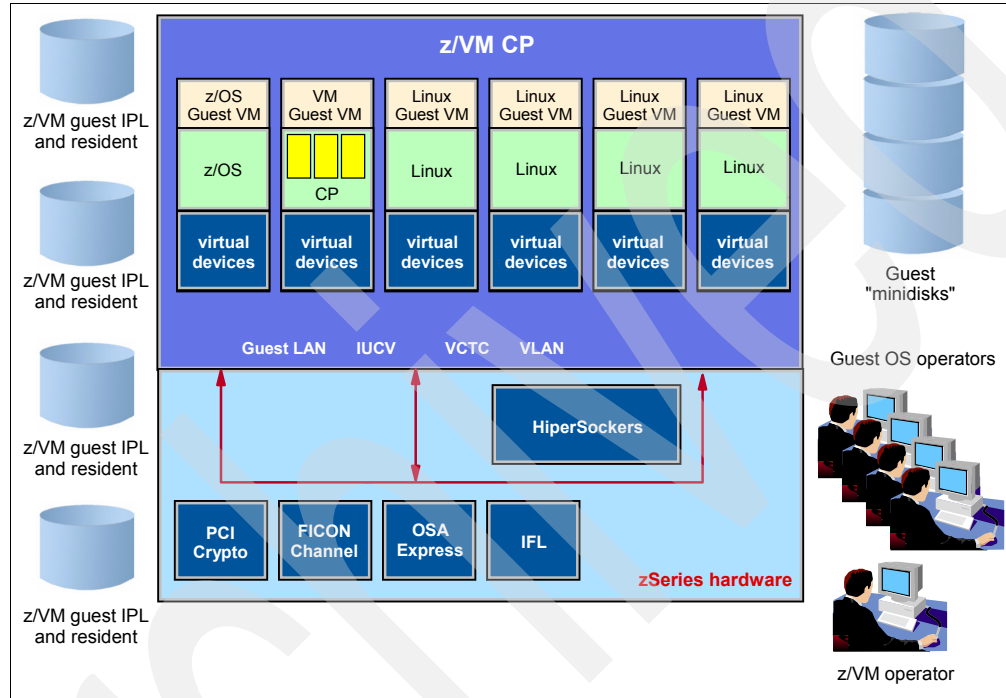


Figure 1-1 z/VM implementation with multiple guests

Note the following points:

- ▶ Only the Control Program is IPLed using the actual hardware IPL sequence. The guest IPL sequence is simulated by the Control Program.
- ▶ The Control Program operator console is actually providing Control Program emulated *hardware consoles* for the guest virtual environment. Control Program commands, on top of the guest IPL command, are providing, for example, the equivalent of a *System reset* or *Restart* hardware functions for the guest machines.
- ▶ The operating systems running in each guest have their usual operator console, which are physically connected through I/O channels to their respective operating systems.
- ▶ z/VM has its own scheme of disk storage partitioning, known as *minidisks*, that the Control Program uses to share one physical disk among several guests.
- ▶ z/VM can itself be running in a guest virtual machine, thus creating *second level* guest z/VMs.
- ▶ The security requirements, as seen from the z/VM software perspective, ensure that the guest z/VMs have access only to the physical resources to which they are entitled and to guarantee inter-guest isolation. Each guest operating system is then responsible for its security environment as seen by its own users.

Note: These Control Program and guest software layers created in the z/VM exploit the z/Architecture® and use the physical CPUs by switching instruction flows and address spaces between Control Program, guests operating systems, and user programs. They all exploit the same System z instruction set, with the exception of Control Program. Control Program uses a very specific instruction that is designed for virtual machine use called the *Start Interpretive Execution* (SIE).

1.2.1 The SIE instruction

By default, z/VM does not allow guest operating systems to be aware of each other. It achieves this by using the *Interpretive Execution Facility* (IEF) of the System z that is built in specifically for this purpose.

IEF executes an entire virtual machine instruction stream as a single instruction called SIE. The virtual machine is dispatched to run by the Control Program in a way that makes the System z firmware aware of the virtual machine details. The guest runs on the hardware until its time slice expires or until an instruction that cannot be virtualized is attempted, such as reference to a real address. At that point, the Control Program regains control to simulate the operation. Should a guest operating system fail, control is always returned to the Control Program for error recovery. In this way, guests are isolated and protected from others. This results in very low overhead for a virtualized environment and delivers confidentiality and integrity among virtual servers.

This feature is described in greater detail in *z/VM Security and Integrity*, GM13-0145.

The operating system running as a guest can perform its own address translation and maintain its own tables of *real* and *virtual* memory without awareness that its entire address space is virtual. What appears to be page or swap space to an operating system can be, in reality, real memory. Performance improvements can be realized in this way with certain operating systems. Conversely, a poorly behaving application will be paged out by z/VM and will not drag down the entire system. In such a protected environment, a guest operating system that is running on z/VM can communicate over a virtual network with another guest of the same mainframe, or with another server externally, and not notice the difference.

Clearly, not all devices can be shared simultaneously as can real memory, network interfaces, and direct access disks. Tape drives, for example do not lend themselves to being used by multiple programs concurrently. Thus, serial devices are attached to a guest for the duration of use, while random access devices are controlled by the Control Program and instructions for each guest interleaved in an efficient manner. Printers and other unit record devices are, in general, owned by the Control Program and the spooling subsystem controls which output set is printing at any given time. In certain circumstances, a serial device is dedicated to a guest with the **attach** command and is unavailable to other guests during that session.

1.2.2 System z cryptographic solution

System z is well equipped to address modern cryptographic security needs. Since early in the 1990s, the hardware has shipped with one form or another of cryptographic processor included, and upgrades were available shortly after to allow you to customize and expand that solution. All the System z operating systems (z/OS, Linux on System z, z/VM, and z/VSE™) provide the software to implement the necessary solutions. Today, the mainframe offers everything that you need for an effective cryptographic solution in your environment.

Cryptographic software support: z/VM

Virtual machines enable the sharing of System z hardware among many operating systems. As a virtualization solution, the z/VM operating system does not provide a direct interface into any of the cryptographic hardware, nor does it require cryptographic services itself. z/VM provides a means of sharing the hardware cryptographic resources among the operating systems that are hosted (known as *guests*). Cryptographic resources can be shared through z/VM as follows:

- ▶ For all available cryptographic accelerators, z/VM provides unlimited access to all guests. This includes access to the CP Assist Cryptographic Function (CPACF), the PCI Cryptographic Accelerator (PCICA), and the CEX2C when configured as a pure accelerator (CEX2A).
- ▶ The secure key *Hardware Security Module* (HSM), also sometimes referred to as a *Tamper Resistant Security Module* (TSRM), is comprised of a highly specialized piece of equipment that is designed to be the basis of your cryptographic security solution. These devices are the *strongboxes* that protect your symmetric keys and our asymmetric private keys. For HSM processors (CEX2C), z/VM can assign them to guests as well, but like logical partitioning, z/VM must assign the domains to each guest. A guest can have more than one domain. Also, multiple guests can be assigned the same domain, but only one guest can be active in a domain at any time.

For z/VM TCP/IP prior to z/VM Version 5.3, z/VM TCP/IP provides Secure Sockets Layer (SSL) support through a program interface called *VMSOCK*. Calls are redirected to a Linux on System z guest under z/VM, which contains special code provided by z/VM. This implementation was available for TN3270 or programs that were written to take advantage of the interface.

With z/VM Version 5.3, the usage of the SSL was extended to Transport Layer Security (TLS) and now provides full support for TN3270, SMTP and FTP.

For addition information, see Chapter 6, “Cryptography on z/VM” on page 201, and consult the z/VM Web page at:

<http://www.vm.ibm.com/related/tcpip/>

Cryptographic software support: z/OS

z/OS has provided cryptographic solutions on System z longer than any other operating system. The Integrated Cryptographic Services Facility (ICSF) solution is available.

In the z/OS world, the Integrated Cryptographic Services Facility (ICSF), an unpriced optional feature of the Cryptographic Services (a base element of z/OS), is considered to be synonymous with the cryptographic solution. In addition to providing the Common Cryptographic Architecture (CCA) APIs and interfacing with the hardware, ICSF interfaces with the External Security Manager to ensure that requesters are authorized to access the cryptographic services and resources that they are requesting.

Cryptographic software support: Linux

The System z platform with its high availability, connectivity, scalability, and virtualization, provides an ideal platform on which to host Linux servers. Linux implementations have access to all of the cryptographic packages that any other Linux deployment might have, which would typically include many software cryptographic toolkits and products. What makes Linux distinct in this area is that it also has access to the System z hardware cryptography environment.

Cryptographic software support: z/VSE

The z/VSE operating system also supports cryptographic processing to support SSL for its TCP/IP-based processes. If cryptographic accelerator hardware is available, z/VSE uses it automatically. This includes CPACF, PCICA, and CEX2C/CEX2A hardware. If there is no cryptographic hardware available, z/VSE performs the necessary cryptography in software.

1.2.3 Intrusion detection

One element of z/VM intrusion detection capabilities is that if a login is denied, the event is tracked and a security journal entry made when the number of denials exceeds an installation-defined maximum. When a second maximum is reached, log on to the user ID is disabled, an operator message is issued, and the terminal session is terminated.

Journaling is supported on z/VM. Virtual machine logon attempts and linking to other virtual machine's minidisks are detected and recorded. Using the recorded information, you can identify attempts to log on to a virtual machine or to link to minidisks using invalid passwords.

1.2.4 Accountability

A special capability available with z/VM is *Logon By*. When users log on to the shared user ID using this option, they provide their own user ID and password. An audit trail is maintained of who is actually logged into a shared user ID, so the problems inherent in sharing passwords are avoided. This audit trail tracks the identity of the user of a shared user ID, ensures user authority is validated, and provides *accountability*.

1.2.5 Certification

The *Common Criteria* is an internationally recognized International Standards Organization (ISO) standard that is used by governments and other organizations to assess the security and assurance of technology products. Under the Common Criteria, products are evaluated according to strict standards for various features, such as security functionality and the handling of security vulnerabilities.

Common Criteria ensures:

- ▶ A set of meaningful security functions
- ▶ Access control
- ▶ Audit
- ▶ Extensive testing of those functions
- ▶ Effective processes
- ▶ Good documentation
- ▶ Assurance levels 1 through 7
- ▶ Evaluation by accredited firms
- ▶ Certification by government agencies

For more information about Common Criteria, refer to:

<http://www.CommonCriteriaPortal.org>

The LSPP security labeling system

Central to LSPP security is its system of security labeling. Each object in a z/VM LSPP-compliant system has a *security label*, or *SECLABEL*, that designates its relative confidentiality and its membership in a security category. An object's security label defines what sort of data it can contain and, by implication, what sort of data it cannot contain.

Note: An object can have one and only one security label at any given time. Similarly, each user (subject) in the system has at least one security label that designates relative power and privilege over objects. That is, a subject's security label specifies whether it can access a given object and, if so, what actions, if any, it can perform upon that object.

Some subjects perform a wide variety of tasks and fulfill many different roles, each with its own security implications. So, it is only natural that some subjects are able to perform work under more than one security label each. However, only one security label can be in effect at any given time, and that security label governs all of the subject's activity until it is changed.

A security relationship, then, always exists between the SECLABEL of any given subject and the SECLABEL of any given object. Within this relationship, the security relationship between subject and object itself is implied. A question to consider is, "Is this relationship conducive to the security of the organization?" The answer to that question is provided by *mandatory access control*.

Mandatory access control (MAC) is a security policy that governs which subjects can access which objects, and in what way, based upon certain rules. These rules are the **-property* and the *simple security property*. RACF commands are used to manage MAC for Control Program commands, DIAGNOSE codes, and system functions. MAC restricts a subject's access to an object.

CAPP

The Controlled Access Protection Profile (CAPP) specifies a set of security functional and assurance requirements, including access controls that are capable of enforcing access limitations on individual users and data objects. CAPP-conforming products also provide an audit capability which records the security-relevant events which occur within the system.

CAPP was derived from the requirements of the C2 class of the U. S. Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), dated December 1985. This protection profile provides security functions and assurances which are equivalent to those provided by the TCSEC and replaces the requirements used for C2 trusted product evaluations.

In z/VM, the CAPP requirements are met through the following specific mechanisms:

- ▶ Discretionary Access Control (DAC)

A method of restricting access to data objects based upon the identity of users or groups to which the users belong. DAC protects system objects from unauthorized access by any user. Normally, permission to access an object is granted by the owner of the object. Occasionally, it can be granted by someone else, such as a privileged administrator.

- ▶ Auditability of Security-Relevant Events

The recording of facts that describe a security-relevant event taking place in a computing system. In general, a security-relevant event is one that occurs in a computing system that, for better or for worse, affects the safety and integrity of the system's processes and data. The facts recorded that describe such an event include the time and date of the event, the name of the event, the name of the system objects affected by the event, the name of the user who caused the event to occur, and additional information about the event.

In general, the security-relevant events in z/VM are:

- CP commands
- DIAGNOSE functions
- Communication among virtual machines

- **Object Reuse**

A practice that prevents any newly-assigned storage object from making available to its new owner any data that belonged to its former owner. This includes any encrypted data. Object reuse also requires the elimination of any residual user authorization access to a previously existing object. This ensures that if another, new object occurs in the system later under the same name, the subjects having access to the old object will not have access to the new one.

- **Identification and Authentication**

A method of enforcing individual accountability by providing a way to authenticate a user's identity uniquely and unambiguously. Thus, any security-relevant action users might take can be attributed to them.

z/VM V5.1 was evaluated for conformance to the Controlled Access Protection Profile (CAPP) and the Labeled Security Protection Profile (LSPP) of the Common Criteria, both at Evaluation Assurance Level (EAL) 3+.

As a Statement of Direction, IBM intends to evaluate z/VM V5.3 with the RACF Security Server optional feature for conformance to the Controlled Access Protection Profile (CAPP) and Labeled Security Protection Profile (LSPP) of the Common Criteria standard for IT security, ISO/IEC 15408, at Evaluation Assurance Level 4 (EAL4). Refer to z/VM General Information Version 5 Release 3 GC24-6095-06 for further details.

For a complete discussion for a Secure Implementation see *z/VM Secure Configuration Guide*, SC24-6138.

1.2.6 Debugging in a virtual environment

Using a virtual environment to debug operating systems and applications gives you the distinct advantage of using Control Program commands to query memory locations while the virtual machine is running, or in a stopped state at some chosen point of interruption. IBM debugs new versions of z/VM in various virtual machine types created by a previous version of z/VM.

Another important advantage is that, because the entire environment is virtual, little harm can come to the system by incorrect device calls and operations. An instruction that could potentially freeze up a device when running on real hardware and require operator invention is trapped by Control Program, recognized as illegal, and an appropriate response code returned to the calling application without affecting hardware availability to other guests of the mainframe.

1.2.7 Virtual networking

Communication between virtual machines is provided by various simulated devices or by facilities that are unique to the z/VM operating environment. Available communications paths include z/VM Guest LANs, Inter-User Communication Vehicle (IUCV), and Virtual Channel-to-Channel Adapter (VCTCA).

Each of these options provides a highly secure communication path that is not detectable or in any way “sniffable” by other virtual machine (that is, no other virtual machine can eavesdrop on the data moving between virtual machines). Of course, these virtual network connections are only as secure as the guests connected to them.

A *virtual network* is connected typically to the outside world using virtual firewalls or routers. These are virtual machines that have both virtual network connections and real network connections, routing traffic as needed between the two.

Virtual networks need to be planned with the same care and attention to security as real, physical networks. Networks, virtual or physical, must be designed and implemented so that unauthorized access to data or resources is not possible. For system administration tasks, a separate network with secure access is recommended. The ability to define multiple virtual routers allows you to completely isolate traffic moving in and out of the mainframe.

The best LAN is one without wires

The z/VM Guest LAN, introduced with z/VM Version 4 Release 2, provides multipoint, any-to-any virtual shared media connections between guests. A virtual machine accesses a Guest LAN using a virtual Network Interface Card (NIC), which emulates either a System z HiperSockets™ adapter or, in later z/VM releases, a QDIO Ethernet adapter. The most recent development is the Virtual Switch, which adds connection of a Virtual QDIO Ethernet NIC to a LAN Segment that is connected to a OSA card, eliminating the need for routing. IPV6 support was added to the HiperSockets and Ethernet virtual NICs.

1.2.8 Compliance to policy

A z/VM system is secured using security features of the System z hardware by maintaining compliance to security policy within operating practices, thus making use of the *user directory* that contains a list of users of the system. The main control point of z/VM security is the user directory file, called *USER DIRECT*. For that reason, it is also called the *Control Program directory*. This directory file is owned by the system administrator, and every user of the system is identified within this file, including the system administrator. Also, each user's resources are defined in the file. The directory and the ability to promote a directory into active state must be the most protected assets of a z/VM system. The system administrator must lead the way in following security standards and guidelines if the community is to be safe. When DirMaint is installed, it owns the directory, and controls each part of the directory that can be changed by a given user in the system.

When Resource Control Facility feature for z/VM (RACF) is installed, it can be configured to control functions normally being checked in the directory for authorization. At a minimum RACF controls the password field but can be used to control minidisk access, spool files, and commands privileges.

In z/VM, a subject is a virtual machine, which is one of four types:

- ▶ General user
- ▶ Privileged user
- ▶ Trusted server
- ▶ System operator

For further information, refer to *IBM System z9 Enterprise Class Technical Guide*, SG24-7124.

Each role has approximately the same logical structure. A general user is defined as a virtual machine that has, at the most, the set of the Control Program commands available in the IBM-defined privilege class G.

In addition, the general user does not have the following characteristics:

- ▶ SPECIAL
- ▶ Group-SPECIAL
- ▶ CLAUTH
- ▶ Group-CLAUTH
- ▶ OPERATIONS authority to RACF
- ▶ No OBEY authority for VM TCP/IP
- ▶ No access to the z/VM directory (source or object forms)
- ▶ No read-write access to the PARM disk(s), or other system areas of CP-owned volumes
- ▶ No read-write access to the source or object code of CP, CMS, RACF, or z/VM TCP/IP
- ▶ No read-write access to the RACF database
- ▶ No read-write access to the RACF audit trail

The general user does not have the following options in the Control Program directory entry:

- ▶ COMSRV
- ▶ DIAG88
- ▶ DIAG98
- ▶ DEVMAINT
- ▶ MAINTCCW
- ▶ SETORIG

z/VM privileges

In the z/VM system of privilege, a user either can have no privileges or can be assigned to one or more privilege classes. Each privilege class represents a subset of Control Program commands that the system permits the user to enter. Each privilege class, sometimes called *CP privilege class*, is defined around a particular job or set of tasks, thereby creating an area outside of which the user cannot go. Of course, it is commonplace for a user to be assigned to more than one CP privilege class. Users are unable to enter commands in privilege classes to which they are not assigned.

Note: Any user, except those with either NO PRIVILEGE or CP privilege class G, is considered part of the configuration but is not necessarily considered trusted.

Here is a summary of CP privilege classes, their associated users, tasks, and security implications:

▶ **Privilege class A**

The primary system operator. The system operator is among the most powerful and privileged of all z/VM users. The system operator is responsible for the system's availability and its resources. The system operator also controls accounting, broadcasts messages, and sets performance parameters.

▶ **Privilege class B**

The system resource operator. The system resource operator controls the allocation and de-allocation of real resources, such as memory, printers, and DASD. Note that the system resource operator does not control any resource already controlled by the system operator or the spooling operator.

▶ **Privilege class C**

System programmer. The system programmer updates the functions of the z/VM system and can change real storage in the real machine.

► **Privilege class D**

Spooling operator. The spooling operator controls spool files and real unit record devices, such as punches, readers, and printers.

► **Privilege class E**

- System analyst. The system analyst has access to real storage and examines dumps to make sure that the system is performing as efficiently and correctly as possible.

► **Privilege class F**

- IBM service representative. The representative of IBM who diagnoses and solves problems by examining and accessing real input and output devices and the data they handle.

► **Privilege class G**

General user. This is the most prevalent and innocuous of the CP privilege classes. The commands that privilege class G users can enter effect only their own virtual machines.

► **Privilege class ANY**

The commands in this privilege class are available to any user.

Privilege classes A, B, C, D, E, and F require individuals worthy of significant trust and whose activities require careful auditing. For example, users with privilege class B or C can modify an installation's system of CP privilege. Because this modification violates the CAPP security policy, system programmers and similarly privileged users must be trusted to not tamper with the system of CP privilege (and auditing must confirm this trust).

As another example, privilege class C users can enter the **cp store host** command that allows them to alter real storage. This command makes it possible for users to negate the CAPP classification.

Privilege class G users have no influence outside their own virtual machines. So, with the exception of access to storage objects, they have very little security relevance.

The ANY privilege class commands cannot violate the security policies of the system. This is because all commands in the ANY privilege class are auditable and subject to either Discretionary Access Control (DAC) or Mandatory Access Control (MAC). Therefore, class ANY users, together with class G users, cannot violate the security policy. In the Control Program, each level of privilege is discrete and not predicated on others. Furthermore, each privilege class (a subset of commands) is related to one or more function types (subsets of users).

The Control Program directory

The Control Program directory is the reference repository which z/VM uses to perform its access control. By default, each z/VM user's address space, DASD, vswitch, and all files are private to the user or virtual server.

Special action is required to expose data to another user, although, as with all platforms, the OPERATOR or superuser is able to gain all access rights to them if they have a need. In this directory, each guest's *privilege class* is assigned.

The privilege class determines their rights to issue certain CP commands and program instructions (diagnose codes) that reference the world outside their own virtual machine. Although a default set of classes is defined when you initially install z/VM, you can also create your own custom classes. You can define up to 32 classes. The standard class for general users is class G. Class G users cannot affect other users or CP operation, although by using certain query commands, they can become aware of other guests.

It is possible, even desirable, to create classes with less than general user privilege which allow certain virtual machines only the minimum functionality required to perform their assigned duty.

The CP directory has basically two forms:

- ▶ Human readable file called USER DIRECT.
- ▶ Machine readable in object form, placed on a reserved area of disk by the privileged CP **directxa** utility command.

In z/VM, access rights can be granted in two ways:

- ▶ Using the privilege class.
- ▶ Granting the access to resources by the owner or by another authorized user. For example, you can grant a user read or write access to virtual disks that are owned by a different user. The resource's owner or a system administrator can grant the permission.

The format of the CP directory

The CP directory file contains groups of statements. Each group is associated with a single and unique user. The user is identified and distinguished from each other by the USER statement; all the records following that statement apply to that particular user. Figure 1-2 shows an example of directory entries.

```
USER LINUX1 xxxxxxxx 256M 1G G
INCLUDE COMMON
ACCOUNT LNX1
IPL 301
MACHINE ESA 4
OPTION ACCT
CONSOLE 0009 3215 T
NICDEF BC0 TYPE QDIO LAN SYSTEM VSWITCH1
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
MDISK 0101 FB-512 V-DISK 131072 M
MDISK 0102 FB-512 V-DISK 262144 M
MDISK 0103 FB-512 V-DISK 131072 M
MDISK 0301 3390 1476 1000 USXF36 MR
MDISK 0302 3390 2201 850 USXF27 M
MDISK 0303 3390 2476 850 USXF36 M
MDISK 0401 3390 601 200 USXF27 MR
MDISK 0407 3390 3031 150 USXF29 M
MDISK 0408 3390 3191 145 USXF35 M
```

Figure 1-2 A user directory entry

Here is an explanation for the most common directory entries within the USER DIRECT file:

- ▶ **USER:** Defines the user ID, password (if not running an external security manager), virtual memory, and privilege class (for example: the USER statement begins a directory entry).

In Figure 1-2, the user ID is LINUX1 with xxxxxxxx as the logon password. The virtual machine has a default storage of 256 megabytes (256M) when initially loaded, but the owner can redefine storage up to a maximum of 1 gigabyte (1G). The *G* means the virtual machine user belong to class G and can execute commands designated only for users within that class.

Refer to *Running Linux Guest in less than CP Privilege Class G*, REDP-3870 Class G for a description on how to define z/VM systems to run Linux virtual machines.

- ▶ **INCLUDE:** The INCLUDE statement specifies the name of a profile entry to be invoked as part of the USER statement. The include entry is specified in the directory file and it is a common statements used by multiple users. For example, usually, the read access to systems minidisks are located on an include entry.
- ▶ **ACCOUNT:** Defines accounting information for the user ID described in USER. This might be the department number of the Linux programmers.
- ▶ **IPL:** Defines the user's program or device to Initially Load (which is known as boot on other platforms).

In our example, the IPL statement indicates which operating system to load when you log on to the virtual machine and that minidisk 301 is the device from which to load, which is the minidisk that contains the Linux.

- ▶ **SPOOL:** Defines virtual I/O devices that are available to this user, with one statement per device. Usual devices and addresses describe a virtual input device, like a card reader, usually with a 00C address, an output device such as a card punch with a 00D address, and a virtual printer with a 00E address. The CP controls the real devices and provides *spool space* for the user's files while they are in transmission between users or waiting for real devices. This means that you can punch a file from your user ID to another user ID to read without a card deck actually being produced.
- ▶ **MDISK:** Defines areas of disk that are owned by this user. The term *minidisk identifies* an area of a real direct access storage device (DASD) that is partitioned and assigned to a specific address and owned by a specific user. From the perspective of that user, it is a volume of DASD. There is virtually no limit to the number of MDISK statements present for each user.
- ▶ **LINK:** Defines access to other user's MDISK statements. Minidisks can be shared in Read or Write mode, but it is the responsibility of the user to ensure proper RESERVE/RELEASE protocols are followed to maintain data integrity. There is one LINK statement per shared minidisk.
- ▶ **MACHINE:** Describes the processor architecture of the virtual machine and the maximum number of virtual CPUs that can be defined for this virtual machine. The default is one.
- ▶ **CONSOLE:** Defines the operating console (virtual console) for the virtual machine. If supported by the operating system, you can specify 3270.
- ▶ **NICDEF:** Defines this virtual machine's attachment to a z/VM virtual switch.

System user IDs involved in security

There are some users that are defined as part of the installation process. The standard user IDs for the default system installation are:

- ▶ **OPERATOR:** System operator and high privilege user ID. Equal to root on UNIX® systems.
- ▶ **MAINT:** Used by a person for system administration and maintenance. Similar to OPERATOR from an authorization point of view.
- ▶ **EREP:** EREP stands for *Environmental Record Editing and Printing Program*. Hardware anomaly detection and predictive failure system.
- ▶ **DISKACNT:** Records events such as logon and logoff.
- ▶ **OPERSYMP:** Used to retrieve symptom records. System dump analyzer and problem tracking system.

System configuration file statements involved in security

Note: For the complete description of syntax and usage for the system configuration file, refer to *z/VM CP Planning and Administration*, SC24-6083.

The system configuration file is located on a partition of a volume allocated as PARM. This minidisk is normally under user ID maint, and it is on minidisk address CF1. The file is called SYSTEM CONFIG by default, although your installation can change the name. The file is read at IPL time by the CP program that uses the statements contained in the file to configure the system.

The following list summarizes the statements that are contained in the configuration file that are relevant to security:

- ▶ **DEFine COMmand:** Use the DEFINE COMMAND or CMD statement to define a new CP command or a new version (by IBM class) of an existing CP command on the system during initialization.
- ▶ **DEFine LAN:** Use the DEFINE LAN statement to create a guest LAN which can be shared among virtual machines on the same VM system. Each guest LAN segment is identified by a unique combination of ownerid and lanname. A VM user can create a simulated network interface card (NIC) and connect it to this LAN segment.
- ▶ **DEFine VSWITCH:** Use the DEFINE VSWITCH statement to create a CP system-owned switch (a virtual switch) to which virtual machines can connect. Each switch is identified by a *switchname*. A z/VM user can create a simulated QDIO network interface card (NIC) and connect it to this switch with the NICDEF directory statement. Under the DEFINE VSWITCH statement, the VLAN parameter is important if you want to isolate guests subnets based on VLAN IDs.
- ▶ **DISable COMmand:** Use the DISABLE COMMAND or CMD statement to prevent CP from processing requests for the specified CP command during and after initialization.
- ▶ **DISable DIAGnose:** Use the Disable Diagnostic statement to prevent CP whether from processing requests for one or more locally-developed DIAGNOSE codes during and after initialization.
- ▶ **ENable DIAGnose:** Use the ENABLE COMMAND or CMD statement to permit CP to process requests for the specified CP command during and after initialization.
- ▶ **ENFORCE_BY_VOLid:** Use the ENFORCE_BY_VOLid configuration statement to enforce attachment of DASD devices by their VOLIDs on the ATTACH command.
- ▶ **FEATURE:** Use the FEATURES statement to set certain attributes of the system at system initialization.
- ▶ **JOURNALing:** Use the JOURNALING statement to tell CP whether to include the journaling facility, whether to enable the system being initialized to set and query the journaling facility, and what to do if someone tries to log on to the system or link to a disk without a valid password.
- ▶ **MODify COMmand:** Use the MODIFY COMMAND or CMD statement to redefine an existing CP command on the system during initialization.
- ▶ **MODify LAN:** Use the MODIFY LAN statement to modify the attributes of an existing guest LAN during initialization.
- ▶ **MODify PRIV_CLASSES:** Use MODIFY PRIV_CLASSES to change the privilege classes authorizing the following CP functions.
- ▶ **MODIFY VSWITCH:** Use the MODIFY VSWITCH statement to modify the attributes of an existing virtual switch.

- ▶ **PRIV_CLASSES:** Use the `PRIV_CLASSES` statement to change the privilege classes authorizing the following CP functions.
- ▶ **SYSTEM_USERids:** Use the `SYSTEM_USERIDS` statement to specify user IDs that will perform special functions during and after IPL. These functions include accumulating accounting records, system dump files, EREP records, and symptom records, and specifying the primary system operator's user ID and disconnect status.
- ▶ **USER_DEFAULTS:** Use the `USER_DEFAULTS` statement to define default attributes and permissions for all users on the system.

1.3 Additional features

This section introduces briefly that additional feature and products that are available in the z/VM system and that are used to build the security solution.

1.3.1 The Resource Access Control Facility

The Resource Access Control Facility (RACF) licensed program can satisfy the preferences of the user without compromising any of the concerns raised by security personnel. The RACF approach to data security is to provide an access control mechanism that offers effective user verification, resource authorization, and logging capabilities. RACF supports the concept of *user accountability*. It is flexible, has little noticeable effect on the majority of end users, and little or no impact on an installation's current operation.

RACF controls access to and protects resources on both multiple virtual storage (z/OS) and virtual machine systems. For a software access control mechanism to work effectively, it must be able to first identify the person who is trying to gain access to the system, and then verify that the user is really that person.

With RACF, you are responsible for protecting the system resources, such as minidisks, terminals, and shared file system (SFS) files and directories, and for issuing the authorities by which those resources are made available to users. RACF records your assignments in *profiles* stored in the RACF database. RACF then refers to the information in the profiles to decide if a user should be permitted to access a system resource.

The ability to log information, such as attempted accesses to a resource, and to generate reports containing that information can prove useful to a resource owner, and is very important to a smoothly functioning security system. Because RACF can identify and verify a user's user ID and recognize which resources the user can access, RACF can record the events where user-resource interaction has been attempted. This function records actual access activities or variances from the expected use of the system.

RACF has a number of logging and reporting functions that allow a resource owner to identify users who attempt to access the resource. In addition, you or your auditor can use these functions to log all detected successful and unsuccessful attempts to access the RACF database and RACF-protected resources. Logging all access attempts allows you to detect possible security exposures or threats. The logging and reporting functions are:

- ▶ **Logging:** RACF writes audit records in a file for detected, unauthorized attempts to enter the system. Optionally, RACF can also writes records for authorized attempts or detected, unauthorized attempts to:
 - Access RACF-protected resources
 - Issue RACF commands
 - Modify profiles on the RACF database.

- ▶ **Sending Messages:** RACF sends messages to the security console for detected, unauthorized attempts to enter the system and for detected, unauthorized attempts to access RACF-protected resources or modify profiles on the RACF database.
- ▶ **Keeping Statistical Information:** Optionally, RACF can keep selected statistical information, such as the date, time, and number of times that a user enters the system and the number of times a single user accesses a specific resource. This information can help the installation analyze and control its computer operations more effectively. In addition, to allow the installation to track and maintain control over its users and resources, RACF provides commands that enable the installation to list the contents of the profiles in the RACF database.

Some features introduced with z/VM Version 5.3 and RACF Feature Level 5.3 include:

- ▶ Mixed-case 8-character passwords
- ▶ Mixed-case password phrases up to 100 characters, including blanks
- ▶ No longer possible to reset password to default group name
- ▶ Audit trail can be unloaded in XML format
- ▶ Remote authorization and audit through z/VM new LDAP server and utilities

For information about how we configured the RACF environment on our ITSO system and for configurations example, refer to Chapter 2, “RACF feature of z/VM” on page 21.

Note: For references on security implementation on z/VM, including RACF, refer to *z/VM Secure Configuration Guide*, SC24-6138.

1.3.2 LDAP

Today, people and businesses rely on networked computer systems to support distributed applications. These distributed applications might interact with computers on the same local area network, within a corporate intranet, within extraneous linking up partners and suppliers, or anywhere on the worldwide Internet. To improve functionality and ease-of-use and to enable cost-effective administration of distributed applications, information about the services, resources, users, and other objects accessible from the applications needs to be organized in a clear and consistent manner. Much of this information can be shared among many applications, but it must also be protected in order to prevent unauthorized modification or the disclosure of private information.

Information describing the various users, applications, files, printers, and other resources accessible from a network is often collected into a special database that is sometimes called a directory. As the number of different networks and applications has grown, the number of specialized directories of information has also grown, resulting in islands of information that are difficult to share and manage. If all of this information could be maintained and accessed in a consistent and controlled manner, it would provide a focal point for integrating a distributed environment into a consistent and seamless system.

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that has evolved to meet these needs. LDAP defines a standard method for accessing and updating information in a directory. LDAP has gained wide acceptance as the directory access method of the Internet and is therefore also becoming strategic within corporate intranets. It is being supported by a growing number of software vendors and is being incorporated into a growing number of applications. For example, the two most popular Web browsers, Netscape Navigator/Communicator and Microsoft® Internet Explorer®, as well as application middleware, such as the IBM WebSphere Application Server or the IBM HTTP server, Linux APACHE Webserver, support LDAP functionality as a base feature.

As part of our environment, we implemented LDAP under z/VM and z/OS managing Linux for System z guests under z/VM. For information about our configuration, refer to Chapter 3, “z/VM LDAP server” on page 111.

1.3.3 z/VM VSWITCH networking

The z/VM Virtual Switch (VSWITCH) introduced with z/VM V4.4 builds on the Guest LAN technology that was delivered in earlier z/VM releases. VSWITCH connects a Guest LAN to an external network using an OSA-Express interface. Two additional OSA-Express devices can be specified as backups in the VSWITCH definition. The Linux guests connected to the VSWITCH are on the same subnet as the OSA-Express interface or interfaces and other machines connected to that physical LAN segment. When VLANs are implemented multiple subnets flows on the same LAN segments.

The z/VM V4.4 implementation of VSWITCH operates at Layer 3 (network layer) of the OSI model. It only supports the transport of IP packets. In other words, it only can be used for TCP/IP applications. All destinations are identified as IP addresses, thus no MAC addresses are used (link layer independent), and ARP processing is off-loaded to the OSA-Express adapter. In this environment, all hosts share the same OSA-Express MAC address. Traffic destined for the physical portion of the LAN segment is encapsulated into an Ethernet frame with the OSA-Express's MAC as the source MAC address. For inbound packets, the OSA-Express strips the Ethernet frame and forwards the IP packet to the Virtual Switch for delivery to the guest by the destination IP address within the IP packet.

In z/VM V5.1, the VSWITCH can operate at Layer 2 (data link layer) of the OSI model.

Because the VSWITCH is essentially connected to the physical LAN, the requirement for an intermediate router between the physical and (internal) Guest LAN segments is removed.

This reduces network latency. It also reduces overall CPU consumption, in some test cases by as much 30%. Removing the router also means that you no longer need specialized skills to configure and administer a z/VM-based or Linux-based router.

Virtual LANs (VLANs) facilitate easy administration of logical groups of machines that can communicate as though they were on the same local area network (LAN). VSWITCH provides support for IEEE 802.1Q VLANs. This means that Linux guests connected to a VSWITCH can participate in a VLAN environment.

z/VM V5.2 introduced support for VLANs with support for Generic Attribute Registration Protocol (GVRP) which can reduce load on the system by filtering only registered VLANs.

z/VM V5.3 introduced support for link aggregation for OSA-Express2, enhanced ease-of-use VLAN and promiscuous mode configuration changes. Link aggregation allows up to eight OSA-Express2 to work as one link, increasing throughput and error recover ability.

The security advantage of VSWITCH is when VLAN are used to isolate guests machine on the same network (Same LAN segment). Virtual LAN segments are controlled by the virtual switch, not the guests, an external security manager (ESM), such as RACF can control access to the virtual switch and VLANs. VLAN in conjunction with the port type designated for a guest, defines the ingress and egress rules that are applied to this connection. The guest VLAN IDs are specified through the grant option of the SET VSWITCH command or system configuration file.

For maximum isolation between guests you should configure your VSWITCH as VLAN AWARE, VLAN ID as the assigned subnet on the LAN segment where the guest will operate and PORTTYPE ACCESS on the Define VSWITCH statement of the System Configuration

File. Use the System Configuration File Statement Modify VSWITCH or SET VSWITCH Command, to GRANT a guest user ID coupling to the VSWITCH. CP will remove and add VLAN tags on ingress and egress from the guest port, guest is configured and will operate with the network adapter as though belonging to a segment not using VLAN IDs.

When using RACF, VMLAN class is used. From an access control perspective, guest LANs and virtual switches are treated the same way. The VMLAN class contains two sets of profiles to protect LANs: base profiles that control the ability of a z/VM user to use a LAN, and for IEEE VLAN-aware virtual switches, VLAN ID-qualified profiles that are used to assign a user to one or more IEEE VLANs. For more information, refer to Chapter 2, “RACF feature of z/VM” on page 21.

For further information about z/VM VSWITCH operation, consult *z/VM Connectivity*, SC24-6080.

For further information about system configuration file statements, consult *z/VM CP Planning and Administration*, SC24-6083.

For further information about VSWITCH commands, refer to *z/VM CP Commands and Utilities*, SC24-6081.

For RACF authorization mechanism for VSWITCH, refer to *z/VM RACF Security Server System Administrator's Guide*, SC24-6142.

Example 1-1 shows an example of creating a VLAN AWARE VSWITCH and granting access to a user through the SET VSWITCH command.

Example 1-1 Creating a VLAN AWARE with tag VLAN 2 controlled by CP

```
Ready; T=0.01/0.01 12:55:28
define vswitch sw00001 rdev 2e28 connect vlan 2 porttype access nat 1
VSWITCH SYSTEM SW00001 is created
Ready; T=0.01/0.01 12:56:00
HCPSWU2859I Device 2E28 for VSWITCH SYSTEM SW00001 does not support GVRP.
HCPSWU2830I VSWITCH SYSTEM SW00001 status is ready.
HCPSWU2830I DTCVSW2 is VSWITCH controller for device 2E28.
set vswitch sw00001 grant lnxsu1
Command complete

Ready; T=0.03/0.03 12:57:53
q vswitch sw00001
VSWITCH SYSTEM SW00001 Type: VSWITCH Connected: 1 Maxconn: INFINITE
PERSISTENT RESTRICTED NONROUTER Accounting: OFF
VLAN Aware Default VLAN: 0002 Default Porttype: Access GVRP: Disabled
Native VLAN: 0001
MAC address: 02-00-00-00-00-07
State: Ready
IPTimeout: 5 QueueStorage: 8
RDEV: 2E28 VDEV: 2E28 Controller: DTCVSW2
Ready; T=0.01/0.01 12:59:01
q vswitch sw00001 details
VSWITCH SYSTEM SW00001 Type: VSWITCH Connected: 1 Maxconn: INFINITE
PERSISTENT RESTRICTED NONROUTER Accounting: OFF
VLAN Aware Default VLAN: 0002 Default Porttype: Access GVRP: Disabled
Native VLAN: 0001
MAC address: 02-00-00-00-00-07
State: Ready
```

```
IPTimeout: 5      QueueStorage: 8
RDEV: 2E28 VDEV: 2E28 Controller: DTCVSW2
VSWITCH Connection:
  MAC address: 00-09-6B-1A-1F-38
  RX Packets: 0      Discarded: 121      Errors: 0
  TX Packets: 0      Discarded: 0        Errors: 0
  RX Bytes: 0        TX Bytes: 0
  Device: 2E2A Unit: 002 Role: DATA      Port: 0001 Index: 0001
  Options: VLAN_ARP
Adapter Connections:
  Adapter Owner: LNXSU1 NIC: C300 Name: UNASSIGNED
  Porttype: Access
Ready; T=0.01/0.01 13:01:10
```

Example 1-2 shows a DEFINE NIC command creating a Network Adapter and a COUPLE command connecting the NIC to the VSWITCH created in Example 1-1.

Example 1-2 Coupling NIC to VSWITCH

```
CP DEFINE NIC C300 TYPE QDIO
NIC C300 is created; devices C300-C302 defined

Ready; T=0.01/0.01 08:57:15
CP COUPLE C300 SYSTEM SW00001
```

To avoid allowing users to define and couple, you can use the NICDEF directory statement and restrict usage of the Define and Couple commands, as shown in Example 1-3.

Example 1-3 NICDEF statement

```
NICDEF C300 TYPE QDIO LAN SYSTEM SW00001
```

Note: You can find additional information in *Networking Overview for Linux on zSeries*, REDP-3901.

RACF feature of z/VM

IBM Resource Access Control Facility (RACF) Security Server feature function level 530 for z/VM 5.3.0 has all the function of the previous release (RACF 1.10) with many new enhancements. New enhancements include:

- ▶ Sniffer support (APAR from 1.10)
- ▶ Mixed case password support
- ▶ Password phrase support (maximum 100 characters)
- ▶ Support for the new z/VM LDAP server
- ▶ Enhancement to user related processing
- ▶ SMF data unload as XML output
- ▶ Protection for the CP FOR command and Diag"x88"

This chapter describes the processes of installing, configuring, managing, monitoring, auditing, and controlling of RACF resources. We chose to use the DirMaint product for managing our system in regards to adding virtual machines to the system directory. Unlike other operating systems, z/VM has used the two phase approach when defining virtual machines. You must first define the virtual machine to the system CP, and then if you are using an external security manager (ESM), you must also add users and define users resources to the ESM database. You can find information about DirMaint installation and use in Appendix A, "DirMaint implementation" on page 301.

This chapter follows the concepts outlined in the chapter on requirements and installation in *Secure Configuration Guide for z/VM*, SC24-6139. This chapter describes the security requirements to keep in mind while installing and customizing z/VM and RACF for a secure environment.

Note: To make local modifications to RACF, you *must* have the high level assembler product installed on your system.

2.1 RACF z/VM concepts

An External Security Manager (ESM) for the z/VM system is software that provides additional functions to the z/VM system for controlling access to resources. Resource Access Control Facility (RACF) for z/VM is an ESM that works with the CP component of z/VM to manage system integrity.

RACF receives request from resource managers on the system (CP, Shared File System, TCP/IP, FTP, and so forth) on behalf of virtual machines to access a resource. In z/VM each of the resource managers will likely have a different virtual machine and person who is responsible for this support, rather than having a single *superuser* that is responsible for these processes. The resource manager checks to see if that resource is being managed by RACF or not. If the resource is managed by RACF, it will then check to see if the virtual machine has the proper authorization to access the resource or if the virtual machine is a member of a group that has access to the resource. If the virtual machine or a group of which the virtual machine is a member has the authority then access is granted to the resource. If they are not, then access will be denied.

The z/VM resource managers interface with the ESM using the RPI interface.

2.2 Installing and configuring RACF

RACF for z/VM is a priced product shipped with the z/VM 5.3.0 system deliverable using Virtual Machine Serviceability Enhancement Staged / Extended (VMSES/E), in a disabled state. It must be enabled and configured by the system programmer prior to using.

The Program Directory for the product describes the installation process and can be downloaded from the following address:

<http://www.ibm.com/servers/eserver/zseries/zos/racf/vm.html>

RACF Security Server for z/VM Program Directory, GI10-0788 describes the step-by-step process to enable and configure the product.

The main steps in the process are as follows:

1. Set RACF to the ENABLED state using the VMSES/E tools.
2. Perform post installation steps - see 2.2.1, "Post-installation tasks" on page 23.
3. Build the RACF enabled CPLOAD MODULE - see 2.2.2, "Build the RACF enabled CPLOAD MODULE" on page 33.
4. Update RACF database and options - see 2.2.3, "Update the RACF database and options" on page 36.
5. Place RACF into production - see 2.2.4, "Place RACF into production" on page 40.
6. Updating the authorization process - see 2.2.5, "Using HCPRWAC" on page 41.

You should print the procedural checklist from the RACF Security Server for z/VM Program Directory so that you do not miss any important steps in the process.

In this chapter we assume the reader has basic z/VM system programming knowledge and experience with VMSES/E product and its processes.

2.2.1 Post-installation tasks

This section describes the tasks that you need to perform after you have installed the RACF code. These tasks are the result of our best practices in our ITSO test environment.

- ▶ Allocate DASD
- ▶ Define RACF user IDs
- ▶ Evaluate minidisk access
- ▶ Update RACF user ID directory entry
- ▶ Execute RPIDIRECT
- ▶ Customize the processing of SMF records (optional)
- ▶ Remove ICHDEX01 and ICHRCX02 (optional)

Allocate RACF DASD

We recommend that you move RACFVM's 300 disk to a different volume on your system. The system is shipped with the 200 and 300 minidisk on the 530W02 volume on a 3390-3 install and both are located on the 530RES volume for a 3390-09 install. It is better to keep these minidisk on separate volumes, so that if volume is damaged, you do not lose all the RACF data.

Define RACF user IDs

The following virtual machines are defined in a default CP directory:

5VMRAC30	Product owning virtual machine
RACFVM	Production virtual machine
RACFSMF	SMF virtual machine
RACMAINT	Backup virtual machine
IBMUSER	Initial RACF administrator
AUTOLOG1	System startup machine
AUTOLOG2	System startup machine
SYSADMIN	Authorized RACF administrator

In our test environment, we noticed that by default, the user IDs are defined with *NOLOG* passwords. The Program Directory recommends that you change these entries to *UNLOG*, which is acceptable to RACF.

To perform the change, you need to determine what file will be used for the source directory while implementing RACF. If you have implemented DirMaint, then you need to obtain a copy of the USER WITHPASS file from the DIRMAINT virtual machine (as shown in Example 2-1). If you have not implemented DirMaint, you can use a copy of the USER DIRECT file on MAINT's 2CC disk. When you receive this file, save it as *A2* file to allow the RACF user ID to access the file in later steps. This file is required when the RPIDIRECT EXEC runs later.

Example 2-1 DIRM USER WITHPASS

dirm user withpass

```
DVHXMT1191I Your USER request has been sent for processing.  
Ready; T=0.03/0.03 11:38:50  
DVHREQ2288I Your USER request for MAINT at * has been accepted.  
RDR FILE 0012 SENT FROM DIRMAINT PUN WAS 0020 RECS 2261 CPY 001 A NOHOLD  
DVHREQ2289I Your USER request for MAINT at * has completed; with RC = 0.
```

receive 12 user withpass a2 (replace

```
File USER WITHPASS A2 replaced USER WITHPASS A0 with USER WITHPASS A0 rec  
from DIRMAINT at VMLINUX5  
Ready; T=0.01/0.01 11:39:15
```

The easy way is to perform a global XEDIT **change** command and change NOLOG to UNLOG within the file, as shown in Figure 2-1.

```
USER      WITHPASS A2  F 80  Trunc=80 Size=220
==> ch /NOLOG/UNLOG/*
  90 USER $ALLOC$  NOLOG
  96 USER $DIRECT$ NOLOG
 100 USER $SYSCKP$ NOLOG
 104 USER $SYSWRM$ NOLOG
 108 USER $PAGE$   NOLOG
 112 USER $SPOOL$  NOLOG
 116 USER $TDISK$  NOLOG
 728 USER ROOT NOLOG 32M 32M G
 732 USER DAEMON NOLOG 32M 32M G
 736 USER BIN NOLOG 32M 32M G
 740 USER SYS NOLOG 32M 32M G
 744 USER ADM NOLOG 32M 32M G
 748 USER NOBODY NOLOG 32M 32M G
 752 USER DEFAULT NOLOG 32M 32M G
2203 * * * End of File * * *
```

Figure 2-1 USER WITHPASS

Evaluate minidisk access

The system ships several users minidisk with a read password of *ALL* because RPIDIRECT EXEC generates statements such as:

```
RDEFINE VMMDISK MAINT.190 OWNER(MAINT) UACC(READ)
```

While there are several minidisks on the system that *must* have a UACC(READ), not all them need to have this access. We suggest that you evaluate whether the disks for all of the OSASF virtual machines really need to have this access (especially if you are not using this product). Most environments will have to justify to the auditors why they have disks that have a UACC(READ). Correcting this problem before running RPIDIRECT makes the task much simpler.

Update RACF user ID directory entry

If you are using DirMaint, the RACF product owner virtual machine has to link and access DIRMAINT's 1DF disk. The system is shipped without a read password for this disk. To solve this issue, we decided to change directory statements for the 5VMRAC30 virtual machine. We found it easier to give the virtual machine 5VMRAC30 the directory statement OPTION LNKNOPAS to make it easier when logged on to that virtual machine and to link and access several other virtual machines disks. We do not see this as a security exposure. As soon as RACF is implemented, this directory entry will be nullified, because the control of minidisks linking will be controlled by RACF authorization.

First, we used the **dirm for 5vmrac30 get** command to retrieve the 5VMRAC30 virtual machine definition. Then, we added the **OPTION LNKNOPAS** statement (as shown in Figure 2-2) and updated the DirMaint database with the **dirm for 5vmrac30 replace** command.

```
5VMRAC30 DIRECT  A0  F 80  Trunc=72 Size=26 Line=0 Col=1
==>
0 * * * Top of File * * *
1 USER 5VMRAC30 5VMRAC30 20M 32M BG
2 ACCOUNT 5VMRAC30
3 MACH XA
4 IPL CMS
5 OPTION LNKNOPAS
6 CONSOLE 009 3215
7 SPOOL 00C 2540 READER *
8 SPOOL 00D 2540 PUNCH A
9 SPOOL 00E 1403 A
10 ----- 17 line(s) not displayed ----
27 * * * End of File * * *
```

Figure 2-2 5VMRAC30 directory entry

Execute RPIDIRCT

RPIDIRCT exec is used to generate the RPIDIRT SYSUT1 file that contains all the RACF commands to add the users, to define **vmmdisk** and **vmrdr**, and to permit the owners to the resources. This exec is run from the product owning virtual machine for RACF.

Prior to running the exec, you must link and access DIRMAINT's 1DF disk and MAINT's 191 disk and the 5VMRAC30's 505 disk where the The RPIDIRECT EXEC is located. We suggest that you issue the `cp term more 0 0` command, which makes the exec run faster, as shown in Figure 2-3.

```
link maint 191 2cc rr
Ready; T=0.01/0.01 12:58:54

acc 2cc f
DMSACP723I F (2CC) R/O
Ready; T=0.01/0.01 12:58:58

link dirmaint 1df 1df rr
DASD 01DF LINKED R/O; R/W BY DIRMAINT
Ready; T=0.01/0.01 12:59:04

ac 1df g
DMSACP723I G (1DF) R/O
Ready; T=0.01/0.01 12:59:09

acc 505 e
Ready; T=0.01/0.01 12:59:14

cp term more 0 0
```

Figure 2-3 Setting up to run RPIDIRECT EXEC

When running the RPIDIRECT EXEC, you must provide the file name and file type of the source directory file. It searches for the file on all accessed disks. The default output file mode is *A*. When the exec starts, it prompts you for the default group ID. We used the default, so we replied *N* to the question.

Figure 2-4 shows an example of the `rpidirct user withpass` command.

```

USER WITHPASS Filemode defaulted to "*".
Output defaulted to "A" disk.
  Default group ID = SYS1.
  Would you like to change this default?
  Enter Y/N
N
  Default group ID = SYS1.
The file, RPIDIRECT CNTRL, was found and will be used to
override the directory specifications.
*****
                        DEFINITION pass begins.....
*****
PROFILE IBMDFLT
  PROFILE TCPCMSU
PROFILE TCPGCSU
----- 2527 line(s) not displayed -----

***** 2202 Directory records processed *****

***** RPIDIRECT SYSUT1 CREATED *****

```

Figure 2-4 Running RPIDIRECT EXEC

Note: The *Secure Configuration Guide for z/VM*, SC24-6139 suggests that, after running RPIDIRECT, you modify the resulting RPIDIRECT SYSUT1 file in the following ways:

- ▶ Alter the VMRDR profile for MAINT to specify UACC(UPDATE)
- ▶ Add any additional PERMITs required

You also need to run the RPIBLDDS exec in order to execute the commands in this file. This step will be done later.

We also suggest that you make the same changes to the following virtual machines:

TCPMAINT	The TCP/IP daemon virtual machines spool their console to TCPMAINT
DIRMAINT	To make the DIRM SEND command work
DATAMOVE	To allow DIRMAINT to send files to DATAMOVE

Customize the processing of SMF records (optional)

One of the reasons that you run RACF on your z/VM system is to be able to audit who is doing what on the system. If you want to use the RACFSMF virtual machine, you must set up the PROFILE EXEC and the SMF CONTROL files. For that you need to perform the following tasks:

- ▶ Create the PROFILE EXEC from the SMFPROF EXEC
- ▶ Modify the SMF CONTROL file

Create the RACFSMF PROFILE EXEC

You create the PROFILE EXEC by copying the SMFPROF EXEC from the 291 minidisk, as shown in Example 2-2.

Example 2-2 Creating the PROFILE EXEC for RACFSMF

```
set pf12 retrieve
Ready; T=0.01/0.01 15:59:03
link racfsmf 191 291 mr
Ready; T=0.01/0.01 15:59:10
ac 291 m
Ready; T=0.01/0.01 15:59:21
copy smfprof exec e profile exec m
Ready; T=0.01/0.01 15:59:51
```

After you copy the file, modify the SMFFREQ and SMFSWTC parameters to match Example 2-3.

Example 2-3 RACFSMF's PROFILE EXEC

```
PROFILE EXEC      M1 V 130 Trunc=130 Size=428 Line=124 Col=1
====>
124 Smfpct      = 80
125 Smfinfo     = 'OPERATOR'      /* Default message \r
126 Smffreq     = ' AUTO '        /* Valid values: DAILY, WEEKLY,
127                                     /*          AUTO
128 Smfday       = 'MONDAY'        /* Valid values: SATURDAY - FRI
129 Smfswtch    = ' NO '          /* Valid values: YES NO
130 /* 1 line deleted
131 hi          = '1de8'x
132 lo          = '1d60'x
```

Modify the SMF CONTROL file

The Program Directory tells you to detach RACFSMF's 191 disk when you complete the work on the PROFILE EXEC. However, the very next step tells you to link and access this disk again, because you need to copy the SMF CONTROL file to several disks. The SMF CONTROL file resides on RACFVM's 191 disk. The directions tell you to copy it to RACFSMF's 191 disk and modify it and then to copy it back to the original disk. We think you will find it easier to modify the one on RACFVM's disk and then copy it to the two other disk.

First, you need to link and access the appropriate disk. Because, you still have RACFSMF's 191 disk, you can complete the steps in Example 2-4.

Example 2-4 Accessing the appropriate disk

```
link racfvm 191 391 mr
Ready; T=0.01/0.01 16:22:22
link racmaint 191 491 mr
Ready; T=0.01/0.01 16:22:29
ac 391 n
Ready; T=0.01/0.01 16:22:34
acc 491 o
Ready; T=0.01/0.01 16:22:37
```

Edit the SMF CONTROL file on the *N* disk (which is RACFVM's 191 disk). Make the change shown in Example 2-5. The file contains only one line, so we split it into two lines for readability.

Example 2-5 SMF CONTROL

```
SMF      CONTROL N1 F 100 Trunc=100 Size=2
====>
* * * Top of File * * *
CURRENT 301 K PRIMARY 301 K SECONDARY 302 K
10000 VMSP CLOSE 001 SEVER YES 0 RACFSMF
* * * End of File * * *
```

After modifying this file, you have to copy it to the *M* and *O* disks. Then the **filelist smf control *** command should return results similar to those shown in Example 2-6.

Example 2-6 File list of SMF CONTROL

```
5VMRAC30 FILELIST A0 V 169 Trunc=169 Size=4 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrec1  Records  Blocks  Date
SMF   CONTROL M1 F      100      1      1  7/13/07
SMF   CONTROL N1 F      100      1      1  7/13/07
SMF   CONTROL O1 F      100      1      1  7/13/07
SMF   CONTROL E1 F      100      1      1 11/29/05
```

The SMF CONTROL file on the *E* disk is your original file located on 5VMRAC30s 505 disk, and it should not be changed. Now you can detach the 291, 391, and 491 disks.

Remove ICHDEX01 and ICHRCX02 (optional)

Even though this step is considered optional, we recommend that you remove the ICHDEX01 (Hashing or DES Encryption) and ICHRCX02 exits that are enabled by default.

- ▶ ICHDEX01 removal enables the DES encryption of passwords and password phrases in the RACF database.
- ▶ ICHRCX02 removal disables batch-mode alternate user IDs user support.

Delete the ICHDEX01 exit

This section shows how to remove the ICHDEX01 member from the RACFLPA LOADLIB. To perform this step, you do *not* need the high level assembler. To delete ICHDEX01 exit, follow the instructions in Appendix B.3 of the Program Directory, “Local Modification to Full Part Replacement Text Files” using the following substitution values:

- ▶ For *fn* use *ICHDEX01*
- ▶ For *blist* use *RPIBLLPA*

You should use the VMSES/E process to create a local modification to this load library. A local copy of the RPIBLLPA EXEC should be created and the local modification should be logged in the local version vector table for the product. The local version vector table is nothing more than a log file of the parts you have performed local service for. It is very important to complete these steps so that future IBM service to this part will not overlay your local modifications.

The first step in deleting this member of the RACFLPA LOADLIB is to establish the 5VMRAC30's minidisk order. We used the VMSES/E exec VMFSETUP to perform this step (Example 2-7).

Example 2-7 VMFSETUP for RACF

```
ac 590 t
DMSACC724I 590 replaces T (590)
Ready; T=0.01/0.01 09:15:46

vmfsetup 5vmrac30 racf
VMFSET2760I VMFSETUP processing started for 5VMRAC30 RACF
VMFUTL2205I Minidisk|Directory Assignments:
      String      Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E    R/W   2C2   RAC2C2
VMFUTL2205I APPLY    F    R/W   2A6   RAC2A6
VMFUTL2205I          G    R/W   2A2   RAC2A2
VMFUTL2205I DELTA    H    R/W   2D2   RAC2D2
VMFUTL2205I BUILD0   I    R/W   29E   RAC29E
VMFUTL2205I BUILD6   J    R/W   599   RAC599
VMFUTL2205I BUILD4   K    R/W   505   RAC505
VMFUTL2205I BUILD2   T    R/W   590   RAC590
VMFUTL2205I BASE     U    R/W   2B2   RAC2B2
VMFUTL2205I -----  A    R/W   191   RAC191
VMFUTL2205I -----  B    R/O   5E5   MNT5E5
VMFUTL2205I -----  D    R/W   51D   MNT51D
VMFUTL2205I -----  S    R/O   190   MNT190
VMFUTL2205I -----  Y/S   R/O   19E   MNT19E
VMFSET2760I VMFSETUP processing completed successfully
```

The next step is to determine the highest level of service to the build list for the RACFLPA load library by using the VMFSIM EXEC with the GETLVL parameter. The exec searches all of the version vector tables for this product and determine the highest level of service. It returns the file name and file type of that part. If you do not execute the VMFSETUP exec before you run the VMFSIM exec, you will not get the correct results.

Example 2-8 shows the **vmfsim getlvl** command. It gives you the file name and file type of the file that you need to copy to create your new file.

Example 2-8 The vmfsim getlvl command

```
vmfsim getlvl 5VMRAC30 RACF tdata :part rpibllpa exc (history
:PART RPIBLLPA EXC00000 BASE-FILETYPE
Ready; T=0.06/0.06 09:20:43
```

The output from the **vmfsim getlvl** command lists this element as BASE-FILETYPE. In VMSES/E terminology it means that there has been no service to this part by IBM or locally by a system programmer (no entries in the IBM and Local Version Vector Tables). In our case, we used the RPIBLLPA EXEC. You need to determine on which disk the base file is located. Copy the highest level of the build list to the 2C2 local disk (E-disk).

Use following syntax:

```
copyfile blist ft * = exclnnnn 2c2_fm
```

where *blist* is the file name to be copied, *ft* is the file type, *nnnn* is a local tracking number that you supply and *2c2_fm* is the filemode of the 2C2 minidisk. Because this is the first modification to this file, we use 0001 as the *nnnn* value and file mode e to reflect the 2C2 minidisk, as follows:

```
copyfile rpihlpa exec u = excl0001 e
```

Modify the RPIBLLPA EXCL0001 on the E disk and comment out the entry for the ICHDEX01 member, as shown in Example 2-9.

Example 2-9 RPIBLLPA EXCL0001

```
RPIBLLPA EXCL0001 E1 F 80 Trunc=80 Size=749 Line=456 C
====>
456 *
457 *:OBJNAME. ICHDEX01 LEPARMS RENT REUS LET NCAL XREF
458 *:BLDREQ. RPIBLOBJ.ICHDEX01
459 *:OPTIONS. CONCAT SYSLIB RACFOBJ
460 *:OPTIONS. INCLUDE RACFOBJ(ICHDEX01)
461 *:OPTIONS. ENTRY ICHDEX01
462 *:EOBJNAME.
463 *
```

Log this local modification to the RPIBLLPA EXEC into the local version vector table. In prior releases of z/VM, the VMFSIM MODIFY command was used. Starting with z/VM 5.2.0, you can use the VMFSIM LOGMOD command with more user-friendly syntax:

```
vmfsim logmod 5VMRAC30 vvtlcl e tdata :mod lcl0001 :part rpihlpa exc
```

The 2C2 disk should now contain 5VMRAC30 VVTLCL and RPIBLLPA EXCL0001 files. Example 2-10 shows the content of the 5VMRAC30 file.

Example 2-10 File list of the 2C2 disk

```
5VMRAC30 FILELIST A0 V 169 Trunc=169 Size=2 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrecl  Records  Blocks  Date
      5VMRAC30 VVTLCL  E1 V      32         1        1  7/14/07
      RPIBLLPA EXCL0001 E1 F      80       749       15  7/14/07

5VMRAC30 VVTLCL  E1 V 80 Trunc=80 Size=1
====>
0 * * * Top of File * * *
1 :PART.RPIBLLPA EXC :MOD.LCL0001
2 * * * End of File * * *
```

Next, generate a new RACFLPA LOADLIB using the VMFBLD command. When you issue the command, make sure that you specify the **blist** parameter (in this case **rpibllpa**). If you do not, then all buildlist listed in the BLD section of the 5VMRAC30 PPF file would be built (Example 2-11).

```
vmfbld ppf 5VMRAC30 RACF rpibllpa (all)
```

Example 2-11 VMFBLD process for loadlib

```
VMFBLD2195I VMFBLD PPF 5VMRAC30 RACF RPIBLLPA ( LOG CNTRL RPIVM NOCKVV
          NOSETUP ALL
VMFBLD2760I VMFBLD processing started
VMFUTL2205I Minidisk|Directory Assignments:
          String      Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E      R/W  2C2   RAC2C2
----- 13 line(s) not displayed -----
VMFBLD1851I Reading build lists
VMFBLD2182I Identifying new build requirements
VMFBLD2182I New build requirements identified
VMFBLD1851I (1 of 1) VMFBDLLB processing RPIBLLPA EXCL0001 E, target
          is BUILD4 505 (K)
VMFLLB2217I RACFLPA LOADLIB will be rebuilt because all members must
          be rebuilt
----- 66 line(s) not displayed -----
VMFBLD1851I (1 of 1) VMFBDLLB completed with return code 0
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
```

To place the new local modification into production, you will need to link to RACFVM's 305 disk and then use the VMFCOPY command to copy the files to the production disk (Example 2-12). The VMFCOPY updates the VMSES PARTCAT file on the 305 disk.

Example 2-12 Place changes into production

```
link RACFVM 305 305 MR
acc 505 e
acc 305 f
vmfcopy RACFLPA * e = f (prodid 5VMRAC30%RACF replace oldd
```

Delete the ICHRCX02 exit

We also suggest that you remove ICHRCX02 member from the loadlib to disable batch-mode alternate user support. Follow the same steps that we describe in "Delete the ICHDEX01 exit" on page 29. Create your new buildlist, log the local modification into the local version vector table, build the loadlib again, and copy the files to RACFVMs 305 disk.

After you finish, you should have three files as shown in Example 2-13.

Example 2-13 5VMRAC30s 2C2 disk

```
5VMRAC30 FILELIST A0 V 169 Trunc=169 Size=3 Line=1 Col=1 A1
Cmd  Filename Filetype Fm Format Lrec1      Records      Blocks
*    5VMRAC30 VVTLCL  E1 V          40          1          1
      RPIBLLPA EXCL0002 E1 F          80         749         15
      RPIBLLPA EXCL0001 E1 F          80         749         15
```

```
5VMRAC30 VVTLCL  E1 V 80 Trunc=80 Size=1 Line=0
```

```

====>
0 * * * Top of File * * *
1 :PART.RPIBLLPA EXC :MOD.LCL0002 LCL0001
2 * * * End of File * * *

```

2.2.2 Build the RACF enabled CPLOAD MODULE

Make sure that you have logged off from the RACF product owner virtual machine and log on to the MAINT virtual machine. When the PROFILE EXEC completes running, you have all the required disks accessed. The RACF product is shipped on the system in a disabled state. The next step uses the CP command **set product** to enable this product to run and generate the CPLOAD MODULE that enables RACF to CP. In other words, the CP nucleus requires the system to have an External Security Manager (ESM) to manage authentication for the z/VM system. The initial settings for the ESM are that if a resource is not defined to the ESM, then it is deferred to CP for a final decision. Later in the process, you change this setting to secure the system so that all resources must be defined to the ESM. Otherwise, the request for accessing will fail.

Issue the **service racf enable** command. See Figure 2-5 for output.

Note: The new CP nucleus, with the RACF CP parts, is placed on the secondary parm disk (default disk address is CF2). For your information, a copy of the previous (or currently running) CPLOAD MODULE is still on the primary (CF1) and tertiary (CF3) parm disks as CPLOAD MODULE. It is also saved on the secondary parm disk as CPLDOLD MODULE.

```

VMFSRV2195I SERVICE RACF ENABLE
VMFSRV2760I SERVICE processing started
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2602R The following components can be enabled for PROD 5VMRAC30
RACF. Enter the number of your choice
(0) Bypass this product
(1) :PPF 5VMRAC30 RACF :PRODID 5VMRAC30%RACF
:DESC RACF Feature of z/VM, FL530
(2) :PPF SERVP2P RACF :PRODID 5VMRAC30%RACF
:DESC RACF Feature for z/VM, FL530
(3) Exit
VMFINS2603I Processing product :PPF 5VMRAC30 RACF :PRODID
5VMRAC30%RACF
VMFINS2603I Enabling product 5VMRAC30%RACF
VMFINS2771I The CP SET PRODUCT command completed successfully for
product 5VMRAC30
VMFINS2772I File 5VMRAC30 PRODSYS created on your A-disk contains the
system configuration PRODUCT statement for product
5VMRAC30
VMFINS2603I PRODUCT ENABLED IN VMSES/E, CP PROCESSING REQUIRED
VMFINS2760I VMFINS PROCESSING COMPLETED SUCCESSFULLY
HCPZAC6730I CPRELEASE REQUEST FOR DISK A COMPLETED.

```

Figure 2-5 CP SET PRODUCT

When the product is enabled dynamically, the configuring of RACF by the service exec sets a flag in a VMSES/E software inventory table. This flag causes the CP nucleus to be built using the RACF versions of the HCPRWA, HCPRPD, HCPRPW, HCPRPI, HCPRPG, and HCPRPF files. The SERVICE EXEC then generates a new CPLOAD MODULE and places it on the CF2 disk only (Figure 2-6). It is moved to the CF1 and CF3 disks in a later step.

```
*****
*           Date: 07/14/07           Time: 10:42:10           **
*****
VMFSRV2195I SERVICE RACF ENABLE
VMFSRV2760I SERVICE processing started
----- 256 line(s) not displayed -----
HCPZAC6730I CPRELEASE request for disk C completed.
DMSACP723I X (CF1) R/O
HCPZAC6730I CPRELEASE request for disk B completed.
HCPZAC6732I CPACCESS request for MAINT's 0CF3 in mode C completed.
VMFSRV2760I SERVICE processing completed successfully for CP BUILD
HCPZAC6732I CPACCESS request for MAINT's 0CF2 in mode B completed.
VMFSRV2760I SERVICE processing started for RACF BUILD
----- 58 line(s) not displayed -----
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
----- 8 line(s) not displayed -----
VMFSRV2760I SERVICE processing completed unsuccessfully
```

Figure 2-6 Generation of the CPLOAD Module on CF2

IPL your system with RACF in a test mode

When the SERVICE EXEC completes, issue the VMFVIEW command to verify that there are no problems. Shut down your system and then, using the LOADPARM option, IPL your system again. From the SAPL screen select extent 2 of your sysres volume (CF2), as shown in Figure 2-7. You must tab to the extent field and change from extent 1 to extent 2 and then press PF10 to start the IPL.

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 5 RELEASE 3.0

DEVICE NUMBER:  0423      MINIDISK OFFSET:  00000000  EXTENT:  2

MODULE NAME:    CPLOAD    LOAD ORIGIN:      2000

-----IPL PARAMETERS-----

-----COMMENTS-----

-----

          9= FILELIST  10= LOAD  11= TOGGLE EXTENT/OFFSET
```

Figure 2-7 IPL from Extent 2

During IPL process you must perform a NOAUTOLOG start and change the time of day if required (Figure 2-8). The NOAUTOLOG option tells system *not* to start the AUTOLOG1 virtual machine. Therefore, no other virtual machines are started automatically. When the IPL completes, you start RACMAINT virtual machine with the **xauto1og** RACMAINT command. The reason for starting RACMAINT, instead of RACFVM is that in a later step, you run the PUT2PROD exec. This exec copies files to the RACFVM virtual machine disks. RACMAINT links to those disk to run in READ ONLY mode, thus allowing MAINT and the PUT2PROD exec to gain write access to the disks owned by RACFVM.

```

11:08:23 z/VM V5 R3.0 SERVICE LEVEL 0701 (64-BIT)
11:08:24 SYSTEM NUCLEUS CREATED ON 2007-07-14 AT 10:43:01, LOADED FROM
LX5RES
11:08:24
11:08:24 *****
11:08:24 * LICENSED MATERIALS - PROPERTY OF IBM* *
11:08:24 * * *
11:08:24 * 5741-A05 (C) COPYRIGHT IBM CORP. 1983, 2007. ALL RIGHTS *
11:08:24 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE, *
11:08:24 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE *
11:08:24 * CONTRACT WITH IBM CORP. *
11:08:24 * *
11:08:24 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES. *
11:08:24 *****
11:08:24
11:08:24 HCPZCO6718I Using parm disk 2 on volume LX5RES (device 0423) .
11:08:24 HCPZCO6718I Parm disk resides on cylinders 159 through 278.
11:08:24 Start ((Warm|Force|COLD|CLEAN) (DRain) (Disable) (NODIRect)
11:08:24 (NOAUTOlog)) or (SHUTDOWN)
11:16:46 NOAUTOLOG
11:16:46 NOW 11:16:46 EDT SATURDAY 2007-07-14
11:16:46 Change TOD clock (Yes|No)
NO

```

Figure 2-8 z/VM IPL

When the RACMAINT virtual machine logs on and runs the PROFILE EXEC, it executes RACSTART EXEC. This causes this virtual machine to be defined as the ESM for your system. You can ignore the messages about the 591 and 505 disk not being accessed. This will not cause a problem. You can now disconnect from the OPERATOR virtual machine.

2.2.3 Update the RACF database and options

The following tasks are needed to update the RACF database with information from the CP directory and to set up options for the RACF environment.

Update the RACF database with existing CP Directory

Log on to the IBMUSER virtual machine. This virtual machine is defined to have RACF special and operations authority in the initial RACF database that was shipped with the system. The password for this virtual machine is *SYS1*, and you have to change the password the first time that you log on.

You perform the following tasks from this virtual machine:

1. Set a PF key to retrieve commands.
2. Run RPIBLDDS to build the RACF database.
3. Define the security administrator.

Before you can build the RACF database, you need to link to several of the product owners' disks and access them (Setting up the IBMUSER virtual machine):

- ▶ 191 - Location of the RPIDIRCT SYSUT1 file
- ▶ 305 - Location of the RPIBLDDS EXEC
- ▶ 29E - Location of the RAC EXEC

```
set pf12 retrieve
Ready; T=0.01/0.01 11:25:46
link 5vmrac30 505 305 rr
RPIMGR031E RESOURCE 5VMRAC30.505 SPECIFIED BY LINK COMMAND NOT FOUND
DASD 0305 LINKED R/O; R/W BY RACMAINT
Ready; T=0.01/0.01 11:33:40
link 5vmrac30 191 192 rr
RPIMGR031E RESOURCE 5VMRAC30.191 SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 11:33:59
link 5vmrac30 29e 29e rr
RPIMGR031E RESOURCE 5VMRAC30.29E SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 11:34:06
ac 305 c
ac 192 b
ac 29e d
```

Figure 2-9 Setting up the IBMUSER virtual machine

The RPIBLDDS EXEC is used to modify the RACF DATABASE. It uses the RPIDIRCT SYSUT1 file as input. This file was created earlier by the 5VMRAC30 virtual machine with the RPIDIRECT EXEC. It contains all the RACF commands to add users, define resources, and authorize users to resources. See Figure 2-10.

```

rpibldds
Using default file RPIDIRCT SYSUT1
Processing batch file RPIDIRCT SYSUT1 using "RAC" command interface
=> RDEFINE VMCMD RACF UACC(READ)
=> RDEFINE VMCMD RAC UACC(READ)
=> ADDGROUP SYSTEM
=> ALTGROUP SYSTEM OVM(GID(0))
=> ADDGROUP STAFF
=> ALTGROUP STAFF OVM(GID(1))
=> ADDGROUP GBIN

=> RDEFINE VMMDISK MAINT.5A2 OWNER(MAINT) UACC(NONE)
=> PERMIT MAINT.5A2 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
=> RDEFINE VMMDISK MAINT.5A4 OWNER(MAINT) UACC(NONE)
=> PERMIT MAINT.5A4 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
=> RDEFINE VMMDISK MAINT.5A6 OWNER(MAINT) UACC(NONE)
=> PERMIT MAINT.5A6 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)

=> PERMIT 5VMTCP30.491 CLASS(VMMDISK) ID(LDAPSRV) ACCESS(READ)
=> PERMIT 5VMTCP30.492 CLASS(VMMDISK) ID(LDAPSRV) ACCESS(READ)
=> PERMIT TCPMAINT.591 CLASS(VMMDISK) ID(LDAPSRV) ACCESS(READ)
=> PERMIT TCPMAINT.198 CLASS(VMMDISK) ID(LDAPSRV) ACCESS(READ)

=> PERMIT TCPMAINT.591 CLASS(VMMDISK) ID(VSMPROXY) ACCESS(READ)
Ready; T=1.32/1.72 11:44:51

```

Figure 2-10 Running RPIBLDDS

When the RPIBLDDS EXEC completes, your RACF database is initialized with all the virtual machines and resources that were shipped with the z/VM system. Now, you create additional RACF administrators. You need to determine what virtual machines are trusted to manage your secure environment.

We suggest the following virtual machines at a minimum:

- ▶ MAINT
- ▶ OPERATOR
- ▶ SYSADMIN
- ▶ BLDSEG

Note: We feel that it is important to give OPERATOR the RACF attributes of SPECIAL and OPERATIONS. The reason is that if during the IPL process of the system, an operator accidentally set the date incorrectly (such as the wrong year) would expire all of the passwords for all the virtual machines. By having the appropriate RACF attributes the OPERATOR virtual machine could set passwords for other virtual machines.

The RACF **altuser** command is used to modify the RACF attributes for a virtual machine (Example 2-14).

Example 2-14 Setting RACF attributes

```
rac alu maint operations special
Ready; T=0.01/0.01 11:49:44
rac alu operator operations special
Ready; T=0.01/0.01 11:49:53
rac alu sysadmin operations special
Ready; T=0.01/0.01 11:50:02
rac alu bldseg operations special
Ready; T=0.01/0.01 11:50:04
```

After the new RACF administrator are defined, log off from IBMUSER. Log on to MAINT, assuming that you gave MAINT RACF authority, and complete the installation of the product.

Because the virtual machine IBMUSER is a well known user, it might be a target for unauthorized accesses to your system. To prevent further use of the IBMUSER virtual machine, it is recommended that you revoke this virtual machine and remove the operations and special attributes (Example 2-15). *Do not delete this virtual machine from your system because IBMUSER ran the exec to generate the RACF database and it is now listed as the owner of all the other virtual machines on the system.*

Example 2-15 RACF alter user for IBMUSER

```
link 5vmrac30 29e 29e rr
RPIMGRO31E RESOURCE 5VMRAC30.29E SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 12:00:18
access 29e 1
DMSACP723I L (29E) R/O
Ready; T=0.01/0.01 12:00:23
rac alu ibmuser revoke
Ready; T=0.01/0.01 12:04:01
rac alu ibmuser nospecial
Ready; T=0.01/0.01 12:04:08
rac alu ibmuser nooperation
Ready; T=0.01/0.01 12:04:14
```

Set RACF options (optional)

At this point, you should also define which resources should be managed by RACF. We suggest the following list as a good starting point:

```
RAC SETROPTS CLASSACT(VMMDISK)
RAC SETROPTS CLASSACT(VMRDR)
RAC SETROPTS CLASSACT(VMLAN)
```

Other options can be VMBATCH and VMSEGMT.

It would be also a good task to make the corresponding updates to the VMXEVENT to tailor this entry to your installation. This avoids RACF calls for resources that are not RACF protected and, consequently, avoids wasting CPU cycles and cause RACF contention.

2.2.4 Place RACF into production

Run **PUT2PROD EXEC** from the MAINT virtual machine.

Note: If you have installed the RACF code in the z/VM shared file system, the *Secure Configuration Guide for z/VM*, SC24-6139 suggests that you install RACF to minidisk and **xauto1og** the AUTOLOG1 virtual machine before running **PUT2PROD EXEC**.

Note: Make sure you run the PUT2PROD EXEC without any parameters, because the \$SERVICE PROD file on MAINT's 191 disk already lists the components that need to be put into production:

```
SERVICE $PRODS A1 V 80 Trunc=80
0 * * * Top of File * * *
1 SERVP2P RACF
2 SERVP2P CP
3 * * * End of File * * *
```

When PUT2PROD has completed, you should run **vmfview put2prod** to verify that everything was successful.

Set up AUTOLOG1 and AUTOLOG2

When doing a normal warm start, the IPL process starts the AUTOLOG1 virtual machine. AUTOLOG1 only starts your ESM (RACFVM). After the RACF environment is initialized, RACF issues the **xauto1og** command for the AUTOLOG2 virtual machine, which starts the remaining servers for the system.

The existing PROFILE EXEC for the AUTOLOG1 virtual machine works perfectly for the AUTOLOG2 virtual machine. So, you can just copy it to the appropriate disk. You then need to modify the PROFILE EXEC for AUTOLOG1 to only start the production ESM (RACFVM). See Figure 2-11.

```
link autolog1 191 11 mr
Ready; T=0.01/0.01 12:25:07
link autolog2 191 12 mr
Ready; T=0.01/0.01 12:25:14
ac 11 x
Ready; T=0.01/0.01 12:25:18
acc 12 z
DMSACC724I 012 replaces Z (011)
Ready; T=0.01/0.01 12:25:29
copy profile exec x = z
Ready; T=0.01/0.01 12:25:53

PROFILE EXEC      X1 V 130 Trunc=130 Size=7
==>
0 * * * Top of File * * *
1 /*****
2 /* Autolog1 Profile Exec */
3 /*****
4 'CP XAUTOLOG RACFVM'
5 'CP LOGOFF'
6 * * * End of File * * *
```

Figure 2-11 Setting up the AUTOLOG1 and AUTOLOG2 virtual machines

You should be able to IPL from the CF1 disk (extent 1) and run in production mode.

2.2.5 Using HCPRWAC

Initially, the system is built with non-aggressive authorization checking with the security parameters in the SYSSEC macro. In fact, most of the entries specify the key word *defer*, which means that if the ESM does not know what to do with a request, the request is routed to the system CP for determination. This is not a very secure model to run the production system. For this reason, we recommend that after everything is working correctly, you might want to change the SYSSEC macro to *fail* instead of *defer*. You must have the high level assembler to perform local modifications to assembler files.

Issue the **vmfupdat sysuf** command (Figure 2-12).

```
HCPRWA  RPIBASE0 E1 F 80          3 Blks 10/25/06 Line    118 of
====>
      SYSSEC ,                      X
      DISK=ALLOW,DISKU=DEFER,DISKF=FAIL,DISKW=DEFER,DISKM=ON,X
      RDRP=ALLOW,RDRU=DEFER,RDRF=FAIL,RDRW=DEFER,RDRM=ON,    X
      NODEP=ALLOW,NODEU=DEFER,NODEF=FAIL,NODEW=DEFER,NODEM=ON,X
      CMDP=ALLOW,CMDU=DEFER,CMDF=FAIL,CMDW=DEFER,CMDM=ON,    X
      LANP=ALLOW,LANU=DEFER,LANF=FAIL,LANW=DEFER,LANM=ON,    X
      DEFLT=ALLOW,DEFLTU=DEFER,DEFLT=FAIL,DEFLT=DEFER @L2C
      SPACE 3
```

Figure 2-12 HCPRWA assemble file

The RACF product is shipped with pre-assembled files for this member of the CP nucleus. The process of how to rebuild your system with the appropriate changes to the HCPRWAC ASSEMBLE file is documented in the *Secure Configuration Guide for z/VM*, SC24-6139 in Appendix C. See also Figure 2-13.

```
HCPRWA  RBOL0001 E1 F 80  Trunc=80 Size=137 Line=120 Col=1 Alt=2
====>
120      SYSSEC ,                      X
121      DISK=ALLOW,DISKU=FAIL,DISKF=FAIL,DISKW=FAIL,DISKM=ON, X
122      RDRP=ALLOW,RDRU=FAIL,RDRF=FAIL,RDRW=FAIL,RDRM=ON,    X
123      NODEP=ALLOW,NODEU=FAIL,NODEF=FAIL,NODEW=FAIL,NODEM=ON, X
124      CMDP=ALLOW,CMDU=FAIL,CMDF=FAIL,CMDW=FAIL,CMDM=ON,    X
125      LANP=ALLOW,LANU=FAIL,LANF=FAIL,LANW=FAIL,LANM=ON      X
126      DEFLT=ALLOW,DEFLTU=DEFER,DEFLT=FAIL,DEFLT=DEFER @L2C
      SPACE 3
```

Figure 2-13 Modified HCPRWAC assemble

HCPRWAC is the IBM-provided modification of HCPRWA that complies with the requirements of LSPP. We will follow these step to further enable our system. This process uses VMFUPDAT to update VM SYSSUF (Figure 2-14).

*** Update SYSSUF Table Entries ***

Update any PPF/component name or YES|NO field. To change all occurrences of a PPF name in the table replace both ***** fields with PPF names.

Compname	Prodlev	Servlev	Prodlev	Description

OSA	4OSASF40	RSU-0701	RSU-0701	OSASF for VM
:INSTALL	YES	:INSPPF	SERV2P	OSA
:BUILD	YES	:BLDPPF	SERV2P	OSA
:INCLUDE	YES	:P2PPF	SERV2P	OSAP2P
PERFTK	5VMPTK30	000-0000	000-0000	Performance Tool Kit
:INSTALL	YES	:INSPPF	SERV2P	PERFTK
:BUILD	NO	:BLDPPF	SERV2P	PERFTK
:INCLUDE	YES	:P2PPF	SERV2P	PERFTKP2P
RACF	5VMRAC30	000-0000	000-0000	RACF Feature of z/VM, FL530
:INSTALL	YES	:INSPPF	SERV2P	RACF
:BUILD	YES	:BLDPPF	SERV2P	RACF
:INCLUDE	CCC	:P2PPF	SERV2P	RACFP2P

Change PPF name ***** to *****

Page 4 of 6

PF1=HELP PF3/PF12=Quit PF5=Process PF6=VMFSUFTB PF7=Backward PF8=Forward

Figure 2-14 VMFUPDAT SYSSUF

After you modify the entry for INCLUDE from *YES* to *CCC*, you select PF5 to process. This raises a flag in the VM SYSSUF file meaning that RACF was updated and to set this product to BUILD (Figure 2-15). This in turn causes the CPLOAD MODULE to be built with the new HCPRWA file (which is actually the HCPRWAC file). This changes the parameters from *defer* to *fail*.

```

VM          SYSSUF   D1  V 100  Trunc=100 Size=41 Line=30 Col=1 Alt=0
====>
30 :PRODID.5684042J%ICKDSF :SERVLEV.ESO-0610 :DESC.ICKDSF DEVICE SUPPORT F
31 :INCLUDE.YES :INSTALL.YES :INSPPF.SERV2P ICKDSF :BUILD.YES :BLDPPF.SER
32 ICKDSFP2P :PRODLEV.ESO-0610
33 :PRODID.5VMDIR30%DIRM :SERVLEV.000-0000 :DESC.Install/service DirMaint
34 :INSTALL.YES :INSPPF.SERV2P DIRM :BUILD.YES :BLDPPF.SERV2P DIRM :P2PP
35 :PRODLEV.000-0000
36 :PRODID.5VMRAC30%RACF :SERVLEV.000-0000 :DESC.RACF Feature of z/VM, FL5
37 :INSPPF.SERV2P RACF :BUILD.YES :BLDPPF.SERV2P RACF :P2PPPF.SERV2P
38 :PRODID.5VMPTK30%PERFTK :SERVLEV.000-0000 :DESC.Performance Tool Kit :I
39 :INSPPF.SERV2P PERFTK :BUILD.NO :BLDPPF.SERV2P PERFTK :P2PPPF.SERV2P
40 :PRODID.5VMHCD20%VMHCD :SERVLEV.RSU-0701 :DESC.VMHCD for z/VM 5.2.0 :IN
41 :INSPPF.SERV2P VMHCD :BUILD.YES :BLDPPF.SERV2P VMHCD :P2PPPF.SERV2P
42 * * * End of File * * *

```

Figure 2-15 VM SYSSUF file

The next step is to force the building of the CP nucleus. Issue the following commands:

```

vmfsetup 5vmrac30 racf (link
vmfrepl rpiblcprn exec 5vmrac30 racf (nocopy $select
vmfsetup detach

```

The VMFREPL EXEC is used to support the local modification of replacement maintained parts. VMFREPL can be used to:

- ▶ Copy the highest level of a part
- ▶ Copy a specified part
- ▶ Update a Version Vector Table
- ▶ Update a Select Data file
- ▶ Display the highest levels of a part

RPIBLCPN EXEC is used to build the CPLOAD MODULE using the RACF files and the version vector tables for RACF. The \$SELECT operand adds an entry into the 5VMRAC30 \$SELECT file (Example 2-16) on RACFVMs apply disk (2A6), which defines to VMSES/E that there has been local service to the RPIBLCPN EXEC.

Example 2-16 5VMRAC30 \$SELECT file

```
5VMRAC30 $SELECT F1 V 80 Trunc=80 Size=2
====>
0 * * * Top of File * * *
1 :APPLYID.07/16/07 16:09:18
2 RPIBLCPN EXC EXC00000 BASE-FILETYPE
3 * * * End of File * * *
```

The SERVICE EXEC is used again, similar to when you enabled the RACF product. This time you use the BUILD operand to create the new CPLOAD MODULE, by issuing the following command:

```
service racf build
```

The new CP nucleus, with the RACF CP parts, is placed on the secondary parm disk (default disk address of CF2). For your information, a copy of the previous (or currently running) CPLOAD MODULE is still on the primary (CF1) and tertiary (CF3) parm disks as CPLOAD MODULE. It is also saved on the secondary parm disk as CPLOLD.

Complete the process by following these steps:

1. Shut down the currently running system.
2. IPL from EXTENT 2 of the 530RES volume.
3. Start the system with the NOAUTOLOG parameter.
4. Perform XAUTOLOG RACMAINT.
5. Run the PUT2PROD EXEC from the MAINT virtual machine.

This completes the installation and configuration of the RACF product for z/VM 5.3.0.

2.3 RACF management processes

In this section, we describe how to make DirMaint and RACF work together and show some basic setup in RACF to protect commonly used resources.

2.3.1 DirMaint changes to work with RACF

There were many enhancements done to DirMaint in z/VM 5.1.0 to eliminate the Interactive System Productivity Facility (ISPF) requirement for dual registration. This support was added through the service stream with VM63733 (PTF UV60921). This APAR provides DirMaint exits that allow the DIRMAINT virtual machine to execute the appropriate RACF commands to perform the following tasks:

- ▶ Add user
- ▶ Define MDISK
- ▶ Define VMRDR
- ▶ Define VMPOSIX
- ▶ Define SURROGAT
- ▶ Define VMBATCH

Note: This is one of the reasons that we recommend to use DirMaint with your RACF environment. Implementing the DirMaint exits is an easier task than customizing the ISPF product. In addition, the ISPF product is a priced product that is not licensed for IFL processor.

The code to exploit this process is shipped with the base system with z/VM 5.2.0 and 5.3.0. To implement this process you just need to update your CONFIGnn DATADVH file with the statements defined in the CONFIGRC SAMPDVH file (Figure 2-18 on page 46). There is *not* a copy of the CONFIGRC SAMPDVH file on the DIRMAINT minidisks. It is located on the 2C2 disk owned by the 5VMDIR30 virtual machine. The best method how to obtain a copy of this file is to use the VMFSETUP command (Figure 2-16) from MAINT and then copy the file to one of MAINT's disks.

```
vmfsetup servp2p dirm (link
VMFSET2760I VMFSETUP processing started for SERVP2P DIRM
VMFSET2204I Linking 5VMDIR30 2B2 as 2B2 with the link mode MR
VMFSET2204I Linking 5VMDIR30 2B1 as 2B1 with the link mode MR
VMFSET2204I Linking 5VMDIR30 2C4 as 2C4 with the link mode MR
VMFSET2204I Linking 5VMDIR30 2C2 as 2C2 with the link mode MR
VMFSET2204I Linking 5VMDIR30 2D2 as 2D2 with the link mode MR
----- 17 line(s) not displayed -----
VMFUTL2205I BUILD6      M      R/W  29D  DRM29D
VMFUTL2205I BUILD6U    N      R/W  502  DRM502
VMFUTL2205I BASE       O      R/W  2B2  DRM2B2
VMFUTL2205I BASE1      P      R/W  2B1  DRM2B1
VMFUTL2205I ----- A      R/W  191  MNT191
VMFUTL2205I ----- B      R/W  5E5  MNT5E5
VMFUTL2205I ----- C      R/W  2CC  MNT2CC
VMFUTL2205I ----- D      R/W  51D  MNT51D
VMFUTL2205I ----- S      R/O  190  MNT190
VMFUTL2205I ----- Y/S    R/O  19E  MNT19E
VMFSET2760I VMFSETUP processing completed successfully
Ready; T=0.23/0.25 08:17:50
```

Figure 2-16 VMFSETUP with the LINK operand

Figure 2-17 shows the output when you issue a **filelist** command for the CONFIGRC file on all disk.

```

MAINT      FILELIST A0  V 169  Trunc=169 Size=7 Line=1
Cmd  Filename Filetype Fm Format Lrecl      Records
CONFIG  SAMPDVH  F2 V          73        1460
CONFIGRC SAMPDVH  F2 V          73         89
CONFIG  DATADVH  L2 V          73        1460
CONFIG  SAMPDVH  O2 V          73        1460
CONFIGRC SAMPDVH  O2 V          73         89
CONFIG  $SAMPDVH P1 F        1024         91
CONFIGRC $SAMPDVH P1 F        1024          5

```

Figure 2-17 Filelist of CONFIGRC

Copy the CONFIGRC DATADVH file to your A disk. When you have finished, execute the **vmfsetup detach** command.

```

CONFIGRC SAMPDVH  F2  V 80  Trunc=80 Size=89 Line=40 Col=1 Alt=0
==>
40 POSIX_CHANGE_NOTIFICATION_EXIT=          DVHXPESM  EXEC
41 LOGONBY_CHANGE_NOTIFICATION_EXIT=        DVHXLB    EXEC
42 ----- 3 line(s) not displayed -----
45 USER_CHANGE_NOTIFICATION_EXIT=          DVHXUN     EXEC
46 DASD_OWNERSHIP_NOTIFICATION_EXIT=        DVHXdN     EXEC
47 PASSWORD_CHANGE_NOTIFICATION_EXIT=        DVHXPn     EXEC
48 ----- 5 line(s) not displayed -----
53 RACF_ADDUSER_DEFAULTS=                   UACC(NONE)
54 RACF_RDEFINE_VMMDISK_DEFAULTS=            UACC(NONE) AUDIT(FAILURES(READ))
55 RACF_RDEFINE_VMPOSIx_POSIXOPT.QUERYDB=    UACC(READ)
56 RACF_RDEFINE_VMPOSIx_POSIXOPT.SETIDS=      UACC(NONE)
57 RACF_RDEFINE_SURROGAT_DEFAULTS=            UACC(NONE) AUDIT(FAILURES(READ))
58 RACF_RDEFINE_VMBATCH_DEFAULTS=            UACC(NONE) AUDIT(FAILURES(READ))
59 RACF_RDEFINE_VMRDR_DEFAULTS=              UACC(NONE) AUDIT(FAILURES(READ))
60 RACF_RDEFINE_VMMDISK_DEFAULTS=            UACC(NONE) AUDIT(FAILURES(READ))
61 ----- 4 line(s) not displayed -----
65 RACF_VMBATCH_DEFAULT_MACHINES=            BATCH1 BATCH2
66 ----- 5 line(s) not displayed -----
71 TREAT_RAC_RC.4=                          0 | 4 | 30
72 /*-----*/
73 ----- 17 line(s) not displayed -----

```

Figure 2-18 CONFIGRC SAMPDVH

Use the DirMaint **send** command to retrieve a copy of the CONFIGnn DATADVH file. After receiving the file, copy the contents of the CONFIGRC SAMPDVH file to your CONFIGnn file and save the updated file. Use the DirMaint command file to return this file back to the DIRMANT virtual machine. Complete the work by executing the FIXDIRM EXEC (Figure A-11 on page 311). You need to give the DIRMANT and DATAMOVE virtual machine RACF attributes of special and operations (Example 2-17).

Example 2-17 RACF authorization for DIRMANT and DATAMOVE

rac alu dirmaint special operations

Ready; T=0.01/0.01 11:47:25

rac alu datamove special operations

Ready; T=0.01/0.01 11:47:33

After completing this work, when you add a new user or minidisk with DirMaint, it is added automatically to the RACF database, *with one exception*. When you add the minidisk or console statement to a virtual machine, those devices are defined to the RACF database with a UACC(NONE). This means that the virtual machine cannot use its own devices. You must issue the RACF **permit** command *manually* to grant the virtual machine an access of ALTER. An APAR for z/VM 5.3.0 has been opened that will correct this problem. The *APAR number is VM64275*. We feel that the easiest way to do this step is to write an exec to perform the task, as shown in Example 2-18.

Example 2-18 PERM EXEC

```

PERM      EXEC      S2  V 130  Trunc=130 Size=10 Line=0 Col=1 Alt=5
====>
0 * * * Top of File * * *
1 /*This exec will issue the RACF PERMIT command for mdisk resources*/
2 /*                                                                    */
3 /* Syntax:      perm   userid   disk-address                        */
4 /*                                                                    */
5
6 Arg   userid   disk
7
8 Aay "Issuing the RACF PERMIT for" userid " mdisk - "   disk
9
10 'rac PERMIT' userid '.'disk 'CLASS(VMMDISK) RESET ID('userid') AC(ALTER)'
11 * * * End of File * * *
```

2.3.2 RACF authorization concepts

Resources are defined to RACF/VM as profiles in the RACF database. There are profiles for all the resources that are defined to a RACF-enabled z/VM system (vmmdisk, vmrdr, vmlan, and so forth). These profiles can be *generic* (MAINT.19*, where the asterisk is one or more characters) or *discrete* (MAINT.CF1). See Figure 2-19.

```
Discrete Profiles

RDEFINE VMMDISK MAINT.CF1 OWNER(MAINT) UACC(NONE)
PERMIT MAINT.CF1 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
RDEFINE VMMDISK MAINT.CF2 OWNER(MAINT) UACC(NONE)
PERMIT MAINT.CF2 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
RDEFINE VMMDISK MAINT.CF3 OWNER(MAINT) UACC(NONE)

Generic Profiles

RDEFINE VMMDISK MAINT.CF% OWNER(MAINT) UACC(NONE)
PERMIT MAINT.CF% CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
RDEFINE VMMDISK MAINT.190 OWNER(MAINT) UACC(READ)
PERMIT MAINT.190 CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
RDEFINE VMMDISK MAINT.19E OWNER(MAINT) UACC(READ)
PERMIT MAINT.19E CLASS(VMMDISK) RESET ID(MAINT) AC(ALTER)
```

Figure 2-19 Discrete and generic profiles

The RPIDIRECT EXEC that was used to create the commands to define the RACF database during the installation and configuration process used discrete profiles. Your installation needs to determine whether you want to continue with this practice or use generic profiles. Both methods or a combination of methods will work. Make sure you issue SETROPTS GENERIC(VMMD) before you define the generic profiles.

2.3.3 RACF passwords and password phrases

It is important to understand that *passwords* and *password phrases* are two different things. *Passwords* can be upper case (default) or mixed case, if enabled with the RAC SETROPTS PASSWORD(MIXEDCASE) command, while *password phrases* are mixed case by default. Passwords are 1 to 8 characters, while password phrases can be 9 to 100 characters.

Password phrases are new with RACF FL 530. They are implemented by simply assigning them to users as desired. RACF has built-in rules for password phrases, for example 14-character minimum length. You can change the default rules if they do not apply to your installation by installing the ICHPWX11 exit, which calls the IRRPHREX EXEC. This allows you to enable 9 to 13 character phrases and to code additional quality rules in REXX™.

To install the exit, you need to make a local modification to a RACF loadlib and update the IRRPHREX SAMPLE file. The sample provided consists of both pieces, the ICHPWX11 exit and the IRRPHREX EXEC. A virtual machine can have a password, a password phrase, or both assigned to it, and CP will accept either one.

This section demonstrates how to implement both of these functions.

Password rules

Your organization's security guidelines can help you make the decisions for password rules that govern your system. These rules are implemented with the RACF SETROPTS commands. There are several parameters that control password requirements.

- ▶ Password change interval
- ▶ Inactive virtual machines intervals
- ▶ When virtual machines are revoked because of unsuccessful log in attempts
- ▶ Password history (password re-use)
- ▶ Password rules (maximum of 8 rules)
 - Password length
 - Password character requirements (vowels, numbers, and so forth)
 - Password in mixed case

For example, here is a list of password rules that you might want to consider:

- ▶ RAC SETROPTS PASSWORD(INTERVAL(90)) to define the password interval to 90 days
- ▶ RAC SETROPTS INACTIVE(30) to revoke a user ID if unused for over 30 days
- ▶ RAC SETROPTS PASSWORD(MIXEDCASE) to allow mixed-case passwords at your installation
- ▶ RAC SETROPTS PASSWORD(REVOKE(4)) to define the limit the user's count of successive incorrect passwords or password phrases if the password or password phrase is not correct before revoking the user
- ▶ RAC SETROPTS PASSWORD(HISTORY(6)) to define the number of previous passwords and password phrases (1-32) that RACF saves for each user to avoid duplication
- ▶ RAC SETROPTS PASSWORD(RULE1(LENGTH(6:8) ALPHA(1) ALPHANUM(3:8)) and RAC SETROPTS PASSWORD(RULE2(LENGTH(8)) to define the syntax of the new passwords for your installation

The RACF **setropts list** command displays the password settings (Example 2-19).

Example 2-19 RACF SETROPTS LIST

PASSWORD PROCESSING OPTIONS:

PASSWORD CHANGE INTERVAL IS 90 DAYS.

MIXED CASE PASSWORD SUPPORT IS IN EFFECT

6 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.

AFTER 4 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,

A USERID WILL BE REVOKED.

NO PASSWORD EXPIRATION WARNING MESSAGES WILL BE ISSUED.

INSTALLATION PASSWORD SYNTAX RULES:

*RULE 1 LENGTH(6:8) A*LLLLLL*

*RULE 2 LENGTH(8) ******

LEGEND:

A-ALPHA C-CONSONANT L-ALPHANUM N-NUMERIC V-VOWEL W-NOVOWEL *-ANYTHING

c-MIXED CONSONANT m-MIXED NUMERIC v-MIXED VOWEL \$-NATIONAL

Password phrases

Password phrases are implemented by default in RACF with default passphrase validation rules. If your installation needs to augment the rules, install the ICHPWX11 exit. This section provides information about how to implement the exit.

RACF has the new option of using the new password phrase exit (ICHPWX11) to augment RACF function when validating a new password phrase. This exit calls a REXX exec IRRPHREX SAMPLE on RACFVM's 305 disk. The exit point does nothing by default. The sample shipped by IBM consists of the exit proper, ICHPWX11, and a REXX exec named IRRPHREX (Example 2-20 on page 50). ICHPWX11 has to be installed as is described in the System Programmer's Guide.

The exit gains control when a new password phrase is processed and can examine the value specified for the password phrase and enforce installation rules in addition to the RACF rules. For example, while RACF does not allow the user ID to be part of the password phrase, the exit could perform more complex tests like disallow the company name, the names of months, and the current year in the password phrase.

The user of the new password phrase exit *augments* the RACF rules, but cannot override them. Be sure that the exit and the RACF rules do not contradict each other. For example, if the exit requires that the pass phrases contain all alphabetic characters, users will not be able to create new password phrases, because RACF requires at least two non-alphabetic characters. If you try to assign a phrase that conflicts the password rules, RACF will not accept the new phrase and displays the following message:

```
ICH21039I NEW PASS PHRASE REJECTED BY RACF RULES
```

The interval value specified on the PASSWORD command applies to both passwords and password phrases. It continues to be processed by the new password exit, ICHPWX01, and is not passed to the ICHPWX11 exit.

Example 2-20 IRRPHREX SAMPLE

```
IRRPHREX SAMPLE  Z1  F 80  Trunc=80 Size=562 Line=0 Col=1 Alt=4
====>
  0 * * * Top of File * * *
  1 ----- 118 line(s) not displayed -----
119 debug = 'off'
120 ----- 8 line(s) not displayed -----
128 Phr_minlen = 9
129 ----- 7 line(s) not displayed -----
136 Phr_maxlen = 100 /* Maximum password phrase length */
137 ----- 2 line(s) not displayed -----
139 /* Allowable characters. */
140 ----- 14 line(s) not displayed -----
154 numbers = '0123456789'
155 letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
156 special = '$@# '
157 Phr_allowed_chars = numbers||letters||special
158 ----- 31 line(s) not displayed -----
189 Phr_triviality = 'no'
190 ----- 22 line(s) not displayed -----
212 Phr_min_unique = 0
213 Phr_min_unique_norm = 'yes' /* @P1C*/
214 ----- 22 line(s) not displayed -----
```

The steps to implement password phrases for RACF are documented in *RACF Security Server System Programmer's Guide*, SC24-6149. The HLASM product is required to assemble the ICHPWX11 file.

These task are performed from the 5VMRAC30 virtual machine:

```
access 590 t
vmfsetup 5vmrac30 racf
copy ichpx11 assemble k = = e
vmfhlasm ichpx11 5vmrac30 racf ($select outmode e
rename ichpx11 txt00000 e = txtl0001 e
rename ichpx11 assemble e = asml0001 e
vmfsim logmod 5VMRAC30 vvtlcl e tdata :mod lcl0001 :part ichpx11 txt
vmfsim logmod 5VMRAC30 vvtlcl e tdata :mod lcl0001 :part ichpx11 asm
vmfbld ppf 5vmrac30 racf (serviced
```

Put the code into production (copy files created by VMFBLD to RACFVM's 305 disk).

Note: The process that is documented in the RACF Security Server System Programmer's Guide will not work as documented. When you link to RACFVM's 305 disk, you cannot get it in write mode, because RACFVM has the disk in write mode. If you force off the RACFVM, then you have no ESM and you cannot autolog RACMAINT after you have forced RACFVM. We describe here how you can put the code into production.

For this process, you have to give 5VMRAC30 the privilege class A or C so that it can execute the **set secuser** command. You can use your normal processes to change the privilege class and then place the directory online. You have to log off and then log on to the 5VMRAC30 virtual machine to pick up the directory change. Issue the **vmfsetup 5vmrac30 racf** command to re-establish your disk search order.

Perform the task as shown in Example 2-21 to gain write access to RACFVM's 305.

Example 2-21 Write access to RACFVM.s 305

```
set secuser racfvm 5vmrac30
HPCPFX6768I SECUSER of RACFVM initiated.
Ready; T=0.01/0.01 15:07:06
send cp racfvm det 305
Ready; T=0.01/0.01 15:07:14
RACFVM : DASD 0305 DETACHED
link racfvm 305 305 mr
RACFVM : (OPERATOR) ICH408I USER(5VMRAC30) GROUP(SYS1 ) NAME(#####
#####)
RACFVM : (OPERATOR) RACFVM.305 CL(VMMDISK )
RACFVM : (OPERATOR) INSUFFICIENT ACCESS AUTHORITY
RACFVM : (OPERATOR) ACCESS INTENT(CONTROL) ACCESS ALLOWED(NONE)
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO RACFVM.305
HCPLNM298E RACFVM 0305 not linked; request denied
Ready(00298); T=0.01/0.01 15:07:22
send cp racfvm link * 305 305 mr
RACFVM : DASD 0305 LINKED R/W
Ready; T=0.01/0.01 15:07:53
send racfvm acc 305 b/a
RACFVM : DMSACC724I 305 replaces B (305)
Ready; T=0.01/0.01 15:08:03
```

As shown, there is a security violation with **link** command. To solve it, use one of your systems RACF administrators and issue the **racf permit** command to allow 5VMRAC30 to have *control* access to RACFVM's 305 disk.

```
rac permit racfvm.305 class(vmmdisk) id(5vmrac30) ac(control)
```

Now you can complete the task of moving files to RACFVM's 305 disk as shown in Example 2-22.

Example 2-22 Moving files to RACFVM's 305 disk

```
send racfvm det 305
RACFVM : DASD 0305 DETACHED
RACFVM : CST
Ready; T=0.01/0.01 15:17:45
link racfvm 305 305 mr
Ready; T=0.01/0.01 15:17:55
acc 305 z
Ready; T=0.01/0.01 15:18:01

vmfcopy * * k = = z (prodid 5vmrac30%racf oldd replace
Ready; T=0.25/0.33 15:19:35
det 305
DASD 0305 DETACHED
Ready; T=0.01/0.01 15:19:46
send racfvm link * 305 305 mr
RACFVM : CST
Ready; T=0.01/0.01 15:19:57
send racfvm access 305 b/a
RACFVM : DMSACC724I 305 replaces B (305)
Ready; T=0.01/0.01 15:20:07
RACFVM : DMSACP723I B (305) R/O
RACFVM : CST
```

Then, send RACFVM **ipl 490** command that restarts RACF (Example 2-23). You cannot IPL CMS or 190 in RACFVM, or RACF will not start correctly.

Example 2-23 RACF IPLing 490

```
send cp racfvm ipl 490 clear parm autocr
Ready; T=0.01/0.01 15:20:51
RACFVM : RACFVM CMS XA Rel 14 03/19/2002
RACFVM : DMSACP723I B (305) R/O
RACFVM : HCPDTV040E Device 0591 does not exist
RACFVM : HCPDTV040E Device 0505 does not exist
RACFVM : HCPDTV040E Device 0590 does not exist
RACFVM : RACF is defined to the Z/VM system and the current product,
RACFVM : status is ENABLED
RACFVM : RACF
RACFVM : Feature for z/VM
RACFVM : Version 5.3.0
RACFVM :
RACFVM : Licensed Materials - Property of IBM
RACFVM : 5741-A05
RACFVM : (C) Copyright IBM CORP. 1981, 2007 All Rights Reserved.
RACFVM : DMSACC723I R (0200) R/W - OS
RACFVM : DMSACC723I Q (0300) R/W - OS
15:20:54 * WNG FROM RACFVM : RACF/VM SERVICES ARE NOW AVAILABLE.
RACFVM : * WNG FROM RACFVM : RACF/VM SERVICES ARE NOW AVAILABLE.
```

Note: This process was the only way that we could allow RACFVM's 305 disk to be updated without a system outage. If you can afford the outage, then you should shut down the system and IPL with the NOAUTOLOG parameter. Then, start RACMAINT as described previously.

This completes the instructions on how to install the exit. At this point, the sample exit does not perform any additional function compared to having no exit. You should now adjust the exit to reflect your installation requirements.

Password phrase syntax rules

Password phrase syntax rules include:

- ▶ Must not contain the user ID (as sequential uppercase or sequential lowercase characters).
- ▶ Must contain at least two alphabetic characters (A-Z, a-z).
- ▶ Must contain at least two non-alphabetic characters (numerics, punctuation, or special characters).
- ▶ Must not contain more than two consecutive characters that are identical.
- ▶ Must be enclosed in single quotation marks, with single quotation marks within the password phrase doubled. Note that the quotation marks must be removed from the password phrases when RACF prompts at logon.
- ▶ Must not contain forward slashes, nulls (X'00'), or leading or trailing blanks.

Only a RACF administrator can assign the initial phrase. When assigned, the user can modify the phrase, and is prompted to change it by default the first time it is used to log on.

Use the following command to disable the password function and enable a phrase:

```
rac alu jigu et nopassword phrase('better then today')
```

When the virtual machine JIGUET logs on to the system, it is prompted to change the password. When changing the password from the logon prompt, do not use the quotation marks (for example: *red white blue/red white blue*).

If the virtual machine wants to change the phrase while logged on to the system, issue:

```
rac phrase phrase('red white blue' 'howdy to everyone in vm land')
```

Although it looks like a mistake, the command is correct. It is **phrase** and it has an operand of **phrase**.

2.3.4 Adding virtual machines and resources to the system and the RACF database

This section describes how to add virtual machine and resources to the system and the RACF database.

Adding virtual machines with DirMaint

As discussed earlier in this chapter, we have chosen DirMaint as the tool to add virtual machines to the system. We chose this method because of the DirMaint exits added in z/VM 5.1.0. These exits allow the DIRMAINT virtual machine to issue the appropriate RACF commands to add users and define resources without the need for ISPF (Figure 2-20).

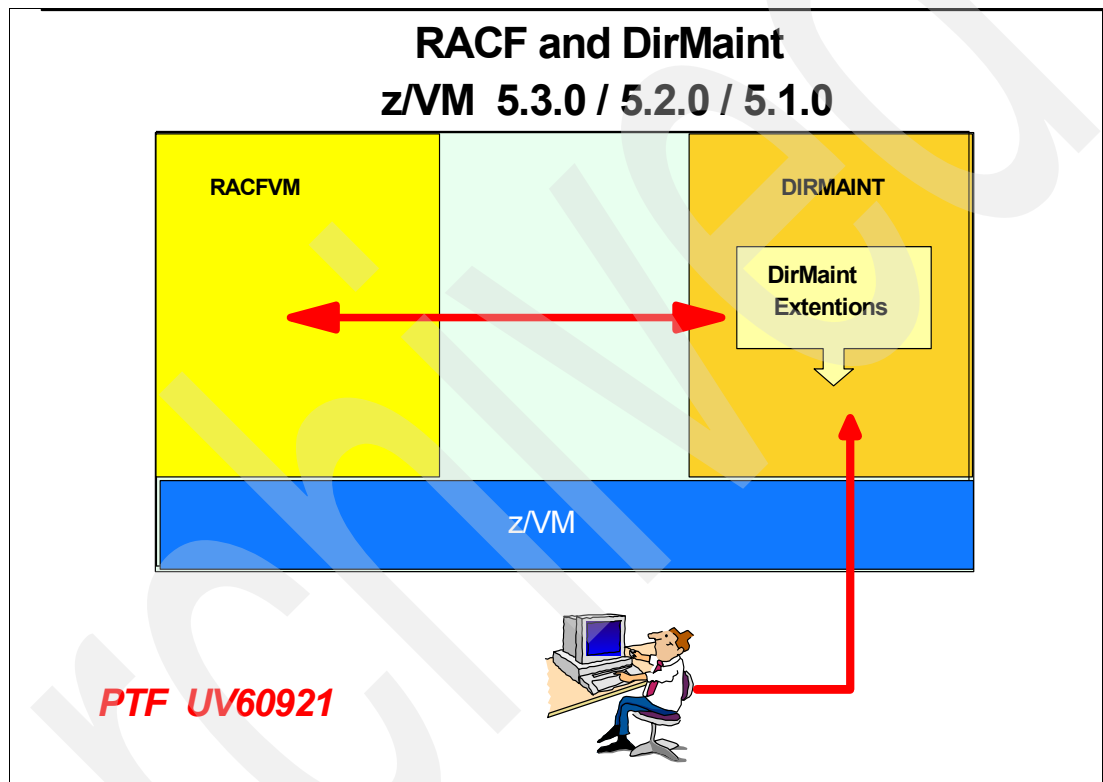


Figure 2-20 DirMaint extensions

To use the DirMaint extensions, you must authorized DIRMAINT and DATAMOVE virtual machines with RACF attributions *special* and *operations*.

When you need to add a new virtual machine to our system, first make sure the virtual machine has not been defined previously (Example 2-24).

Example 2-24 Verification of a virtual machine

rac lu userbob

ICH30001I *UNABLE TO LOCATE USER* ENTRY USERBOB

Ready(00004); T=0.01/0.01 08:43:53

dirm for userbob get nolog

DVHXMT1191I Your GET request has been sent for processing.

Ready; T=0.03/0.03 08:44:09

DVHREQ2288I Your GET request for USERBOB at * has been accepted.

DVHBDG6209E *Specified user USERBOB does not exist*, request GET failed.

DVHGET3212E Unexpected RC= 6209, from: EXEC DVHBBDDGT USERBOB DIRECT A0

DVHREQ2289E Your GET request for USERBOB at * has failed; with RC =

DVHREQ2289E 3212.

To create a new virtual machine, create a file on the A disk of a DirMaint administrator, which contains new virtual machine definition (see Figure 2-21).

```
USERBOB DIRECT  A0  F 80  Trunc=72 Size=5 Line=0
====>
0 * * * Top of File * * *
1 USER USERBOB TEXAS  32m  100m  BCDG
2   INCLUDE IBMDFLT
3   IPL CMS PARM AUTOGR
4   MACHINE XA
5   LINK TCPMAINT 0592 0592 RR
6 * * * End of File * * *
```

Figure 2-21 USERBOB DIRECT

Issue the command **dirm add**. It displays a panel similar to the one shown in Figure 2-22.

```
-----DirMaint ADD-----

Add an entry to the directory for a new USERID or Profile.

Fill in the USERID or PROFILE being added:
    ==> userbob

Optionally fill in the following when using a prototype:
    LIKE ==>          (file name of prototype)
    PW   ==>          (password for new user)
    VPW  ==>          (password again for verification)
    ACCT ==>          (account value for new user - optional)

Notes:
    - If a value is given for any one of PW, VPW, or ACCT,
      then a value is required for LIKE.
    - If a value is given for either PW or VPW,
      then a value is required for both of them.

741-A05 (c) Copyright IBM Corporation 1979, 2007.
    1= Help      2= Prefix Operands    3= Quit      5=Submit
==>
```

Figure 2-22 DIRMAINT ADD

After filling in the name of the virtual machine, press PF5. You receive the messages shown in Example 2-25.

Example 2-25 DirMaint Output

```
PUN FILE 0013 SENT TO   DIRMAINT RDR AS  0037 RECS 0013 CPY  001 0 NOHOLD NOKEEP
DVHXM1191I Your ADD request has been sent for processing.
Ready; T=0.07/0.08 08:51:11
DVHREQ2288I Your ADD request for USERBOB at * has been accepted.
DVHBIU3450I The source for directory entry USERBOB has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHDRC3451I The next ONLINE will take place via delta object directory.
DVHBIU3428I Changes made to directory entry USERBOB have been placed
DVHBIU3428I online.
DVHBIU3450I The source for directory entry USERBOB has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHDRC3451I The next ONLINE will take place via delta object directory.
DVHBIU3428I Changes made to directory entry USERBOB have been placed
DVHBIU3428I online.
DVHREQ2289I Your ADD request for USERBOB at * has completed; with RC
DVHREQ2289I = 0.
```

If you issue the **rac lu** command and the **dirm for userbob get no lock**, you find it has been defined.

When you add a minidisk to this user, the minidisk address is added to the RACF database as well. When the resource is added to the database, the process does *not* currently give the user authority (**rac permit**) to use their own device. This problem is corrected with *APAR VM62475* for z/VM 5.3.0. So, the final step is to issue the **rac permit** command for the resource:

```
rac permit userbob.191 class(vmmdisk) reset id(userbob) ac(alter)
```

Adding virtual machines without DirMaint

If you decide not to use the DirMaint product on your system, then there is an automated process that also updates the RACF database. This method is not as automated as the **dirm add**, but it works well.

You use the same processes that you use to build the initial RACF database. The processes are the RPIDIRECT and RPIBLDDS execs. These processes have to be complete from the MAINT virtual machine, because MAINT is the owner of the USER DIRECT file found on the 2CC disk.

Add the new user to the USER DIRECT FILE (Figure 2-23).

```
USER      DIRECT  C1  F 80  Trunc=80 Size=2215 Line=2203 Col=1
====>
2203 *
2204 USER USERBOB 18FUMDIM 32M 100M BCDG
2205     INCLUDE IBMDFLT
2206     IPL CMS PARM AUTOGR
2207     MACHINE XA
2208     LINK TCPMAINT 0592 0592 RR
2209 *
2210 USER RAICHER MICHAEL 32M 100M BCDG
2211     INCLUDE IBMDFLT
2212     IPL CMS PARM AUTOGR
2213     MACHINE XA
2214     LINK TCPMAINT 0592 0592 RR
2215 MDISK 191 3390 2220 10 530W99 MR ALL GO4IT WHYNOT
2216 * * * End of File * * *
```

Figure 2-23 USER DIRECT on the 2CC disk

Then, put the directory online with the **directxa** command and copy the directory entry for the new user to the *userid direct A* file (Figure 2-24).

```
RAICHER  DIRECT  A1  F 80  Trunc=72 Size=6 Line=0 Col=1 Alt=0
==>
0 * * * Top of File * * *
1 USER RAICHER MICHAEL  32M 100M BCDG
2   INCLUDE IBMDFLT
3   IPL CMS PARM AUTO CR
4   MACHINE XA
5   LINK TCPMAINT 0592 0592 RR
6   MDISK 191 3390 2220 10 530W99 MR  ALL  GO4IT  WHYNOT
7 * * * End of File * * *
```

Figure 2-24 RAICHER DIRECT on the 191 disk

Now, the new virtual machine is added to the system directory. However, if you try to log on to the virtual machine, it fails (as shown in Example 2-26), because the virtual machine is not defined in the RACF database and because you are no longer deferring the request to CP.

Example 2-26 Logon RAICHER

```
logon raicher
HCPLGA053E RAICHER not in CP directory
```

Enter one of the following commands:

LOGON userid	(Example: LOGON VMUSER1)
DIAL userid	(Example: DIAL VMUSER2)
MSG userid message	(Example: MSG VMUSER2 GOOD MORNING)
LOGOFF	
UNDIAL	

You need to update the RACF database with information about this virtual machine. To do so, link and access the 505 disk that is owned by 5VMRAC30. You need this disk because that is where the RPIDIRECT and RPIBLDDS execs are located.

Then, run the RPIDIRECT EXEC against the RAICHER DIRECT file (Example 2-27) to generate a new RPIDIRECT SYSUT1 file (Figure 2-25).

Example 2-27 Running RPIDIRECT

```
rpirect raicher direct
RAICHER DIRECT Filemode defaulted to "*".
Output defaulted to "A" disk.
  Default group ID = SYS1.
  Would you like to change this default?
  Enter Y/N
n
  Default group ID = SYS1.
The file, RPIDIRECT CNTRL, was found and will be used to override
the directory specifications.
*****
                DEFINITION pass begins.....
*****
USER RAICHER XXXXXXX  32M 100M BCDG
  INCLUDE IBMDFLT

Profile IBMDFLT does not exist in Directory.

MDISK 191 3390 2220 10 530W99 MR  ALL  GO4IT  WHYNOT
*****
DEFINITION pass complete - PERMIT command generation begins...
*****
      processing LINK TCPMAINT 0592 0592 RR for user RAICHER
*****
***** Scan ended *****
*****
***** 5 Directory records processed *****
*****
***** RPIDIRECT SYSUT1 CREATED *****
```

```
RPIDIRECT SYSUT1  A1  V 80  Trunc=80 Size=16 Line=0 Col=1 Alt=0
==>
0 * * * Top of File * * *
1 ***** RAICHER
2 *
3 ADDUSER RAICHER DFLTGRP(SYS1) UACC(NONE) PASSWORD(MICHAEL)
4 RDEFINE VMBATCH RAICHER OWNER(RAICHER) UACC(NONE)
5 PERMIT RAICHER CLASS(VMBATCH) ACCESS(ALTER) RESET
6 RDEFINE VMRDR RAICHER UACC(NONE) OWNER(RAICHER)
7 PERMIT RAICHER CLASS(VMRDR) ID(RAICHER) ACCESS(ALTER) RESET
8 RDEFINE VMMDISK RAICHER.191 OWNER(RAICHER) UACC(READ)
9 PERMIT RAICHER.191 CLASS(VMMDISK) RESET ID(RAICHER) AC(ALTER)
10 *
11 *****
12 *
13 *                      PERMIT DIRECTORY LINKS
14 *
15 *****
16 *
17 * * * End of File * * *
```

Figure 2-25 New RPIDIRECT SYSUT1 file

Now, run the RPIBLDDS exec using the new RPIDIRECT SYSUT1 file and update the RACF database with the commands shown in Figure 2-26.

```

rpihldds rpidirect
Processing batch file RPIDIRECT SYSUT1 using "RAC" command interface
*
=> ADDUSER RAICHER DFLTGRP(SYS1) UACC(NONE) PASSWORD(NOLOG)
=> RDEFINE VMBATCH RAICHER OWNER(RAICHER) UACC(NONE)
=> PERMIT RAICHER CLASS(VMBATCH) ACCESS(ALTER) RESET
=> RDEFINE VMRDR RAICHER UACC(NONE) OWNER(RAICHER)
=> PERMIT RAICHER CLASS(VMRDR) ID(RAICHER) ACCESS(ALTER) RESET
=> RDEFINE VMMDISK RAICHER.191 OWNER(RAICHER) UACC(NONE)
=> PERMIT RAICHER.191 CLASS(VMMDISK) RESET ID(RAICHER) AC(ALTER)

Ready; T=0.02/0.03 11:11:45

```

Figure 2-26 Running RPIBLDDS

You can now issue the **rac lu** command to check how virtual machine is defined (Example 2-28).

Example 2-28 RAC LU RAICHER

```

rac lu raicher
USER=RAICHER NAME=UNKNOWN OWNER=IBMUSER CREATED=07.195
DEFAULT-GROUP=SYS1 PASSDATE=00.000 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=UNKNOWN
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
-----
ANYDAY ANYTIME
GROUP=SYS1 AUTH=USE CONNECT-OWNER=IBMUSER CONNECT-DATE=07.195
CONNECTS= 00 UACC=UPDATE LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
GROUP=SYSTEM AUTH=USE CONNECT-OWNER=IBMUSER CONNECT-DATE=07.195
CONNECTS= 00 UACC=NONE LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED

```

As shown in the output, the virtual machine RAICHER was added to the RACF database.

2.3.5 Implementing LOGONBY with RACF

The CP LOGONBY directory statement designates up to eight virtual machines to another virtual machine. This function was originally a DirMaint implementation and was added to VM a number of releases ago (according to VM historians it was VM/ESA Version 2 Release 1). RACF has support for this function with the SURROGAT class facility but is not limited to the maximum of eight surrogate virtual machines. The RACF LOGON BY function allows authorized virtual machines to log on to a shared virtual machine using their own password. This is extremely handy when you have several virtual machines that need to share the MAINT virtual machine, but only one person can be logged on at a given time.

To fully understand this function, you need to become familiar with the following terms:

- ▶ *Shared user*: User ID that has the capability of being logged onto by a different user.
- ▶ *Surrogate user*: Person logging on to the shared user ID.
- ▶ *Direct logon*: A traditional logon, in which you log on to your own user ID.
- ▶ *Shared logon*: A logon in which a surrogate user uses the BY option of the LOGON command to log on to a different user ID. The surrogate user operates with the RACF authority of the shared user ID.

To implement the RACF LOGON BY facility, perform the following tasks:

1. Use the **setropts** command to activate the CLASSACT(SURROGAT) class.

```
rac setropts class(surrogat)
```

2. Verify SURROGAT class is active.

```
rac setr list
```

```
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM)
```

```
STATISTICS = NONE
```

```
ACTIVE CLASSES = USER GROUP VMMDISK VMRDR VMLAN SURROGAT
```

3. Define profiles of the form LOGONBY.shared_userid in the SURROGAT class for each user ID that is to be shared.
4. Permit specific users to the appropriate SURROGAT profiles.
5. List the information with the RLIST command.

Logonby processing

We suggest that you create a sample file from which to copy to implement the LOGON BY function (see Example 2-29) where you change *shrduser* and *surrogat-id1*.

Example 2-29 RPIDIRCT SURROGAT

```
RPIDIRCT SURROGAT A1 F 80 Trunc=80 Size=5 Line=0 Col=1 Alt=0
====>
0 * * * Top of File * * *
1 RDEFINE SURROGAT LOGONBY.shrdusr UACC(NONE) AUDIT(ALL)
2 PERMIT LOGONBY.shrdusr CL(SURROGAT) RESET(ALL)
3 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surrogat-id1)
4 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surrogat-id2)
5 RL SURROGAT LOGONBY.shrdusr AUTH
6 * * * End of File * * *
```

When you need to add surrogate users to the RACF database, copy this file to RPIDIRECT SYSUT1 on your A disk and then modify that file as shown in Example 2-30.

Example 2-30 RPIDIRECT SYSUT1 before the changes

```
RPIDIRECT SYSUT1  A1  F 80  Trunc=80 Size=5 Line=0 Col=1 Alt=0
====>  ch /shrdusr/MAINT/* *
        0 * * * Top of File * * *
        1 RDEFINE SURROGAT LOGONBY.shrdusr UACC(NONE) AUDIT(ALL)
        2 2 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(ALTER) ID(shrdusr) RESET(ALL)
        3 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surg-id1)
        4 PERMIT LOGONBY.shrdusr CL(SURROGAT) ACC(READ) ID(surg-id2)
        5 RL SURROGAT LOGONBY.shrdusr AUTH
        6 * * * End of File * * *
```

If you want to add SURROGAT support for the MAINT virtual machine, tailor the file to look like Example 2-31.

Example 2-31 RPIDIRECT SYSUT1 after the changes

```
RPIDIRECT SYSUT1  A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0
====>
        0 * * * Top of File * * *
        1 ALTUSER MAINT NOPASSWORD NOPHRASE
        2 RDEFINE SURROGAT LOGONBY.MAINT UACC(NONE) AUDIT(ALL)
        3 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(ALTER) ID(MAINT) RESET(ALL)
        4 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(DETRO)
        5 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(VELLOSO)
        6 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(JIGUET)
        7 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(GNIRSS)
        8 PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(RAICHER)
        9 RL SURROGAT LOGONBY.MAINT AUTH
       10 * * * End of File * * *
```

The ALTUSER MAINT NOPASSWORD NOPHRASE is a good way to protect the MAINT user ID from being revoked because of too many attempts with the wrong password. Prior to z/VM 5.3, MAINT could be revoked by logging on directly with too many incorrect passwords. With the 5.3 release, if you set MAINT NOPASSWORD, the ID is protected from this type of attack. In our example, we show the permit for each user, although defining the permission by group (for example, ITSGRP) would be better practice. Use the RPIBLDDS EXEC again to execute these definitions (Example 2-32).

Example 2-32 Output of RPIBLDDS

```
rpibldds rpidirect sysut1
Processing batch file RPIDIRECT SYSUT1 using "RAC" command interface
=> RDEFINE SURROGAT LOGONBY.MAINT UACC(NONE) AUDIT(ALL)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(ALTER) ID(MAINT) RESET(ALL)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(DETRO)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(VELLOSO)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(JIGUET)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(GNIRSS)
=> PERMIT LOGONBY.MAINT CL(SURROGAT) ACC(READ) ID(RAICHER)
=> RL SURROGAT LOGONBY.MAINT AUTH
CLASS      NAME
-----
SURROGAT   LOGONBY.MAINT
```

LEVEL	OWNER	UNIVERSAL ACCESS	YOUR ACCESS	WARNING
00	DETRO	NONE	READ	NO
INSTALLATION DATA				

NONE				
APPLICATION DATA				
USER	ACCESS	ACCESS COUNT		
----	-----	-----		
MAINT	ALTER	000000		
DETRO	READ	000000		
VELLOSO	READ	000000		
JIGUET	READ	000000		
GNIRSS	READ	000000		
RAICHER	READ	000000		

To use the logon BY function you log on with BY keyword as shown in Figure 2-27.

```

z/VM ONLINE

      /  VV      VVV MM      MM
     /  VV      VVV MMM     MMM
    /  VV      VVV MMMM    MMMM
   /  VV      VVV MM MM MM MM
  /  VV      VVV MM  MMM  MM
 /  VVVVVV   MM  M  MM
/  VVV      MM      MM
ZZZZZZ /      V      MM      MM

      built on IBM Virtualization Technology

z/VM 5.3.0

Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID  ==>
PASSWORD ==>

COMMAND ==>  logon maint by detro

```

Figure 2-27 Log on with the BY option

When prompted for the password, the password for the virtual machine DETRO is supplied (Example 2-33), although the virtual machine MAINT is logged on.

Example 2-33 Logon complete

```
logon maint by detro
```

Enter your password,

or

To change your password, enter: ccc/nnn/nnn

where ccc = current password, and nnn = new password

```
ICH70001I MAINT      LAST ACCESS AT 16:43:07 ON TUESDAY, JULY 17, 2007
```

```
HCPLNM102E DASD 0123 forced R/O; R/W by DIRMAINT
```

```
HCPLNM108E MAINT 0126 not linked; volid LN5PG2 not mounted
```

```
z/VM Version 5 Release 3.0, Service Level 0701 (64-bit),
```

```
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0007 RDR, NO PRT, NO PUN
LOGON AT 11:11:44 EDT WEDNESDAY 07/18/07
z/VM V5.3.0 2007-06-14 11:51
Ready; T=0.01/0.01 11:11:45
```

Review the *RACF Security Server System Programmer's Guide*, SC24-6149 or the *RACF Security Server Administrator's Guide*, SC24-6142 for additional information about this topic.

2.3.6 Managing VSWITCH and Guest LANS

Guest LAN and z/VM Virtual Switch (VSWITCH) were introduced to z/VM in version 4. They are not to be confused with IEEE Virtual LANS, although a VSWITCH can be VLAN aware. (This is a topic discussed in *Linux on IBM eServer zSeries and S/390: VSWITCH and VLAN Features of z/VM 4.4*, REDP-3719.)

z/VM Guest LAN is a z/VM networking function that was added in z/VM Version 4 Release 1 and that was further enhanced in Version 4 Release 2.

VSWITCH is a z/VM networking function announced by IBM for z/VM Version 4 Release 4. This new feature is designed to improve the interaction between guests running under z/VM and the physical network connected to the IBM System z processor.

IBM also announced IEEE 802.1Q VLAN support for z/VM Virtual Switch, z/VM QDIO Guest LAN, and z/VM HiperSockets Guest LAN, allowing z/VM guests to create and participate in virtual LAN configurations.

A VSWITCH is always owned by the SYSTEM (another name for the system control program) and is always restricted by CP for coupling by a virtual NIC. Guest LANs can be owned by a virtual machine or the SYSTEM. Guest LANs can be restricted or non-restricted. From a security point of view the Guest LANs should always be restricted to keep virtual machines from using them without explicit authorization.

When Guest LANs are restricted, RACF takes care about authorization. VMLAN is one of the classes that can be activated with the SETROPTS command,

You can activate the class with the following command:

```
rac setropts class(vmlan)
```

The **rac setropts list** command then displays the following:

```
ACTIVE CLASSES = DATASET USER GROUP VMMDISK VMRDR VMLAN
```

The **query lan detail** command displays the status of each guest LAN and VSWITCH (Example 2-34). You can also determine whether they are restricted.

Example 2-34 The query lan detail command

```
q lan detail
VSWITCH SYSTEM VSWITCH1 Type: VSWITCH Connected: 0    Maxconn: INFINITE
  PERSISTENT RESTRICTED    NONROUTER                Accounting: OFF
  VLAN Aware  Default VLAN: 0001    Default Porttype: Access  GVRP: Disabled
                Native VLAN: 0001
  MAC address: 02-00-00-00-00-01
  State: Defined
  IPTimeout: 5                QueueStorage: 8
  RDEV: 2E24                  Controller: NONE      Error: No RDEV
VSWITCH SYSTEM VSWITCH2 Type: VSWITCH Connected: 0    Maxconn: INFINITE
  PERSISTENT RESTRICTED    NONROUTER                Accounting: OFF
  VLAN Unaware
  MAC address: 02-00-00-00-00-02
  State: Defined
  IPTimeout: 5                QueueStorage: 8
Ready; T=0.01/0.01 11:45:20
```

If you have a virtual machine that has defined a virtual network interface card (NIC) and you issued the CP COUPLE command to connect your NIC to the virtual lan segment, it fails unless you have been authorized (granted permission). See Example 2-35.

Example 2-35 Define and couple NIC to a restricted LAN

```
def nic 900 qdio
NIC 0900 is created; devices 0900-0902 defined
Ready; T=0.01/0.01 11:50:13
couple 900 system vswitch1
RPIMGRO31E RESOURCE SYSTEM.VSWITCH1 SPECIFIED BY COUPLE COMMAND NOT FOUND
HCPCPL6011E You are not authorized to COUPLE to SYSTEM VSWITCH1
Ready(06011); T=0.01/0.01 11:50:26
```

If RACF is not managing the control of a guest LAN and VSWITCH, then a virtual machine that has B privilege class uses the following command to allow the couple to happen:

```
CP SET VSWITCH vswitchname GRANT userid
```

If you decide to enable the RACF class VMLAN with the RAC SETROPTS command:

- ▶ The resource must be defined to the RACF database.
- ▶ Virtual machines must have RAC PERMIT commands issued to allow the coupling to happen.

Figure 2-28 shows the commands to define and authorize virtual machines to VSWITCH1.

```
RAC RDEFINE VMLAN SYSTEM.vswitch1 UACC(NONE)
Ready; T=0.01/0.01 11:56:00

RAC RDEFINE VMLAN SYSTEM.vswitch2 UACC(NONE)
Ready; T=0.01/0.01 11:56:05

rac search class(vmlan)
SYSTEM.VSWITCH1
SYSTEM.VSWITCH2
Ready; T=0.01/0.01 11:56:22

RAC PERMIT SYSTEM.vswitch1 CLASS(VMLAN) RESET(ALL)
RAC PERMIT SYSTEM.vswitch1 CLASS(VMLAN) ID(maint) ACCESS(UPDATE)
Ready; T=0.01/0.01 11:57:49
RAC PERMIT SYSTEM.vswitch1 CLASS(VMLAN) ID(1nxsu1) ACCESS(UPDATE)
Ready; T=0.01/0.01 11:58:38
RAC PERMIT SYSTEM.vswitch1 CLASS(VMLAN) ID(1nxsu2) ACCESS(UPDATE)
Ready; T=0.01/0.01 11:58:42
```

Figure 2-28 RACF Authorization of VSWITCH1

In this example, we issue the PERMIT with the RESET option before allowing the permit. Because z/VM does not have the SETROPTS NOADDCREATOR, the user defining the profile is put on the ACL with ALTER. This might not be within your installation policy.

If MAINT issues the couple command now, it will be successful (Example 2-36).

Example 2-36 Coupled to VSWITCH1

```
couple 900 system vswitch1
NIC 0900 is connected to VSWITCH SYSTEM VSWITCH1
Ready; T=0.01/0.01 12:04:23
query vswitch vswitch1 detail
VSWITCH SYSTEM VSWITCH1 Type: VSWITCH Connected: 1 Maxconn: INFINITE
  PERSISTENT RESTRICTED NONROUTER Accounting: OFF
  VLAN Aware Default VLAN: 0001 Default Porttype: Access GVRP: Disabled
    Native VLAN: 0001
  MAC address: 02-00-00-00-00-01
  State: Defined
  ITimeout: 5 QueueStorage: 8
  RDEV: 2E24 Controller: NONE Error: No RDEV
Adapter Connections:
Adapter Owner: MAINT NIC: 0900 Name: UNASSIGNED
Porttype: Access
Ready; T=0.01/0.01 12:04:40
```

There is one additional topic that we need to discuss about the management of the RACF class VMLAN. If the VSWITCH is connected to a real OSA adapter that is connected to a network switch enabled for VLAN support (in network terms this is a connection to a TRUNK port), you must specify the VLAN number when issuing the **rac permit** command. For example, if the virtual machine LNXUS1 had IP addresses in VLANs 53 and 22, and LNXUS2 had an IP address that was in VLAN 22, the commands shown in Example 2-37 are issued.

Example 2-37 RACF Permit with VLAN support

```
RAC PERMIT SYSTEM.vswitch1.53 CLASS(VMLAN) ID(lnxsu1) ACCESS(UPDATE)
Ready; T=0.01/0.01 11:58:38
RAC PERMIT SYSTEM.vswitch1.22 CLASS(VMLAN) ID(lnxsu1 lnxu2) ACCESS(UPDATE)
Ready; T=0.01/0.01 11:58:42
```

You can obtain information about the RACF managed VMLANs with the **rac rlist** command. See Example 2-38.

Example 2-38 RAC RLIST command

```
rac r1 vmlan system.vswitch1 auth
CLASS      NAME
-----
VLAN       SYSTEM.VSWITCH1
LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
  00    MAINT              NONE              UPDATE      NO
-----
27 line(s) not displayed
-----
USER      ACCESS  ACCESS COUNT
-----
MAINT     UPDATE   000000
LNXSU1    UPDATE   000000
LNXSU2    UPDATE   000000

      ID      ACCESS  ACCESS COUNT  CLASS      ENTITY  NAME
-----
NO ENTRIES IN CONDITIONAL ACCESS LIST
```

For more information about this subject, see the *RACF Security Server Administrator's Guide*, SC24-6142.

2.3.7 Managing RSCS nodes

When the RACF CLASS VMNODE is activated on your z/VM system, then you must provide authorization to allow virtual machines to send files to other RSCS nodes. This should only be implemented if you are running the Remote Spooling Communications Subsystem (RSCS) product.

To complete this process you must perform these steps:

1. Use the RAC SETROPTS command to activate the class VMNODE.
2. Use the RAC RDEFINE command to define each of the target RSCS nodes, UACC(NONE).
3. Use the RAC PERMIT command to authorize virtual machines to the RSCS node.

If you follow these steps, you see there is a problem with RSCS when you send a file that causes the RSCS virtual machine to abend. The CP command that is causing the problem is the CP TAG command. This command is controlled by RACF by default and only comes into play when you are using RSCS (Figure 2-29).

```
rac setevent list

----- 9 line(s) not displayed -----
CONTROLLABLE VM EVENTS

VM EVENT          STATUS    VM EVENT          STA
-----          -
COUPLE.G          CONTROL   FOR.C             CONTROL
FOR.G             CONTROL   LINK              CONTROL
STORE.C           CONTROL   TAG               CONTROL
TRANSFER.D        CONTROL   TRANSFER.G        CONTROL
TRSOURCE          CONTROL   DIAG088           CONTROL
DIAG0A0           CONTROL   DIAG0D4           CONTROL
DIAG0E4           CONTROL   DIAG280           CONTROL
DIAG290           CONTROL   APPCPWVL          CONTROL
MDISK             CONTROL   RSTDSEG           CONTROL
```

Figure 2-29 RACF SETEVENT command

The solution is to create a RACF VMXEVENT class profile to manage the commands that are required by the RSCS virtual machine (Figure 2-30). You also need to set the SECLABEL for the RSCS virtual machine to SYSNONE if using SECLABELs.

```
rac rdefine vmxevent usersel.rscs
Ready; T=0.01/0.01 08:55:52

rac ralter vmxevent usersel.rscs addmem(tag/noctl transfer.g/noctl
transfer.d/noctl)
Ready; T=0.01/0.01 08:56:32

rac setevent refresh usersel.rscs
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: COUPLE
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: FOR.C
----- 10 line(s) not displayed -----
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: APPCPWVL
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: MDISK
RPISET113W TURNING CONTROL ON AUTOMATICALLY FOR: RSTDSEG
RPISET126I SETEVENT COMPLETED SUCCESSFULLY.
Ready; T=0.01/0.01 08:56:55

rac setropts class(vmxevent)
Ready; T=0.01/0.01 08:57:23

rac alu rscs seclabel(sysnone)
Ready; T=0.01/0.01 08:57:23
```

Figure 2-30 Defining RACF VMXEVENT Class

After completing this process you can perform the first step, enabling the class VMNODE, as shown in Figure 2-31.

```
rac setr list
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM)
STATISTICS = NONE
ACTIVE CLASSES = DATASET USER GROUP VMMDISK VMRDR VMLAN SURROGAT
GENERIC PROFILE CLASSES = NONE
GENERIC COMMAND CLASSES = NONE
GENLIST CLASSES = NONE
GLOBAL CHECKING CLASSES = NONE
RACLIST CLASSES = NONE
AUTOMATIC DATASET PROTECTION IS IN EFFECT
ENHANCED GENERIC NAMING IS NOT IN EFFECT
REAL DATA SET NAMES OPTION IS INACTIVE

rac setropts class(vmnode)
Ready 0.01/0.01 08:59:23

rac setr list
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM)
STATISTICS = NONE
ACTIVE CLASSES = USER GROUP VMMDISK VMNODE VMRDR VMLAN SURROGAT
GENERIC PROFILE CLASSES = NONE
GENERIC COMMAND CLASSES = NONE
GENLIST CLASSES = NONE
```

Figure 2-31 RACF VMNODE class activation

After activating the RACF class VMNODE, you must define the target RSCS nodes to which you will be sending files. You have to decide how you want to manage outbound traffic to these RSCS nodes. You can do it with a global setting (UACC-update), or you can use a very strict setting of (UACC-none) as shown in Figure 2-32. The setting of this parameter will have to meet your security policies for your environment.

```
rac rdefine vmnode vmlinux5 uacc(none)
Ready; T=0.01/0.01 07:02:42

rac permit vmlinux5 class(vmnode) id(maint) uacc(update)
Ready; T=0.01/0.01 07:03:02

rac permit vmlinux5 class(vmnode) id(rscs) ac(update)
Ready; T=0.01/0.01 07:03:13

rac permit vmlinux5 class(vmnode) id(sendrscs) ac(update)
Ready; T=0.01/0.01 07:03:13

sf perm exec a maint at vmlinux5
File PERM EXEC A2 sent to MAINT at VMLINUX5 on 07/19/07 07:02:57
Ready; T=0.01/0.01 07:03:57

sf perm exec a maint at wtscvmxa
RPIMGR031E RESOURCE WTSCVMXA SPECIFIED BY TAG COMMAND NOT FOUND
HCPCST003E Invalid option - WTSCVMXA
Ready(00003); T=0.01/0.01 07:04:55
```

Figure 2-32 Defining VMNODE

When issuing the permits, you can issue the permit to either a virtual machine name defined to RACF or to a RACF group name. In Figure 2-32, SENDRSCS is a group name and any member of that group would be allowed to send files to the RSCS node for which the permit was issued.

You can see an example of non-authorized access in Example 2-39.

Example 2-39 Sendfile from a non-authorized virtual machine

```
id
GUMBY    AT 2NDLEVEL VIA RSCS      07/19/07 09:57:43 EDT    THURSDAY
Ready; T=0.01/0.01 09:57:43

sf profile exec a to maint at vmlinux5
RPIMGR032E YOU ARE NOT AUTHORIZED TO SPOOL TO RSCS
HCPSPL007E Invalid userid - RSCS
Ready(00007); T=0.01/0.01 09:56:29
```

From a virtual machine with RACF administrator attributes, you can:

- ▶ Create a new group
- ▶ Connect a virtual machine to the group
- ▶ Authorize the new group to the RSCS node

You create the group SENDRSCS and connect a virtual machine to this group, as shown in Figure 2-33.

```
rac addgroup sendrscs owner(sys1)
Ready; T=0.01/0.01 10:06:54

rac listgrp sendrscs
INFORMATION FOR GROUP SENDRSCS
      SUPERIOR GROUP=SYS1          OWNER=SYS1
      NO INSTALLATION DATA
      NO MODEL DATA SET
      TERMUACC
      NO SUBGROUPS
      NO USERS
Ready; T=0.01/0.01 10:07:06

rac connect gumby group(sendrscs)
Ready; T=0.01/0.01 10:09:42
```

Figure 2-33 Create Group SENDRSCS

Next, display the attributes of the virtual machine with the RACF LISTUSER command (Figure 2-34).

```
rac lu gumby
USER=GUMBY  NAME=UNKNOWN  OWNER=IBMUSER  CREATED=07.195
  DEFAULT-GROUP=SYS1      PASSDATE=07.200  PASS-INTERVAL= 30  PHRASE
  ATTRIBUTES=PASSPHRASE
----- 8 line(s) not displayed -----
  GROUP=SYS1      AUTH=USE      CONNECT-OWNER=IBMUSER  CONNECT-
CONNECTS=        06  UACC=UPDATE  LAST-CONNECT=07.200/09:55:37
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
  GROUP=SENDERSCS AUTH=USE      CONNECT-OWNER=DETRO  CONNECT-
CONNECTS=        00  UACC=NONE   LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
  NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
Ready; T=0.01/0.01 10:10:16
```

Figure 2-34 RACF Listuser

With these steps completed, the virtual machine can now use the SENDFILE command to transfer a file with RSCS (Example 2-40). If you created RSCS's VMRDR to have a UACC(NONE), then you also have to permit the virtual machine to have UPDATE access to the VMRDR. You can use the RACF RALTER command to make this change. In our tests, we did not see an exposure in doing this, because we secured VMNODE with RACF.

Example 2-40 Successfully sent file using the GROUP authorization

```
sf profile exec a to maint at vmlinux5
RPIMGR032E YOU ARE NOT AUTHORIZED TO SPOOL TO RSCS
HCPSPLO07E Invalid userid - RSCS
Ready(00007); T=0.01/0.01 10:36:48

(**** From a RACF administrator we changed the UACC to update for RSCS****)

sf profile exec a to maint at vmlinux5
File PROFILE EXEC A1 sent to MAINT at VMLINUX5 on 07/19/07 10:38:03
Ready; T=0.01/0.01 10:38:03
From VMLINUX5: DMTAXM104I File (0063) spooled to MAINT -- origin 2NDLEVEL
(GUMBY) 07/19/07 10:38:02 EDT
```

2.4 RACF security labels

A security label, or *SECLABEL*, designates an object's relative confidentiality and its membership in a security category. An object's security label defines what sort of data it can contain and, by implication, what sort of data it cannot contain.

2.4.1 Security labels overview

Use a *security label* to associate a specific security level with a set of (zero or more) security categories. Security labels, when associated with resources and users, provide several advantages over security levels and security categories, which is an older RACF mechanism which could be used to classify users and resources.

You can assign security labels to data that is not necessarily protected by a resource profile. For example, spool files are assigned the security label of their creators. In many cases, data that has been assigned a security label retains that security label from the time the data is created until the data is deleted. It means, when a spool file is created by a user that is running under a security label, the spool file is assigned the security label of the user. The spool file retains that security label until the spool file itself is deleted (which can be long after the user logs off).

Users can log on with different security labels at different times but with the same user ID. Without security labels, a user always has the same security level and categories. Output printed for a user or job by the Print Services Facility™ can have a PSF identification label related to the security label of the user or job printed on every page.

Output printed for a user by the z/VM system or by RSCS Secure Printing can have an identification label related to the security label of the user printed for each output file.

It is easy to maintain the security classification of users and data. Changing the definition of a security label affects all users and resources that have that security label and you need not make the same change for many different profiles as you would have to for security levels and categories.

2.4.2 Creating a security label

When creating a security label, you must create a profile in the SECLABEL class. The name of the profile is the security label.

To create a SECLABEL profile, do the following:

1. Define the SECLEVEL profile to the SECDATA class using the RDEFINE command:

```
RDEFINE SECDATA SECLEVEL UACC(NONE)
```

2. Define security levels as members of the SECLEVEL profile in the SECDATA class:

```
RALTER SECDATA SECLEVEL ADDMEM(seclevel-name/seclevel-number ...)
```

3. Define the CATEGORY profile to the SECDATA class using the RDEFINE command:

```
RDEFINE SECDATA CATEGORY UACC(NONE)
```

4. Define categories as members of the CATEGORY profile in the SECDATA class:

```
RALTER SECDATA CATEGORY ADDMEM(category-1 category-2 ...)
```

5. For each security label, define a profile in the SECLABEL class. The profile names are the security labels available on your system, and must be no longer than eight characters. For each SECLABEL profile, specify a security level and (optionally) a set of categories. For example:

```
RDEFINE SECLABEL security-label SECLEVEL(seclevel-name)  
ADDCATEGORY(category-1 category-2 ...)
```

6. Users cannot use a particular security label (for logging on, for submitting a job, or for specifying in a RACF profile) unless they have at least READ access authority to the SECLABEL profile of that name. For example, if the SECLABEL profile named EAGLE has UACC(NONE) specified, and you wanted user AHLEE and group GROUP1 to be able to log on with a security label of EAGLE, issue the following command:

```
PERMIT EAGLE CLASS(SECLABEL) ACCESS(READ) ID(AHLEE GROUP1)
```

7. When you are ready to start using security labels, activate the SECLABEL class and activate SETROPTS RACLIST processing for the class. SETROPTS RACLIST processing is required for the SECLABEL class. You can do these two actions in one command:

```
SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)
```

2.4.3 Security label naming restrictions

Security labels SYSHIGH, SYSLOW, and SYSNONE are security labels that you can specify for resource and user profiles. However, you cannot create them directly. At RACF initialization, RACF creates SECLABEL profiles with the following names if they do not already exist:

- ▶ *SYSHIGH* combines the highest security level specified in the SECLEVEL profile with all the categories defined in the CATEGORY profile.
- ▶ *SYSLOW* is the lowest security level specified in the SECLEVEL profile and no categories.
- ▶ *SYSNONE* is the same as SYSLOW, but is intended for use on resources that must be written to at different security labels when the SETROPTS MLS option is in effect.

2.4.4 Security label NONE

You should not define a security label on z/VM with the name NONE for the following reasons:

- ▶ NONE is the default security label for z/VM system printers.
- ▶ VM does not permit jobs to print on a printer with a security label of NONE.
- ▶ When a spool file has no security label, the response to a QUERY READER, QUERY PRINTER, QUERY PUNCH or QUERY TRFILES command will contain the character string NONE in the SECLABEL field.

2.5 RACF auditing

One of the many reasons to run an ESM on your system is to have a tool that can help you to determine whether there have been any attempts to bypass system integrity. RACF audit data is a record of an installation's security-relevant events. This data is used to verify the effectiveness of an installation's security policy, determine whether the installation's security objectives are being met, and identify unexpected security relevant events.

RACF, like all other ESMs, has the ability to customize auditing requirements for your installation, thus allowing you to meet your corporate security policies. RACF uses attributes to allow or disallow RACF command execution. The SPECIAL attribute allows a virtual machine to issue all RACF commands, except for those assigned to the AUDITOR class (but with SPECIAL you can assign the AUDITOR attribute - this action is also audited). This separation of powers is necessary because it is the security administrator's job to establish RACF controls; it is the auditor's job to test the adequacy and effectiveness of these controls. Auditor is responsible for checking that RACF is meeting the installation's needs for access control and accountability.

2.5.1 Enabling auditing

You can enable (audit) or disable (noaudit) functions dynamically to meet the needs of your installation. When you enable to collect the audit records, SMF records will be generated. This was an optional step in the configuration of your RACF environment (Example 2-2 on page 28 and Example 2-3 on page 28). If you elected not to perform that step previously, you need to implement it now, before continuing.

RACF always logs information about certain events which are essential to an effective data-security mechanism. The events that RACF always logs are:

- ▶ Every use of the RVARY or SETROPTS command.
- ▶ Every time a RACROUTE REQUEST=VERIFY request fails.
- ▶ Every time the console operator grants access to a resource as part of the failsoft processing performed when RACF is inactive.

RACF never logs some events, because knowing about these events is not essential to effective data security. RACF never logs any use of the following RACF commands:

- ▶ LISTGRP
- ▶ LISTUSER
- ▶ RLIST
- ▶ LDIRECT
- ▶ LFILE
- ▶ SRFIL
- ▶ SRDIR
- ▶ SEARCH

In addition to the events that RACF always logs and never logs, there are other events RACF can log optionally. Optional logging is under the control of either a resource-profile owner or the virtual machine with the auditor attribute.

The first step in establishing the auditing environment is to activate the RACF class for auditing with the SETROPTS command. With this command you specify what functions within the AUDIT facility you want to monitor (above what RACF always monitors). These functions include:

- ▶ USERS
- ▶ VMMDISK
- ▶ VMLAN
- ▶ VMRDR
- ▶ VMCMD
- ▶ VMNODES
- ▶ SURROGAT

Use the SETROPTS LIST command to determine your current AUDIT environment, as shown in Example 2-41.

Example 2-41 AUDIT CLASS functions

```
rac setropts list
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM) SAUDIT CMDVIOL NOOPERAUDIT
STATISTICS = NONE
AUDIT CLASSES = NONE
ACTIVE CLASSES = DATASET USER GROUP VMMDISK VMRDR VMNODE VMLAN SURROGAT
                  VXMBR VMXEVENT
GENERIC PROFILE CLASSES = NONE
GENERIC COMMAND CLASSES = NONE
GENLIST CLASSES = NONE
GLOBAL CHECKING CLASSES = NONE
RACLIST CLASSES = NONE
```

As shown in this example, auditing for RACF classes is not enabled. Prior to enabling any of the functions, you need to start the RACFSMF virtual machine and update the PROFILE EXEC for the AUTOLOG2 virtual machine to start RACFSMF when the system is IPLed.

There are two main utilities that are used to manage the RACF generated SMF records in the z/VM environment.

- ▶ RACF Report Writer
- ▶ RACF SMF Data Unload

The report writer utility supports audit records for RACF 1.9.2 and earlier. It does not support most of the audit records introduced in RACF 1.10 for z/VM or later releases. RACF Report

Writer require the use of *tdisk* space on your system. You need to discuss with your z/VM system programmer if *tdisk* space has been defined on your system. If it has not, then it needs to be added.

These utilities are located on RACFVM's 305 disk, and the disk must be linked and accessed prior to execution.

You start by turning on a few other AUDIT features on the z/VM system prior to running these programs.

Issue the following command, as shown in Figure 2-35:

```
rac setropts audit (user group vmmdisk vmrdr vmlan surrogat)
```

```
rac setropts audit (user group vmmdisk vmrdr vmlan surrogat)
ICH14002I NOT AUTHORIZED TO SPECIFY AUDIT; KEYWORD IGNORED.
Ready(00004); T=0.01/0.01 07:37:25

rac alu maint audit
Ready; T=0.01/0.01 07:37:36

rac setropts audit (user group vmmdisk vmrdr vmlan surrogat)
Ready; T=0.01/0.01 07:37:42

rac setropts list
ATTRIBUTES = INITSTATS NOWHEN(PROGRAM) SAUDIT CMDVIOL NOOPERAUDIT
STATISTICS = NONE
AUDIT CLASSES = USER GROUP SURROGAT VMMDISK VMRDR VMLAN
ACTIVE CLASSES = DATASET USER GROUP VMMDISK VMRDR VMNODE VMLAN SURROGAT
                  VXMBR VMXEVENT
GENERIC PROFILE CLASSES = NONE
GENERIC COMMAND CLASSES = NONE
GENLIST CLASSES = NONE
GLOBAL CHECKING CLASSES = NONE
RACLIST CLASSES = NONE
```

Figure 2-35 RACF SETROPTS AUDIT controls

2.5.2 RACF Data Security Monitor Utility (RACDSMON)

DSMON is a program that produces reports on the status of the security environment of your installation and the status of resources that RACF controls. You can use the reports to audit the current status of your installation's system security environment by comparing the actual system characteristics and resource-protection levels with the intended characteristics and levels. You can also control the reporting that DSMON does by specifying control statements that request certain functions for user input.

Prior to running the RACDSMON EXEC, you have to link and access several disks owned by RACFVM. This exec also requires that you have a virtual storage of twenty megabytes. Perform the steps in Figure 2-36, when you IPL the 490 disk depending upon what CMS commands are executed from the PROFILE EXEC you might receive some errors You can disregard those error messages.

```
define 490 590
08:29:53 DASD 0590 DEFINED
Ready; T=0.01/0.01 09:29:44

def stor 20m
STORAGE = 20M
Storage cleared - system reset.

link racfvm 490 490 rr
link racfvm 200 200 rr
link racfvm 300 300 rr
link racfvm 305 305 rr
08:30:02 DASD 0490 LINKED R/O; R/W BY RACFVM
Ready; T=0.01/0.01 08:30:02

ipl 490
RACFVM CMS XA Rel 14 03/19/2002
Ready; T=0.01/0.01 08:30:09

access 305 1
```

Figure 2-36 RACDSMON step-up procedures

When the RACDSMON EXEC runs, the generated output is placed in your virtual printer. You should issue the **cp spool print *** command, so that you can receive the files when this process completes. (You might want to add this to your PROFILE EXEC.)

RACFVM's 200 and 300 disks are the locations of the RACF database. The RACDSMON generated reports will be pulled from those disks. When the exec is executed, you need to create a tdisk of the same disk type as these 200 and 300 disks. You can issue the CP commands **query virtual 200** and **query virtual 300** to determine this information. You can install the z/VM system on 3390 DASD or on simulated 9330 Fixed Block SCSI disk. Therefore, one of these is the type of tdisk space that you need to define. In our example, we installed the system on 3390 DASD.

To run the RACDSMON utility, enter the CP command **racdsmon**. It displays several panels. The first panels are only informational in nature. One allows you to go into a CMS SUBSET environment, where you can perform tasks such as linking to the disk that you should have linked prior to executing the exec. The first panels where you need to provide information is the panel that prompts you for the address of the INPUT RACF database device (see Figure 2-37). When the exec runs, these prompts are actually displayed on three separate panels.

```
Enter the INPUT RACF dataset device address one at a time.

Enter END when all input data sets are entered.
or
Enter QUIT to terminate processing.
200

Enter the NEXT INPUT RACF dataset device address .

Enter END when all input data sets are entered.
or
Enter QUIT to terminate processing.
300

Enter the NEXT INPUT RACF dataset device address .

Enter END when all input data sets are entered.
end
```

Figure 2-37 INPUT RACF Datasets

The next prompt from the exec asks whether you want to use a tdisk or a minidisk. If your system does not have tdisk space defined, then you can use existing minidisk. These disks have to be defined in the system directory and need to be the same size and geometry as the 200 and 300 disk that is owned by RACFVM. We chose to use tdisk for our process, as shown in Figure 2-38.

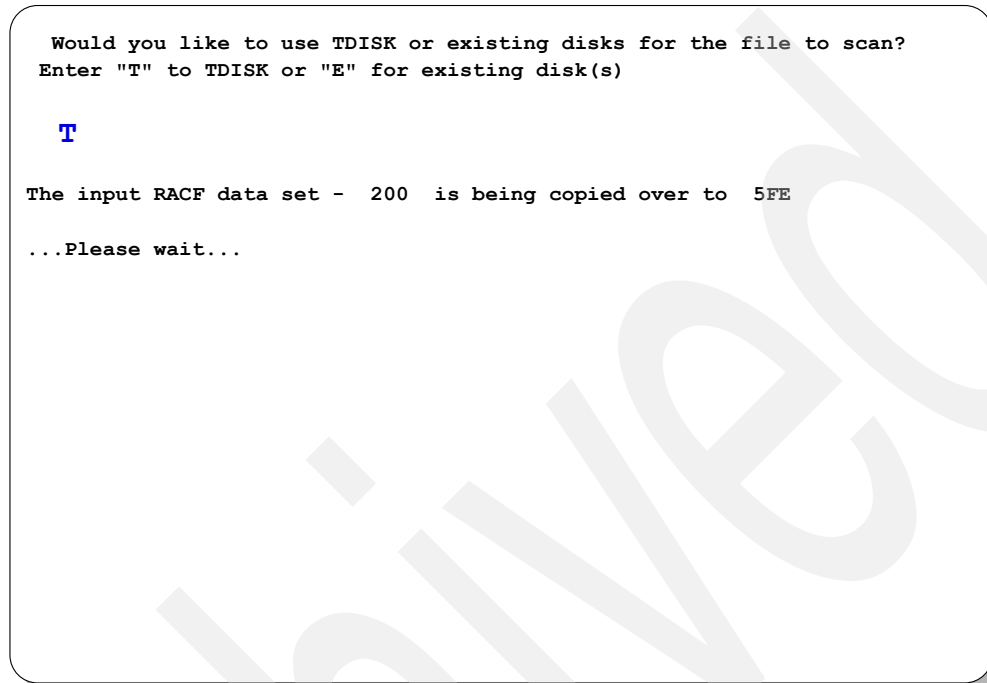


Figure 2-38 TDISK or Existing Disk

The next two screens display messages about the copy of the 200 and 300 disks to the 5FD and 5FE disks. You should then receive a message about the ICHDSM00 SYSIN file and have an opportunity to edit the file (Example 2-42). Accept the default on this panel, and edit the file.

Example 2-42 ICHDSM00 SYSIN file message

The ICHDSM00 SYSIN file will initially contain all DSMON FUNCTION control statements that are applicable to VM .

XEDIT will be invoked in order to tailor the ICHDSM00 SYSIN file.

Please be sure to issue the FILE command when edits are completed.

Press Enter to go into XEDIT

You need to modify the ICHDSM00 SYSIN file. It is shipped with one option that is not supported on RACF for z/VM. Figure 2-39 shows the correct modifications for this file. After you modify the file, save it. When running RACDSMON in the future, you can respond to the question about editing this file with NO (which will then use the file on your A disk).

```
ICHDSM00 SYSIN      A1  F 80  Trunc=80 Size=7
==>
0 * * * Top of File * * *
1 FUNCTION SYSTEM
2 FUNCTION RACGRP
3 FUNCTION RACCDT
4 FUNCTION RACEXT
5 FUNCTION RACGAC
6 FUNCTION RACUSR
7 FUNCTION RACDST
8 * * * End of File * * *
```

Figure 2-39 Update ICHDSM00 SYSIN

After a few minutes, you receive the messages shown in Figure 2-40.

```
Program ICHDSM00 is being executed - Please wait

DMSACC723I R (0200) R/W - OS
DMSACC723I Q (0300) R/W - OS
CSTSET001I CMS SUB-TASKING SUPERVISOR INITIALIZED.
CSTINT003I INITIATOR ACTIVATED.
RPISMF050E Syntax error in SMF control card
RPISMF054I SMF recording not started
  ICH508I ACTIVE RACF EXITS: ICHPWX11
  ICH520I RACF 5.3.0 IS ACTIVE.
RDR FILE 0084 SENT FROM DETRO      PRT WAS 0084 RECS 0354 CPY  001 A NOHOLD
NOKEEP
CSTINT004I PROGRAM 'RACFIPL' ENDED. COMPLETION CODE = 000000.
CSTINT006I NO MORE SUB-TASKS.
CSTTER001I CST TERMINATED.

Return code from ICHDSM00 = 0
RDR FILE 0085 SENT FROM DETRO      PRT WAS 0085 RECS 0030 CPY  001 A NOHOLD
NOKEEP
Ready; T=0.08/0.15 11:22:41
```

Figure 2-40 RACDSMON completion

As shown in Figure 2-40, it looks as though there is a problem, because of the RPISMF050E and RPISMF054I messages. However, these messages are a little misleading, as discussed in *Security Server Messages and Codes*, SC24-6148. The RPISMF050E is as follows:

```
RPISMF050E SYNTAX ERROR IN SMF CONTROL CARD
```

This error message occurs because the card in the file named SMF CONTROL is in error. You can ignore this message if it is issued while you are running the RACUT100 exec or the RACDSMON exec.

When the RACDSMON EXEC has completed, you need to IPL CMS or 190 to have full CMS function. The 490 disk owned by 5VMRAC30 does not include all of the CMS executable that you have with normal CMS. After IPLing the 490 disk you do not have access to FIELIST, RDRLIST, FULLIST, XEDIT, and so forth.

The RACDSMON EXEC creates two files in your VMRDR. The file named (none) is the actual report generated. You should receive a file that provides a file name and file type that is used by the security audit team. You can discard the file ICHDSM00 \$\$\$\$\$\$.

The audit report includes the following information:

- ▶ RACF System Report (Example 2-43)
- ▶ RACF Exits Report (Example 2-44)
- ▶ Selected User Attribute Report (Example 2-45)
- ▶ RACF Class Descriptor Table Report (Example 2-46)
- ▶ RACF Global Resource Table Report (Example 2-47)
- ▶ RACF Group Tree Report (Example 2-48)

Example 2-43 RACF System Report

```

DSMON    RPT0723  A1  F 132  Trunc=132 Size=354 Line=0 Col=1 Alt=13
====>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
0 * * * Top of File * * *
1 RACF DATA SECURITY MONITOR
2   S Y S T E M       R E P O R T
3 -----
4 CPU-ID                      006A3A
5 CPU MODEL                   2084
6 OPERATING SYSTEM/LEVEL      z/VM Version 5 Release 3.0, service 1
7 LAST SYSTEM GENERATION       Generated at 07/17/07 12:53:12 EDT
8 LAST SYSTEM IPL              IPL at 07/19/07 13:01:53 EDT
9 RACF VERSION 5 RELEASE 3 IS ACTIVE

```

Example 2-44 RACF Exits Report

```

DSMON    RPT0723  A1  F 132  Trunc=132 Size=356 Line=1
====>
11   R A C F       E X I T S       R E P O R T
12 EXIT MODULE          MODULE
13 NAME                 LENGTH
14 -----
15 ICHPWX11             1,520
16 RACF DATA SECURITY MONITOR

```

Example 2-45 Selected User Attribute Report

```

DSMON    RPT0723  A1  F 132  Trunc=132 Size=353 Line=19 Col=1 Alt=3
====>
19   S E L E C T E D   U S E R   A T T R I B U T E   R E P O R T
20 USERID             ----- ATTRIBUTE TYPE -----
21                   SPECIAL      OPERATIONS    AUDITOR      REVOKE
22 -----
33 DATAMOVE           SYSTEM      SYSTEM
34 DEFAULT
35 DETRO              SYSTEM      SYSTEM      SYSTEM
36 DIRMAINT           SYSTEM      SYSTEM
37 IBMUSER
38 MAINT              SYSTEM      SYSTEM      SYSTEM
39 NOBODY
40 OPERATOR           SYSTEM      SYSTEM
41 SYSADMIN           SYSTEM      SYSTEM

```

Example 2-46 RACF Class Descriptor Table

DSMON	RPT0723	A1	F 132	Trunc=132	Size=354	Line=53	Col=1	Alt=2
====>								
53	R A C F	C L A S S	D E S C R I P T O R	T A B L E	R E			
54	CLASS				DEFAULT			
55	NAME	STATUS	AUDITING	STATISTICS	UACC			
56	-----							
57	RVARSMBR	INACTIVE	NO	NO	NONE			
58	RACFVARS	INACTIVE	NO	NO	NONE			
59	SECLABEL	INACTIVE	NO	NO	NONE			
60	VMMDISK	ACTIVE	YES	NO	NONE			
61	VMRDR	ACTIVE	YES	NO	NONE			
62	VMCMD	INACTIVE	NO	NO	NONE			
63	VMNODE	ACTIVE	NO	NO	NONE			
64	VMBATCH	INACTIVE	NO	NO	NONE			
65	FILE	INACTIVE	NO	NO	NONE			
66	DIRECTRY	INACTIVE	NO	NO	NONE			
67	SFSCMD	INACTIVE	NO	NO	NONE			
68	VMPOSIX	INACTIVE	NO	NO	NONE			
69	VMLAN	ACTIVE	YES	NO	NONE			
70	VMMAC	INACTIVE	NO	NO	NONE			
71	VMSEGMT	INACTIVE	NO	NO	NONE			

Example 2-47 RACF Global Resource Table

DSMON	RPT0723	A1	F 132	Trunc=132	Size=353	Line=264	Col=1	Alt=13
====>								
264	R A C F	G L O B A L	A C C E S S	T A B L E				
265	CLASS	ACCESS	ENTRY					
266	NAME	LEVEL	NAME					
267	-----							
268	PROPCNTL		-- GLOBAL INACTIVE	--				
269	APPCLU		-- GLOBAL INACTIVE	--				
270	SMESAGE		-- GLOBAL INACTIVE	--				
271	DEVICES		-- GLOBAL INACTIVE	--				
272	VTAMAPPL		-- GLOBAL INACTIVE	--				
273	PSFMPL		-- GLOBAL INACTIVE	--				
274	OPERCMDS		-- GLOBAL INACTIVE	--				
275	WRITER		-- GLOBAL INACTIVE	--				
276	JESSPOOL		-- GLOBAL INACTIVE	--				
277	JESJOBS		-- GLOBAL INACTIVE	--				
278	JESINPUT		-- GLOBAL INACTIVE	--				
279	CONSOLE		-- GLOBAL INACTIVE	--				
280	TEMPDSN		-- GLOBAL INACTIVE	--				
281	DIRAUTH		-- GLOBAL INACTIVE	--				
282	SURROGAT		-- GLOBAL INACTIVE	--				
283	NODMBR		-- GLOBAL INACTIVE	--				

Example 2-48 RACF Group Tree Report

DSMON RPT0723 A1 F 132 Trunc=132 Size=353

====>

320 R A C F G R O U P T R E E
321 LEVEL GROUP (OWNER)

322 -----

323

324 1 | SYS1 (IBMUSER)

325

326 2 | GADM (IBMUSER)

327

328 2 | GBIN (IBMUSER)

329

330 2 | GNOBODY (IBMUSER)

331

332 2 | GSYS (IBMUSER)

333

334 2 | MAIL (IBMUSER)

335

336 2 | SECURITY (IBMUSER)

337

338 2 | SENDRSCS

339

340 2 | STAFF (IBMUSER)

2.5.3 RACF SMF Data Unload Utility (RACFADU)

The RACF SMF data unload utility, available with RACF/VM 1.10 and RACF FL 530, is the IBM-recommended utility for processing RACF audit records. With it, you can create a sequential file from the security relevant SMF data. You can use the sequential file in several ways:

- ▶ View the file directly
- ▶ Use the file as input for installation-written programs
- ▶ Manipulate the file with sort/merge utilities
- ▶ Output to an XML-formatted file for viewing with a Web browser

If the output is loaded into a database management system, for example DB2® or SQL/DS™, you can issue your own queries. RACF ships the sample statements required to define and load the DB2 tables.

Before you can execute the RACFADU EXEC, you need to:

- ▶ Link and access RACFVM's 305
- ▶ Link RACFVM's 301 and 302
- ▶ Have adequate free space on your A disk for the output file (30 cylinders worked)

You can issue RACFADU EXEC with or without any parameters. Without any parameters, it opens the RACFADU Panel (Figure 2-41).

RACF SMF Unload Utility - Input Panel

. Virtual address of input SMF data minidisk	_____	
. Virtual address of output minidisk	_____	
. Filename and filetype of sequential output file	RACFADU	OUTPUT
. Filename and filetype of XML easily readable output file	_____	_____
. Filename and filetype of XML compressed output file	_____	_____

PF1 = Help PF2 = Execute PF3 = Quit
ENTER = Verify input fields

Enter CP/CMS Commands below:
====>

Figure 2-41 RACFADU panel

Example 2-49 shows the command issued with all the required options for the command and bypasses the input panel.

Example 2-49 RACFADU without input panel

```

Ready; T=0.02/0.03 07:45:25
racfadu 301 191
RACFADU OUTPUT
RPIADU033I SMF unload completed successfully.
View the RACFADU MESSAGES file for additional details.
Ready; T=0.10/0.13 07:45:35

```

When you execute the exec in either mode, two files are created on your A disk by default. The files created are:

```

RACFADU MESSAGES A1
RACFADU OUTPUT A1

```

The RACFADU MESSAGES file describes how many of each type of SMF records were processed, as shown in Example 2-50.

Example 2-50 RACFADU MESSAGES file

```

RACFADU  MESSAGES A1  F 132  Trunc=132 Size=9 Line=0 Col=1 Alt=0
====>
0 * * * Top of File * * *
1 IRR67652I The utility processed 0 SMF type 30 records.
2 IRR67652I The utility processed 2355 SMF type 80 records.
3 IRR67652I The utility processed 21 SMF type 81 records.
4 IRR67655I The utility processed 0 SMF type 83 subtype 1 records.
5 IRR67655I The utility processed 0 SMF type 83 subtype 2 records.
6 IRR67655I The utility processed 0 SMF type 83 subtype 3 records.
7 IRR67655I The utility processed 0 SMF type 83 subtype 4 records.
8 IRR67653I The utility bypassed 0 SMF records not related to IRRADU00.
9 IRR67650I SMF data unload utility has successfully completed.
10 * * * End of File * * *

```

The RACFADU OUTPUT file is the readable output of all the SMF data records (Example 2-51). If this file already exist on the output disk, you are prompted to rename or replace the old file before continuing. These files can be used by DB2, SQL/DS, and DFSORT/CMS.

Example 2-51 RACFADU OUTPUT file

```

RACFADU  OUTPUT  A1  V 5331  Trunc=5331 Size=5043 Line=3972 Col=1 Alt=0
====>
3972 DEFINE    SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3973 RDEFINE   SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3974 PERMIT    SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3975 PERMIT    SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3976 PERMIT    SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3977 PERMIT    SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3978 PERMIT    SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3979 PERMIT    SUCCESS  11:00:34 2007-07-18 VMSP NO NO NO DETRO SYS1
3980 ACCESS    SUCCESS  11:11:44 2007-07-18 VMSP NO NO NO DETRO SYS1
4598 ACCESS    SUCCESS  17:18:19 2007-07-18 VMSP NO NO NO DETRO SYS1
4751 ACCESS    SUCCESS  17:24:04 2007-07-18 VMSP NO NO NO DETRO SYS1
4993 ACCESS    SUCCESS  07:08:29 2007-07-23 VMSP NO NO NO DETRO SYS1

```

The utility can also be used to generate a XML file that is readable with a browser. To create the XML file, you must pass the file name and file type for the XML file. The parameters are:

```

OUTXRN filename File name of output XML easily readable file.
OUTXRT filename File type of output XML easily readable file.
OUTXCN filename File name of output XML compressed.
OUTXCT filename File type of output XML compressed.

```

We found it much easier to use the panel when generating the XML files from your SMF data (Figure 2-42). Also, if you are using the XML function, we found it worked better with the compressed version of this process.

```

                                RACF SMF Unload Utility - Input Panel

. Virtual address of input SMF data minidisk          0301
. Virtual address of output minidisk                  0191
. Filename and filetype of sequential
  output file
. Filename and filetype of XML easily readable
  output file
. Filename and filetype of XML compressed          RACFADU1 XML
  output file

                                PF1 = Help    PF2 = Execute    PF3 = Quit
                                ENTER = Verify input fields

Enter CP/CMS Commands below:
```

Figure 2-42 Creating the XML file

When the file was created, use the IBM Personal Communications program to download the file from the A disk to the desktop in *text* mode. After you have downloaded the file, open it with an editor and change the encoding value on the first line as shown in Figure 2-43 (and as documented in *RACF Security Server Auditor's Guide*, SC24-6143). This is required because of the mis-match between z/VM's use of EBCDIC versus the PC using ASCII.

```

RACFADU9 XML      A1  V 191  Trunc=191 Size=209591 Line=0 Col=1 Alt=0
=====
0 * * * Top of File * * *
1 <?xml version='1.0' encoding='ISO8859-1 ?>
2 <securityEventLog xmlns='http://www.ibm.com/xmlns/zOS/IRRSchema'>
3
4   <rdf:Description rdf:about=''
5                       xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns
6                       xmlns:dc='http://purl.org/dc/elements/1.1/'>
7       <dc:creator>      RACF for z/VM      SMF Unload (HRF5030)</dc:creat
8       <dc:subject>RACF Security Event Log 2007-07-24 11:49:54</dc:subject>
9       <dc:language>en</dc:language>
10    </rdf:Description>
11
12    <event>
13      <eventType>RACFINIT</eventType>
14      <timeWritten>11:19:46.54</timeWritten>
15      <systemSmfid>VMSP</systemSmfid>
16      <prodName>RACF</prodName>
17      <details>
18        <datasetName>RACF.DATASET</datasetName>
19        <datasetVol>RACF</datasetVol>

```

Figure 2-43 Modify the encoding value

You can view the audit report on personal computers and workstations using an XML-capable Web browser. Many browsers available today have the ability to correctly parse and render XML documents. Therefore, when the audit report is on that system, you can read it as easily as any other Web document. Simply open a listing of the files and single- or double-click the file to open it in the browser window.

When we opened this file with FireFox, we received message in reference to a missing style file (Figure 2-44). In this case, you need to combine this file with a customized stylesheet to get the browser to filter and display the panels in a more acceptable format.

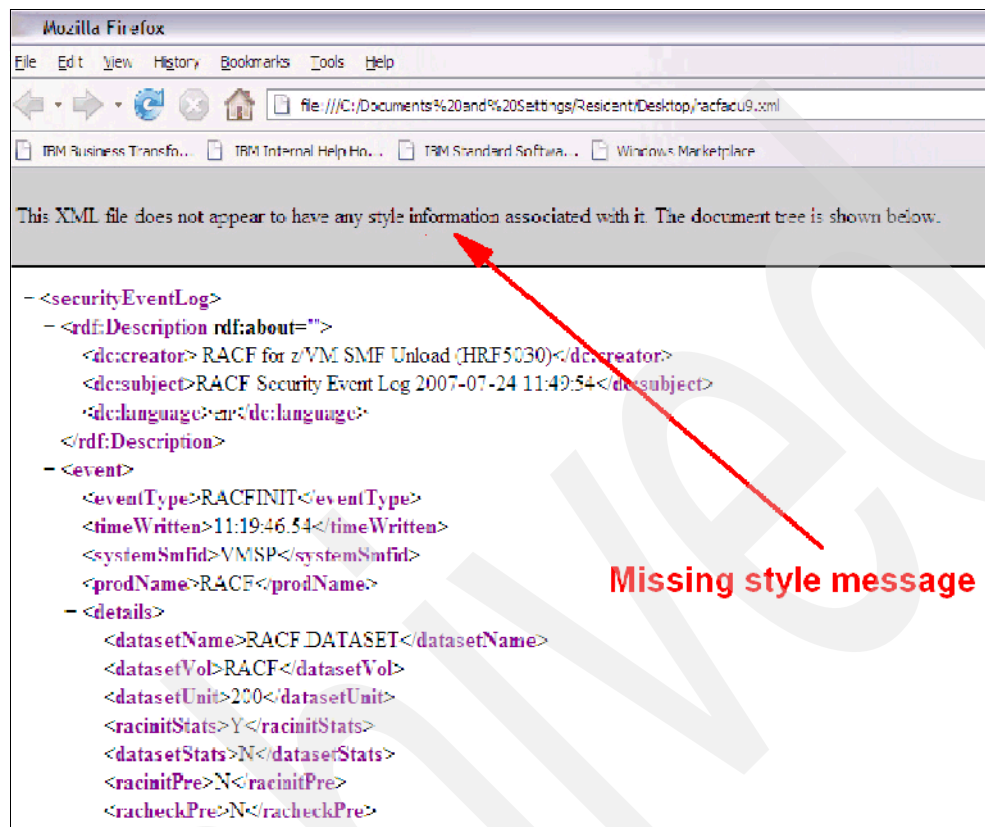


Figure 2-44 Opening file with FireFox

When opened with Internet Explorer, we received a message about ActiveX® needing to be allowed (Figure 2-45). However, in both cases we only saw the output in tree format.

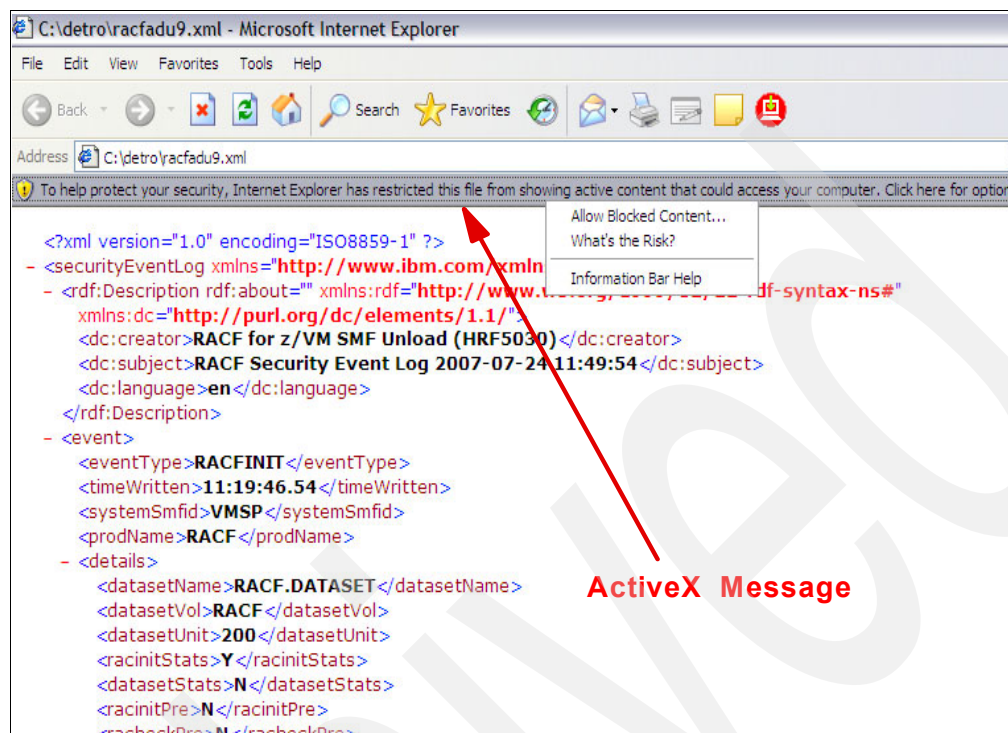


Figure 2-45 Internet Explorer view

2.5.4 RACF report writer utility (RACFRW)

Note: The report writer is no longer the IBM-recommended utility for processing RACF audit records. The RACF SMF data unload utility is the preferred reporting utility. The report writer does not support many of the audit records introduced after RACF 1.9.2.

The RACF report writer lists the contents of System Management Facilities (SMF) records in a format that is easy to read. SMF records reside in the SMF data file. You can also tailor the reports to select specific SMF records that contain certain kinds of RACF information. With the RACF report writer, you can obtain:

- ▶ Reports that describe attempts to access a particular RACF-protected resource in terms of user identity, number and type of successful accesses, and number and type of attempted security violations.
- ▶ Reports that describe user and group activity.
- ▶ Reports that summarize system use and resource use.

The RACF report writer lists the contents of SMF records in a format that is easy to read. It provides a wide range of reports that enable you to monitor and verify the use of the system and resources.

The RACF report writer consists of three phases:

1. Command and sub-command processing
2. Record selection
3. Report generation

The steps to perform when running the RACRPORT EXEC are similar to the steps in running other RACF utilities. You must link and access several disk owned by RACFVM and then execute the exec (Figure 2-46). Unlike the RACDSMON where you have to link to RACFVM's 200 and 300 (location of the RACF database), this time, you need access to the SMF records which are created on RACFVM's 301 and 302 disk.

While working with RACRPORT, we observed that the tdisk used by this utility was hardcoded in the program with virtual address 5FF. If you use that virtual address for something else, you must detach it or re-define the disk to another address.

```
link racfvm 191 111 rr
link racfvm 301 301 rr
link racfvm 302 302 rr
link racfvm 305 305 rr

17:06:09 DASD 0111 LINKED R/O; R/W BY RACFVM
Ready; T=0.01/0.01 17:06:09
17:06:09 DASD 0301 LINKED R/O; R/W BY RACFVM
Ready; T=0.01/0.01 17:06:09
17:06:09 DASD 0302 LINKED R/O; R/W BY RACFVM
Ready; T=0.01/0.01 17:06:09
17:06:09 DASD 0305 LINKED R/O; R/W BY RACFVM
Ready; T=0.01/0.01 17:06:09
```

Figure 2-46 Linking required disk for RACRPORT

The RACRPORT EXEC generates the reports in your virtual printer. So, spool your printer to your reader to make things easier.

The SET SECUSER command is used to change the secondary user ID that is associated with your virtual machine or another user's virtual machine. To issue the command for the OPERATOR virtual machine, you must be authorized to execute class A privileged commands (Example 2-52). This command is the command that additionally authorizes the use of the CP SEND command.

Example 2-52 CP Query Privclass

```
id
DETRO    AT 2NDLEVEL VIA RSCS    07/23/07 14:14:19 EDT    MONDAY
Ready; T=0.01/0.01 14:14:19
q privclass
Privilege classes for user DETRO
    Currently: ABCDEFG
    Directory: ABCDEFG
Ready; T=0.01/0.01 14:14:22
```

Before, you can run this utility, RACFVM has to switch from the primary SMF disk to the alternate SMF disk. This is accomplished with the RACF SMF SWITCH command. During the RACF implementation, there was an optional step to change the Message Routing Table, which allowed to define additional virtual machines that would be allowed to request the SMF

SWTICH of RACFVM. At that time, you stated that you really did not need to add additional virtual machines to this list. Now, you are going to implement this process by having the OPERATOR virtual machine issue the SMF SWITCH, as shown in Figure 2-47.

```
smsg racfvm smf switch
Ready; T=0.01/0.01 17:31:35
  You are not authorized to issue SMSG to a RACF server

set secuser operator maint
17:10:17 HCPCFX6768I SECUSER of OPERATOR initiated.
17:10:17 HCPQCS150A User OPERATOR has issued a VM read
Ready; T=0.01/0.01 17:10:17
send cp operator smsg racfvm smf switch
```

OPERATOR and RACFSMF are the only virtual machines authorized to send RACFVM an SMSG command. This is one solution to solve this problem.

Figure 2-47 RACF SMF switch

RACFSMF has not been given the authority to link to RACFVM's 301 and 302 disk so the SMF SWITCH is going to fail. The solution is to issue RACF PERMITS commands for the those disks.

Issue the following commands:

```
rac permit racfvm.301 class(vmmdisk) id(racfsmf) ac(update)
rac permit racfvm.302 class(vmmdisk) id(racfsmf) ac(update)
```

Now when you issue the SMF SWITCH command through the CP SEND command, it will be successful. With this step completed, you can issue the RACRPORT command after you have accessed the 305 disk (Figure 2-48).

```
racrport
DMSACC724I 305 replaces B (305)
DMSACP723I B (305) R/O
DMSACP725I 305 also = L disk
The RACFRW CONTROL file cannot be located and is required for execution
to continue.

Do you wish to create a RACFRW CONTROL file?
Please enter YES or NO

yes

XEDIT will be invoked in order to tailor the RACFRW CONTROL file.

Please be sure to issue the FILE command when edits are completed.
Please press Enter to continue
```

Figure 2-48 RACRPORT exec

The RACFRW CONTROL file must contain the input required by the report writer, including the RACFRW command and sub-commands. This file resides on your A disk and is created with XEDIT when RACFRW is executed. The statements included in Example 2-53 generate a detailed report.

Example 2-53 RACFRW CONTROL file

```
RACFRW  CONTROL  A1  F 80  Trunc=80 Size=8 Line=1
====>
===== * * * Top of File * * *
===== RACFRW TITLE ('MY Systems RACF REPORT Output 07/22/07')
===== SELECT VIOLATIONS
===== SELECT SUCCESSES
===== EVENT LOGON
===== EVENT SETROPTS
===== LIST
===== END
===== * * * End of File * * *
```

After modifying the RACFRW CONTROL file and saving it to your A disk, you are prompted to define where the work disk is located. Your options are on a tdisk or your A disk. We feel that because the A disk is usually a small disk, you would use the tdisk, much like we did when we discussed the RACDSMON exec. Respond to the prompt as shown in Figure 2-49.

```
The RACF Report Writer requires Disk Space for a Sort work file.
You may wish to use Tdisk for this function.

Note: If Tdisk is not used, the Sort work file will be written on
the A disk.

Do you wish to use Tdisk for the Sort work file?

Please enter YES or NO

YES
```

Figure 2-49 Using TDISK for RACRPORT

The RACRPORT exec does not contain the logic that we saw in the RACDSMON exec, where RACDSMON could determine what type of tdisk space was defined on your system. With the RACRPORT exec you have to specify the tdisk disk type.

The **query tdisk** command (Example 2-54) gives you information about system defined tdisk space. However, it does not provide the characteristics of the disk device. You need to query the devices to obtain the disk type.

Example 2-54 The query tdisk command

```
query tdisk

DASD 0429 ATTACHED CPVOL 0000 LX5PG2
DASD MDISKS NOT FOUND
DASD TDSKS NOT FOUND
Ready; T=0.01/0.01 08:02:10

cp query 429 id
DASD 0429 3390-0A CU: 2105-E8
Ready; T=0.01/0.01 08:02:19
```

With this information, you can select the proper type of tdisk to have created as the sort disk. The exec then provides you with the information about the SMF disk that is being used for the generation of this report (Figure 2-50). The requirement is that the sort disk must be the same size or larger than the source disk. We have always made the sort disk the same size as the source disk.

```

You will now be prompted for Tdisk space

Since the number of cylinders or blocks required depends on the size of
the SMF DATA file, it is recommended that you allocate a
temporary disk that is at least as large as the SMF DATA file.

The disk containing the SMF DATA file will be displayed below

LABEL  VDEV M  STAT  CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK
RCF301 301  C/A R/O   7 3390 4096     1      189-15    1071    1260

Please enter the number of cylinders or blocks for Tdisk allocation.

Format of 5FF Begins

007

```

Figure 2-50 Sort disk size definition

After you have defined the size of the source disk, the tdisk is created as address 5FF (this address must be available). It then uses the definitions you created in the RACFRW CONTROL file and generates a console file in your VMRDR. You can peek this file or receive it to disk. If you print it, it should be printed on a printer that supports logical record lengths of 132 characters.

Because of variations from one installation to another, it is not possible to identify all of the ways an auditor might use the RACF report writer. The following list, however, identifies some possibilities, which are described in *RACF Security Server Auditor's Guide*, SC24-6143:

- ▶ Monitoring Password Violation Levels
- ▶ Monitoring Access Attempts in WARNING Mode
- ▶ Monitoring Access Violations
- ▶ Monitoring the Use of RACF Commands
- ▶ Monitoring Specific Users
- ▶ Monitoring SPECIAL Users
- ▶ Monitoring OPERATIONS Users
- ▶ Monitoring Failed Accesses to Resources Protected by a Security Level
- ▶ Monitoring Accesses to Resources Protected by a Security Label

2.6 RACF database backup

We describe RACUT200 and RACUT400 programs in this section.

2.6.1 RACF database verification utility program (IRRUT200)

The RACF database verification utility program, IRRUT200, can obtain a working copy of the RACF database. RACUT200 is the exec that will call IRRUT200, a RACF utility program that you can use to identify inconsistencies in the internal organization of a RACF database and also to make an exact copy of the RACF database (Figure 2-51). To examine your database, you must use the release level of IRRUT200 that corresponds to the release level of your database. It performs the following functions:

- ▶ Validates and reports errors found in the relative byte addresses (RBAs) of each segment of all profiles.
- ▶ Validates that index entries point to the correct profile.
- ▶ Validates the database format.
- ▶ Issues return codes to signal validation errors.
- ▶ Scans the index blocks and prints information about problems with the index-block chains.
- ▶ Compares the segments of the database that are actually in use to the segments allocated according to the BAM blocks, and prints information about inconsistencies.
- ▶ Creates a backup copy of a RACF database.
- ▶ Creates an enhanced, formatted index report displaying the 255-byte profile name and profile type information.

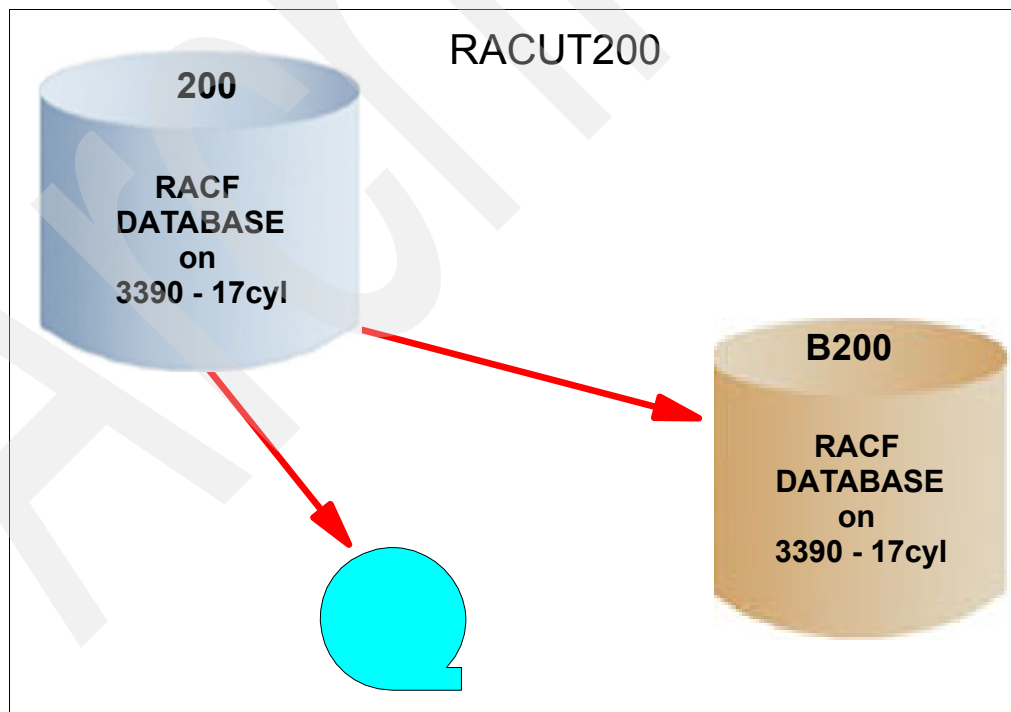


Figure 2-51 RACUT200 utility

2.6.2 RACF database Split/Merge/Extend utility program (IRRUT400)

RACUT400 is the exec that calls IRRUT400. It is similar to IRRUT200 in that it copies a RACF database, but it provides additional functions not available with IRRUT200 (Figure 2-52). It performs the following functions:

- Copies a RACF database to a larger or smaller database, provided there is enough space for the copy.
- Identifies inconsistencies, such as duplicate profiles appearing in different data sets.
- Physically reorganizes the database by bringing all segments of a given profile together.
- Copies a database to a different device type.

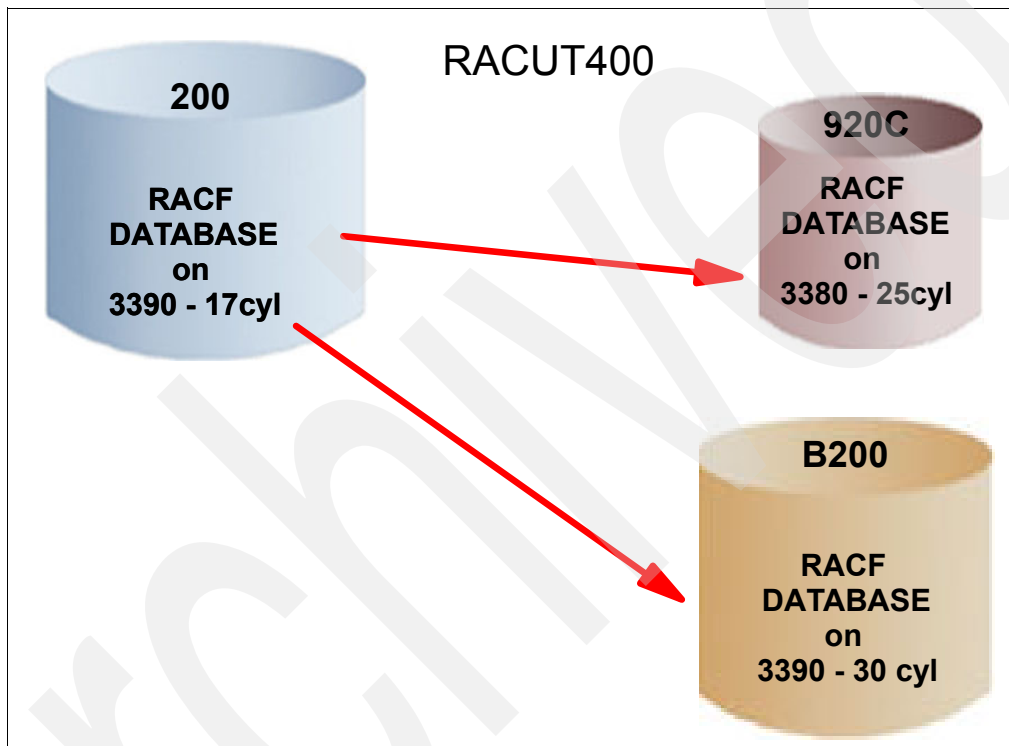


Figure 2-52 RACUT400 utility

When running these utilities, you have to make a decision, where you want to run them. They can be run from the RACFVM virtual machine or from another virtual machine that has linked and accessed the appropriate disks. If you run the utility from RACFVM, you must deactivate the RACF environment. This action is disruptive to your system. When running them from another virtual machine, the RACF environment does not have to be disabled. We feel it is best to back up the database without disruption.

To run RACUT200:

1. Log on to RACMAINT (this virtual machine has most of the disks required).
2. IPL CMS.
3. Create the control file called RACVERIFY FILE, this file resides on RACFVM's 191 disk.
4. Link and access RACFVM's 305 disk.
5. Obtain write access to the output disk (our examples we will use the B200 disk).

We have used the DirMaint process to create the B200 disk (output disk). This also defined the new device and issued the permit to the disk in the RACF database. The database disk or the backup disk must be formatted with a blocksize of 4096. We found it easier to just use the z/VM DASD Dump Restore utility (DDR) to get the proper blocksize, as shown in Example 2-53.

```

link racfvm b200 b200 mr
Ready; T=0.01/0.01 08:44:58
ddr
z/VM DASD DUMP/RESTORE PROGRAM
ENTER:
sysprint cons
ENTER:
input 300 dasd
ENTER:
out b200 dasd
ENTER:
copy all
HCPDDR711D VOLID READ IS RACFBK
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
yes
HCPDDR711D VOLID READ IS RACFBK
DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:
yes
COPYING RACFBK
COPYING DATA 07/25/07 AT 12.45.12 GMT FROM RACFBK TO RACFBK
INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS
START STOP START STOP
0 16 0 16
END OF COPY
ENTER:
END OF JOB

```

Figure 2-53 Using DDR to initialize the B200 disk

Now that the B200 disk has a blocksize of 4096 we can use the RACUT200 utility to verify and copy the RACF database. We found that we could not use the B200 disk address because there was an internal relationship of addresses within this utility. So we needed to define B200 as 300. To do this from the RACMAINT virtual machine we needed to free up address 300. We chose to detach the 300 disk and then define the B200 as 300 (Example 2-55).

Example 2-55 Disk address changes

```

detach 300
DASD 0300 DETACHED
Ready; T=0.01/0.01 08:53:02
define b200 300
DASD 0300 DEFINED
Ready; T=0.01/0.01 08:53:07
q v dasd
DASD 0190 3390 LX5RES R/O      107 CYL ON DASD 0423 SUBCHANNEL = 0004
DASD 0191 3390 LX5W02 R/W      9 CYL ON DASD 0428 SUBCHANNEL = 000D
DASD 019D 3390 LX5W01 R/O     146 CYL ON DASD 0427 SUBCHANNEL = 0005
DASD 019E 3390 LX5W01 R/O    250 CYL ON DASD 0427 SUBCHANNEL = 0006
DASD 0200 3390 LX5W02 R/O     17 CYL ON DASD 0428 SUBCHANNEL = 0009
DASD 0300 3390 LX5W02 R/W     17 CYL ON DASD 0428 SUBCHANNEL = 000E
DASD 0301 3390 LX5W02 R/W      7 CYL ON DASD 0428 SUBCHANNEL = 000B
DASD 0302 3390 LX5W02 R/O      7 CYL ON DASD 0428 SUBCHANNEL = 000C
DASD 0305 3390 LX5W02 R/W     41 CYL ON DASD 0428 SUBCHANNEL = 0008

```


Prior to running RACUT200, you should **spool print ***, which sends the output to your virtual printer. When the RACUT200 EXEC is executed, you are prompted for all the required parameters (Figure 2-54). Issue the **racut200** command.

**This exec is used to Verify and create a backup
Copy of a RACF data base.**

Press Enter to continue....

Figure 2-54 Executing RACUT200

You are prompted as to verify the RACF database, as shown in Figure 2-55.

Do you want to Verify a RACF data base?

Enter YES or NO or QUIT

YES

Figure 2-55 Verification question

Always respond *YES*, to this question (Figure 2-55).

IRRUT200 is controlled by utility control statements that have the following format:

INDEX (FORMAT)

I

where:

INDEX specifies you want the index scan function performed.

FORMAT specifies a formatted listing of all the index blocks.

Only one blank can separate INDEX and FORMAT

Press Enter to continue....

Figure 2-56 RACUT200 index

The next three screens are informational screens about the RACVERIFY FILE (Figure 2-56, Figure 2-57, and Figure 2-58).

```
MAP (ALL)
M
where:
MAP specifies you want the BAM /allocation verification performed.
ALL specifies that you want the encoded map for each BAM block
in the RACF data set printed.
Only one blank can separate MAP and ALL

END
where:
END terminates the utility program. An end-of-file on
SYSIN also terminates the program.

Press Enter to continue....
```

Figure 2-57 MAP and END information

```
XEDIT will be invoked in order to tailor the RACVERIFY FILE

Please be sure to issue the FILE command when edits are

Press Enter to go into XEDIT
```

Figure 2-58 Xedit to be invoked

The RACVERIFY FILE was created by the exec. It can be used with these three parameters. In the future, when you execute the RACUT200 EXEC, you are prompted whether you want to overlay this file. If you specify NO, it continues to use this file (Figure 2-59).

```
RACVERIFY FILE      A1  F 80  Trunc=80 Size=3 Line=3 Col=1
DMSXIN571I Creating new file:
DMSXMD587I XEDIT:

===== * * * Top of File * * *
===== INDEX FORMAT
===== MAP ALL
===== END
      |...+....1....+....2....+....3....+....4....+....5.
===== * * * End of File * * *

====>  FILE
```

Figure 2-59 RACVERIFY FILE

The process then prompts you for the input addresses (Figure 2-60). The input is 200 (the RACF primary disk), and the output is 300 (which is really the B200 address).

```
Enter the  Input  device address

200

Enter the  Output device address
      or
Enter a Null line to Bypass Copy

300
```

Figure 2-60 Input and output devices

Finally, you are prompted one last time before the backup occurs (Figure 2-61).

```
RACF.DATASET  is the input Racf data set at virtual address '200'
RACF.BACKUP   is the output Racf data set at virtual address '300'
Do you wish to continue?
Enter YES or NO

YES
```

Figure 2-61 Address confirmation

When the program completes, you notice the two NOTES messages. These are expected and this is not a problem. Important thing is that you receive a return code of zero (Figure 2-62).

```
Processing of RACF.DATASET begins
Program 'IRRUT200' is being executed - Please wait -

NOTE: Message RPIRND003E for TTR conversion failure on input disk is
expected

NOTE: Message MSDMSLIO201W is expected when ICHRRCDE is not available
on your system.

RPIRND003E - TTR conversion failed for block number 00000BDC on disk 0200.
DMSLIO201W The following names are undefined:
  ICHRRCDE
PRT FILE 0028 SENT FROM RACMAINT PRT WAS 0028 RECS 1586 CPY 001 A NOHOLD
NOKEEP

Processing of RACF.DATASET completes
Return code from 'IRRUT200' = 0
Ready; T=0.09/0.45 09:00:52
```

Figure 2-62 RACUT200 completion

When RACUT200 completes successfully, the verification information is created in a virtual printer spool file. With your printer spooled to your reader, you should be able to use the PEEK command to view the output. The size of this file varies, but it is larger than 200 records. Therefore, when you peek the file, use the option **PEEK (for *)** (Figure 2-63).

```

0028      PEEK      A0 V 132 Trunc=132 Size=1586 Line=0 Col=1 Alt=0
File (none) (none) from RACMAINT at 2NDLEVEL Format is PRINT.
* * * Top of File * * *
          **** INDEX BLOCK VERIFICATION
          **** SCAN OF INDEX BLOCKS AT LEVEL

BLOCK WITH RBA OF 000000017000

OFFSET  COMP.                      ENTRY NAME
COUNT
00E     000 RF
023     000 VMBATCH -E
040     000 VMCMD -RAC
05F     000 VMMDISK -MAINT.190
084     000 VMMDISK -MAINT.7B
0A8     000 VMMDISK -SYSB
0C8     000 VMMDISK -4OSASF40.3
0EE     000 VMPOSIX -U22
10D     000 VMRDR -RSCS
1= Help      2= Add line  3= Quit      4= Tab      5= Clocate      6= ?/Change
7= Backward  8= Forward   9= Receive  10= Rgtright 11= Spltjoin  12=
Cursor

```

Figure 2-63 PEEK output file

2.6.3 The RACF database unload utility (IRRDBU00)

The RACF database unload utility enables installations to create a sequential file from a RACF database. The sequential file can be used in several ways:

- ▶ Viewed directly
- ▶ Used as input for installation-written programs
- ▶ Manipulated with sort/merge utilities
- ▶ It can also be uploaded to a database manager (for example, SQL/DS) to process complex inquiries and create installation-tailored reports

The use of this program is documented in *RACF Security Server Administrator's Guide*, SC24-6142.

To use the RACF Database Unload Utility, execute the RACFDBU exec. To run this, you must perform the following steps:

1. Link and access RACFVM's 305 disk.
2. Obtain a link to the RACF database or backup database (if it is the active database the link must be in write mode and if you use the backup database you need only read mode).
3. Ensure that there is adequate free space on the output minidisk to contain the utility output file.

4. Run in a virtual machine with at least 20 MB of virtual storage.
5. IPL RACFVM's 490 disk.

Note: It is recommended that you run the utility against the RACF database using the NOLOCKINPUT parameter (which is the default). If you use this method it should not affect the performance of your RACF environment.

We found that it was easiest to run the RACFDBU exec from the virtual machine RACMAINT because the directory entry for this virtual machine already has links to all the required disks, it has 20 MB of virtual storage and it can IPL from address 490.

Issuing the command without any parameters displays the RACF Database Unload Utility Input Panel. When this utility executes, it will create two files on the output disk.

The files are:

- ▶ RACFDBU MESSAGES
- ▶ RACFDBU OUTPUT (by default, but you can change the file name or file type)

Issue the following command, as shown in Figure 2-64:

racfdbu

```

      RACF Database Unload Utility - Input Panel

. Status of input database:  1 = NOLOCKINPUT
                           2 = LOCKINPUT
                           3 = UNLOCKINPUT      1

. Virtual address of input database      200

. Virtual address(es) of input database (optional)  ____  ____  ____

. Virtual address of output minidisk      191

. Filename and filetype of output file      RACFDBU  OUTPUT

      PF1 = Help    PF2 = Execute    PF3 = Quit
      ENTER = Verify input fields

Enter CP/CMS Commands below:
=====>
```

Figure 2-64 Executing RACFDBU

When the RACFDBU exec starts, it defaults to NOLOCKINPUT, which is what you want to happen. If you chose to lock the input file, you can change that value, but for options 2 and 3, you must have the input disk in write mode.

After filling in the input database address and output minidisk values, you press PF2 (Figure 2-64). If the output file exist on the output disk, you will receive a message stating it

will be backed up prior to creating a new copy of the file, but you will have to press the PF2 key again to perform the back up of the file.

The processing takes a couple of minutes and when the utility finishes, you receive a validation message (Figure 2-65).

RACF Database Unload Utility - Input Panel

. Status of input database:

1 = NOLOCKINPUT
2 = LOCKINPUT
3 = UNLOCKINPUT

1

. Virtual address of input database

0200

. Virtual address(es) of input database (optional)

. Virtual address of output minidisk

0191

. Filename and filetype of output file

RACFDBU OUTPUT

RPIDBU037I RACFDBU is creating a R/W copy of 1 or more RACF databases that are currently R/O. Please wait.

RPIDBU033I Database unload completed successfully.

View the RACFDBU MESSAGES file for additional details.

PF1 = Help PF2 = Execute PF3 = Quit
ENTER = Verify input fields

Enter CP/CMS Commands below:

Figure 2-65 RACFDBU completion

As discussed earlier in this topic, the process creates two files on RACMAINT's 191 disk. If you want to examine either of these files, you need to IPL CMS into the virtual machine or access the 190 disk. For several releases of RACF for z/VM, the 490 disk was only shipped with a subset of the CMS code that is on MAINT's 190 disk. You are severely limited in the use of CMS commands that can be executed when you IPL the 490 disk (Figure 2-66).

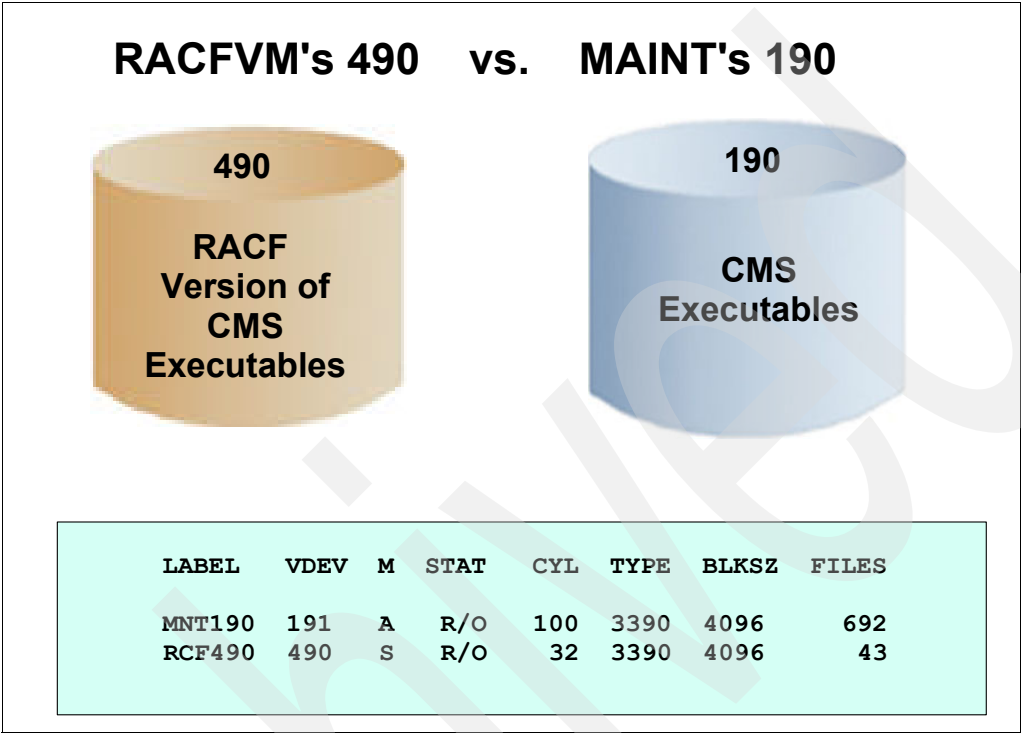


Figure 2-66 490 verses 190

You can now view the output file from the RACFDBU exec. First, look at the RACFDBU MESSAGE file (Figure 2-67). This file contains a summary of what was found in the database. We found this helpful so that we can determine how many groups, users, SECLABEL, VMMDISKS, VMRDRS, VMNODES, VMLANS, and so forth are defined in the RACF database for statistical reporting.

```

RACFDBU  MESSAGES A1  F 132  Trunc=132 Size=24 Line=1
====>
0 * * * Top of File * * *
1 IRR67010I Specified option: NOLOCKINPUT
2 IRR67013I Option in effect: NOLOCKINPUT
3 IRR67182I RACF.DATASET associated with DD INDD1 has been successfully op
4 IRR67007I The blocksize was taken from DD INDD1 and the data set was clo
5 IRR67150I Processing 1 RACF data set(s).
6 IRR67182I RACF.DATASET associated with DD INDD1 has been successfully op
7 IRR67163I INDD1 is a primary data set. All input data sets must be prima
8 IRR67093I Processing group profiles.
9 IRR67494I 12 group profile(s) have been unloaded.
10 IRR67093I Processing user profiles.
11 IRR67494I 124 user profile(s) have been unloaded.
12 IRR67093I Processing dataset profiles.
13 IRR67093I Processing general profiles.
14 IRR67494I 3 general SECLABEL profile(s) have been unloaded.
15 IRR67494I 330 general VMMDISK profile(s) have been unloaded.
16 IRR67494I 124 general VMRDR profile(s) have been unloaded.
17 IRR67494I 2 general VMCMD profile(s) have been unloaded.
18 IRR67494I 2 general VMNODE profile(s) have been unloaded.
19 IRR67494I 124 general VMBATCH profile(s) have been unloaded.
20 IRR67494I 16 general VMPOSIX profile(s) have been unloaded.
21 IRR67494I 2 general VMLAN profile(s) have been unloaded.

```

Figure 2-67 RACFDBU MESSAGES A

The RACFDBU OUTPUT file (Figure 2-68) actually contains the sequential output from the RACF database. Although the design rational of the IRRDBU00 utility is not diagnosis, this utility does provide some useful diagnostic capabilities. Because the IRRDBU00 utility must read every profile in the entire RACF database, it provides a side effect of validating profile data. While this is not a comprehensive validation of every field value, it is a validation of many lengths and count fields which are needed to read each profile successfully. The side effect might help to identify a profile in error.

RACFDBU OUTPUT A1 V 3096 Trunc=3096 Size=2162 Line=1									
====>									
0	*	*	*	Top of File	*	*	*		
1	0102	GADM	ADM	USE					
2	0100	GADM	SYS1	2007-07-14	IBMUSER	NONE		NO	
3	0130	GADM	0000000004						
4	0102	GBIN	BIN	USE					
5	0100	GBIN	SYS1	2007-07-14	IBMUSER	NONE		NO	
6	0130	GBIN	0000000002						
7	0102	GNOBODY	NOBODY	USE					
8	0100	GNOBODY	SYS1	2007-07-14	IBMUSER	NONE		NO	
9	0130	GNOBODY	2147483647						
10	0102	GSYS	SYS	USE					
11	0100	GSYS	SYS1	2007-07-14	IBMUSER	NONE		NO	
12	0130	GSYS	0000000003						
13	0100	MAIL	SYS1	2007-07-14	IBMUSER	NONE		NO	
14	0130	MAIL	0000000006						
15	0100	SECURITY	SYS1	2007-07-14	IBMUSER	NONE		NO	
16	0130	SECURITY	0000000007						
17	0102	SENDRSCS	GUMBY	USE					
18	0100	SENDRSCS	SYS1	2007-07-19	SYS1	NONE		NO	
19	0102	STAFF	DAEMON	USE					
20	0100	STAFF	SYS1	2007-07-14	IBMUSER	NONE		NO	
21	0130	STAFF	0000000001						

Figure 2-68 RACFDBU OUTPUT file

Archived

z/VM LDAP server

This chapter introduces z/VM Lightweight Directory Access Protocol (LDAP) and describes the installation and configuration process.

z/VM V5.3 introduces a z/VM LDAP server and client. It is a subcomponent of TCP/IP. The z/VM LDAP server has been adapted from the IBM Tivoli Directory Server for z/OS (delivered in z/OS V1.8).

z/VM LDAP can help to simplify administration tasks when a Linux farm is installed on z/VM.

The z/VM LDAP server helps administrators:

- ▶ Improve Linux logon security using RACF services to validate user and password.
- ▶ Reduce repetitive tasks, such as defining the same Linux user in multiple Linux images.
- ▶ Implement the RACF database sharing coupled with a z/VM LDAP server using native authentication permits to have a single point to administrate multiple z/VM and Linux servers.
- ▶ Gives a better business continuity solution when using multiple mainframes in conjunction with the hiperswap function.

3.1 LDAP terminology

When you start to use LDAP server and clients, it is helpful to understand some terms and processes.

LDAP defines the content of messages exchanged between an LDAP client and an LDAP server. The messages specify the operations that are requested by the client (for example search, modify, and delete), the responses from the server, and the format of data that is carried in the messages. LDAP messages are carried over TCP/IP, a connection-oriented protocol, so there are also operations to establish and disconnect a session between the client and server.

Figure 3-1 gives an overview of the main LDAP actors.

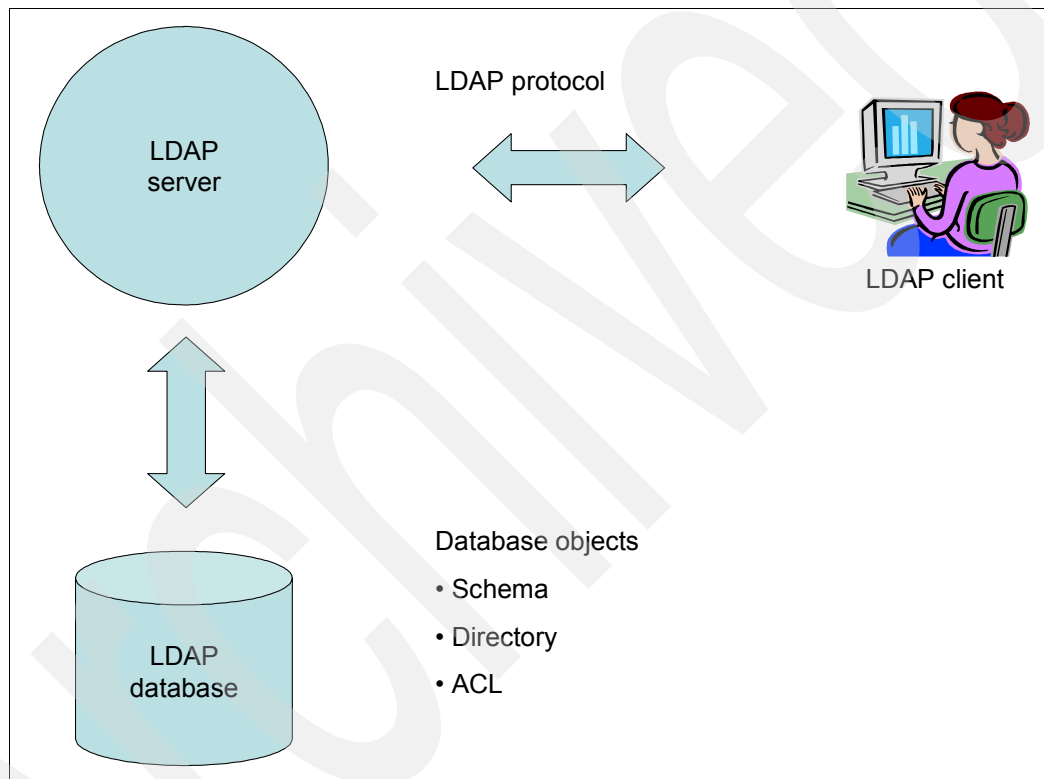


Figure 3-1 LDAP overview

However, for the designer of an LDAP directory, it is not so much the structure of the messages that are sent and received over the wire that are of interest. What is important is the logical model that is defined by these messages and data types, how the directory is organized, what operations are possible, how information is protected, and so forth.

The general interaction between an LDAP client and an LDAP server takes the following form:

1. The client establishes a session with an LDAP server. This is known as *binding* to the server. The client specifies the host name or IP address and TCP/IP port number where the LDAP server is listening.
2. The client can provide a user name and a password to authenticate properly with the server, or the client can establish an anonymous session with default access rights. The client and server can also establish a session that uses stronger security methods such as encryption of data.

3. The client then performs operations on directory data. LDAP offers both read and update capabilities. This allows directory information to be managed as well as queried. LDAP also supports searching the directory for data meeting arbitrary user-specified criteria. Searching is a very common operation in LDAP. A user can specify what part of the directory to search and what information to return. A search filter that uses boolean conditions specifies what directory data matches the search.
4. When the client is finished making requests, it closes the session with the server. This is also known as *unbinding*.

The philosophy of the LDAP API is to keep simple things simple. This means that adding directory support to existing applications can be done with low overhead. Because LDAP was originally intended as a lightweight alternative to DAP for accessing X.500 directories, it follows an X.500 model.

The directory stores and organizes data structures known as *entries*. A directory entry usually describes an object such as a person, device, a location, and so on, as shown in Figure 3-2.

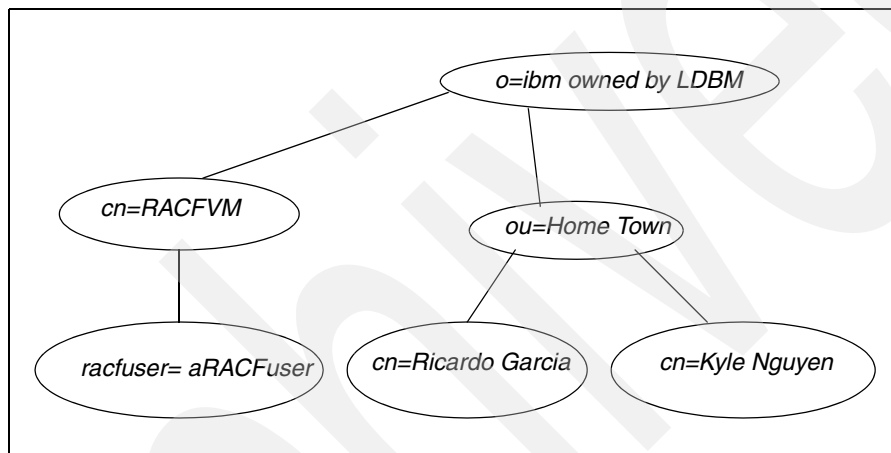


Figure 3-2 LDAP directory

Each entry has a name called a *distinguished name* (DN) that uniquely identifies it. The DN consists of a sequence of parts called relative distinguished names (RDNs), much like a file name consists of a path of directory names in many operating systems such as z/OS, z/VM, Linux, UNIX, and Windows®. The entries can be arranged into a hierarchical tree-like structure based on their distinguished names. This tree of directory entries is called the *Directory Information Tree* (DIT).

Each entry contains one or more attributes that describe the entry. Each attribute has a type and a value. For example, the directory entry for a person might have an attribute called *telephoneNumber*. The syntax of the *telephoneNumber* attribute would specify that a telephone number must be a string of numbers that can contain spaces and hyphens. The value of the attribute would be the person's telephone number, such as 800-555-1234.

A directory entry describes some object. An object class is a general description, sometimes called a *template*, of an object, as opposed to the description of a particular object. For instance, the object class *person* has a *surname* attribute, whereas the object describing John Smith has a *surname* attribute with the value Smith. The object classes that a directory server can store and the attributes they contain are described by schema. Schema define what object classes are allowed where in the directory, what attributes they must contain, what attributes are optional, and the syntax of each attribute. For example, a schema could define a *person* object class. The *person* schema might require that a person have a *surname*

attribute that is a character string, specify that a person entry can optionally have a telephoneNumber attribute that is a string of numbers with spaces and hyphens, and so on.

Object classes

An object class is an LDAP term that denotes the type of object being represented by a directory entry or record. Some typical object types are person, organization, organizational unit, domain component and groupOfNames. There are also object classes that define an object's relationship to other objects, such as object class top denotes that the object can have subordinate objects under it in a hierarchical tree structure. Note that some LDAP object classes can be combined, for example, an object class of organizational unit will most often also be simultaneously defined as a top object class because it will have entries beneath it.

Attributes

All the object class does is define the attributes, or types of data items contained in that type of object. Some examples of typical attributes are cn (common name), sn (surname), givenName, mail, UID, and userPassword. Just as the object classes are defined with unique OIDs, each attribute also has a unique OID number assigned to it. LDAP V3 attributes follow a notation similar (ASN.1) to object classes.

LDAP operations

LDAP defines operations for accessing and modifying directory entries such as:

- ▶ Binding and unbinding
- ▶ Searching for entries meeting user-specified criteria
- ▶ Adding an entry
- ▶ Deleting an entry
- ▶ Modifying an entry
- ▶ Modifying the distinguished name or relative distinguished name of an entry (move)
- ▶ Comparing an entry

Today, all the directory servers use the version 3 of LDAP.

LDAP V3 is documented in several IETF RFCs (2251, 2252, 2253, 2254, 2255, and 2256).

3.1.1 LDIF files

LDAP Data Interchange Format (LDIF) is used as input or output file during LDAP server commands. When an LDAP directory is loaded for the first time or when many entries have to be changed at once, it is not very convenient to change every single entry on a one-by-one basis. For this purpose, LDAP supports the LDAP Data Interchange Format (LDIF) that can be seen as a convenient, yet necessary, data management mechanism. It enables easy manipulation of mass amounts of data.

Figure 3-3 describes a LDAP server export followed by an import to an another LDAP server.

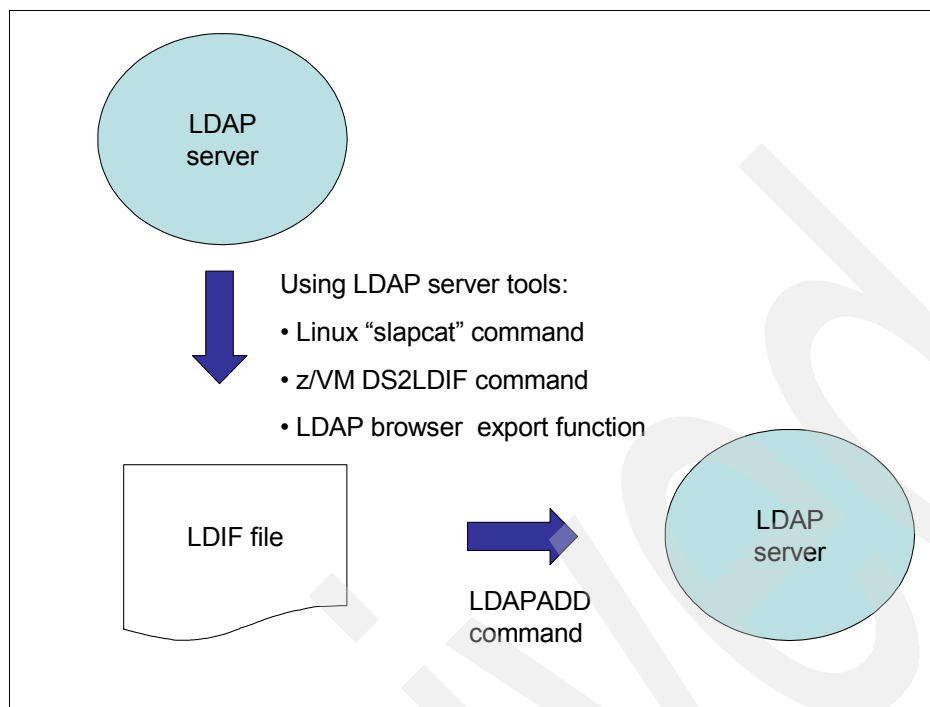


Figure 3-3 Export and import LDAP database

Depending of the client operating system and the server operating system, different tools are used to export or import LDAP server data. For example, on Linux, the main tool is slapcat, and it is available in the openldap2 package. On Windows, tools such as LDAP Browser/Editor by Jarek Gavor includes a database export/import tool. With z/VM, DS2LDIF tools can also be used to copy a LDAP directory to a CMS files.

3.2 z/VM LDAP

z/VM LDAP server is a ported version of the IBM Tivoli Directory Server for z/OS server. LDAP server uses z/VM OpenExtensions and BFS files.

3.2.1 LDAP client

The LDAP client runs in a CMS virtual machine. The LDAP client functions are available in the TCPMAINT 592 minidisk. Each z/VM user could inquire, insert, update, or delete data in LDAP database depending on its permissions. The LDAP client functions include:

- ▶ LDAPSRCH: Performs a search using filter parameters.
- ▶ LDAPMDFY: Updates entries. With -a option, data could be added if they do not exist.
- ▶ LDAPADD: Adds entries.
- ▶ LDAPDLET: Deletes entries.
- ▶ LDAPCMPR: Compares LDAP entries with a file or from your standard input.

3.2.2 z/VM LDAP back end services

The LDAP server could be implemented with different back end services. We describe those in this section

LDBM back end service

In Figure 3-4, the LDAP server stores the database in the file pool controlled by the BFS server. When there is only the LDBM back end active, the Linux passwords and users are stored in the LDBM database. The password encryption (if it is activated) is made by LDAP server. VM users are administrated separately.

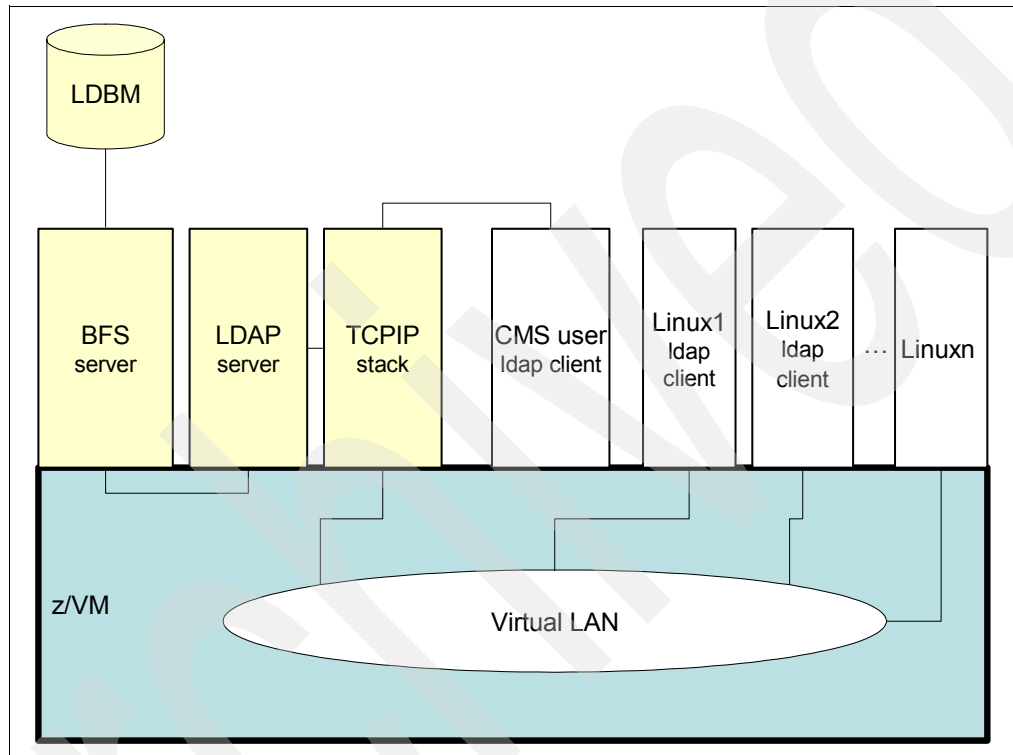


Figure 3-4 LDBM backend

GDBM back end service

GDBM back end service is used to log changes made to LDBM and schema entries.

SDBM back end service

Secured Database Manager (SDBM) gives access to the data stored in the RACF database. Its main function is the directory authentication (bind) based on the user ID and the password.

Depending on the permissions, the user could also perform some change in the RACF database like ADDUSER, ADDGROUP, ALTUSER, ALTGROUP, DELUSER, DELGROUP, CONNECT, REMOVE, SEARCH, and so forth.

When only SDBM is used, every Linux user is defined in each Linux image. SDBM back end service is not able to create the user directory and to define the Linux shell used during the logon because this information could not be stored in the RACF database.

LDAP commands such as LDAPSRCH, LDAPADD, LDAPMDFY can change a user logon password using the following command:

```
LDAPSRCH -p host -w oldpass/newpass ...
```

Figure 3-5 shows the main z/VM servers used.

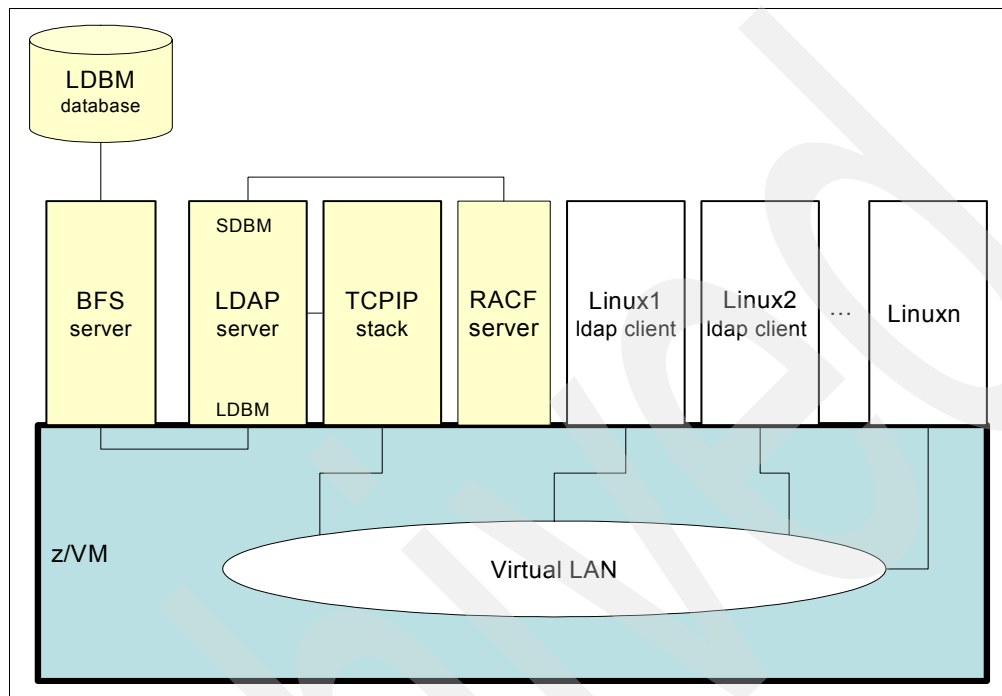


Figure 3-5 SDBM back end service

ICTX back end service

Remoteable service support (ICTX) enable audit and authorization requests to be resolved using RACF Security Server for z/VM. The remote application connects to the z/VM LDAP server using XML-binary Optimized Packaging (XOP) packets.

3.2.3 Native authentication

LDBM native authentication uses RACF to qualify the user ID and password that is received. The LDBM database owns complementary informations such as UID, shell used, home directory, and others. This configuration allows the best centralization and security control.

A user can have the same Linux and z/VM password while the Linux logon user ID could be the same or different.

User passwords can be changed using a LDAP command from a CMS user, as follows:

```
LDAPSRCH -h 9.12.4.191 -D "cn=user1,ou=Home Town,o=ibm" -w  
"oldpassword/newpassword" -s base -b "o=ibm"
```

Figure 3-6 describe the main z/VM users.

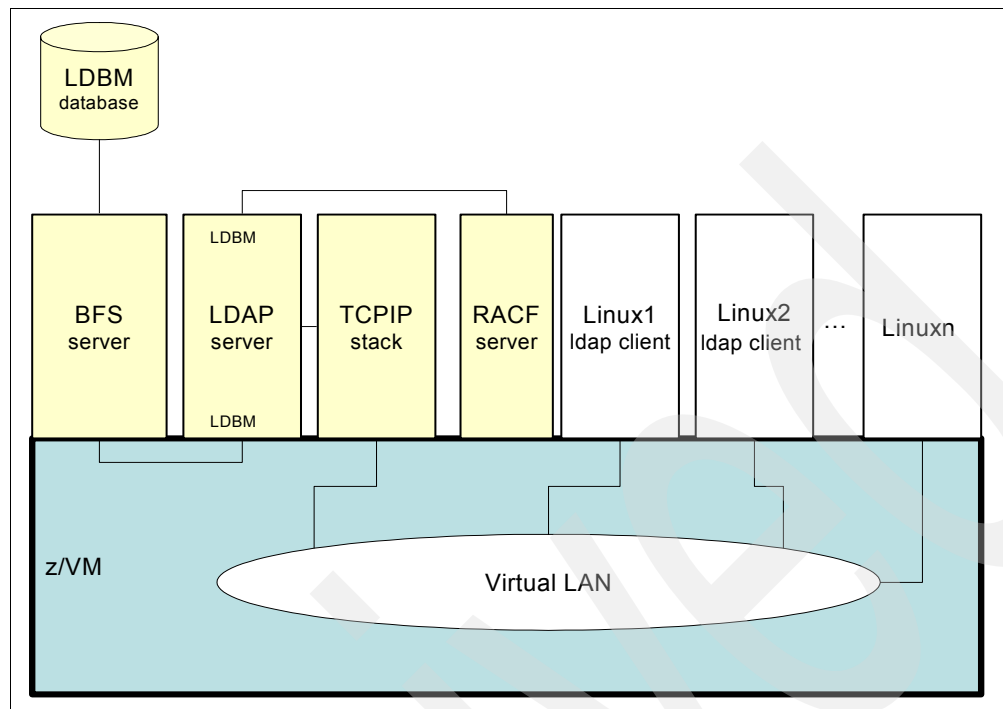


Figure 3-6 Native authentication

3.2.4 Multiple back end services

A LDAP server can use multiple back end services (LDBM, SDBM, and GDBM). If your company contains multiple organizations, multiple LDBM databases could be defined.

A LDAP server can have only one SDBM back end service and one GDBM back end service.

Multiple LDBM back end instances can be defined.

3.2.5 Multiple servers

You can define multiple LDAP servers on z/VM, but the database cannot be shared. To mirror the LDAP database, you need to activate the replication function.

Replication

There are two ways to replicate LDAP data:

- ▶ *Peer-to-peer*, where data available in each LDAP server are merged. A minimal conflict resolution is done to try to give the administrators non overlapping update possibilities.
- ▶ *Read only replication*, where the master server will replicate its data to slave servers. This permits to increase the security level because LDAP servers can read only the data and updates are centralized.

3.3 Installing z/VM LDAP server

In our installation, we choose to implement the native authentication for the z/VM LDAP because this option provides more facilities for the Linux servers.

To implement the LDAP server, the following products are mandatory:

- ▶ z/VM 530
- ▶ RACF (see the installation description in Chapter 2, “RACF feature of z/VM” on page 21)
- ▶ TCP/IP (in our case, we choose to install a new stack as described in 3.3.1, “Implementing a new TCP/IP stack” on page 120)
- ▶ LDAP server (see 3.3.2, “Creating the LDAP server” on page 123)
- ▶ BFS server (see 3.3.3, “Creating a BFS file pool VMSEVL” on page 126)

DirMaint is optional if it is already installed.

To minimize the impact of the new installation, we decided to create a new TCP/IP stack and a new BFS server.

Figure 3-7 describes the main z/VM user IDs needed for a z/VM LDAP server.

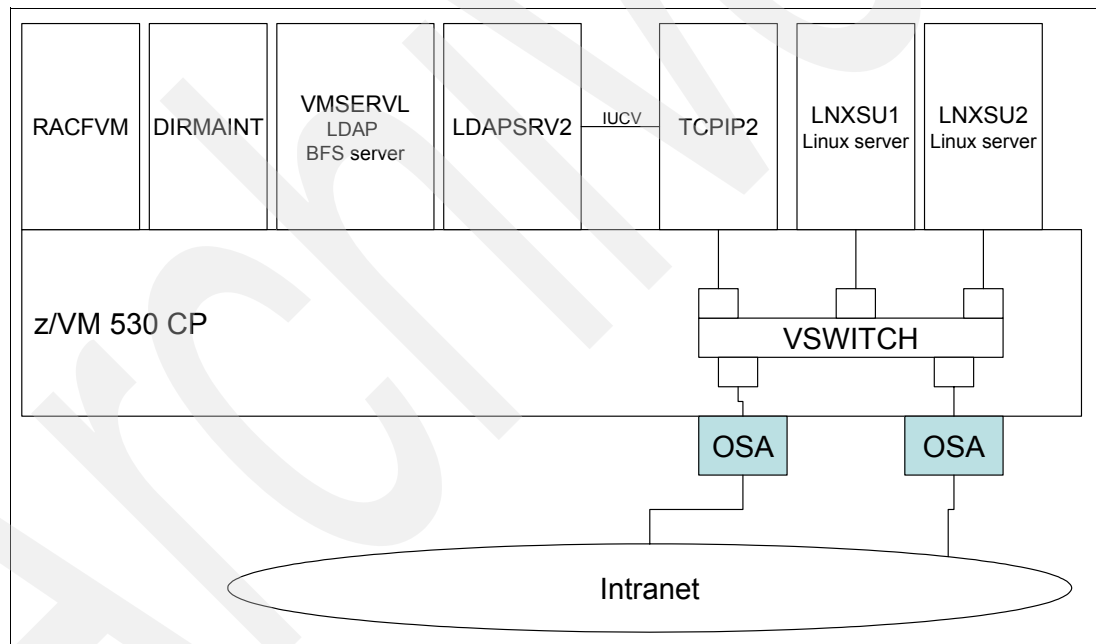


Figure 3-7 z/VM user IDs used with LDAP server

3.3.1 Implementing a new TCP/IP stack

z/VM TCP/IP is usually running in a production configuration. In our case, we decided to define a new TCP/IP stack to avoid interferences with the production side. The new TCP/IP stack is installed in the TCPIP2 user, and its HOME ADDRESS is 9.12.4.191, as shown in Figure 3-8.

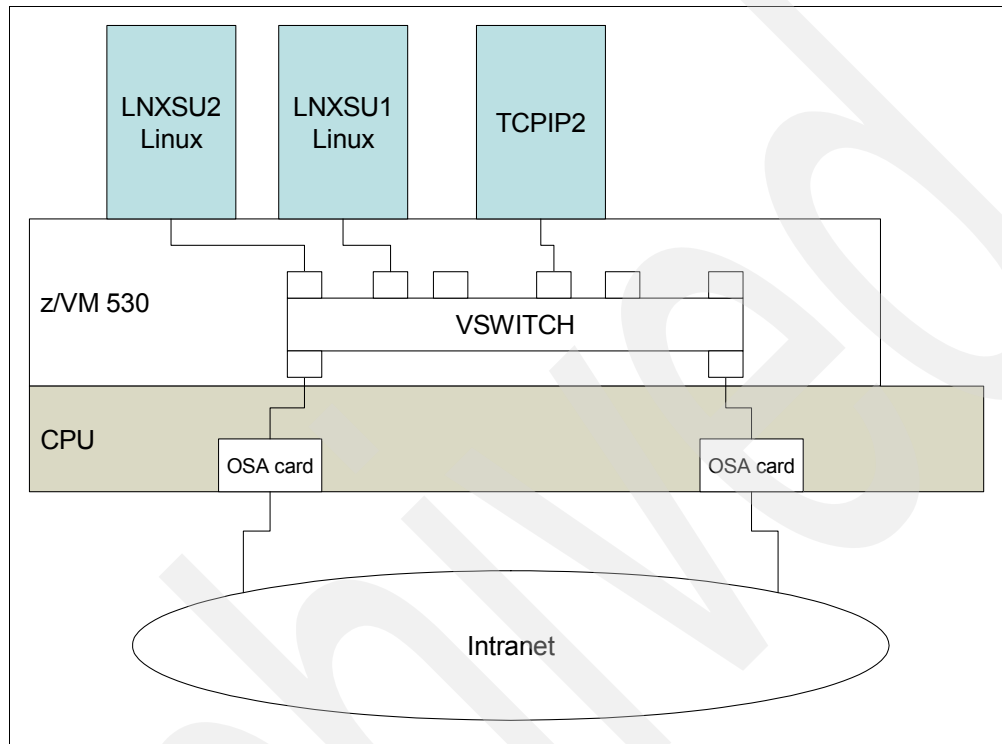


Figure 3-8 z/VM network configuration

To define the TCP/IP stack, follow these steps:

1. Define the VM TCPIP2 user.

Figure 3-9 provides a TCPIP2 directory sample. You can use XEDIT to create the TCPIP2 DIRECT file, and you can use the DirMaint to define TCPIP2 in the user directory with the command **DIRM ADD TCPIP2**.

```
USER TCPIP2 TEST 32M 128M ABG
INCLUDE TCPCMSU
OPTION QUICKDSP SVMSTAT MAXCONN 1024 DIAG98 APPLMON
SHARE RELATIVE 3000
IUCV ALLOW
IUCV ANY PRIORITY
IUCV *CCS PRIORITY MSGLIMIT 255
IUCV *VSWITCH MSGLIMIT 65535
LINK 5VMTCP30 491 491 RR
LINK 5VMTCP30 492 492 RR
LINK TCPMAINT 591 591 RR
LINK TCPMAINT 592 592 RR
LINK TCPMAINT 198 198 RR
```

Figure 3-9 Sample TCPIP2 directory

You can also add a 191 minidisk to the TCPIP2 user with the following command:

```
DIRM FOR TCPIP2 AMD 191 XXXX AUTOG 10 SYSTEM MR
```

2. Provide the RACF permissions.

If you try to log on with the TCPIP2 user, you receive RACF violation messages such as those shown in Figure 3-10 because the user is not allowed to access the TCP minidisks.

```
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO TCPIP2.191
HCPLNM298E TCPIP2 0191 not linked; request denied
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO 5VMTCP30.491
HCPLNM298E TCPIP2 0491 not linked; request denied
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO 5VMTCP30.492
HCPLNM298E TCPIP2 0492 not linked; request denied
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO TCPMAINT.591
HCPLNM298E TCPIP2 0591 not linked; request denied
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO TCPMAINT.198
HCPLNM298E TCPIP2 0198 not linked; request denied
```

Figure 3-10 RACF messages received during log on of TCPIP2

You need to provide RACF access to these minidisks for the TCPIP2 user by issuing the following commands from a user with RACF administrator authority:

```
RAC PERMIT TCPIP2.191 CLASS(VMMDISK) ACCESS(ALTER) ID(TCPIP2)
RAC PERMIT 5VMTCP30.491 CLASS(VMMDISK) ACCESS(READ) ID(TCPIP2)
RAC PERMIT 5VMTCP30.492 CLASS(VMMDISK) ACCESS(READ) ID(TCPIP2)
RAC PERMIT TCPMAINT.591 CLASS(VMMDISK) ACCESS(READ) ID(TCPIP2)
RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ACCESS(READ) ID(TCPIP2)
```

TCPIP2 also needs an access to the VSWITCH:

```
RAC RDEFINE VMLAN SYSTEM.VSWITCH1 UACC(NONE)
RAC PERMIT SYSTEM.VSWITCH1 ACC(UPDATE) CLASS(VMLAN) ID(TCPIP2)
```

Now, TCPIP2 is able to access its minidisks and the VSWITCH.

3. Copy PROFILE EXEC in TCPIP2.

Log on with the TCPIP2 user and format the TCPIP2 191 minidisk. When the task is completed, copy the PROFILE EXEC from the TCP/IP user191 minidisk.

4. Create the new configuration files for TCPIP2.

All the configuration files for a TCP/IP stack are usually stored in the TCPMAINT 198 minidisk.

To create the configuration file for TCPIP2, log on first with the TCPMAINT user ID. Then, create a new file called TCPIP2 TCP/IP as shown in Figure 3-11. This file contains the TCP/IP network definitions.

```
; -----  
; - PROFILE TCPIP for TCPIP2  
; -----  
ASSORTEDPARMS  
ENDASSORTEDPARMS  
;  
OBEY  
OPERATOR TCPMAINT MAINT  
ENDOBEY  
;  
PORT  
    23    TCP INTCLIEN          ; TELNET Server  
;  
DEVICE DEVC202  OSD C202  NONROUTER  
LINK OSAC202L QDIOETHERNET DEVC202  MTU 4096  
;  
HOME  
9.12.4.191 255.255.252.0 OSAC202L  
;  
GATEWAY  
; Network      Subnet      First      Link      MTU  
; Address      Mask        Hop        Name      Size  
; -----  
DEFAULTNET          9.12.4.1      OSAC202L      4096  
;  
START DEVC202
```

Figure 3-11 TCPIP2 TCP/IP file

TCP/IP also needs a file, in our case TCPIP2 DTCPARMS, that defines the virtual and real devices that are used by the stack, as shown in Figure 3-12.

```
.*****  
.* SYSTEM DTCPARMS  
.*  
.******  
:nick.TCPIP2 :type.server  
              :class.stack  
              :vnic.C202 SYSTEM VSWITCH1
```

Figure 3-12 TCPIP2 DTCPARMS file

5. Start TCPIP2.

Now that the base configuration files are available, you can start TCPIP2. Log off and then log on the TCPIP2 user and check that there are no errors during the start time. When the TELNET server is ready, a TN3270 session can be open to the 9.12.4.191 address.

3.3.2 Creating the LDAP server

LDAP server is part of the TCP/IP and runs in a CMS user ID. The standard user ID name is *LDAPSRV*. In our configuration, we used *LDAPSRV2*.

To define the LDAP server, follow these steps:

1. Define the LDAPSRV2 user ID.
 - a. Log on as DIRMAINT administrator and create a file to define the new LDAPSRV2 user as shown in Figure 3-13.

```
USER LDAPSRV2 LDAPSRV 128M 256M BG
INCLUDE TCPCMSU
POSIXINFO UID 5 GID 0
IUCV ALLOW
IUCV RACFVM PRIORITY MSGLIMIT 255
OPTION QUICKDSP SVMSTAT
LINK 5VMTCP30 491 491 RR
LINK 5VMTCP30 492 492 RR
LINK TCPMAINT 591 591 RR
LINK TCPMAINT 592 592 RR
LINK TCPMAINT 198 198 RR
```

Figure 3-13 LDAPSRV2 DIRECT file

Note that the posix information could have been kept in RACF in the OVM segment.

- b. Add the user ID to the VM directory with the DirMaint command **DIRM ADD LDAPSRV2**.
 - c. Add a 191 minidisk to LDAPSRV2 user ID with the following command:

```
DIRM FOR LDAPSRV2 AMD 191 XXXX AUTOG 10 SYSTEM MR
```
 - d. After you define the user ID, you need to provide RACF authorization to the resources it is going to access. Perform the following commands:

```
RAC PERMIT LDAPSRV2.191 CLASS(VMMDISK) ID(LDAPSRV2) ACCESS(ALTER)
RAC PERMIT 5VMTCP30.491 CLASS(VMMDISK) ID(LDAPSRV2) ACCESS(READ)
RAC PERMIT 5VMTCP30.492 CLASS(VMMDISK) ID(LDAPSRV2) ACCESS(READ)
RAC PERMIT TCPMAINT.591 CLASS(VMMDISK) ID(LDAPSRV2) ACCESS(READ)
RAC PERMIT TCPMAINT.198 CLASS(VMMDISK) ID(LDAPSRV2) ACCESS(READ)
```
 - e. Now you can log off and log on with the LDAPSRV2 user ID. Format its 191 minidisk and copy the PROFILE EXEC from the LDAPSRV user191 minidisk.

- f. Because you are sharing TCPMAINT 591, 592 and 198, you need a private TCP/IP DATA file. To do so, copy it from the TCPMAINT 592 minidisk to LDAPSRV2 191 and modify the TCPIPUSERID, the HOSTNAME, and the NSINTERADDR as shown in Figure 3-14.

```
; -----  
; - TCPIP DATA created by DTCIPWIZ EXEC on 20 Jun 2007  
; - Configuration program run by MAINT at 14:26:47  
; -----  
; %%File Origin Indicator - DO NOT REMOVE OR ALTER the next line%%  
; %%TCPIP%%TCPIP%%SDATA%%  
; -----  
TCPIPUSERID TCPIP2  
; -----  
HOSTNAME VMLINUX5  
; -----  
DOMAINORIGIN ITS0.IBM.COM  
; -----  
NSINTERADDR 9.12.6.7  
; -----
```

Figure 3-14 partial TCP/IP DATA

2. Modify TCPIP2 configuration files.

Before activating the new LDAP server, you have to change the TCPIP2 configuration files to add the PORT, OBEY, AUTOLOG statements as shown in Figure 3-15.

```
; -----  
; - PROFILE TCPIP for TCPIP2  
; -----  
ASSORTEDPARMS  
ENDASSORTEDPARMS  
;  
OBEY  
OPERATOR TCPMAINT MAINT LDAPSRV2  
ENDOBEY  
;  
AUTOLOG  
    LDAPSRV2  0          ; LDAP Server  
ENDAUTOLOG  
;  
PORT  
    23  TCP INTCLIEN      ; TELNET Server  
    389 TCP LDAPSRV2      ; LDAP Server  
    636 TCP LDAPSRV2 NOAUTOLOG ; LDAP Server (Secure)  
;  
DEVICE DEVC202 OSD C202 NONROUTER  
LINK OSAC202L QDIOETHERNET DEVC202 MTU 4096  
;  
HOME  
9.12.4.191 255.255.252.0 OSAC202L  
;  
GATEWAY  
; Network      Subnet      First      Link      MTU  
; Address      Mask        Hop        Name      Size  
; -----  
DEFAULTNET          9.12.4.1      OSAC202L    4096  
;  
START DEVC202
```

Figure 3-15 TCPIP2 CONFIG

You also need to create a LDAPSRV2 DTCPARMS to contain LDAP server parameters as shown in Figure 3-16. In this file, the file pool used to store the LDAP data is defined. A new file pool, VMSYSL, will be used. Refer to 3.3.3, “Creating a BFS file pool VMSERVL” on page 126 for information about how to define it.

```

*****
.* LDAPSRV2 DTCPARMS
*****
:nick.LDAPSRV2 :type.server :class.ldap
.*-----
.* Note: The Byte File System directory path name specified for the
.*       'var/ldap/' mount purposely omits the file space ID portion of
.*       this path name. This has been done to allow for substitution
.*       of the subject server user ID when this entry is processed.
.*-----
:nick.ldap      :type.class
                :name.LDAP daemon
                :command.LDAPSRV
                :runtime.C
                :memory.128M
                :mixedcaseparms.YES
                :mount. ../VMBFS:VMSYSL:ROOT/  /
                :ESM_Enable.NO
                :ESM_Racroute.LDAPESM

```

Figure 3-16 LDAPSRV2 DTCPARMS

You need to restart the TCPIP2 user to refresh the new definitions by issuing the **FORCE TCPIP2** and **XAUTOLOG TCPIP2** commands.

3.3.3 Creating a BFS file pool VMSERVL

The LDAP server needs a Byte File System (BFS) to store his work files and the LDAP databases. For security reasons, it is suggested to create a specific BFS for the LDAP server instead of using the VMSYS file pool.

To define the BFS file pool, follow these steps:

1. Define the VMSEVL user.

To define the new VMSEVL user, use the VMSEVU user directory as a sample. Figure 3-17 shows the VMSEVL DIRECT file.

- a. Log on with a DirMaint administrator to create the VMSEVL DIRECT file.

```
USER VMSEVL TEST      32M 32M BG
  INCLUDE IBMDFLT
  ACCOUNT 1 VMSEVU
  MACH XC
  OPTION MAXCONN 2000 NOMDCFS APPLMON ACCT QUICKDSP SVMSTAT
  SHARE REL 1500
  XCONFIG ADDRSPACE MAXNUMBER 100 TOTSIZE 8192G SHARE
  XCONFIG ACCESSLIST ALSIZE 1022
  IUCV ALLOW
  IUCV *IDENT RESANY GLOBAL
  IPL CMS
  POSIXOPT SETIDS ALLOW
  CONSOLE 009 3215 T MAINT
  LINK MAINT 193 193 RR
  MDISK 191 3390 1852 003 LX5W02 MR RSERVER WSERVER
  MDISK 302 3390 1855 014 LX5W02 WR RLOG1 WLOG1
  MINIOPT NOMDC
  MDISK 301 3390 1869 009 LX5W02 WR RCONTROL WCONTROL
  MINIOPT NOMDC
  MDISK 303 3390 1878 014 LX5W02 WR RLOG2 WLOG2
  MINIOPT NOMDC
  MDISK 304 3390 1892 003 LX5W02 WR RCATALOG WCATALOG
  MDISK 305 3390 1895 100 LX5W02 WR RDATA WDATA
```

Figure 3-17 VMSEVL user ID

- b. Use the DirMaint command **DIRM ADD VMSEVL** to create the new VM user.
- c. Log on with a user ID with RACF administrator authority to provide the correct access to VMSEVL:

```
RAC PERMIT VMSEVL.191 CLASS(VMDISK) ID(VMSEVL) ACCESS(ALTER)
RAC PERMIT VMSEVL.302 CLASS(VMDISK) ID(VMSEVL) ACCESS(ALTER)
RAC PERMIT VMSEVL.301 CLASS(VMDISK) ID(VMSEVL) ACCESS(ALTER)
RAC PERMIT VMSEVL.303 CLASS(VMDISK) ID(VMSEVL) ACCESS(ALTER)
RAC PERMIT VMSEVL.304 CLASS(VMDISK) ID(VMSEVL) ACCESS(ALTER)
RAC PERMIT VMSEVL.305 CLASS(VMDISK) ID(VMSEVL) ACCESS(ALTER)
```

- d. Log on with the VMSEVL user and **FORMAT** each VMSEVL minidisks.

- e. Create a VMSEVL DMSPARMS file on VMSEVL 191 minidisk as shown in Figure 3-18.

```
ADMIN MAINT JIGUET
NOBACKUP
SAVESEGID CMSFILES
FILEPOOLID VMSYSL
USERS 100
NODFSMS
```

Figure 3-18 VMSEVL DMSPARMS

- f. Create a PROFILE EXEC on VMSEVL 191 minidisk as shown in Figure 3-19.

```
/* PROFILE EXEC for Shared File System Server */
'CP SET MSG ON'
'CP SET RUN ON'
'SET AUTOREAD OFF'
'ACCESS 193 B'
'EXEC FILESERV START'
```

Figure 3-19 PROFILE EXEC

- g. Build the file pool using the **FILESERV GENERATE** command.
- h. When the **GENERATE** process completed, start the server by issuing the **IPL CMS** command.
- i. Perform a CP **DISCONNECT**.
- j. From a VMSEVL administrator user, enroll the LDAPSRV2 user by issuing the following command:

```
ENROLL USER LDAPSRV2 VMSYSL (BFS USER LDAPSRV2 BLOCKS 16000
```

Note: File pool administrator is defined in VMSEVL DMSPARMS.

2. Build the BFS inside VMSEVL.

A BFS file pool is usable with OpenExtensions as soon as a directory is defined in the file pool space. The LOADBFS EXEC is used to create the directory and to load the files in it. Sample files are available with CMS to create the directory and to load the OpenExtensions shell and utilities. TCP/IP have sample files to load LDAP base files.

- a. Log on as one of the BFS administrators described in VMSEVL DMSPARMS and perform the following commands:

```
LINK MAINT 193 193 RR
LINK MAINT 400 400 RR
LINK 5VMTCP30 491 491 RR
LINK 5VMTCP30 492 492 RR
ACCESS 193 B
ACCESS 400 C
ACCESS 491 D
ACCESS 492 E
COPY BFS LOADBFS * = = A
COPY SHELL LOADBFS * = = A
COPY LDAPSRV LOADBFS* = = A
```

- b. Update the file pool name from VMSYS to VMSYSL in all the three files (BFS LOADBFS, SHELL LOADBFS, and LDAPSRV LOADBFS).
- c. Create the BFS directory by issuing the LOADBFS BFS command.
- d. Load the OpenExtensions shell and utilities with the LOADBFS SHELL command.
- e. Install the LDAP code with the LOADBFS LDAPSRV command.

The permission given on /var/ldap directory needs to be verified. To do this, you need a CMS user defined with UID 0. This value is defined in the POSIXINFO directory parameter as POSIXINFO UID 0 GNAME system.

- f. Log on the user ID with UID 0 and issue the following commands:

```
openvm mount ../VMBFS:VMSYSL:ROOT/ /
openvm shell
cd var
ls -al command returns:
drwxr-xr-x  1 jiguet  system      0 Jul 13 20:23 ldap
drwxr-xr-x  1 jiguet  system      0 Jul 13 19:48 spool
```

LDAPSRV2 needs to be able to create files on /var/ldap. In our configuration, LDAPSRV2 is a member of the system group.

The `ls -al` command shows that the `w` permission is missing to the system group. You can change the value using the `chmod` command:

```
chmod 775 /var/ldap
```

Then, check whether it is OK by issuing the `ls -al` command as follows:

```
drwxrwxr-x  1 jiguet  system      0 Jul 13 20:23 ldap
drwxr-xr-x  1 jiguet  system      0 Jul 13 19:48 spool
```

3.3.4 Configuring the LDAP server

The LDAP server backend uses RACF to grant the logon user ID and password. You need to configure the LDAP server to exchange data with RACF server. Permission is given in LDAPSRV2 DTCPARMS file.

To configure the LDAP server, follow these steps:

1. LDAP configuration files.

- a. Log on with the TCPMAINT user ID and check that the ESM_Enable parameter in LDAPSRV2 DTCPARMS is correctly set as shown in Figure 3-20.

```
.*****
.* LDAPSRV2 DTCPARMS
.******
:nick.LDAPSRV2 :type.server :class.ldap
.*-----
.* Note: The Byte File System directory path name specified for the
.*       'var/ldap/' mount purposely omits the file space ID portion of
.*       this path name. This has been done to allow for substitution
.*       of the subject server user ID when this entry is processed.
.*-----
:nick.ldap      :type.class
                :name.LDAP daemon
                :command.LDAPSRV
                :runtime.C
                :memory.128M
                :mixedcaseparms.YES
                :mount. ../VMBFS:VMSYSL:ROOT/  /
                :ESM_Enable.YES
                :ESM_Racroute.LDAPESM
```

Figure 3-20 LDAPSRV2 DTCPARMS

- b. Edit the DS CONF file that contains the main LDAP parameters and modify the values as shown in Figure 3-21.

```
adminDN "cn=ldapadm2"
adminPW secret
allowAnonymousBinds on
commThreads 10
listen ldap://:389
logfile /var/ldap/gldlog.output
maxConnections 65535
sendV3StringsOverV2As UTF-8
sizeLimit 500
timeLimit 3600
validateIncomingV2strings on
database LDBM GLDBLD31
suffix "o=ibm"
```

Figure 3-21 DS CONF file

If you plan to use the SDBM back end, you need to uncomment the line database SDBM GLDBSD31 and the line suffix "cn=RACFVM" in the DS CONF file.

The ldapadm2 user specified in the adminDN parameter should exist in RACF. Otherwise, you will not be able to create your LDAP directory. If you want to use SDBM, uncomment the first line adminDN and comment the second line. The user specified in the racfid=LDAPADM2 parameter should be a known RACF user ID.

Create the LDAPADM2 user in RACF by logging on with a user with RACF administrator authority and issues the following command:

```
RAC ADDUSER LDAPADM2 PASSWORD(PASS)
```

Note: You will change the user password during the LDAP test verification.

- c. The DS ENVVARS file is the another configuration file used by the LDAP server that needs to be customized. See Figure 3-22 for a sample.

```
#
NLSPATH=/usr/lib/nls/msg/%L/%N
#
LANG=En_US.IBM-1047
#
#LIBPATH=/usr/lib
#
#TZ=GMT0
#
#=====
#  END OF CONFIGURATION FILE
#=====
```

Figure 3-22 DS ENVVARS file

- d. From a RACF administrator, authorize LDAPSRV2 to open IUCV session with the RACF server.

```
RAC RDEFINE FACILITY ICHCONN UACC(NONE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(LDAPSRV2) ACCESS(UPDATE)
RAC SETR CLASS(FACILITY)
```

- e. If, during LDAPSRV2 logon, you receive the message RPIMGR032E YOU ARE NOT AUTHORIZED TO SPOOL TO TCPMAINT, run the following commands from a RACF administrator to fix the problem:

```
RAC RDEFINE VMXEVENT EVENTS1
RAC RALTER VMXEVENT EVENTS1 ADDMEM(TAG/NOCTL)
RAC RALTER VMXEVENT EVENTS1 ADDMEM(TRANSFER.D/NOCTL)
RAC RALTER VMXEVENT EVENTS1 ADDMEM(TRANSFER.G/NOCTL)
RAC SETEVENT REFRESH EVENTS1
```

- f. Log on with the LDAPSRV2 user ID and check if there are error messages.

Example 3-1 shows the LDAP server start up messages.

Example 3-1 LDAP server console messages

```
DTCRUN1022I Console log will be sent to default owner ID: TCPMAINT
RPICMS016I USER/RACF VM Racroute communication path is established.
DTCRUN1027I Server will use TcpipUserid TCPIP2
DTCRUN1021R To cancel LDAP daemon startup, type any non-blank character and
           press ENTER. To continue startup, just press ENTER.

DTCRUN1011I Server started at 08:58:07 on 21 Aug 2007 (Tuesday)
DTCRUN1011I Running "LDAPSRV"
DTCLDP2106I Debug setting: 0
DTCLDP2107I Using server configuration file: DS CONF D2
DTCLDP2107I Using environment variable file: DS ENVVARS D1
```

DTCLDP2107I Using server module: GLDSRV31 MODULE E2
 070821 12:58:08.126346 GLD1003I LDAP server is starting.
 070821 12:58:08.126687 GLD1001I LDAP server version 3.18, Service level 0A19849
 Build date Mar 22 2007, Time 22:58:27.
 08:58:08 * MSG FROM TCPIP2 : Not forcing you because you're connected
 070821 12:58:08.126811 GLD1002I LDAP runtime version 3.18, Service level
 0A19849
 , Build date Mar 22 2007, Time 23:25:52.
 070821 12:58:08.155760 GLD1023I Processing configuration file //DD:CONFIG.
 070821 12:58:08.156076 GLD1024I Configuration file //DD:CONFIG processed.

Server Configuration
 adminDN: cn=ldapadm2
 adminPW: *configured*
 allowAnonymousBinds: on
 armName: GLDSRVR
 audit 1: off
 commThreads: 10
 db2Terminate: recover
 dnCacheSize: 1000
 idleConnectionTimeout: 0
 listen 1: ldap://:389
 logfile: /var/ldap/gldlog.output
 maxConnections: 65535
 pcIdleConnectionTimeout: 0
 pcThreads: 10
 schemaPath: /var/ldap/schema
 schemaReplaceByValue: on
 securityLabel: off
 sendV3StringsOverV2As: UTF-8
 serverEtherAddr: 402084006A3A
 serverSysplexGroup: undefined
 sizeLimit: 500
 srvStartUpError: terminate
 supportKrb5: off
 tcpTerminate: recover
 timeLimit: 3600
 validateIncomingV2Strings: on

database LDBM GLDBLD31 LDBM-0001
 changeLoggingParticipant: on
 commitCheckpointEntries: 10000
 commitCheckpointTOD: 00:00
 databaseDirectory: /var/ldap/ldbm
 extendedGroupSearching: off
 fileTerminate: recover
 filterCacheBypassLimit: 100
 filterCacheSize: 5000
 krbIdentityMap: off
 multiServer: off
 nativeAuthSubtree: all
 nativeUpdateAllowed: off
 persistentSearch: off
 pwEncryption: none
 pwCryptCompat: on

```
readOnly: off
secretEncryption: none
sizeLimit: 500
suffix 1: o=ibm
timeLimit: 3600
useNativeAuth: off
070821 12:58:40.726975 GLD1074W Maximum client connections changed from 65535
to
65523.
070821 12:58:53.668618 GLD1004I LDAP server is ready for requests.
070821 12:58:53.709227 GLD1059I Listening for requests on 9.12.4.191 port 389.
070821 12:58:53.710868 GLD1059I Listening for requests on 127.0.0.1 port 389.
```

2. Access the LDAP server.

When the LDAP server is started, you can perform some check using an LDAP client to verify the LDAP is working correctly.

Note: The first time a client connects to RACF, it has to change the logon password.

3. To verify the functionality, you can use the following command from a CMS LDAP client:

```
LDAPSRCH -h 9.12.4.191 -D "cn=ldapadm2" -w secret -s base -b "o=ibm
objectclass=*"
```

You should receive the following from the LDAP search:

```
ldap_search: No such object
ldap_search: additional info: R004071 DN 'o=ibm' does not exist
(ldbm_process_request)
```

This message is received because the LDAP database is empty.

4. Load schemes in the LDBM database.

The new LDAP server is running, but no data has been loaded yet. Before using an LDBM database, you need to load the schemes.

z/VM TCP/IP provides some schema to populate the LDAP database (USRSCHM LDIF and IBMSCHM LDIF).

Note: In our configuration, we detected compatibility problems while loading the schemes with **ldapmodify** from Linux or from CMS. To avoid these problems:

- ▶ Transfer carefully the LDIF files using FTP in ASCII mode. Other methods give truncated records (for example cut and past or tn3270 file transfer).
- ▶ The CMS LDAPMDFY gives trouble when a minus sign is used between each replace or add block. Replacing the minus sign by a blank character solves the error.

Load the schema with the LDAPMDFY command.

5. Log on with your preferred user ID:

```
LINK TCPMAINT 591 591 RR
ACCESS 591 K
LINK TCPMAINT 592 592 RR
ACCESS 592 L
```

6. Load the schema in the LDBM database:

```
LDAPMDFY -h 9.12.4.191 -D "cn=ldapadm2" -w secret -f //USRSCHM.LDIF -u on
LDAPMDFY -h 9.12.4.191 -D "cn=ldapadm2" -w secret -f //USRSCHM.LDIF -u on
```

7. Load sample user IDs in LDBM database.

z/VM TCP/IP contains a LDIF file containing sample user ID. This data could be added to the LDBM database using:

```
COPY SAMPSEV LDIF K = = A
```

In SAMPSEV LDIF, change all occurrences of o=Your Company to o=ibm.

The updated SAMPSEV LDIF is written in the LDBM database using:

```
LDAPADD -h 9.12.4.191 -D "cn=ldapadm2" -w secret -f //SAMPSEV.LDIF
```

To check the update made in LDAP database, issue:

```
LDAPSRCH -h 9.12.4.191 -D "cn=ldapadm2" -w secret -b o=ibm "objectclass=*"
```

8. A LDAP administrator is created using a LDIF file. This user will be used to administrate the LDAP database.

Insert in the ITSO LDIF file the following data:

```
dn: cn=ldapadm2, o=ibm
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: ibm-nativeAuthentication
cn: ldapadm2
sn: ldapadm2
ibm-nativeId: LDAPADM2
```

And then run:

```
LDAPADD -h 9.12.4.191 -D "cn=ldapadm2" -w secret -f //ITSO.LDIF
```

9. From the TCPMAINT user ID, modify the DS CONF file to:

```
adminDN "cn=ldapadm2"
adminPW secret
allowAnonymousBinds on
commThreads 10
listen ldap://:389
logfile /var/ldap/gldlog.output
maxConnections 65535
sendV3StringsOverV2As UTF-8
sizeLimit 500
timeLimit 3600
validateIncomingV2strings on
database LDBM GLDBLD31
suffix "o=ibm"
useNativeAuth all
nativeUpdateAllowed on
nativeAuthSubtree "o=ibm"
```

10. Restart LDAPSrv2 user ID using a FORCE followed by an XAUTOLOG.

11. From an RACF administrator, create LINUX1 user:

```
RAC ADDUSER LINUX1 PASS(TEST123)
```

12. Verify LDAPADM2 is able to use native authentication:

```
LDAPSRCH -h 9.12.4.191 -D "cn=linux1,o=ibm" -w "test123/itso7471" -s base  
-b o=ibm "objectclass=*
```

During this command, the LINUX1 password is changed from test123 to itso7471. If you run the LDAPSRCH multiple times, use the current password in **-w parm**.

LINUX1 is now available with native authentication. The adminDN is usable with the password secret. To protect your LDBM database, adminDN will be with Native Authentication using the RACF database.

13. From TCPMAINT, modify the DS CONF file to:

```
adminDN "cn=linadm,o=ibm"  
allowAnonymousBinds on  
commThreads 10  
listen ldap://:389  
logfile /var/ldap/gldlog.output  
maxConnections 65535  
sendV3StringsOverV2As UTF-8  
sizeLimit 500  
timeLimit 3600  
validateIncomingV2strings on  
database LDBM GLDBLD31  
suffix "o=ibm"  
useNativeAuth all  
nativeUpdateAllowed on  
nativeAuthSubtree "o=ibm"
```

14. Restart the LDAPSRV2 server to use the parameters defined in the DS CONF file. The adminDN cannot use the password secret any longer.

In the next chapter, we describe how to adapt a Linux server to LDAP. This Linux server uses the LDAPSRV2 server and the z/VM RACF during the Linux user logon.

Archived

Implementing Pluggable Authentication Modules LDAP for Linux servers

The z/VM architecture allows you to install multiple Linux instances on the same logical partition. To manage and administer these configurations easily, the following facilities have been developed to help with large number of System z Linux images:

- ▶ Cloning tools to generate new Linux servers quickly (in less than a minute)
- ▶ XAUTOLOG to allow automatic start for a large number of servers. The SIGNAL command informs each Linux of a need to shut down, and each Linux starts its shutdown. When all Linux is stopped, z/VM stops itself.
- ▶ A Pluggable Authentication Module (PAM) LDAP function to improve security and to reduce administrative tasks. With the PAM function, you can avoid defining users under a System z Linux in every Linux server and can have a unique user ID and password on z/VM, z/OS, and Linux.

This chapter discusses the PAM LDAP function, how to configure it, and how you can take advantage of this feature for your installation operations.

4.1 PAM and Name Service Switch

To use LDAP services, you need to change the PAM. PAM modules are used to control services such as smtp, crond, FTP, atd, sshd, and so forth. Each of these services uses a PAM configuration file. All the PAM configuration files are stored in `/etc/pam.d/` directory.

4.1.1 PAM configuration files

PAM configuration files include information about:

- ▶ The order of authentication checking (auth keyword)
- ▶ Where to get accounting information (account keyword)
- ▶ How a password change is performed (password keyword)
- ▶ Information about session (session keyword)

Note: If you plan to control Samba services with LDAP, you need to deactivate Windows encryption. RACF cannot decrypt user and password information if information is encrypted using the Windows encryption format.

4.1.2 Linux Name Service Switch

To avoid having to store and maintain user informations (`/etc/passwd/`, `/etc/shadow`, and `/etc/group`) on each Linux system, you need to install the Name Service Switch (NSS) package in each Linux during installation and before cloning. The data contained in `/passwd`, `/shadow`, and `/group` is stored in the LDAP server as UID, UID number, GID number, and so forth. The object class `posixAccount` is needed in the LDAP server. NSS and LDAP permit to reduce drastically the administration effort.

4.2 Configuring PAM LDAP and NSS

To use PAM for authentication through LDAP, you must install the appropriate PAM LDAP package on your Linux for System z. Depending on the Linux for System z distribution that you are using on your server, the package itself might already be available, although its name might be slightly different:

- ▶ Red Hat: `openldap`, `openldap-clients`, and `nss_ldap` packages
- ▶ SUSE: `openldap2`, `yast2-ldap-client`, `pam_ldap`, and `nss_ldap` packages

4.2.1 SUSE Linux

In our configuration, we use SUSE Linux Enterprise Server 10 accessed with ssh. Telnet service is not available for TCP/IP terminal for security reason.

You can use the PuTTY program to access a Linux server through ssh from a Windows workstation.

If you have a Linux workstation, ssh is included in Linux distributions.

You need to use the YaST tool to implement the PAM LDAP. Follow these steps:

1. Open an ssh session with the Linux image to be modified.
2. Log on as root user and start YaST by entering **yast**. You are prompted with a panel as shown in Figure 4-1.

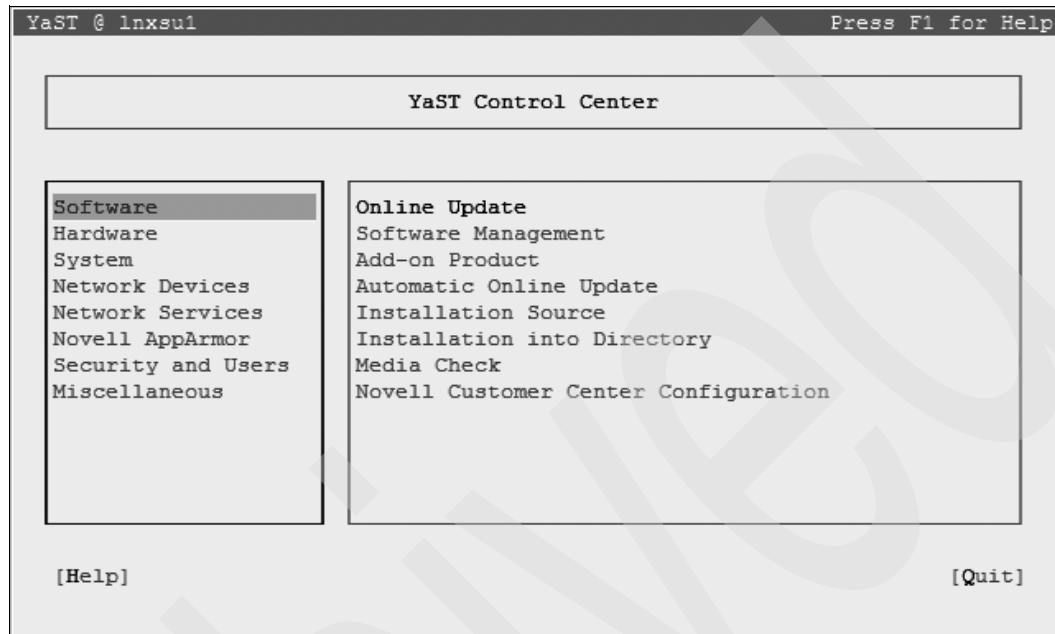


Figure 4-1 Network Services

3. Select Network Services and then select LDAP Client as shown in Figure 4-2.

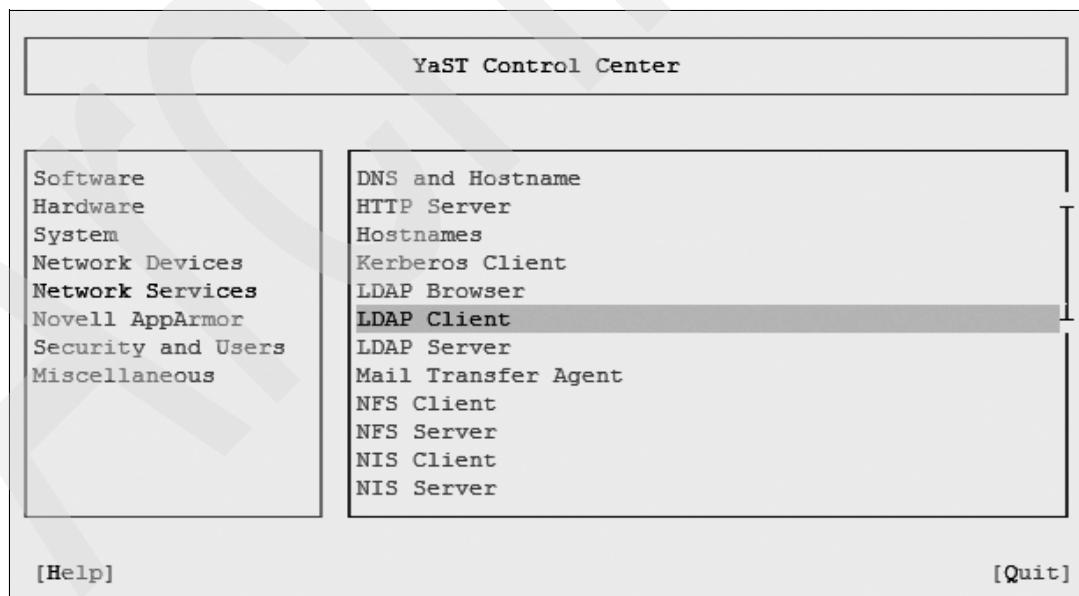


Figure 4-2 Select LDAP Client

4. The LDAP Client Configuration panel displays as shown in Figure 4-3.

The image shows a terminal window titled "LDAP Client Configuration". It contains two main sections: "User Authentication" and "LDAP Client". In the "User Authentication" section, the option "(x) Do Not Use LDAP" is selected. In the "LDAP Client" section, the "Addresses of LDAP Servers" field contains "127.0.0.1", the "LDAP Base DN" field contains "dc=example,dc=com", and the "LDAP TLS/SSL" option is selected with "(x)". At the bottom, there are buttons for "[Back]", "[Abort]", and "[Finish]", along with a link for "[Advanced Configuration...]" and checkboxes for "[] Start Automounter" and "[] Create Home Directory on Login".

```
LDAP Client Configuration
User Authentication
(x) Do Not Use LDAP
( ) Use LDAP
( ) Use LDAP but Disable Logins

LDAP Client
Addresses of LDAP Servers
127.0.0.1 [Find]
LDAP Base DN
dc=example,dc=com [Fetch DN]
(x) LDAP TLS/SSL
[ ] LDAP Version 2

[ ] Start Automounter
[ ] Create Home Directory on Login
[Advanced Configuration...]
[ Back ] [ Abort ] [ Finish ]
```

Figure 4-3 LDAP Client Configuration original field

5. Update the fields as shown in Figure 4-4.

The image shows the same "LDAP Client Configuration" terminal window, but with updated values. In the "User Authentication" section, the option "(x) Use LDAP" is now selected. In the "LDAP Client" section, the "Addresses of LDAP Servers" field contains "9.12.4.191", the "LDAP Base DN" field contains "o=ibm", and the "LDAP TLS/SSL" option is now unselected with "[]". The other elements, including the bottom buttons and checkboxes, remain the same as in Figure 4-3.

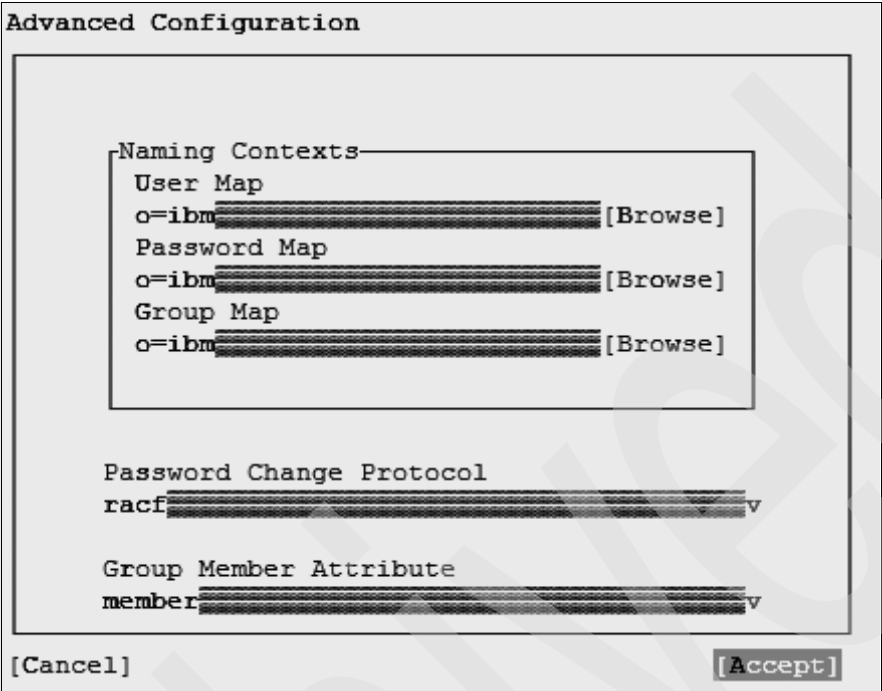
```
LDAP Client Configuration
User Authentication
( ) Do Not Use LDAP
(x) Use LDAP
( ) Use LDAP but Disable Logins

LDAP Client
Addresses of LDAP Servers
9.12.4.191 [Find]
LDAP Base DN
o=ibm [Fetch DN]
[ ] LDAP TLS/SSL
[ ] LDAP Version 2

[ ] Start Automounter
(x) Create Home Directory on Login
[Advanced Configuration...]
[ Back ] [ Abort ] [ Finish ]
```

Figure 4-4 LDAP Client Configuration after update

- When you have updated all the fields, select the field **Advanced Configuration**. This option opens another panel with complementary parameters. Complete these fields and select **Accept** as shown in Figure 4-5.



Advanced Configuration

Naming Contexts—

User Map
o=ibm [Browse]

Password Map
o=ibm [Browse]

Group Map
o=ibm [Browse]

Password Change Protocol
racf

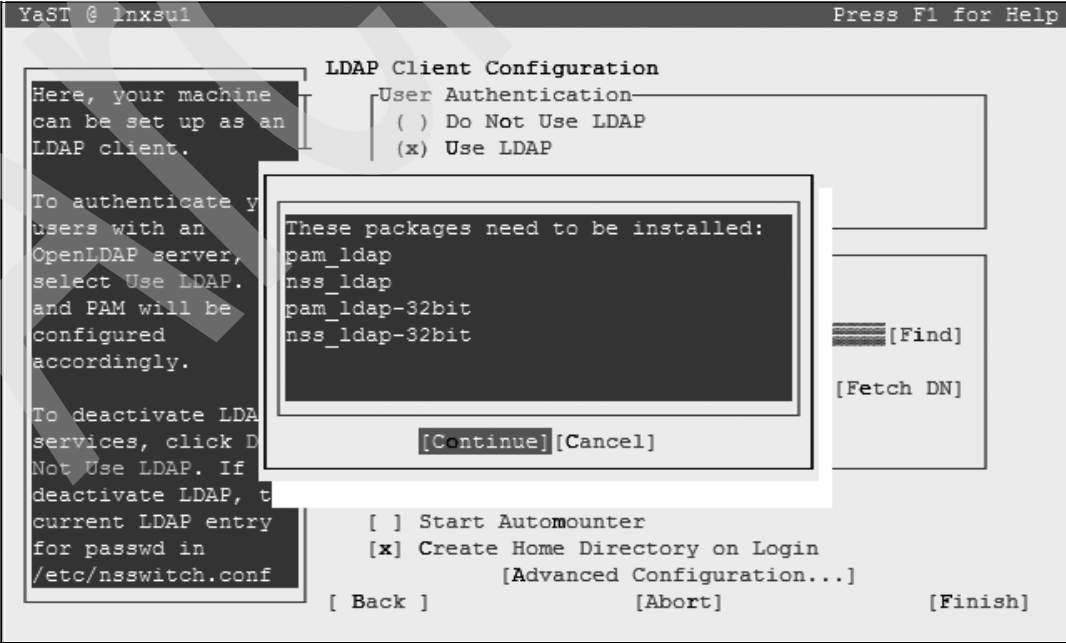
Group Member Attribute
member

[Cancel] [Accept]

Figure 4-5 Advanced Configuration parameters

- At this point, the LDAP Client Configuration panel displays. Select **Finish** to install the LDAP client.

When you receive the panel shown in Figure 4-6, select **Continue** to start the install of the missing Linux packages and to modify the configuration files.



YaST @ lnxsul Press F1 for Help

LDAP Client Configuration

Here, your machine can be set up as an LDAP client.

To authenticate y users with an OpenLDAP server, select Use LDAP. and PAM will be configured accordingly.

To deactivate LDA services, click D Not Use LDAP. If deactivate LDAP, t current LDAP entry for passwd in /etc/nsswitch.conf

User Authentication—
() Do Not Use LDAP
(x) Use LDAP

These packages need to be installed:
pam_ldap
nss_ldap
pam_ldap-32bit
nss_ldap-32bit

[Find]
[Fetch DN]

[Continue] [Cancel]

[] Start Automounter
[x] Create Home Directory on Login
[Advanced Configuration...]

[Back] [Abort] [Finish]

Figure 4-6 LDAP Client installation

8. When the YaST process is complete, perform the following steps to verify the PAM LDAP configuration:
 - a. Open a new ssh session with Linux.
 - b. Log on as linux1. This user is defined in the LDAP server but does not exist in the Linux server.

The following messages are received during the logon:

```
login as: linux1
Using keyboard-interactive authentication.
Password:
Creating directory '/home/linux1'.
Creating directory '/home/linux1/.fonts'.
Creating directory '/home/linux1/.mozilla'.
Creating directory '/home/linux1/.xemacs'.
Creating directory '/home/linux1/bin'.
Creating directory '/home/linux1/Documents'.
Creating directory '/home/linux1/public_html'.
linux1@lnxsul:~> ls
bin Documents public_html
```

PAM LDAP configuration is completed.

4.2.2 Red Hat Linux

In case you are running on Red Hat Linux, check that the following packages are installed using RPM commands:

```
rpm -qa | grep ldap or yum list *ldap*
```

Verify that you have the following packages:

- ▶ openldap
- ▶ openldap-client
- ▶ nss_ldap

If any of the packages are not installed, retrieve them and install them using the following command:

```
rpm -ivh packagename or yum install openldap openldap-clients nss_ldap
```

When all packages are installed, use the following command to activate the LDAP client functions:

```
authconfig --enableldap --enableldapauth --ldapsrv=9.101.54.73
--ldapbasedn=o=ibm --update
```

Then, follow these steps:

1. Update the /etc/ldap.conf (only the uncommented options are shown):

```
# @(#) $Id: ldap.conf,v 1.38 2006/05/15 08:13:31 lukeh Exp $
#
# This is the configuration file for the LDAP nameservice
# switch library and the LDAP PAM module.
#
# The man pages for this file are nss_ldap(5) and pam_ldap(5)
#
# PADL Software
```

```
# http://www.padl.com
#

base o=cs1

ldap_version 3
# Search timelimit
#timelimit 30
timelimit 120

# Bind/connect timelimit
#bind_timelimit 30
bind_timelimit 120

# Reconnect policy: hard (default) will retry connecting to
# the software with exponential backoff, soft will fail
# immediately.
bind_policy soft

# Idle timelimit; client will close connections
# (nss_ldap only) if the server has not been contacted
# for the number of seconds specified below.
#idle_timelimit 3600
idle_timelimit 3600
# RACF is an alias for the above. For use with
# IBM RACF
pam_password racf
# Just assume that there are no supplemental groups for these named users
nss_initgroups_ignoreusers root,ldap,named,avahi,haldaemon
# NDS mappings
nss_map_attribute uniqueMember member
pam_filter objectclass=posixAccount
uri ldap://9.101.54.73/
ssl no
tls_cacertdir /etc/openldap/cacerts
pam_password md5
nss_base_passwd o=cs1
nss_base_shadow o=cs1
nss_base_group o=cs1
```

2. Add `pam_mkhomedir.so` in the `/etc/pam.d/system-auth` file. This option creates the user home directory if it does not exist and the file stored in `/etc/skel` is copied.

```
##PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth required pam_env.so
auth sufficient pam_unix.so nullok try_first_pass
auth requisite pam_succeed_if.so uid >= 500 quiet
auth sufficient pam_ldap.so use_first_pass
auth required pam_deny.so

account required pam_unix.so broken_shadow
account sufficient pam_succeed_if.so uid < 500 quiet
account [default=bad success=ok user_unknown=ignore] pam_ldap.so
account required pam_permit.so
```

password	requisite	pam_cracklib.so try_first_pass retry=3
password	sufficient	pam_unix.so md5 shadow nullok try_first_pass
use_authok		
password	sufficient	pam_ldap.so use_authok
password	required	pam_deny.so
session	optional	pam_keyinit.so revoke
session	required	pam_mkhomedir.so skel=/etc/skel umask=0077
session	required	pam_limits.so

When these changes are activated, a Linux user defined in the LDAP server can log on. If everything is successful, at logon time the user home directory is created automatically if it does not already exist.

4.3 Changing the password

When LDAP PAM controls are active, a Linux user can change a password using the following methods:

► From the Linux system:

- With the **passwd** command
- With the **ldapsearch** command, for example:

```
ldapsearch -h 9.12.4.191 -D "cn=ldapadm2,o=ibm" -w "oldpass/newpass" -x
-s base -b "o=ibm" "objectclass=*" "(")
```

► From a CMS user (if the user ID is shared between z/VM and Linux)

- If the CMS user is the same as the Linux user, then the password can be changed during logon:

```
LOGON cmsuser oldpassword/newpassword/newpassword
```

- If the Linux user is not equal to a CMS user, then use the **ldapsearch** command:

```
LDAPSRCH -h 9.12.4.191 -D "cn=ldapadm2,o=ibm" -w "oldpass/newpass" -s
base -b "o=ibm" "objectclass=*" "(")
```

► From a z/OS system (if the user ID is shared between z/OS and Linux)

- If the z/OS user is the same as the Linux user, then the password can be changed in the logon panel specifying old password, new password, and then new password again
- If the Linux user is not equal to a z/OS user, then use the **ldapsearch** command:

```
LDAPSEARCH -h 9.12.4.191 -D "cn=ldapadm2,o=ibm" -w "oldpass/newpass" -s
base -b "o=ibm" "objectclass=*" "(")
```

Enterprise integration

Running a large number of distributed servers involves a great deal of effort to install, administer, maintain, and provide security for them. To contain this total effort, many enterprises have been consolidating these servers under Linux for System z and taking advantage of the virtualization technologies to use the hardware effectively and to simplify administration tasks. Furthermore, using a *centralized repository* to handle and maintain user information for multiple Linux systems can reduce both the complexity and the effort involved in user administration.

In this chapter, we describe how to plan and implement a central security solution for multiple Linux systems in a z/VM and a z/OS configuration.

To run multiple Linux for System z servers on one mainframe securely, you can use virtualization technology (z/VM or LPAR).

Another aspect of security is to provide secure access to the Linux systems only for authorized users, and the most common way is by using a user ID and password for authentication. If there are users for multiple Linux systems to maintain, it is advantageous to keep this sensitive information in a central place and not spread among various systems.

In a z/VM and z/OS environment, the natural place where user information and passwords are kept centrally already exists—the Resource Access Control Facility (RACF) database.

In this chapter, we explore different scenarios on how to centralize security information to contain and simplify the management task. We also provide the list of tasks that are required to implement the scenarios:

- ▶ Using a central z/VM LDAP server
- ▶ Sharing a RACF database with another z/VM system
- ▶ Sharing a RACF database with z/OS
- ▶ Using a central IBM Tivoli Directory Server for z/OS server
- ▶ Synchronizing LDAP/RACF database with IBM Tivoli Directory Integrator

5.1 Using a central z/VM LDAP server

In this scenario, the z/VM LDAP server will be used as focal point for the Linux server farm. VSWITCH, HiperSocket, Guest LAN permits to concentrate LDAP exchanges within the z/VM system. All the LDAP client and server exchanges happen inside the z/VM system. The local network traffic is not increased. This solution is z/VM centric. Linux and z/VM administration are concentrated on z/VM.

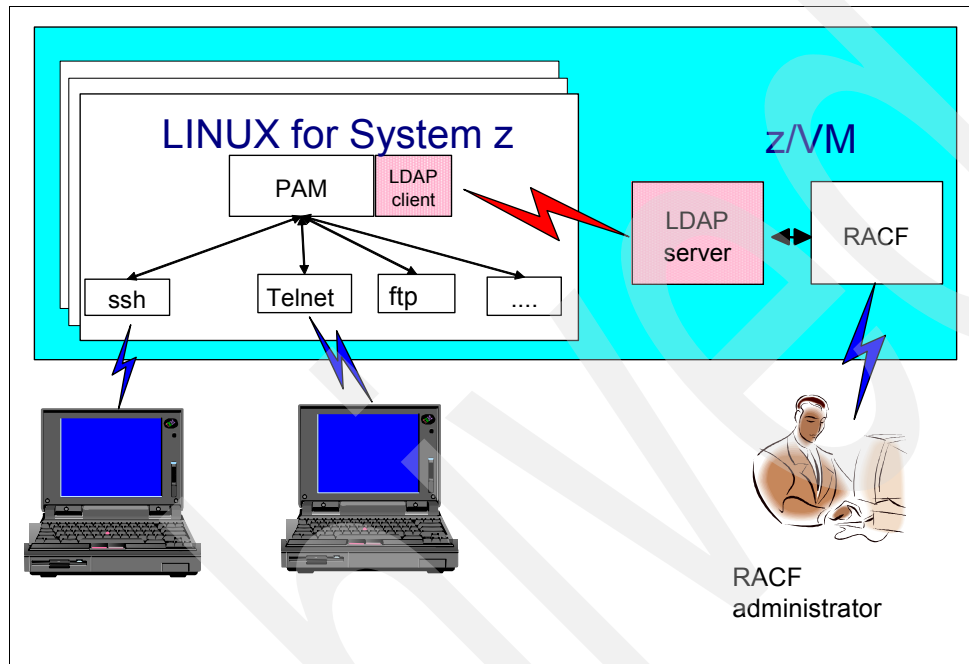


Figure 5-1 z/VM LDAP

5.1.1 LDAP architecture choices

In an environment with many Linux for System z servers, it is easy to manage users from a single management point (add, delete, change user account information, reset of passwords, and so on). This capability is based on the fact that there is a single instance of user information in a central data store (a directory with this information). If the information is stored in an LDAP directory centrally in a network, Linux for System z can access this information using LDAP clients to send messages and requests to the central LDAP server.

Pluggable Authentication Modules (PAMs) are the standard authentication framework for many Linux distributions. They allow integration of various authentication technologies into Linux system services, such as login, passwd, ssh, ftp, su, rlogin, and so on, without changing these services. The modules can be configured to pass authentication requests to LDAP.

When a user is authenticated in a Linux system, there are many services and applications that need access to user information and resolution of user information, such as resolving a user name from the UID number. This service is provided by NSS, which can also be configured to use LDAP to access information.

In such a configuration, it is possible to have the central LDAP server on the z/VM system, and combine and share the existing information of RACF users with Linux accounts, while simultaneously keeping the passwords protected by RACF. This enables you to provide the same high quality of service in both your z/VM environment and Linux for System z user

administration. Also, users with access to multiple Linux for System z can use the same account information about all these systems.

If the Linux for System z images do not reside on the same System z server as the z/VM LDAP server (and therefore use an external network connection) and, depending on your environment, needs, and enterprise policies, consider encrypting the communication between the LDAP client and the LDAP server using Secure Socket Layer (SSL), to avoid sending plain text password information over the network.

Note: In our configuration, we choose to implement the native authentication because this option gives the more facility for Linux servers.

5.1.2 Configuring z/VM LDAP server

Figure 5-2 shows the LDAP configuration. For details about the LDAP installation, refer to Chapter 5, “Enterprise integration” on page 145.

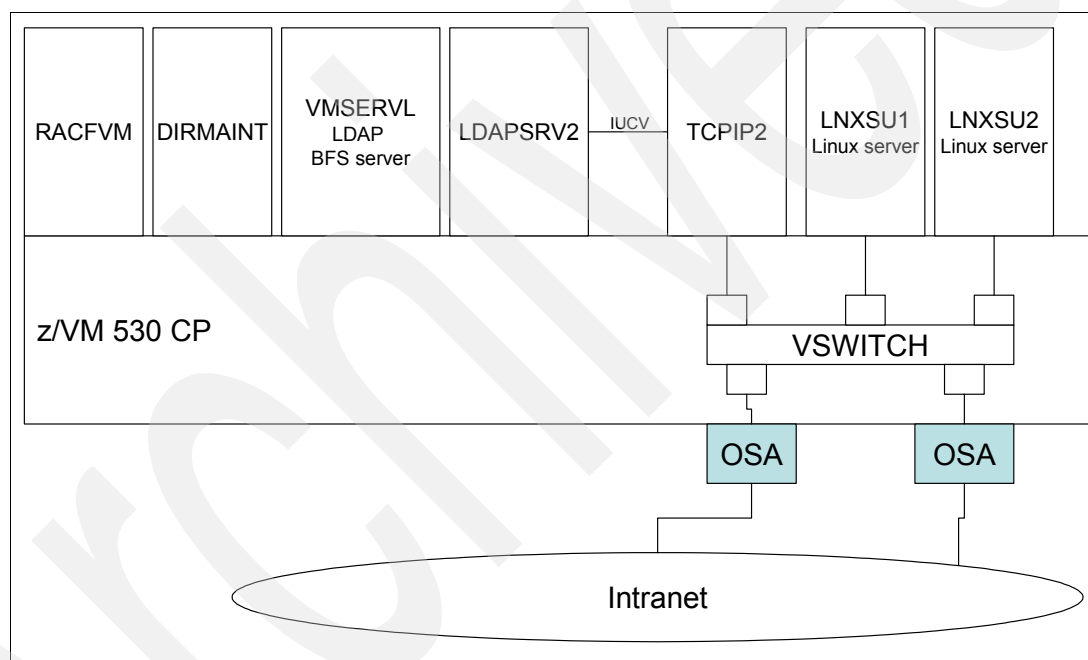


Figure 5-2 z/VM user IDs used with LDAP server

The LDAPSRV2 user ID needs to use RACF to grant the logon user ID and password. So, it needs to be authorized to dialog with the RACF server (LDAPSRV2 DTCPARMS).

To configure the LDAP server for our configuration, perform the following tasks:

1. Log on to the TCPMAINT user ID.
2. Verify the ESM_Enable parameter, as shown in Example 5-1.

Example 5-1 The ESM_Enable parameter

```
*****
.* LDAPSRV2 DTCPARMS
*****
:nick.LDAPSRV2 :type.server :class.ldap
.*=====
```

```

.* Note: The Byte File System directory path name specified for the
.*      'var/ldap/' mount purposely omits the file space ID portion of
.*      this path name. This has been done to allow for substitution
.*      of the subject server user ID when this entry is processed.
.*-----
:nick.ldap      :type.class
                :name.LDAP daemon
                :command.LDAPSRV
                :runtime.C
                :memory.128M
                :mixedcaseparms.YES
                :mount. ../VMBFS:VMSYSL:ROOT/    /
                :ESM_Enable.YES
                :ESM_Racroute.LDAPESM

```

3. Adapt the DS CONF file to activate the LDBM backend and the native authentication, as shown in Example 5-2.

Example 5-2 Activate the LDBM backend

```

adminDN "cn=ldapadm2,o=ibm"
allowAnonymousBinds on
commThreads 10
listen ldap://:389
logfile /var/ldap/gldlog.output
maxConnections 65535
sendV3StringsOverV2As UTF-8
sizeLimit 500
timeLimit 3600
validateIncomingV2strings on
database LDBM GLDBLD31
suffix "o=ibm"
useNativeAuth all
nativeUpdateAllowed on
nativeAuthSubtree "o=ibm"

```

4. Edit the ENVVARS LDAP server configuration file, as shown in Example 5-3.

Example 5-3 The ENVVARS file

```

DS ENVVARS
#
NLSPATH=/usr/lib/nls/msg/%L/%N
#
LANG=En_US.IBM-1047
#
#LIBPATH=/usr/lib
#
#TZ=GMT0
#
#=====
#  END OF CONFIGURATION FILE
#=====

```

5. Verify that the `ldapadm2` user that is specified in the `adminDN` parameter exists in RACF. Otherwise, you will not be able to create your LDAP directory.

From a RACF administrator, define LDAPADM2 user in RACF database:

```
RAC ADDUSER LDAPADM2 PASSWORD(PASS)
```

Note: You will change the user password during the LDAP test verification.

6. Grant the capability to open an IUCV session for the LDAPSRV2 with the RACF:

```
RAC RDEFINE FACILITY ICHCONN UACC(NONE)
RAC PERMIT ICHCONN CLASS(FACILITY) ID(LDAPSRV2) ACCESS(UPDATE)
RAC SETR CLASS(FACILITY)
```

After completing these steps, the LDAPSRV2 user ID should be able to log on.

7. Check on the console for any error messages.

Example 5-4 shows the messages received during the LDAP server start up.

Example 5-4 LDAP server console messages

```
DTCRUN1022I Console log will be sent to default owner ID: TCPMAINT
RPICMS016I USER/RACF VM Racroute communication path is established.
DTCRUN1027I Server will use TcpiUserid TCP/IP2
DTCRUN1021R To cancel LDAP daemon startup, type any non-blank character and
press ENTER. To continue startup, just press ENTER.

DTCRUN1011I Server started at 11:48:28 on 23 Jul 2007 (Monday)
DTCRUN1011I Running "LDAPSRV"
DTCLDP2106I Debug setting: 0
DTCLDP2107I Using server configuration file: DS CONF D2
DTCLDP2107I Using environment variable file: DS ENVVARS D1
DTCLDP2107I Using server module: GLDSRV31 MODULE E2
070723 15:48:29.253094 GLD1003I LDAP server is starting.
070723 15:48:29.253384 GLD1001I LDAP server version 3.18, Service level
0A19849, Build date Mar 22 2007, Time 22:58:27.
070723 15:48:29.253535 GLD1002I LDAP runtime version 3.18, Service level
0A19849, Build date Mar 22 2007, Time 23:25:52.
070723 15:48:29.303678 GLD1023I Processing configuration file //DD:CONFIG.
070723 15:48:29.304269 GLD1024I Configuration file //DD:CONFIG processed.
```

```
Server Configuration
adminDN: cn=ldapadm2,o=ibm
adminPW: *not configured*
allowAnonymousBinds: on
armName: GLDSRV31
audit 1: off
commThreads: 10
db2Terminate: recover
dnCacheSize: 1000
idleConnectionTimeout: 0
listen 1: ldap://:389
logfile: /var/ldap/gldlog.output
maxConnections: 65535
pcIdleConnectionTimeout: 0
pcThreads: 10
schemaPath: /var/ldap/schema
```

```

schemaReplaceByValue: on
securityLabel: off
sendV3StringsOverV2As: UTF-8
serverEtherAddr: 402084006A3A
11:48:29 * MSG FROM TCPIP2 : Not forcing you because you're connected
serverSysplexGroup: undefined
sizeLimit: 500
srvStartUpError: terminate
supportKrb5: off
tcpTerminate: recover
timeLimit: 3600
validateIncomingV2Strings: on

database LDBM GLDBLD31 LDBM-0001
changeLoggingParticipant: on
commitCheckpointEntries: 10000
commitCheckpointTOD: 00:00
databaseDirectory: /var/ldap/ldbm
extendedGroupSearching: off
fileTerminate: recover
filterCacheBypassLimit: 100
filterCacheSize: 5000
krbIdentityMap: off
multiServer: off
nativeAuthSubtree 1: 0=IBM
nativeUpdateAllowed: on
persistentSearch: off
pwEncryption: none
pwCryptCompat: on
readOnly: off
secretEncryption: none
sizeLimit: 500
suffix 1: o=ibm
timeLimit: 3600
useNativeAuth: all
070723 15:48:30.263166 GLD1074W Maximum client connections changed from 65535
to 65523.
070723 15:48:30.265281 GLD1004I LDAP server is ready for requests.
070723 15:48:31.135578 GLD1059I Listening for requests on 9.12.4.191 port 389.
070723 15:48:31.137170 GLD1059I Listening for requests on 127.0.0.1 port 389.

```

5.1.3 Verifying the LDAP server

When the LDAP server is running, you can check its availability. At the first logon, you are required to change the password.

To check the availability of the LDAP sever, follow these steps:

1. Log on as a CMS user.
2. Link and access TCPMAINT 592 minidisk. You can add this minidisk in the CMS PROFILE EXEC so that LDAP commands are available immediately.

3. Run the LDAPSrch command:

```
LDAPSrch -h 9.12.4.191 -D cn=ldapadm2,o=ibm -w PASS/ITS07471 -s base -b  
o=ibm objectclass=*
```

LDAPSrch should return:

```
o=ibm  
objectclass=top  
objectclass=organization  
o=ibm
```

5.1.4 Loading Linux schema and sample data in the LDBM

The LDAP server has previously loaded with sample schemes during the installation step. The Linux server requires one more schema. This schema provides the possibility to store in the LDAP database specific Linux data such as UID, GID number, home directory, login shell, and so forth.

This new schema is available on the FTP site:

<ftp://www.redbooks.ibm.com/redbooks/REDP0221>

Note:

You should transfer the LDIF file to VM using FTP in ASCII mode. Using cut and paste or TN3270 file transfer gives truncated records.

If you observe the following error, you need to modify the ldif file:

```
ldapmodify -a -h 9.12.4.126 -D cn=admin -w secret -f ./nisSchema.ldif -x  
ldapmodify: invalid format (line 52) entry: "cn=schema"
```

Modify the ldif file as follows:

- ▶ Suppress the minus sign between **changetype: modify**
- ▶ Add a **dn:cn=schema** line and **changetype: modify**

```
ibmattributetypes: (  
  1.3.6.1.1.1.1.0  
  DBNAME( 'uidNumber' 'uidNumber' )  
  ACCESS-CLASS normal  
  LENGTH 11  
)
```

```
dn:cn=schema  
changetype: modify  
replace: objectclasses  
objectclasses: (  
  1.3.6.1.1.1.2.12  
  NAME 'bootableDevice'  
  DESC 'A device with boot parameters; device SHOULD be used as a  
structural class'  
  SUP top  
  AUXILIARY  
  MAY ( bootFile $ bootParameter )  
)
```

To load the schema, follow these steps:

1. Log on a CMS user and issue the following commands:

```
LINK TCPMAINT 592 592 RR
ACCESS 592 L
```

2. When the LDIF schema file is available in VM, XEDIT it and modify:

```
dn:cn=schema, <suffix> to dn:cn=schema
```

3. Load the schema file in the LDBM database using the following command:

```
LDAPMDFY -h 9.12.4.191 -D cn=ldapadm2,o=ibm -w ITS07471 -f //NISSCHMA.LDIF
-u on
```

We use the linux1 user here as an example because it was previously defined as LDAP administrator using Native Authentication.

5.1.5 Adding a Linux user in z/VM LDAP

Before you modify the Linux servers, you need to add a test user in the LDAP server. You create it using ITSO LDIF file.

To add a Linux user in z/VM LDAP, follow these steps:

1. Insert the data shown in Example 5-5 in the ITSO LDIF file.

Example 5-5 The data for the LDIF file

```
dn: cn=linux1, ou=Home Town, o=ibm
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: ibm-nativeAuthentication
objectclass: posixAccount
cn: linux1
sn: linux1
ibm-nativeId: LINUX1
uid: linux1
uidnumber: 42123
gidnumber: 100
homedirectory: /home/linux1
loginshell: /bin/bash
```

2. Load the user definition in the z/VM LDAP server:

```
LDAPADD -h 9.12.4.191 -D cn=ldapadm2,o=ibm -w ITS07471 -f //ITS0.LDIF
```

3. From a RACF administrator, define the LINUX1 user in RACF database using the following command:

```
RAC ADDUSER LINUX1 PASSWORD(PASS)
```

4. Because the user LINUX1 is not known in z/VM directory, you cannot change the password with a LOGON. To change the temporary password, use this command:

```
LDAPSRCH -h 9.12.4.191 -D "cn=linux1,o=ibm" -w "PASS/newpass" -s base -b
"o=ibm" "objectclass="
```

Alternatively, you could follow up the ADDUSER command with the following command:

```
ALTUSER LINUX1 PASSWORD(NEWPASS) NOEXPIRED
```

5.1.6 Adapting Linux servers to use LDAP service

Note: If your Linux server is not customized to work with LDAP, refer to the method described in Chapter 4, “Implementing Pluggable Authentication Modules LDAP for Linux servers” on page 137.

If the Linux servers are already working with PAM/LDAP service, you only need to change the LDAP configuration files with either one of the following methods:

- ▶ With YaST using the method described in 4.2, “Configuring PAM LDAP and NSS” on page 138. Because PAM LDAP and NSS are already installed, you only need to change the IP address and the DN information.
- ▶ With vi. You only need to change the `/etc/ldap.conf` file. Example 5-6 shows an example of the changes, in bold.

Example 5-6 Changes to the `/etc/ldap.conf` file

```
#
# This is the configuration file for the LDAP nameservice
# switch library, the LDAP PAM module and the shadow package.
#
# Your LDAP server. Must be resolvable without using LDAP.
host      9.12.4.191
# The distinguished name of the search base.
base o=ibm

# The LDAP version to use (defaults to 3
# if supported by client library)
ldap_version 3

# Don't try forever if the LDAP server is not reachable
bind_policy    soft

# Hash password locally; required for University of
# Michigan LDAP server, and works with Netscape
# Directory Server if you're using the UNIX-Crypt
# hash mechanism and not using the NT Synchronization
# service.
pam_password    racf

# Redirect users to a URL or somesuch on password
# changes.
#pam_password_prohibit_message Please visit http://internal to change your
# password.

# OpenLDAP SSL mechanism
# start_tls mechanism uses the normal LDAP port, LDAPS typically 636
ssl            no
nss_map_attribute    uniqueMember member
pam_filter      objectclass=posixAccount
nss_base_passwd o=ibm
nss_base_shadow o=ibm
nss_base_group  o=ibm
tls_checkpeer   no
#ssl on
```

5.1.7 Linux logon test

At this point, the Linux server is ready to work with PAM LDAP and NSS. A z/VM LDAP server and RACF are ready to handle the Linux logon.

To test the functionality, open a ssh session to the Linux modified for PAM LDAP and logon as linux1.

If successful, you should receive the messages shown in Example 5-7.

Example 5-7 Linux logon test successful

```
login as: linux1
Using keyboard-interactive authentication.
Password:
Creating directory '/home/linux1'.
Creating directory '/home/linux1/.fonts'.
Creating directory '/home/linux1/.mozilla'.
Creating directory '/home/linux1/.xemacs'.
Creating directory '/home/linux1/bin'.
Creating directory '/home/linux1/Documents'.
Creating directory '/home/linux1/public_html'.
linuxusr1@lnxsul:~>
```

The messages that identify the creating of the directories are only valid for the first logon.

5.1.8 Summary

In a configuration with z/VM and multiple Linux images, the z/VM is the best place to centralize both Linux and z/VM administration. In such a configuration, the RACF administrator is able to administrate Linux users, z/VM users and also able to administrate z/VM resources using DIRMAINT.

5.2 Sharing RACF database with another z/VM system

This scenario shows how to provide a single point of authentication in a configuration with two or more z/VM systems. This scenario would allow a virtual machine to change the password on a single system and have the change propagated to the other z/VM systems through the RACF configuration (Figure 5-3).

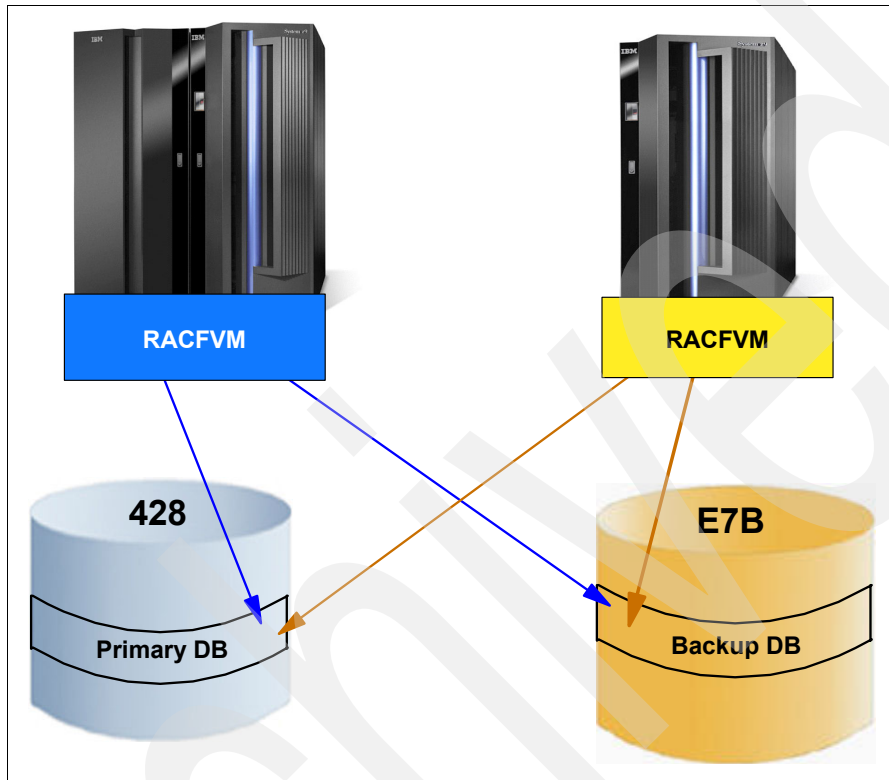


Figure 5-3 z/VM sharing the RACF database

A RACF database can be shared with another operating system, either z/OS or z/VM. In general, a RACF database can be shared with systems that have different levels of RACF installed, but you would have to run the RACFCONV EXEC from the higher level system to ensure the RACF templates are at the same level. A RACF database residing on FBA DASD cannot be shared. See *RACF Security Server System Programmer's Guide*, SC24-6149 for more information about sharing a RACF database.

The first step to take in such a configuration is to decide where the RACF databases and libraries should be located and whether there will be a single database or multiple databases. You can find information about splitting the RACF database in *RACF Security Server System Programmer's Guide*, SC24-6149 and in *Program Directory for RACF Security Server for z/VM*, G110-0788.

The VM restrictions and requirements for sharing DASD must be met. Information concerning VM requirements can be found in the following VM manuals and should be reviewed for planning purposes:

- ▶ z/VM System Operation Guide
- ▶ z/VM CP Planning and Administration Guide
- ▶ z/VM Running Guest Operating Systems Guide

Consider the following elements if your installation plans to share the RACF database between two or more systems:

- ▶ System design in terms of DASD mapping
- ▶ Resource and load balancing
- ▶ Recovery and restart
- ▶ Operational control of the multi-processor environment
- ▶ Programming considerations for user resource protection
- ▶ Data integrity

RACF utilizes serialization, through hardware RESERVE and RELEASE channel programs, to ensure the integrity of data stored in the RACF database between sharing RACF instances. Therefore, if you intend to share the RACF database and the warning message CSTERP001W is issued during RACFVM initialization, you must correct the situation to prevent database damage. (We address this issue in more detail in 5.2.1, “Preparing SYSTEM1 for RACF database sharing” on page 157.) The shared DASD environment must be properly defined to the z/VM Control Program in order for the corresponding serialization (for example, virtual or real RESERVE/RELEASE channel programs) to actually be reflected to the device on which the RACF database resides.

Our scenario is based on two z/VM systems. The first system has RACF implemented successfully. This system is defined with a system identifier of SYSTEM1. We show what has to be converted for SYSTEM1 to make it ready to share the RACF database. The second system currently does not have RACF implemented. It is defined with a system identifier of SYSTEM2. We discuss the process of allowing SYSTEM2 to utilize the RACF database on SYSTEM1.

For the basic RACF implementation process, refer to 2.2, “Installing and configuring RACF” on page 22. In this chapter, we only discuss the differences when implementing RACF in a shared environment.

5.2.1 Preparing SYSTEM1 for RACF database sharing

The first step in the process is to make sure that the database mdisks are not being cached. This is implemented with the directory statement MINIOPT. The MINIOPT statement is an extension to the MDISK statement and it must immediately follow the MDISK statement that defines a non-full-pack mdisk. We used the NOCACHE and NOMDC options to stop caching at the control unit level and at the mdisk level (Figure 5-4).

```
RACFVFM  DIRECT  A0  F 80  Trunc=72 Size=30 Line=0 Col=1 Alt=4
====>
0 * * * Top of File * * *
1 USER RACFVM RACFVM 20M 20M ABCDEGH
2   ACCOUNT SYSTEMS
3   IPL 490 PARM AUTOGR
4   IUCV *RPI PRIORITY MSGLIMIT 100
5   IUCV ANY PRIORITY MSGLIMIT 50
6 ----- 12 line(s) not displayed -----
18  LINK RACMAINT 0191 0591 MR
19  MDISK 0200 3390 0838 017 LX5W02 MWV READ WRITE MULTIPLE
20  MINIOPT NOCACHE NOMDC
21  MDISK 0300 3390 1001 017 LX5W09 MWV READ WRITE MULTIPLE
22  MINIOPT NOCACHE NOMDC
23  MDISK B200 3390 2000 017 LX5W02 MW READ WRITE MULTIPLE
24  MDISK 0191 3390 0886 009 LX5W02 MR READ WRITE MULTIPLE
25  MDISK 0490 3390 0895 038 LX5W02 MR READ WRITE MULTIPLE
26  MDISK 0305 3390 0933 068 LX5W02 MR READ WRITE MULTIPLE
27  MDISK 0301 3390 1018 007 LX5W02 MR READ WRITE MULTIPLE
28  MDISK 0302 3390 1025 007 LX5W02 MR READ WRITE MULTIPLE
29 *
```

SYSTEM1

Figure 5-4 NOCACHE and NOMDC

We decided that the database is going to reside on the *SYSTEM1* volumes that were initialized previously. The 200 and 300 disk are defined on the LX5W02 and LX5W09 DASD which are address 428 and E7B. This environment has shared DASD defined which supports the RESERVE/RELEASE function. The LX5W02 and LX5W09 volumes are CPOWNERD volumes on SYSTEM1 and USER volumes on SYSTEM2.

5.2.2 Re-instating IBMUSER

When we completed the installation of RACF on SYSTEM1, we issued the following commands:

```
rac altuser ibmuser revoke
rac altuser ibmuser nooperation nospecial
```

We issued these commands because the IBMUSER virtual machine is an IBM-defined user ID and it might be a target for unauthorized accesses to your system. To prevent further use of the IBMUSER user ID, we recommend that you revoke the user.

If you do not correct these issues, after RACMAINT is **xauto1oged** on SYSTEM2, you will not be able to link the required disks and run the RPIBLDDS EXEC to initial the database on SYSTEM2.

To correct this problem, perform the steps shown in Figure 5-5.

```
rac alu ibmuser resume
Ready; T=0.01/0.01 16:32:44
rac alu ibmuser operations special
Ready; T=0.01/0.01 16:32:57
rac alu ibmuser password(sys1)
Ready; T=0.01/0.01 16:33:07

CP TERM MORE 1 2
rac lu ibmuser
USER=IBMUSER NAME= OWNER=IBMUSER CREATED=07.195
DEFAULT-GROUP=SYS1 PASSDATE=00.000 PASS-INTERVAL= 30 PHRASEDATE=N/A
ATTRIBUTES=SPECIAL OPERATIONS
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=07.206/16:33:07
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
-----
ANYDAY ANYTIME
GROUP=SYSCTLG AUTH=JOIN CONNECT-OWNER=IBMUSER
CONNECT-DATE=07.195
CONNECTS= 00 UACC=READ LAST-CONNECT=UNKNOWN
```

Figure 5-5 Re-instate IBMUSER

5.2.3 Directory entries

During installation, if the database is not being shared, RACF issues the following warning message:

```
CSTERP__1W - Warning: Device xxx was configured as shared;
now configured as non-shared.
```

If you are not sharing a database, you can ignore the message. If you are sharing a database and receive the message, you have not set up your database correctly.

Both systems directory entries for the RACFVM virtual machine must have the directory entries for the 200 and 300 disks set with a mode of *MWV* (Figure 5-6 and Figure 5-7). This allows the disks to support Virtual RESERVE/RELEASE as described in CP Planning and Administration - SC24-6083.

```

RACFVM   DIRECT   A0  F 80  Trunc=72 Size=30 Line=0 Col=1 Alt=4
====>
0 * * * Top of File * * *
1 USER RACFVM RACFVM 20M 20M ABCDEGH
2   ACCOUNT SYSTEMS
3   IPL 490 PARM AUTO CR
4   IUCV *RPI PRIORITY MSGLIMIT 100
5   IUCV ANY PRIORITY MSGLIMIT 50
6 ----- 12 line(s) not displayed -----
18  LINK RACMAINT 0191 0591 MR
19  MDISK 0200 3390 0838 017 LX5W02 MWV READ WRITE MULTIPLE
20  MINIOPT NOCACHE NOMDC
21  MDISK 0300 3390 1001 017 LX5W09 MWV READ WRITE MULTIPLE
22  MINIOPT NOCACHE NOMDC
23  MDISK B200 3390 2000 017 LX5W02 MW READ WRITE MULTIPLE
24  MDISK 0191 3390 0886 009 LX5W02 MR READ WRITE MULTIPLE
25  MDISK 0490 3390 0895 038 LX5W02 MR READ WRITE MULTIPLE
26  MDISK 0305 3390 0933 068 LX5W02 MR READ WRITE MULTIPLE
27  MDISK 0301 3390 1018 007 LX5W02 MR READ WRITE MULTIPLE
28  MDISK 0302 3390 1025 007 LX5W02 MR READ WRITE MULTIPLE
29 *

```

SYSTEM1

Figure 5-6 SYSTEM1 RACFVM definition

```

RACFVM   DIRECT   A0  F 80  Trunc=72 Size=30 Line=0 Col=1 Alt=4
====>
0 * * * Top of File * * *
1 USER RACFVM RACFVM 20M 20M ABCDEGH
2   ACCOUNT SYSTEMS
3   IPL 490 PARM AUTO CR
4   IUCV *RPI PRIORITY MSGLIMIT 100
5   IUCV ANY PRIORITY MSGLIMIT 50
6 ----- 12 line(s) not displayed -----
18  LINK RACMAINT 0191 0591 MR
19  MDISK 0200 3390 0838 017 LX5W02 MWV READ WRITE MULTIPLE
20  MINIOPT NOCACHE NOMDC
21  MDISK 0300 3390 1001 017 LX5W09 MWV READ WRITE MULTIPLE
22  MINIOPT NOCACHE NOMDC
23  MDISK B200 3390 2000 017 LX5W02 MW READ WRITE MULTIPLE
24  MDISK 0191 3390 0886 009 LX5W02 MR READ WRITE MULTIPLE
25  MDISK 0490 3390 0895 038 LX5W02 MR READ WRITE MULTIPLE
26  MDISK 0305 3390 0933 068 LX5W02 MR READ WRITE MULTIPLE
27  MDISK 0301 3390 1018 007 LX5W02 MR READ WRITE MULTIPLE
28  MDISK 0302 3390 1025 007 LX5W02 MR READ WRITE MULTIPLE
29 *

```

SYSTEM2

Figure 5-7 SYSTEM2 RACFVM definition

This must be done before you create the RPIDIRECT SYSUT1 file. When you restart the system using the CF2 parm disk after having run the SERVICE RACF ENABLE, you should *not* receive the message shown in Figure 5-8. If you receive this message, shut down the system, IPL from CF1, and correct the definitions.

```
RACFVM :  
RACFVM : DMSACC723I R (0200) R/W - OS  
RACFVM : DMSACC723I Q (0300) R/W - OS  
RACFVM : CSTSET001I CMS SUB-TASKING SUPERVISOR INITIALIZED.  
RACFVM : CSTINT003I INITIATOR ACTIVATED.  
RACFVM : ICH508I ACTIVE RACF EXITS: NONE  
RACFVM : CSTERP001W - Warning: Device 200 was configured as shared; now  
configured as non-shared.  
RACFVM : CSTERP001W - Warning: Device 300 was configured as shared; now  
configured as non-shared.  
RACFVM : ICH520I RACF 5.3.0 IS ACTIVE.  
RACFVM : RPISTR001I Program CSTDYNST Initiated.  
RACFVM : RPISTR002I Program CSTDYNST Ended. Completion code = 000000.  
RACFVM : RPISTR003I Subtask RPIMSG Initiated.  
RACFVM : RPISTR003I Subtask RPIINIT Initiated.
```

Figure 5-8 Device 200 and 300 configured NON-SHARED

You must define the addresses of the LX5W02 and LX5W09 volumes with *one* of the following statements in the I/O subsystems for both processors.

RDEVICE statement coded **"SHARED=YES"**

IODEVICE statement for device number 428 coded **"SHARED=YES"**

These definitions are found in the file that creates the Input/Output Configuration Data Set (IOCDS) described in Figure 5-9. Also, notice in Figure 5-6 and Figure 5-7 that both SYSTEM1 and SYSTEM2 have defined the primary and backup database on different volumes.

```

ITSO      IOCP      A1  F 80  Trunc=80 Size=4 Line=0 Col=1 Alt=0
====>

|...+...1...+...2...+...3...+...4...+...5...+...6...+...7..
0 * * * Top of File * * *
1      CHPID PATH=(CSS(0),01), SHARED, PCHID=111, TYPE=FC,          X
2      PARTITION=((SYSTEM1,SYSTEM2,ZOSSYS)(=))
3      CNTLUNIT CUNUMBR=400, PATH=((CSS(0),01)), UNIT=2107
4      IODEVICE ADDRESS=(400,064), CUNUMBR=(400), UNIT=3390
5      CHPID PATH=(CSS(0),1B), SHARED, PCHID=138, TYPE=FC,          X
6      PARTITION=((SYSTEM1,SYSTEM2,ZOSSYS)(=))
7      CNTLUNIT CUNUMBR=E00, PATH=((CSS(0),1B)), UNIT=2107
8      IODEVICE ADDRESS=(E00,064), CUNUMBR=(E00), UNIT=3390
9 * * * End of File * * *

```

Figure 5-9 IOCDS Definitions

5.2.4 DES encryption

If you choose to use the data encryption standard (DES) algorithm in encrypting RACF passwords, you must use it on all the systems that share the RACF database. The encryption method must be the same for all systems sharing the database.

When RACF was implemented on *SYSTEM1*, we executed the step to allow for DES encryption. We will perform these same steps on *SYSTEM2* at the appropriate time.

5.2.5 Initializing the RACF database on SYSTEM2

After the IPL from the CF2 disk, you start the system with the NOAUTOLOG parameter, XAUTOLOG the RACMAINT virtual machine, and log on to the IBMUSER virtual machine (Figure 5-10). When you log on, you use the password for IBMUSER that is defined on SYSTEM1.

```
z/VM ONLINE
This system is for RACF DB Sharing VM to VM

      / VV      VVV MM      MM
     /  VV      VVV  MMM    MMM
    /   VV      VVV   MMMM   MMMM
   /    VV      VVV    MM MM MM MM
  /     VV VVV   MM   MMM  MM
 /      VVVVV   MM   M   MM
/       VVV     MM     MM
/       V      MM      MM

      Detro's System

Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID  ==> ibmuser
PASSWORD ==> xxxxxxxx

COMMAND ==>

RUNNING  SYSTEM2
```

Figure 5-10 Log on IBMUSER on SYSTEM2

You can see differences when you log on in this environment and link the mdisk required to run the RPIBLDDS EXEC. In our environment, we did not receive messages about resources not defined to the RACF database (Figure 5-11).

```
LOGON IBMUSER
ICH70001I IBMUSER  LAST ACCESS AT 07:57:06 ON THURSDAY, JULY 26,
2007
z/VM Version 5 Release 3.0, Service Level 0701 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:  NO RDR,  NO PRT,  NO PUN
LOGON AT 08:02:11 EDT THURSDAY 07/26/07
z/VM V5.3.0    2007-06-14 11:51
DMSACP723I D (192) R/O
Ready; T=0.01/0.01 08:02:12
set pf12 retrieve
Ready; T=0.01/0.01 08:02:16
link 5vmrac30 505 305 rr
DASD 0305 LINKED R/O; R/W BY RACMAINT
Ready; T=0.01/0.01 08:02:25
link 5vmrac30 191 192 rr
Ready; T=0.01/0.01 08:02:32
link 5vmrac30 29e 29e rr
Ready; T=0.01/0.01 08:02:38
```

Figure 5-11 IBMUSER log on to SYSTEM2

The messages displayed in Example 5-12 are shown as an example of what we received originally when we configured RACF for SYSTEM1. The non-existence of the RPIMGR031E messages as shown in Figure 5-12 is good confirmation from SYSTEM2 that we are sharing the SYSTEM1 RACF database.

```
set pf12 retrieve
Ready; T=0.01/0.01 11:25:46
link 5vmrac30 505 305 rr
RPIMGR031E RESOURCE 5VMRAC30.505 SPECIFIED BY LINK COMMAND NOT FOUND
DASD 0305 LINKED R/O; R/W BY RACMAINT
Ready; T=0.01/0.01 11:33:40
link 5vmrac30 191 192 rr
RPIMGR031E RESOURCE 5VMRAC30.191 SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 11:33:59
link 5vmrac30 29e 29e rr
RPIMGR031E RESOURCE 5VMRAC30.29E SPECIFIED BY LINK COMMAND NOT FOUND
Ready; T=0.01/0.01 11:34:06
ac 305 c
ac 192 b
ac 29e d
```

Figure 5-12 RPIMGR031E messages from SYSTEM1 RACF implementation

When the RPIBLDDS EXEC was executed, messages were received about virtual machine resources that had previously been defined. The exec ends with a non-zero return code (Example 5-8). This is acceptable, because these were duplicated entries defined previously in the RACF database on SYSTEM1.

Example 5-8 RPIBLDDS Error Messages

Some RACF commands contained in RPIDIRECT SYSUT1 had non-zero return codes. File RPIBLDDS OUTPUT has been created on your "A" disk with pertinent "RAC" output. Please investigate.
Ready; T=1.35/1.92 08:37:46

When we examined the RPIBLDDS OUTPUT file, we found what we really had expected. There were 492 occurrences of resources *already defined*, which seemed reasonable because of the duplicate virtual machine mdisk addresses being defined in both systems (Figure 5-13).

```
RPIBLDDS OUTPUT   Z1  V 80  Trunc=80 Size=1637 Line=126 Col=1
A1
====>
126 => RDEFINE VMBATCH MAINT OWNER(MAINT) UACC(NONE)
127 ICH10102I MAINT ALREADY DEFINED TO CLASS VMBATCH.
128 => RDEFINE VMRDR MAINT UACC(NONE) OWNER(MAINT)
129 ICH10102I MAINT ALREADY DEFINED TO CLASS VMRDR.
130 => CONNECT MAINT GROUP(SYSTEM)
131 ICH02005I MAINT      CONNECTION NOT MODIFIED.
132 => RDEFINE VMMDISK MAINT.CF1 OWNER(MAINT) UACC(NONE)
133 ICH10102I MAINT.CF1 ALREADY DEFINED TO CLASS VMMDISK.
134 => RDEFINE VMMDISK MAINT.CF2 OWNER(MAINT) UACC(NONE)
135 ICH10102I MAINT.CF2 ALREADY DEFINED TO CLASS VMMDISK.
136 => RDEFINE VMMDISK MAINT.CF3 OWNER(MAINT) UACC(NONE)
137 ICH10102I MAINT.CF3 ALREADY DEFINED TO CLASS VMMDISK.
138 => RDEFINE VMMDISK MAINT.190 OWNER(MAINT) UACC(READ)
139 ICH10102I MAINT.190 ALREADY DEFINED TO CLASS VMMDISK.
```

Figure 5-13 RPIBLDDS OUTPUT file

The messages are very different when a virtual machine is previously added to the RACF database on SYSTEM1. Even though the **adduser** command syntax was defined correctly, we received the messages shown in Figure 5-14. We verified that this is the expected behavior.

```

RPIBLDDS OUTPUT   Z1  V 80  Trunc=80 Size=1637 Line=0 Col=1 Alt=0
====>
 0 * * * Top of File * * *
 1 **** RACF commands returning non-zero return codes (via RPIBLDDS)
 2 ----- 108 line(s) not displayed -----
110 => ADDUSER LGLOPR DFLTGRP(SYS1) UACC(NONE) PASSWORD(LGLOPR)
111 IKJ56702I INVALID USERID, LGLOPR
112 IKJ56702I INVALID USERID, DFLTGRP(SYS1
113 IKJ56701I MISSING USERID+
114 IKJ56712I INVALID KEYWORD, )
115 ----- 6 line(s) not displayed -----
121 => ADDUSER MAINT DFLTGRP(SYS1) UACC(NONE) PASSWORD(MAINT)
122 IKJ56702I INVALID USERID, MAINT
123 IKJ56702I INVALID USERID, DFLTGRP(SYS1
124 IKJ56701I MISSING USERID+
125 IKJ56712I INVALID KEYWORD, )
126 ----- 1153 line(s) not displayed -----
1279 => ADDUSER RACFVM DFLTGRP(SYS1) UACC(NONE) PASSWORD(RACFVM)
1280 IKJ56702I INVALID USERID, RACFVM
1281 IKJ56702I INVALID USERID, DFLTGRP(SYS1
1282 IKJ56701I MISSING USERID+
1283 IKJ56712I INVALID KEYWORD, )

```

Figure 5-14 New user added to RACF database

Notice that the password will be as it exists in the RACF database from SYSTEM1. If the directory password on SYSTEM2 was different, the user needs to use the SYSTEM1 password when logging on to SYSTEM2.

5.2.6 Database verification

We had defined the virtual machine MARIAN to SYSTEM2 prior to implementation of RACF on the system. When we ran the RPIDIRCT EXEC, this virtual machine was found in the system directory. When the RPIBLDDS EXEC ran semi-successfully, this virtual machine was added to the RACF database. If we issue the RACF LISTUSER command on both systems, we should see MARIAN listed (Example 5-9 and Example 5-10).

Example 5-9 RACF LISTUSER SYSTEM1

```

RAC LU MARIAN
USER=MARIAN NAME=UNKNOWN OWNER=DIRMAINT CREATED=07.207
DEFAULT-GROUP=SYS1 PASSDATE=00.000 PASS-INTERVAL= 90 PHRASEDATE=N/A
ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=UNKNOWN
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
-----
ANYDAY ANYTIME

```

Example 5-10 RACF LISTUSER SYSTEM2

```
RAC LU MARIAN
USER=MARIAN  NAME=UNKNOWN  OWNER=DIRMAINT  CREATED=07.207
DEFAULT-GROUP=SYS1      PASSDATE=00.000 PASS-INTERVAL= 90 PHRASEDATE=N/A
ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=UNKNOWN
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY              ANYTIME
GROUP=SYS1          AUTH=USE        CONNECT-OWNER=DIRMAINT  CONNECT-DATE=07.207
CONNECTS=           00  UACC=NONE    LAST-CONNECT=UNKNOWN
```

If we try to log on to the virtual machine MARIAN from SYSTEM1, we receive the message shown in Example 5-11.

Example 5-11 Logon failure from SYSTEM1

```
LOGON MARIAN
HCPLGA053E MARIAN not in CP directory

Enter one of the following commands:

LOGON userid          (Example: LOGON VMUSER1)
DIAL userid           (Example: DIAL VMUSER2)
MSG userid message    (Example: MSG VMUSER2 GOOD MORNING)
LOGOFF
UNDIAL
CP READ  SYSTEM1
```

However, we were successful in logging on to MARIAN on SYSTEM2 (Figure 5-15).

```
LOGON MARIAN
RPIMGR042I PASSWORD EXPIRED

To change your password - enter: nnn/nnn where nnn = new password
or,
enter LOGOFF to cancel

  ICH70001I MARIAN   LAST ACCESS AT 08:51:38 ON THURSDAY, JULY 26,
2007
  HCPRPW004I Password changed
RPIMGR032E YOU ARE NOT AUTHORIZED TO LINK TO MARIAN.191
HCPLNM298E MARIAN 0191 not linked; request denied
z/VM Version 5 Release 3.0, Service Level 0701 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:  NO RDR,  NO PRT,  NO PUN
LOGON AT 09:00:40 EDT THURSDAY 07/26/07
z/VM V5.3.0    2007-06-14 11:51
DMSACP113S A(191) not attached or invalid device address
Ready; T=0.01/0.01 09:00:40
```

Figure 5-15 Log on to MARIAN on SYSTEM2

If we want MARIAN to be able to log on to SYSTEM1, we need to add the virtual machine MARIAN and the required mdisk to the system directory using the normal DirMaint processes. If using DirMaint on SYSTEM2, you will *not* need to add the RACF attributes of *SPECIAL* and *OPERATIONS* to the DIRMAINT and DATAMOVE virtual machines, because they were already defined in the RACF database on SYSTEM1. Use your normal processes to add the new virtual machine on one of the systems. For example, we add a new virtual machine TESTP to SYSTEM1.

Follow these steps to define the virtual machine TESTP to both systems:

1. Create the TESTP DIRECT file (Example 5-12). Remember that the file name and the virtual machine name must match.

Example 5-12 TESTP DIRECT A

```
TESTP DIRECT  A1  F 80  Trunc=72 Size=12
====>
0 * * * Top of File * * *
1 USER TESTP new1 32M 1000M BCDEFG
2 MACH XA
3 OPTION LNKNOPAS
4 IPL CMS PARM AUTO CR
5 CONSOLE 009 3215 T
6 SPOOL 00C 2540 READER *
7 SPOOL 00D 2540 PUNCH A
8 SPOOL 00E 1403 A
9 LINK MAINT 190 190 RR
10 LINK MAINT 19E 19E RR
11 LINK MAINT 19D 19D RR
12 LINK TCPMAINT 592 592 RR
```

2. Issue the DIRM ADD command (Figure 5-16) and enter TESTP as the virtual machine being added and press PF5. You receive messages and if everything is correct you should receive a return code 0 message.

```
-----DirMaint ADD-----  
  
Add an entry to the directory for a new USERID or Profile.  
  
Fill in the USERID or PROFILE being added:  
    ===>  TESTP  
  
Optionally fill in the following when using a prototype:  
    LIKE ===>      (file name of prototype)  
    PW  ===>      (password for new user)  
    VPW ===>      (password again for verification)  
    ACCT ===>     (account value for new user - optional)  
  
Notes:  
    - If a value is given for any one of PW, VPW, or ACCT,  
      then a value is required for LIKE.  
    - If a value is given for either PW or VPW,  
      then a value is required for both of them.  
  
741-A05 (c) Copyright IBM Corporation 1979, 2007.  
    1= Help      2= Prefix Operands      3= Quit      5=Submit  
==>
```

Figure 5-16 DIRM ADD

3. Add a mdisk for the new virtual machine by issuing **dirm for testp amdisk**. The Add Mdisk panel displays. Complete the required information and press PF5. You should see a message about the mdisk being defined successfully.
4. When logging on to the virtual machine, you are required to change the password. You see the mdisk was defined to the RACF database, but no RACF PERMIT was performed (Figure 5-17). To correct the problem, a RACF administrator must issue the **permit** command to allow the owner of the disk to have alter access to their own resource (Example 5-13). In 2.1, "RACF z/VM concepts" on page 22, we discussed how to create a REXX exec to issue the RACF PERMIT commands automatically (see Example 2-18 on page 47). If you have created the exec, then you would need to authorize the virtual machine to have update access to the disk.

```

LOGON TESTP
ICH70001I TESTP  LAST ACCESS AT 12:50:10 ON THURS, JULY 26, 2007
RPIMGRO32E YOU ARE NOT AUTHORIZED TO LINK TO TESTP.191
HCPLNM298E TESTP 0191 not linked; request denied
z/VM Version 5 Release 3.0, Service Level 0701 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:  NO RDR,  NO PRT,  NO PUN
LOGON AT 12:50:26 EDT THURSDAY 07/26/07
z/VM V5.3.0    2007-06-14 11:51
DMSACP113S A(191) not attached or invalid device address
Ready; T=0.01/0.01 12:50:26

```

Figure 5-17 Logon TESTP

Example 5-13 RACF PERMIT 191 MDISK

```

LOGON MAINT
ICH70001I MAINT  LAST ACCESS AT 12:55:22 ON THURSDAY, JULY 26, 2007
HCPLNM102E DASD 0123 forced R/O; R/W by DIRMAINT
z/VM Version 5 Release 3.0, Service Level 0701 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: 0014 RDR,  NO PRT,  NO PUN
LOGON AT 12:56:28 EDT THURSDAY 07/26/07
z/VM V5.3.0    2007-06-14 11:51
Ready; T=0.01/0.01 12:56:28
rac permit testp.191 class(vmmdisk) id(testp) ac(alter)
Ready; T=0.01/0.01 12:56:55

```

5. The final steps are to perform these tasks on SYSTEM2 so that this virtual machine can log on to either system.

If your system environment was set up properly, you can also share the DASD where mdisk are defined. This requires a high level of coordination when defining mdisk, but it is possible. The only problem is that you cannot log on to the virtual machine on both systems at the same time. If you did, the file status table on the mdisk will be corrupted, and all the data will be lost because the CMS file system does *not* support virtual RESERVE/RELEASE for mdisk.

5.2.7 Summary

In this section, we described how to implement RACF database sharing between two systems. The process is the same for additional z/VM systems added to share a single z/VM RACF database.

5.3 Sharing a RACF database with z/OS

In an enterprise that uses z/OS and z/VM operating systems, it can be interesting to share the RACF database. RACF administration for both z/OS and z/VM can be done in a single point without any requirement of duplicating the data. Figure 5-18 shows z/VM and z/OS with access to the two volumes that contain the RACF primary and secondary databases.

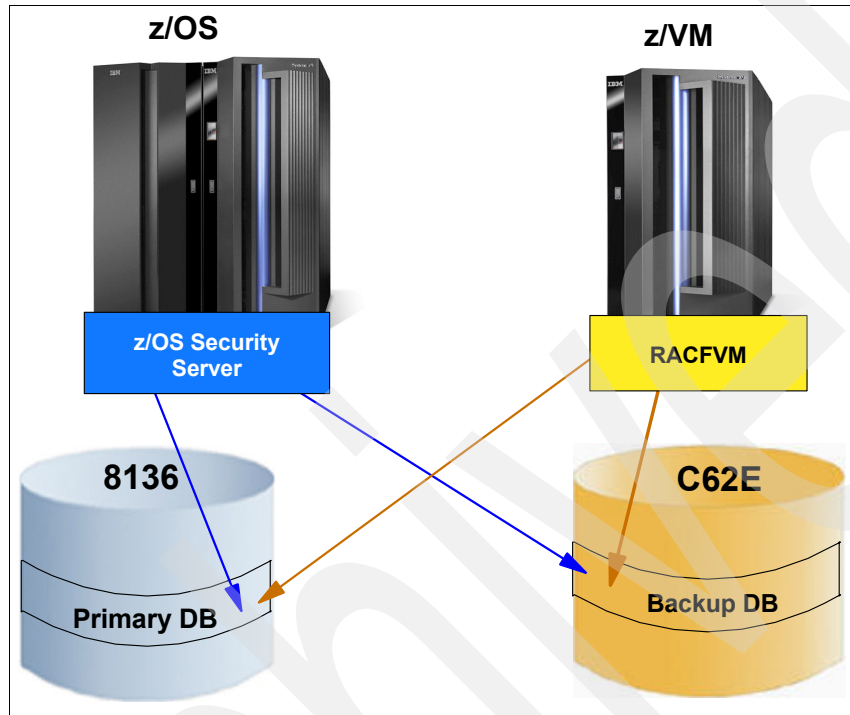


Figure 5-18 Sharing RACF database with z/OS

In this section, we discuss how to set up RACF to share the database with an existing z/OS RACF database. In our configuration, our target z/OS system is a z/OS V1R9. Sharing the database is possible without any further actions because the RACF database templates are compatible in both RACF's.

Note: We only include here the steps that are different from those that we describe in Chapter 2, “RACF feature of z/VM” on page 21.

Note: The RACF database sharing between z/OS and z/VM can be used only if RACF data sharing is not enabled on the z/OS system. When data sharing is used, RACF data is cached and maintained in sync by the Sysplex protocol.

Another limitation in sharing the RACF database between the two operating systems might depend on the GRS set up on the z/OS system. You need to make sure that GRS is not converting the hardware RESERVE operation to a global ENQ in order to maintain integrity between the z/OS and z/VM systems.

If your installation cannot share the database but still desire to share the RACF information, you might want to consider using the IBM Tivoli Directory Integrator product for synchronization between the z/OS and z/VM.

5.3.1 Planning RACF/VM installation

To configure RACF on z/VM for the scenario that we discuss, be sure to check the following steps before you begin the installation process:

1. Before running the RPIDIRECT task, as described in “Execute RPIDIRECT” on page 25, you need to check for user ID conflicts with the z/OS system. The ADDUSER issued to add the VM users to the z/OS database will fail if the user already exists on z/OS. Analyze every case for the need of renaming the user on one of the systems or be sure sharing will not create any problems. In our case, we had user ID *TCPIP* and *LDAPSRV* already defined on z/OS.
2. The primary and backup RACF databases on z/VM are required to be in separate volumes. If your z/OS has the databases on same volume, you need to move one of the databases to a different volume.
3. z/VM LPAR must have access to the z/OS volumes containing the databases.
4. You must have a user ID IBMUSER defined on z/OS, and you need to know the user ID’s password or need to define the user.
5. When sharing the z/VM database with z/OS V1.8 or higher, it is not necessary to update the database templates because they are already present in the shared z/OS RACF database.
6. You need to have High Level Assembler Product (HLASM) installed on z/VM.
7. Remember that if you are planning to use password phrase support on z/VM, the user could not have a password phrase-only user that would work on z/OS. The solution is to give the user both a password and a password phrase. Remember that there is no way you could force the user to use his password phrase on the VM side (unless you plan to have a password exit on the VM side which could enforce this).

5.3.2 Installing RACF on z/VM

To install RACF on z/VM, follow these steps:

1. Perform the tasks that we describe in Chapter 2, “RACF feature of z/VM” on page 21 until you reach “Remove ICHDEX01 and ICHRCX02 (optional)” on page 29. You need to verify whether your z/OS Security Server is using DES. If so, perform the step described in “Remove ICHDEX01 and ICHRCX02 (optional)” on page 29.
2. Prepare RACF/VM to access the z/OS database. Read the RACF 530 Program Directory “Sharing a RACF Database Information.” Issue the command to find the volumes where the z/OS database are allocated.

On z/OS issue the TSO RRVARY command shown in Example 5-14 (this command defaults to RRVARY LIST).

Example 5-14 RRVARY command sample

```
TSO RRVARY
RACF DATABASE STATUS:
ACTIVE USE  NUM VOLUME  DATASET
-----
YES  PRIM   1 BH6CAT   SYS1.RACFESA
YES  BACK   1 BH6ST2   SYS1.RACFBK
RRVARY COMMAND HAS FINISHED PROCESSING.
```

Enter the following command from the z/OS console (shown in Example 5-15):

```
d u,vol=BH6CAT and d u,vol=BH6ST2
```

Example 5-15 Sample of the Display Unit command

IEE457I	10.00.38	UNIT	STATUS	539
UNIT	TYPE	STATUS	VOLSER	VOLSTATE
8136	3390	A	BH6CAT	PRIV/RSDNT

IEE457I	10.01.23	UNIT	STATUS	541
UNIT	TYPE	STATUS	VOLSER	VOLSTATE
C62E	3390	A	BH6ST2	STRG/RSDNT

The command output shows that the primary database is on DASD 8136 and the backup database is on C62E. Their data set names are SYS1.RACFESA and SYS1.RACFBK. Remember that if the z/OS primary and backup databases are on *same volume*, you need to move one of the databases on z/OS to a different volume because RACF/VM needs them in separated volumes.

3. Change your system configuration file on Maint CF1 minidisk to allow the two z/OS volumes to be online to z/VM (Example 5-16). If the z/VM LPAR does not have access to the z/OS volumes, you need to use HCD to allow access. If they are on different machines, then you need to connect the machine where the z/VM runs to have access through ESCON® or FICON® to the DASD controller where the z/OS volumes reside.

Example 5-16 System configuration file sample

Online_at_IPL	2E20-2E2F	5A90-5A9F	CC31-CC49	C504-C505,		
	F101	F181	AB23-AB27	AC23-AC27	8136	C62E,

4. Change the system configuration file on Maint CF1 minidisk to set the DASD as shared, as shown in Example 5-17.

Example 5-17 System configuration file to define shared mdisk

Rdevice	8136	Type	DASD SHARED	YES
Rdevice	C62E	Type	DASD SHARED	YES

5. Verify that your system configuration file has been updated without errors:
 - access 193 z
 - cpsyntax system config
6. Copy the system configuration file to maint CF2 minidisk (Example 5-18).

Example 5-18 Copying the system configuration file

```
cprelease b
CPRELEASE request for disk B scheduled.
HCPZAC6730I CPRELEASE request for disk B completed.
Ready; T=0.01/0.01 14:09:25
acc cf1 z
DMSACP723I Z (CF1) R/O
Ready; T=0.01/0.01 14:09:36
cp link * cf2 cf2 mr
Ready; T=0.01/0.01 14:09:44
acc cf2 w
Ready; T=0.01/0.01 14:09:49
copyf system config z = w(rep
Ready; T=0.01/0.01 14:10:07
```

```

det cf2
DASD OCF2 DETACHED
Ready; T=0.01/0.01 14:10:28
cpaccess * cf2 b
CPACCESS request for mode B scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF2 in mode B completed.

```

7. Issue the command shown in Example 5-19 to update the system configuration file dynamically.

Example 5-19 Updating the system configuration file

```

cp set rdevice 8136 type dasd shared yes
HCPZRP6722I Created RDEV for device 8132.
1 RDEV(s) specified; 0 RDEV(s) changed; 1 RDEV(s) created
Ready; T=0.01/0.01 13:03:08
cp set rdevice C62E type dasd shared yes
HCPZRP6722I Created RDEV for device 8133.
1 RDEV(s) specified; 0 RDEV(s) changed; 1 RDEV(s) created
Ready; T=0.01/0.01 13:03:18
vary on 8136
8136 varied online
1 device(s) specified; 1 device(s) successfully varied online
Ready; T=0.01/0.01 13:04:15
vary on C62E
C62E varied online
1 device(s) specified; 1 device(s) successfully varied online
Ready; T=0.01/0.01 13:04:21

```

8. At this point, you can either dedicate the two volumes to the user ID RACFVM and user ID RACMAINT, or you can make a full-pack mdisk. In our configuration, we decided to use the DEDICATE statement on the directory. We comment out the minidisk 200 and 300 on both user IDs directory and add two DEDICATE statements (Example 5-20).

Example 5-20 Sample directory

```

USER RACFVM RACFVM 20M 20M ABCDEGH
ACCOUNT SYSTEMS
----- 5 line(s) not displayed -----
OPTION QUICKDSP MAXCONN 300
DEDICATE 0200 8136 BH6CAT
DEDICATE 0300 C62E BH6ST2
CONSOLE 0009 3215 T OPERATOR
----- 9 line(s) not displayed -----
*MDISK 200 3390 0838 017 53GW02 MW READ WRITE MULTIPLE
----- 3 line(s) not displayed -----
*MDISK 300 3390 1001 017 53GW02 MW READ WRITE MULTIPLE
----- 4 line(s) not displayed -----

USER RACMAINT RACMAINT 20M 20M ABCDEGH
----- 5 line(s) not displayed -----
OPTION QUICKDSP MAXCONN 300
DEDICATE 0200 8136 BH6CAT
DEDICATE 0300 C62E BH6ST2
CONSOLE 009 3215 T OPERATOR
----- 8 line(s) not displayed -----
*LINK RACFVM 200 200 MR
*LINK RACFVM 300 300 MR
----- 5 line(s) not displayed -----

```

9. You also need to update, assemble, linkedit, and activate ICHRDSNT to contain the z/OS database names (Example 5-21, Example 5-22, Example 5-23, and Example 5-24). This step is considered a local modification to RACF/VM, and you need to have High Level Assembler Product (HLASM) installed.

Example 5-21 Sample ICHRDSNT

```

ICHRDSNT CSECT
----- 10 line(s) not displayed -----
*****
*      DATA SET NAMES
*      FORMAT:
*          CL44'PRIMARY DSNAME',CL44'SECONDARY DSNAME',X'N1,X'N2'
*
*          N1=NUMBER OF RESIDENT BLOCKS
*          N2=FLAG FIELD
----- 7 line(s) not displayed -----
*****
NAME1   DC   CL44'SYS1.RACFESA',CL44'SYS1.RACFBK',X'64',X'81'
*****
                        END   ICHRDSNT

```

Example 5-22 Sample for assembling ICHRDSNT

```

Ready; T=0.01/0.01 11:50:29
vmfsetup 5vmrac30 racf
VMFSET2760I VMFSETUP processing started for 5VMRAC30 RACF
VMFUTL2205I Minidisk|Directory Assignments:
          String   Mode  Stat  Vdev  Label/Directory
VMFUTL2205I LOCALSAM  E    R/W  2C2   RAC2C2
----- 13 line(s) not displayed -----
VMFSET2760I VMFSETUP processing completed successfully
Ready; T=0.07/0.07 11:50:51
listf ichrdsnt assemble *
ICHRDSNT ASSEMBLE K1
ICHRDSNT ASSEMBLE U1
Ready; T=0.01/0.01 11:51:45
copyf ichrdsnt assemble k = = e
Ready; T=0.01/0.01 11:52:04
**we now change ichrdsnt assemble to match dataset names**
xedit ichrdsnt assemble e
Ready; T=0.01/0.01 11:55:10
q disk
LABEL  VDEV M  STAT  CYL TYPE BLKSZ  FILES  BLKS USED-(%) BLKS LEFT  B
----- 3 line(s) not displayed -----
RAC2C2 2C2  E   R/W   9 3390 4096      2        9-01      1611
----- 5 line(s) not displayed -----
RAC505 505  K   R/W  41 3390 4096     128      5058-69     2322
Ready; T=0.01/0.01 13:04:48
vmfhlasml ichrdsnt 5VMRAC30 RACF
VMFASM2760I VMFHLASM processing started
DMSUPD181E No update files were found
VMFASM1907I Assembling ICHRDSNT
"Assembler Done No Statements Flagged
VMFASM2507I ICHRDSNT TXT00000 created on your A-disk for use in a VMSES/
environment

```

```

VMFASM2760I VMFHLASM processing completed successfully
Ready; T=0.01/0.01 13:06:30
copyf ichrdsnt txt00000 a = txtl0001 e
Ready; T=0.01/0.01 13:07:36
rename ichrdsnt assemble e = asml0001 e
Ready; T=0.01/0.01 13:08:52
erase ichrdsnt txt00000 a
Ready; T=0.01/0.01 13:09:20

```

Example 5-23 ICHRDSNT linkedit sample

```

vmfsim logmod 5vmrac30 vvtlcl e tdata :mod lcl0001 :part ichrdsnt txt
Ready; T=0.02/0.02 13:10:21
vmfsim logmod 5vmrac30 vvtlcl e tdata :mod lcl0001 :part ichrdsnt asm
Ready; T=0.02/0.02 13:10:33
vmfbld ppf 5vmrac30 racf RPIBL505 ichrdsnt (all
VMFBLD2760I VMFBLD processing started
----- 7 line(s) not displayed -----
VMFBLD1851I (2 of 2) VMFBDCOM processing RPIBL505 EXEC U, target is BUILD
----- 3 line(s) not displayed -----
VMFBLD2180I There are 1 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
Ready; T=2.10/2.15 13:19:29
vmfbld ppf 5vmrac30 racf RPIBL505 ichrdsnt (all
VMFBLD2760I VMFBLD processing started
----- 7 line(s) not displayed -----
VMFBLD2180I There are 0 build requirements remaining
VMFBLD2760I VMFBLD processing completed successfully
Ready; T=1.99/2.06 13:20:49
listf ichrdsnt * k
ICHRDSNT ASSEMBLE K2
ICHRDSNT TEXT K2
Ready; T=0.01/0.01 13:44:28
listf racflink * *
RACFLINK LKEDIT K5
RACFLINK LOADLIB K2

```

Example 5-24 Enabling the new ICHRDSNT

```

link RACFVM 305 305 MR
acc 505 e
acc 305 f
vmfcopy racflink * e = f (prodid 5VMRAC30%ACF replace oldd

```

10. Delete the ICHRCX02 exit as described in “Delete the ICHRCX02 exit” on page 32.
11. Make sure that you have all the information to log on as IBMUSER.
12. Perform the task described in 2.2.3, “Update the RACF database and options” on page 36 using the user ID IBMUSER.
13. Now continue with the list of tasks described beginning with “Set RACF options (optional)” on page 39 until you complete the installation.

5.3.3 Summary

This section described how to implement RACF database sharing between a z/VM system and a z/OS system where the RACF data sharing feature is not activated.

5.4 Using a central z/OS IBM Tivoli Directory Server

When a Linux farm on System z is implemented, it is very common that it is running in a logical partition that resides on the same server as a z/OS image, where the z/OS image could be either a monoplex or a member of a sysplex. The scenario that we discuss in this section describes how you can administer the Linux users and z/OS RACF environment from the z/OS system. z/VM is administrated separately.

For this environment, we use IBM Tivoli Directory Server for z/OS server and RACF to control the Linux logon. Figure 5-19 describes the LDAP structure. If the LDAP server and the Linux farm are installed on the same System z, you can use a HiperSocket LAN to connect each Linux server to the LDAP server. This HiperSocket LAN is used for the local network traffic while all the LDAP exchanges are done within the CPU without increase on the local network traffic.

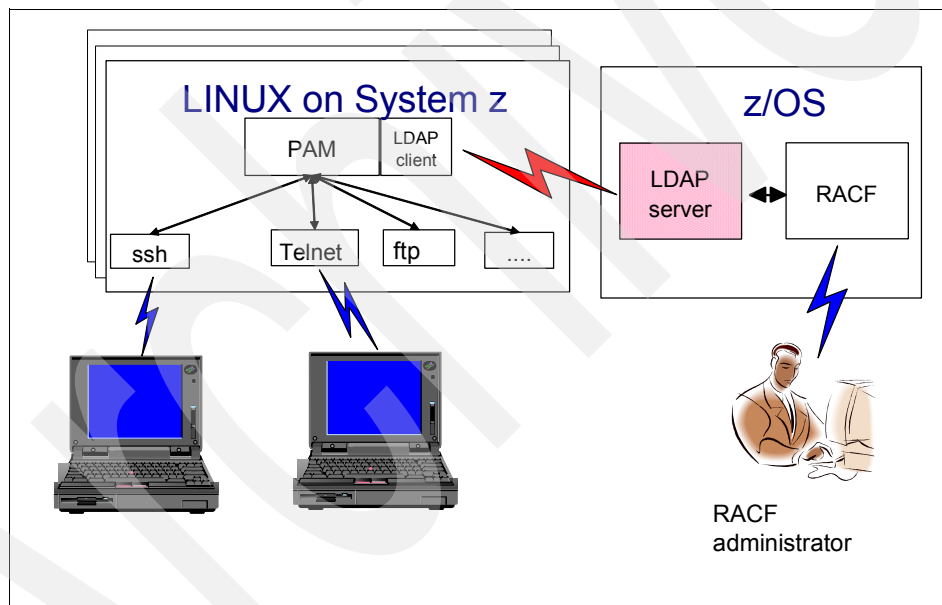


Figure 5-19 Using IBM Tivoli Directory Server for z/OS server

5.4.1 LDAP architecture

For this scenario, IBM Tivoli Directory Server for z/OS server is used by the Linux farm during the Linux logon and to complement the logon information with additional parameters, such as the shell Linux used, the user home directory, the UID, and the GID. The IBM Tivoli Directory Server for z/OS server runs off the UNIX System Services.

IBM Tivoli Directory Server for z/OS contains multiple backends that are used by LDAP server to access the different LDAP databases. Figure 5-20 shows an overall picture of the IBM Tivoli Directory Server for z/OS.

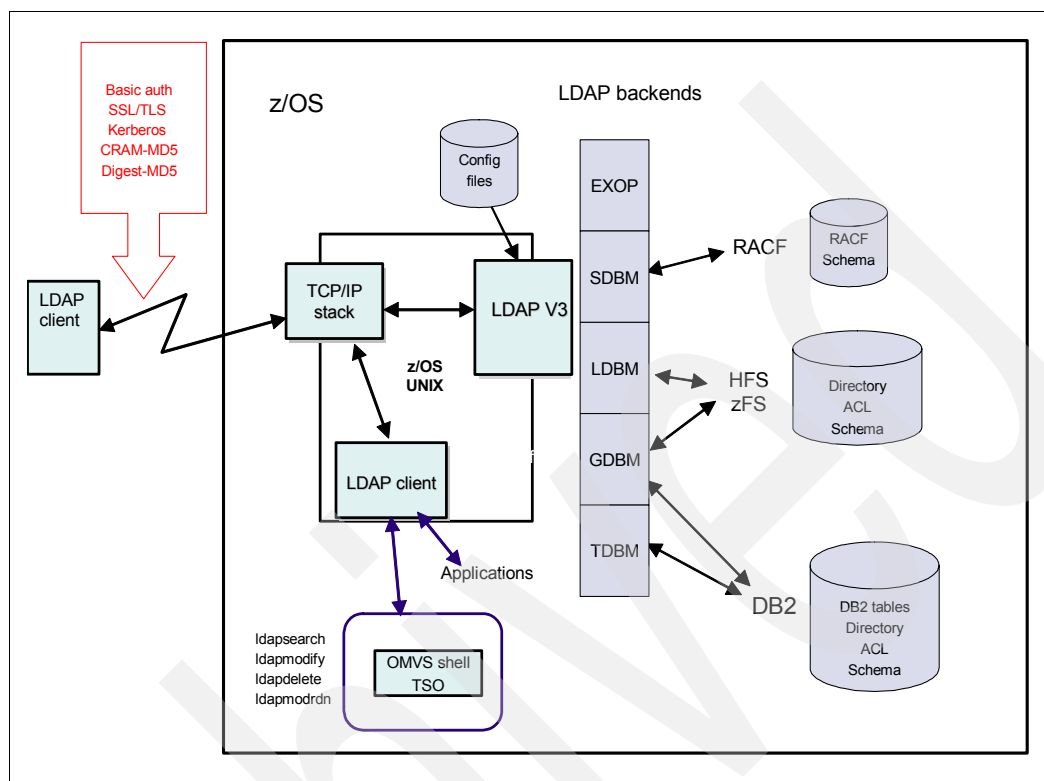


Figure 5-20 IBM Tivoli Directory Server for z/OS architecture

The z/OS backends are:

- ▶ LDBM is the basic LDAP database as in z/VM. It is used in conjunction with native authentication. Native authentication permits to store the logon passwords in RACF and uses RACF to validate user ID and password for a Linux logon.
- ▶ GDBM permits to track the changes made by the different LDAP backends (LDBM, SDBM and TDBM).
- ▶ SDBM is a RACF interface and permits some limited RACF administration. It could be use to delegate some parts of the RACF Linux servers administration.
- ▶ TDBM is specific to z/OS. LDAP will store data in z/OS DB2 table. TDBM could use native authentication (RACF) as LDBM.
- ▶ EXOP backend permits to offer LDAP service to applications using XML-binary Optimized Packaging (XOP) protocol like for example WebSphere MQ and CICS®.

5.4.2 Implementing the IBM Tivoli Directory Server for z/OS

In our sample configuration, we implemented a LDBM server with native authentication. Native authentication permits to check with RACF the Linux logon user ID and password. z/OS user and password could be used to log on to the Linux server or you could use a generic RACF user for multiple Linux users.

Figure 5-21, shows a sample of the LDAP directory.

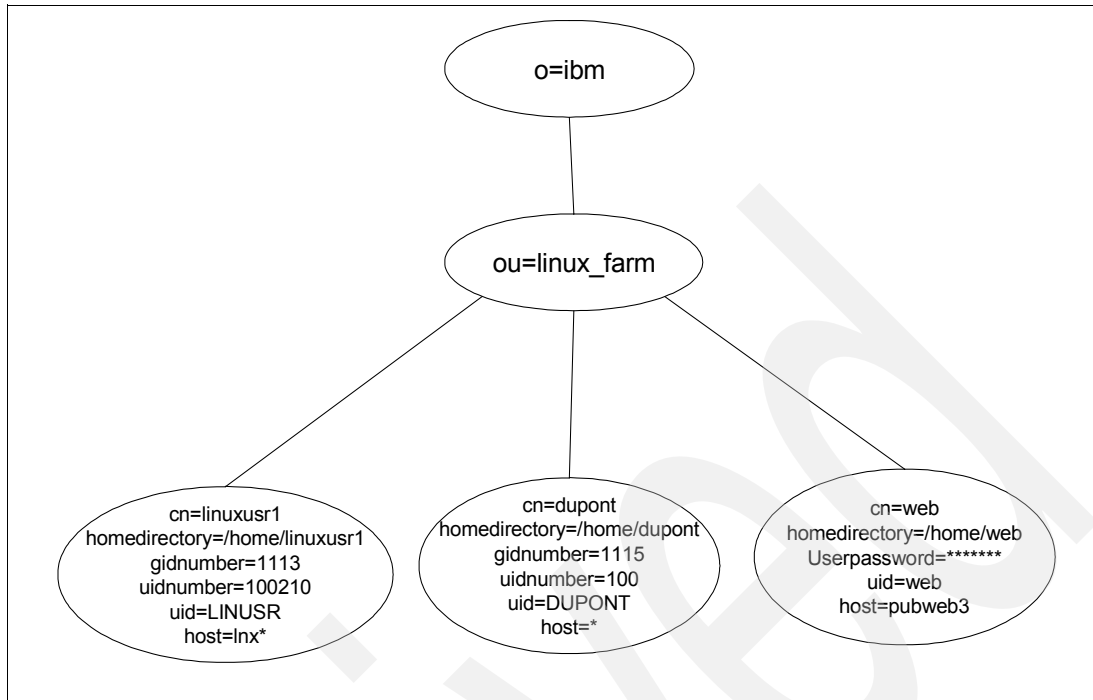


Figure 5-21 LDBM with native authentication

The top directory is *ibm* and the subdirectory is *linux_farm*.

Different possibilities are available in defining a user in the LDAP directory, for example:

► user linuxusr1

This user can log on every and only system matching *lnx**. When the *linuxusr1* logs on for the first time on a *lnx** system, for example on *lnx006* system, the *linux* directory */home/linuxusr1* is created during the initial logon. The user ID sent to RACF for authentication will be *LINUXR* as coded in *UID*. The password checked by RACF is the one given during the Linux logon.

► user dupont

This user could be used on every system under the LDAP server control.

► user web

This user could be used only on the *pubweb3* server and the password is stored directly in the LDAP server. In this case, RACF doesn't control the logon.

The LDAP configuration files represent this configuration:

► ds.envvars (Example 5-25)

Example 5-25 The *ds.envvars* file

```

# @(#)98
# 1.21.2.3
# 2/6/07 11:28:40
# *****
# Licensed Materials - Property of IBM
# 5694-A01
# (C) Copyright IBM Corp. 2005
# *****

```

```
#
# *****
# This was generated by dsconfig on Wed Mar 14 11:53:09 2007.
# This was generated by dsconfig with the input
# file: '/u/bari/ds.profile'.
# WARNING: Any manual updates to this file will be lost
# if dsconfig is executed again and the OUTPUT_DATASET option is
# set to 'BARI.GLD.CNFOUT'
#
# The following substitutions occurred in this file:
#   LDAP_LANG - En_US.IBM-1047
#   USR_LPP_ROOT - /usr/lpp/ldap
#   TZ - GMT0
# *****
NLSPATH=/usr/lpp/ldap/lib/nls/msg/%L/%N
LANG=En_US.IBM-1047
#LIBPATH=/usr/lpp/ldap/lib
#TZ=GMT0
```

- ds.config (Example 5-26, only the active parameters are shown)

Example 5-26 The ds.config file

```
# =====
#
# GLOBAL SECTION
#
# =====
adminDN cn=Admin
adminPW secret
commThreads 10
listen ldap://:389
maxConnections 65535
schemaPath /var/ldap1/schema
sendV3StringsOverV2As UTF-8
sizeLimit 500
timeLimit 3600
validateIncomingV2strings on

#-----
# LDBM-specific CONFIGURATION SETTINGS
#-----
database LDBM GLDBLD31/GLDBLD64

suffix "o=itso"
suffix "f=FR"
suffix "ou=pssc montpellier,o=ibm,c=fr"
suffix "ou=MG_NATIVE,o=IBM,c=US"
suffix "ou=linux_farm,o=IBM"

databaseDirectory /var/ldap1/ldbm
extendedGroupSearching on
nativeAuthSubtree "ou=MG_NATIVE,o=IBM,c=US"
nativeAuthSubtree "ou=linux_farm,o=IBM"
nativeUpdateAllowed on
pwEncryption SHA
useNativeAuth all
```

5.4.3 Verifying the LDAP server

After you start the IBM Tivoli Directory Server for z/OS server, you can check that the access to the LDBM backend is successful using the **ldapsearch** command from a Linux LDAP client:

```
ldapsearch -h 9.12.4.126 -D cn=admin -w secret -b "ou=linux_farm,o=ibm"
"objectclass=*"
```

Note: Remember to add the **-x** option to **ldapsearch** in case you receive the following message:

```
ldap_sasl_interactive_bind_s: Unknown authentication method (-6)
additional info: SASL(-4): no mechanism available: No worthy mechs found
```

In our case, we expected to see as a result from the **ldapsearch** command **-h 9.12.4.126 -D cn=admin -w secret -x -b "ou=linux_farm,o=ibm" "objectclass=*" :**

```
dn: ou=linux_farm,o=ibm
objectclass: top
objectclass: organizationalUnit
ou: linux_farm
```

5.4.4 Loading the schema

If LDAP LDBM database is used for the first time, you have to load two LDIF files that contain schemes. To load the schema, you can use either the z/OS client or the Linux client:

```
ldapmodify -h 9.12.4.126 -D cn=admin -w secret -a -f ./user.schema.ldif
ldapmodify -h 9.12.4.126 -D cn=admin -w secret -a -f ./ibm.schema.ldif
```

5.4.5 Loading the Linux specific schema

Linux server requires one additional schema to provide the capability of storing specific Linux data such as UID, GID number, home directory, login shell, and so forth in the LDAP database.

This new schema is available on the FTP site:

<ftp://www.redbooks.ibm.com/redbooks/REDP0221>

This file is loaded on a Linux for System z. We use a Linux LDAP client to load the schema data in z/OS LDBM database.

Note: The LDIF file has to be transferred using FTP in ASCII mode. Using cut and paste or TN3270 file transfer results in truncated records.

Some changes are required before loading the schema file on z/OS to avoid errors during the **ldapmodify** command execution. Follow these steps:

1. Edit the file as shown in Example 5-27 using vi and modify the lines in bold.

Example 5-27 Changes to the schema file

```
# File generated at 6:31:34 PM on 2/14/02 from IBM LDAP schema version 1.5
# Module Name: nisSchema (v 2) ( nisSchema-oid )
# Dependencies:
#
```

```

dn:cn=schema
changetype: modify
replace: attributetypes
attributetypes: (
    1.3.6.1.1.1.1.24
    ...
    ...
ibmattributetypes: (
    1.3.6.1.1.1.1.0
    DBNAME( 'uidNumber' 'uidNumber' )
    ACCESS-CLASS normal
    LENGTH 11
    )

dn:cn=schema
changetype: modify
replace: objectclasses
objectclasses: (
    1.3.6.1.1.1.2.12
    NAME 'bootableDevice'
    ...

```

2. Change the minus sign before the second `dn:cn=schema` by one blank character.
3. When the file is changed, use the `ldapmodify` command to write the new schema in the LDAP z/OS server:

```
ldapmodify -h 9.12.4.126 -D cn=admin -w secret -f ./nisSchema.ldif -x -a
```

5.4.6 Adding a Linux user in IBM Tivoli Directory Server for z/OS

To add a Linux user, perform the following:

1. Create a `user.ldif` files containing the lines shown in Example 5-28.

Example 5-28 The user.ldif file

```

dn: cn=linuxusr1,ou=linux_farm,o=ibm
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: ibm-nativeAuthentication
objectclass: posixAccount
cn: linuxusr1
sn: linuxusr1
loginshell: /bin/bash
uidnumber: 10020210
gidnumber: 1113
uid: linuxusr2
homedirectory: /home/linuxusr2
ibm-nativeId: LAMBDA

```

2. Load the user definition in the z/OS LDBM server:

```
ldapmodify -h 9.12.4.126 -D cn=admin -w secret -x -a -f ./user.ldif
```

3. From a z/OS RACF administrator session, verify the existence of the LAMBDA user, as defined in the `ibm-nativeId` field of the LDAP entry. If the user does not exist, define the LAMBDA user to RACF.

5.4.7 Adapting Linux servers to use LDAP service

If your Linux servers are not customized to work with LDAP, use the method described in Chapter 4, “Implementing Pluggable Authentication Modules LDAP for Linux servers” on page 137.

If the Linux servers are already working with PAM/LDAP service, LDAP configuration files are changed with either one of the following methods:

- ▶ With YaST as described in 4.2, “Configuring PAM LDAP and NSS” on page 138. Because PAM LDAP and NSS are already installed, you only need to change the IP address and the DN information.
- ▶ With vi. Only the `/etc/ldap.conf` file needs to be changed. Changes are in bold in Example 5-29.

Example 5-29 The `/etc/ldap.conf` file

```
#
# This is the configuration file for the LDAP nameservice
# switch library, the LDAP PAM module and the shadow package.
#
# Your LDAP server. Must be resolvable without using LDAP.
host      9.12.4.126
# The distinguished name of the search base.
base      ou=linux_farm,o=ibm

# The LDAP version to use (defaults to 3
# if supported by client library)
ldap_version 3

# Don't try forever if the LDAP server is not reachable
bind_policy  soft

# Hash password locally; required for University of
# Michigan LDAP server, and works with Netscape
# Directory Server if you're using the UNIX-Crypt
# hash mechanism and not using the NT Synchronization
# service.
pam_password  racf

# Redirect users to a URL or somesuch on password
# changes.
#pam_password_prohibit_message Please visit http://internal to change your
# password.

# OpenLDAP SSL mechanism
# start_tls mechanism uses the normal LDAP port, LDAPS typically 636
ssl        no
nss_map_attribute  uniqueMember member
pam_filter    objectclass=posixAccount
nss_base_passwd ou=linux_farm,o=ibm
nss_base_shadow ou=linux_farm,o=ibm
```

```
nss_base_group  ou=linux_farm,o=ibm
tls_checkpeer  no
#ssl on
```

5.4.8 Linux login test

At this point, the Linux servers are modified to work with PAM LDAP and NSS. An IBM Tivoli Directory Server for z/OS server and RACF are ready to handle the Linux login.

To test, open an ssh session to the Linux where the modified for PAM LDAP is available and log on as linuxusr1.

If everything is correct, you should receive the following message:

```
login as: linuxusr1
Using keyboard-interactive authentication.
Password:
Creating directory '/home/linuxusr1'.
Creating directory '/home/linuxusr1/.fonts'.
Creating directory '/home/linuxusr1/.mozilla'.
Creating directory '/home/linuxusr1/.xemacs'.
Creating directory '/home/linuxusr1/bin'.
Creating directory '/home/linuxusr1/Documents'.
Creating directory '/home/linuxusr1/public_html'.
linuxusr1@lnxsul:~>
```

5.4.9 Summary

This scenario shows that you can use z/OS to control Linux login. z/OS and Linux security administration is performed on the z/OS side. Linux and z/OS users are able to use the same user and password on both operating systems.

You should also read 5.2, “Sharing RACF database with another z/VM system” on page 155 or 5.5, “Synchronizing LDAP/RACF database with IBM Tivoli Directory Integrator” on page 184 if your installation is looking at a solution where the z/VM RACF administration is centralized.

5.5 Synchronizing LDAP/RACF database with IBM Tivoli Directory Integrator

If your configuration is running z/OS and z/VM with multiple Linux instances and does not have the capability to share the RACF database, you can still centralize the RACF administration task by synchronizing data using IBM Tivoli Directory Integrator (Directory Integrator). This product is a powerful and compact middleware component that can retrieve, process and deliver data using a wide variety of protocols and methods.

Directory Integrator can synchronize user and group adds, group connections, user password changes and other user profile changes. In most cases, you manage RACF on z/OS and synchronize those changes to z/VM because its LDAP does offer GDBM and the associated changelog which tracks changes to SDBM.

In this configuration, Directory Integrator binds to both the z/OS and z/VM LDAP servers. It tracks changes made to z/OS and reflected in GDBM, retrieves the changed data from the z/OS SDBM and propagates those changes to the z/VM SDBM. RACF password enveloping makes password changes available and Directory Integrator can decrypt the password envelope and update z/VM passwords to match z/OS passwords for corresponding user identities. The main RACF administration tasks like user creation, password change, connect to groups, group definition are more likely performed on the z/OS side because the z/VM LDAP does not provide the GDBM backend capability.

Figure 5-22 presents an overview of the configuration.

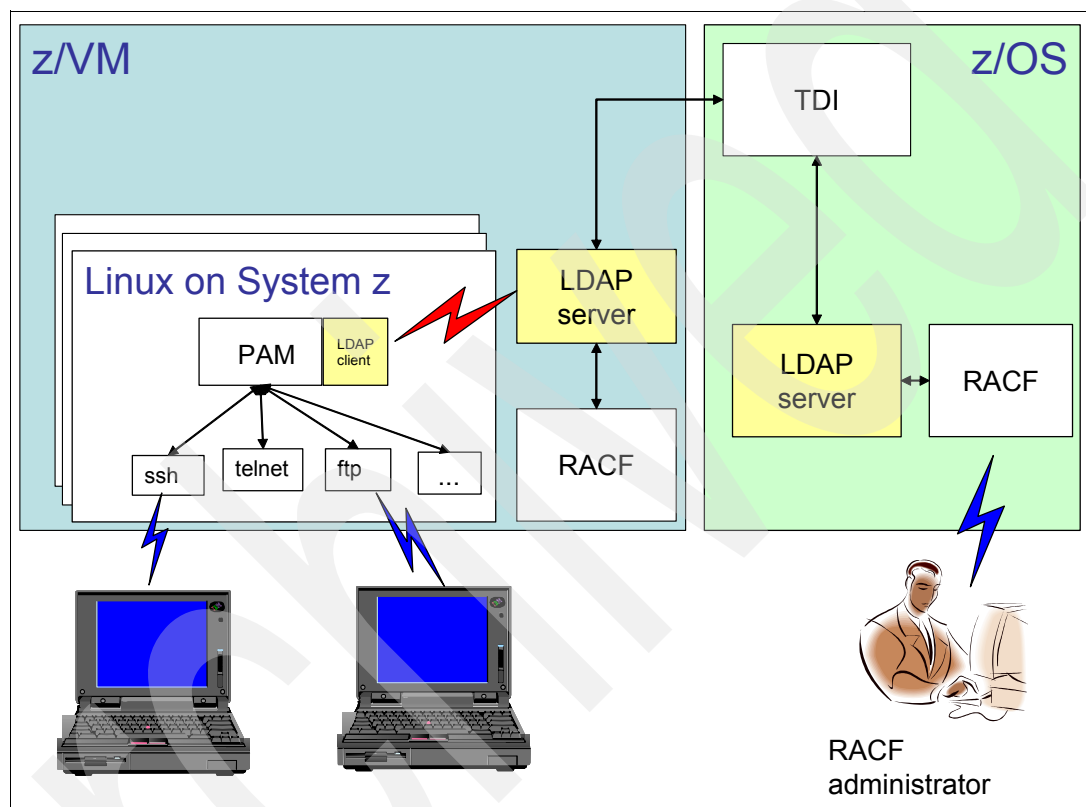


Figure 5-22 IBM Tivoli Directory Integrator configuration overview

In our configuration, we implement two modes of operation, a batch synchronizer and an event-driven synchronizer. Directory Integrator does this work with a set of components linked into a procedure called an AssemblyLine. A developer or implementer builds AssemblyLines from (mostly) pre-built components to meet specific business requirements. The key component type is called a Connector. Connectors read, write or search data sources and move data into or out of AssemblyLines. 5.5.1, “Architecture” has more information about Directory Integrator and 5.5.8, “IBM Tivoli Directory Integrator configuration” details the AssemblyLines built for this solution. Appendix C, “Additional material” on page 321 contains the instructions on how to download the Directory Integrator configuration file.

One AssemblyLine synchronizes existing information as a batch operation. It read user profile data for all z/OS RACF users specifically designated for synchronization and writes this information into z/VM RACF. Alternatively all z/OS users not specifically excluded are synchronized. The Directory Integrator sample AssemblyLine built for this book—called syncExisting in the configuration—implements the latter approach. Users already existing in z/VM get updated if the attributes in z/VM do not match z/OS profile.

If z/OS passwords are enveloped for existing users as described in 5.5.7, “RACF changes” on page 194 this AssemblyLine could propagate those passwords to z/VM. However, this might surprise users in many organizations unless they are fully briefed in advance. Therefore, as a suggested best practice, the syncExisting AssemblyLine does not synchronize passwords.

The second AssemblyLine—called syncChanges in the example configuration—pushes data to z/VM only in response to z/OS RACF profile changes. It watches the z/OS LDAP changelog, which is linked to the RACF GDBM backend. It writes changed z/OS data, including enveloped passwords to z/VM.

The sample configuration AssemblyLines only synchronizes RACF users, but they can be extended to perform more complex operations, including group updates.

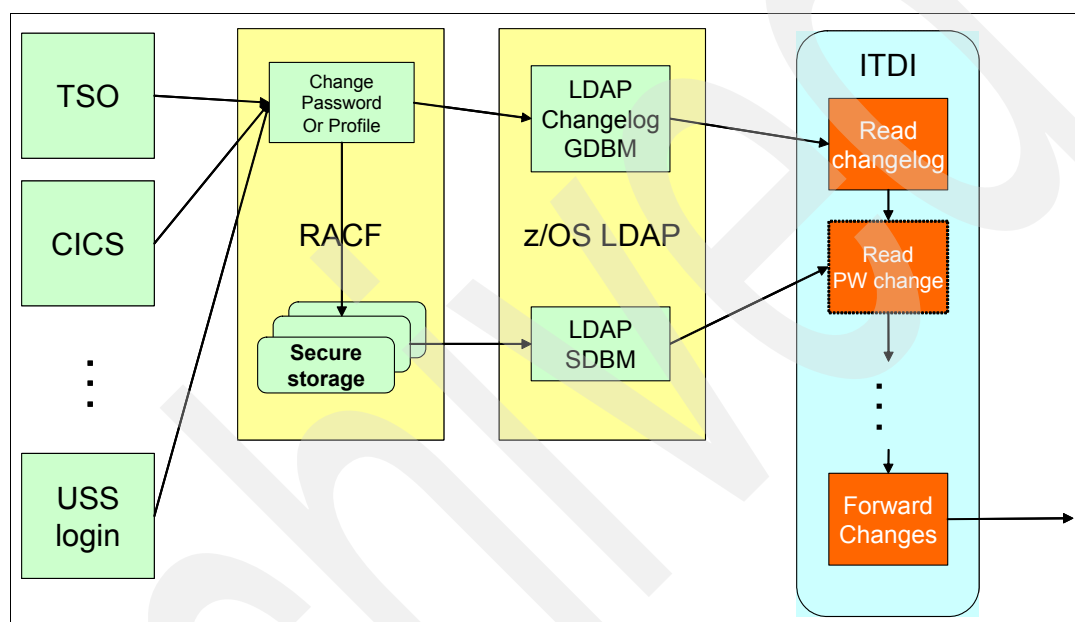


Figure 5-23 IBM Tivoli Directory Integrator and z/OS

5.5.1 Architecture

Tivoli Directory Integrator reads and writes RACF data through the z/OS and z/VM LDAP servers. In both case, RACF profile data is mirrored in the LDAP SDBM backend.

Because RACF exposes this data through LDAP, Directory Integrator can access it by binding to LDAP with a RACF identity. In general, this identity needs the RACF attribute SPECIAL to have the access rights it needs. Always, give Directory Integrator a unique RACF ID so logs will reflect all user profile changes performed by this automated process. All the available RACF fields are described in *z/VM LDAP Administration Guide*, SC24-6140.

Directory Integrator is Tivoli Security application that is bundled with several other IBM Tivoli, WebSphere, and Lotus® products as well as sold as a standalone product to customers. It is widely used within IBM itself also because it is flexible solution to a variety of data synchronization problems. Directory Integrator is a Java™ application and runs on distributed platforms including Windows, Linux and major UNIX implementations as well as i5/OS®, z/OS, and Linux on System z.

The key concept in Tivoli Directory Integrator is the AssemblyLine, which is a thread of execution moving and transforming data. The term AssemblyLine also refers the configuration item detailing the makeup and operation of a specific AssemblyLine. The actual configuration

is stored in a XML-formatted file which is loaded into memory by Directory Integrator and then executed. As noted above, AssemblyLines are built with Connectors that touch both external systems and the AssemblyLine itself. Figure 5-24 shows a typical AssemblyLine. The yellow-colored bus in the diagram represents the a package of data moving down the AssemblyLine called the work entry object. Each Connector uses or manipulates the work entry with the goal of delivering the right data to the target system at the end of the AssemblyLine.

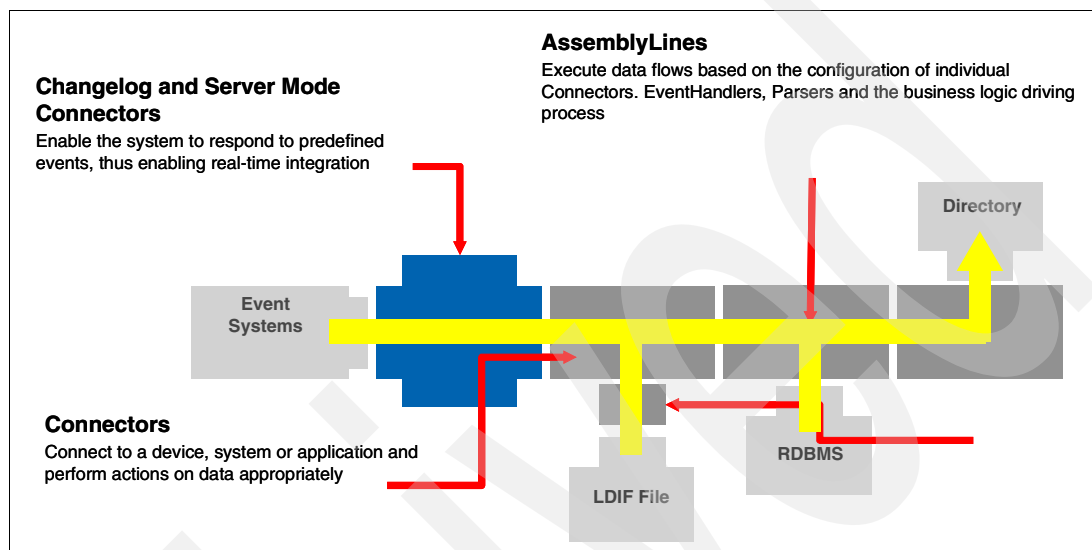


Figure 5-24 IBM Tivoli Directory Integrator Assembly Line

In our configuration, Directory Integrator makes connections to z/OS and z/OS with its built-in LDAP Connector. This Connector has performs all the functions of a suite of command line LDAP utilities or custom-written program. Also, the event-driven AssemblyLine syncChanges polls the z/OS GDBM IBM Tivoli Directory Server backend or changelog with a specialized LDAP Connector called the z/OS IBM Tivoli Directory Server Changelog Connector. This connector keeps track of changes it has already processed by writing the changenumber into a special database called the System Store. This database persists this information even if Directory Integrator goes offline, so when Directory Integrator restarts it will process only new changes.

AssemblyLines contain several different types of components in addition to Connector. Our example solution uses an Attribute Map component, which holds transformations of z/OS into corresponding z/VM data. For example, one line in the Attribute Map creates a RACFOWNER attribute for z/VM to correspond to the one in z/OS but inserts a default value if the z/OS RACFOWNER does not have an account on z/VM.

Directory Integrator solutions include some scripting. Most of the scripting occurs in Connector hooks, which allow the implementer to control and modify any aspect of the AssemblyLine operation. The Before Execute hook of one Connector in syncChanges, for example, tests a value to determine if the change being processed is a password change or other data. If it is a password change, the script dynamically changes the configuration of the LDAP connection so it will retrieve the enveloped password. See 5.5.8, "IBM Tivoli Directory Integrator configuration" and Appendix C, "Additional material" on page 321 for details.

5.5.2 RACF password mirroring

To implement password mirroring between RACF servers, the password enveloping feature needs to be activated in z/OS RACF. Current level of z/VM RACF does not support Password Enveloping, so forth the mirroring can only happen in one direction. Only z/OS password could be mirrored to the z/VM RACF server.

Figure 5-25 explains the steps to mirror a RACF password.

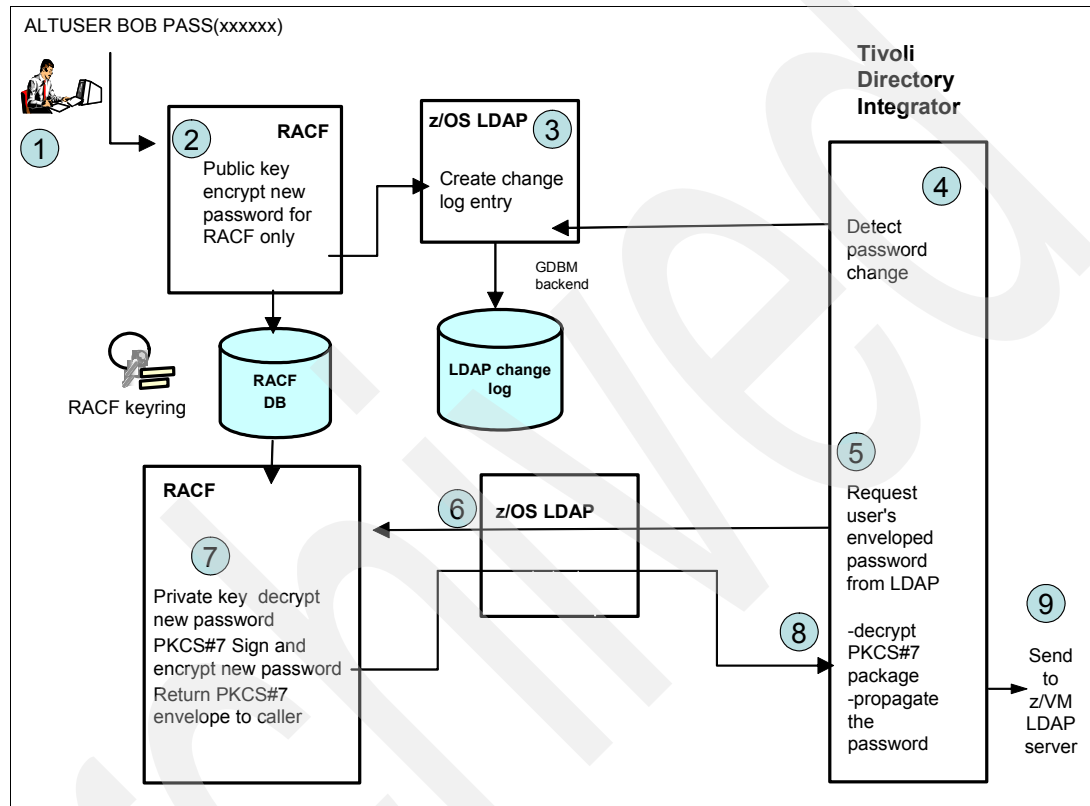


Figure 5-25 Password enveloping

When a password is changed on z/OS side, the process is:

1. RACF administrator or the user changes a password.
2. RACF server encrypts the password received with a public key.
3. RACF send a log entry to GDBM backend.
4. IBM Tivoli Directory Integrator server is tracking GDBM log change.
5. As soon as a change in GDBM log is detected, IBM Tivoli Directory Integrator analyze it. If it is a password change, IBM Tivoli Directory Integrator requests to the LDAP server an access to the RACF password changed.
6. LDAP SDBM backend requests the new password to RACF.
7. RACF verify the correct grant of the requester and if it is ok, it decrypts the new password using its private key and encrypt the new user password with a PKCS #7 key and send it to the IBM Tivoli Directory Integrator server.
8. IBM Tivoli Directory Integrator decrypts the password and propagates it to z/VM RACF.

5.5.3 LDAP backends

IBM Tivoli Directory Integrator is able to work with any backend that is available in the LDAP servers. The dialog between IBM Tivoli Directory Integrator and the LDAP servers uses LDAP protocol. To mirror RACF data, IBM Tivoli Directory Integrator uses LDAP SDBM backend and it requires RACF Special authority to be able to read or write RACF user information.

GDBM backend is also used to retrieve the changes. In our scenario, GDBM is activated only on the z/OS side because we decided to process only the change from z/OS to z/VM and because RACF change logging has not been implemented on z/VM.

5.5.4 z/VM LDAP configuration

In z/VM LDAP server, the configuration file is changed to add the SDBM backend function.

To adapt the z/VM LDAP server parameters:

1. Log on as user TCPMAINT.
2. XEDIT the DS CONF file, as shown in Example 5-30.

The first line is changed. LDAP administrator is checked directly by RACF rather than to use Native Authentication as before. This change is optional when SDBM and LDBM are running, but if you have only SDBM backend then Native Authentication is no more available.

The database and suffix keywords describe the SDBM backend.

Example 5-30 The DS CONF file

```
adminDN "racfid=LDAPADM2,profiletype=user,cn=RACFVM"
allowAnonymousBinds on
commThreads 10
listen ldap://:389
logfile /var/ldap/gldlog.output
maxConnections 65535
sendV3StringsOverV2As UTF-8
sizeLimit 500
timeLimit 3600
validateIncomingV2strings on
database SDBM GLDBSD31
suffix "cn=RACFVM"
database LDBM GLDBLD31
suffix "o=ibm"
useNativeAuth all
nativeUpdateAllowed on
nativeAuthSubtree "o=ibm"
```

3. Save the DS CONF file and restart the LDAPSRV2 server to activate the new configuration file.

4. Verify the changes using one of the following methods:

– a CMS user:

```
LDAPSRCH -h 9.12.4.191 -D racfid=LDAPADM2,profiletype=user,cn=RACFVM -w  
ITS07471 -b racfid=LDAPADM2,profiletype=user,cn=RACFVM "objectclass=*
```

– a Linux user:

```
ldapsearch -h 9.12.4.191 -x -D racfid=LDAPADM2, profiletype=user,  
cn=RACFVM -w ITS07471 -b racfid=LDAPADM2, profiletype=user, cn=RACFVM  
"objectclass=*
```

If the changes were successful, you should receive the messages shown in Example 5-31.

Example 5-31 Changes to configuration file successful

```
# extended LDIF  
#  
# LDAPv3  
# base <racfid=LDAPADM2,profiletype=user,cn=RACFVM> with scope subtree  
# filter: objectclass=*  
# requesting: ALL  
#  
# LDAPADM2, USER, RACFVM  
dn: racfid=LDAPADM2,profiletype=USER,cn=RACFVM  
racfid: LDAPADM2  
racfauthorizationdate: 07/17/07  
racfowner: RACFID=JIGUET,PROFILETYPE=USER,CN=RACFVM  
racfpasswordinterval: 30  
racfpasswordchangedate: 07/23/07  
racfdefaultgroup: RACFID=SYS1,PROFILETYPE=GROUP,CN=RACFVM  
racflastaccess: 07/30/07/11:36:17  
racflogondays: SUNDAY  
racflogondays: MONDAY  
racflogondays: TUESDAY  
racflogondays: WEDNESDAY  
racflogondays: THURSDAY  
racflogondays: FRIDAY  
racflogondays: SATURDAY  
racflogontime: ANYTIME  
racfconnectgroupname: RACFID=SYS1,PROFILETYPE=GROUP,CN=RACFVM  
racfhavepasswordenvelope: NO  
racfattributes: PASSWORD  
objectclass: RACFBASECOMMON  
objectclass: RACFUSER  
  
# search result  
search: 2  
result: 0 Success  
  
# numResponses: 2  
# numEntries: 1
```

5.5.5 IBM Tivoli Directory Server for z/OS configuration

On the z/OS side, you change the LDAP server to activate SDBM and GDBM backends as shown in Example 5-32.

Example 5-32 Changes for z/OS configuration

```
# =====
#
# GLOBAL SECTION
#
# =====
adminDN cn=Admin
adminPW secret
commThreads 10
listen ldap://:389
maxConnections 65535
schemaPath /var/ldap1/schema
sendV3StringsOverV2As UTF-8
sizeLimit 500
timeLimit 3600
validateIncomingV2strings on

#-----
# SDBM-specific CONFIGURATION SETTINGS
#-----
database SDBM GLDBSD31/GLDBSD64

suffix "o=RACFZOS"

#-----
# GDBM-specific CONFIGURATION SETTINGS
#-----
database GDBM GLDBGD31/GLDBGD64
changeLogging on

#-----
# LDBM-specific CONFIGURATION SETTINGS
#-----
database LDBM GLDBLD31/GLDBLD64

suffix "o=itso"
suffix "f=FR"
suffix "ou=pssc montpellier,o=ibm,c=fr"
suffix "ou=MG_NATIVE,o=IBM,c=US"
suffix "ou=linux_farm,o=IBM"

databaseDirectory /var/ldap1/ldb
extendedGroupSearching on
nativeAuthSubtree "ou=MG_NATIVE,o=IBM,c=US"
nativeAuthSubtree "ou=linux_farm,o=IBM"
nativeUpdateAllowed on
pwEncryption SHA
useNativeAuth all
```

After you have made the changes, check the SDBM availability using the following command:

```
ldapsearch -h 9.12.4.126 -x -D "racfid=jjiguet,profiletype=user,cn=RACFZOS" -w  
ITS07471 -b "racfid=LDAPSRV,profiletype=user,cn=RACFZOS" "objectclass=*
```

You should receive the messages shown in Example 5-33.

Example 5-33 Messages

```
# extended LDIF  
#  
# LDAPv3  
# base <racfid=LDAPSRV,profiletype=user,cn=RACFZOS> with scope subtree  
# filter: objectclass=*  
# requesting: ALL  
#  
  
# LDAPSRV, USER, RACFZOS  
dn: racfid=LDAPSRV,profiletype=USER,cn=RACFZOS  
racfid: LDAPSRV  
racauthorizationdate: 03/14/07  
racfowner: RACFID=HAIMO,PROFILETYPE=USER,CN=RACFZOS  
racpasswordinterval: 254  
racprogrammername: LDAP ID  
racdefaultgroup: RACFID=SYS1,PROFILETYPE=GROUP,CN=RACFZOS  
racflastaccess: 07/27/07/10:40:55  
racflogondays: SUNDAY  
racflogondays: MONDAY  
racflogondays: TUESDAY  
racflogondays: WEDNESDAY  
racflogondays: THURSDAY  
racflogondays: FRIDAY  
racflogondays: SATURDAY  
racflogontime: ANYTIME  
racconnectgroupname: RACFID=LDAPGRP,PROFILETYPE=GROUP,CN=RACFZOS  
racconnectgroupname: RACFID=SYSTEM,PROFILETYPE=GROUP,CN=RACFZOS  
racconnectgroupname: RACFID=SYS1,PROFILETYPE=GROUP,CN=RACFZOS  
racfhavepasswordenvelope: NO  
racfattributes: SPECIAL  
racfattributes: OPERATIONS  
racfattributes: GRPACC  
racfattributes: PASSWORD  
safaccountnumber: ACCNT#  
safdefaultcommand: ISPPDF  
safdefaultloginproc: IKJACCNT  
saflogonsize: 860000  
safmaximumregionsize: 0  
safuserdata: 0000  
safdefaultunit: SYSALLDA  
racfomvsuid: 6  
racfomvshome: /u/ldapsrv  
racfomvsinitialprogram: /bin/sh  
racfovuid: 5  
objectclass: RACFBASECOMMON  
objectclass: RACFUSER  
objectclass: SAFTSOSEGMENT  
objectclass: RACFUSEROMVSSEGMENT
```

```
objectclass: RACFUSEROVMSEGMENT
```

```
# search result  
search: 2  
result: 0 Success
```

```
# numResponses: 2  
# numEntries: 1
```

5.5.6 Verifying the GDBM backend

To verify the GDBM backend, you can do an **ldapsearch** command against a `cn=changelog` entry:

```
ldapsearch -h 9.12.4.126 -x -D "cn=admin" -w secret -b "cn=changelog"  
"objectclass=*
```

The command should return:

```
# extended LDIF  
#  
# LDAPv3  
# base <cn=changelog> with scope subtree  
# filter: objectclass=*  
# requesting: ALL  
#  
  
# 1, changelog  
dn: changeNumber=1,cn=changelog  
objectclass: top  
objectclass: changeLogEntry  
objectclass: ibm-changeLog  
changenumber: 1  
changetype: add  
targetdn: RACFID=POKUS,PROFILETYPE=GROUP,CN=RACFZOS  
ibm-changeinitiatorsname: RACFID=MARIAN,PROFILETYPE=USER,CN=RACFZOS  
changetime: 20070727144119.086213Z  
  
# 2, changelog  
dn: changeNumber=2,cn=changelog  
objectclass: top  
...
```

You need to modify the IBM Tivoli Directory Integrator user (TDIUSER) as described in the following **acldbmbm ldif** entry to obtain the following access:

```
dn: cn=changelog  
changetype: modify  
add: aclentry  
aclentry:access-id:racfid=tdiuser,profiletype=user,cn=RACFZOS:normal:rscw
```

Then, you need to modify the GDBM ACL permission using:

```
ldapmodify -h 9.12.4.126 -x -D "cn=admin" -w secret -f ./acldbmbm.ldif
```

Verify that the ACL are correct from a linux server using:

```
ldapsearch -h 9.12.4.126 -x -D "racfid=tdiuser,profiletype=user,cn=racfzos" -w  
itso7471 -b "cn=changelog" "objectclass=*" > searchgdbm.txt
```

Edit searchgdbm.txt to verify the command executed successfully:

```
# extended LDIF  
#  
# LDAPv3  
# base <cn=changelog> with scope subtree  
# filter: objectclass=*<br># requesting: ALL<br>#<br><br># 1, changelog<br>dn: changeNumber=1,cn=changelog<br>objectclass: top<br>objectclass: changeLogEntry<br>objectclass: ibm-changeLog<br>changenumber: 1<br>changetype: add<br>targetdn: RACFID=POKUS,PROFILETYPE=GROUP,CN=RACFZOS<br>ibm-changeinitiatorsname: RACFID=MARIAN,PROFILETYPE=USER,CN=RACFZOS<br>changetime: 20070727144119.086213Z<br><br># 2, changelog<br>dn: changeNumber=2,cn=changelog<br>objectclass: top<br>...
```

5.5.7 RACF changes

IBM Tivoli Directory Integrator needs special authority to be able to change user permissions. In our case, IBM Tivoli Directory Integrator server user ID has RACF special authority in both RACF servers.

We created the z/OS and z/VM RACF user for IBM Tivoli Directory Integrator with the following commands in both RACF:

```
AU ITDI PASSWORD(ITDI) SPECIAL<br>ALU ITDI PASSWORD(password) NOEXPIRED
```

Activating RACF password envelope

To replicate the z/OS user password to z/VM, perform the following tasks on the z/OS RACF server to activate password enveloping:

1. Define, if not existing yet, a user for the RACF subsystem:

```
AU RACF DFLTGRP(SYS1) NOPASSWORD OMVS(AUTOUID HOME(/) PROGRAM(/bin/sh))<br>RALTER STARTED RACF.* STDATA(USER(RACF) GROUP(SYS1) TRUSTED(YES))
```

Use the RDEFINE command instead of RALTER if this is the first time that you have defined the RACF in the STARTED class.

2. Generate a certificate for the RACF server:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('RACFCA') O('ibm') C('us'))<br>WITHLABEL('RACFCA') NOTAFTER(DATE(2008-12-31))
```


3. Generate a certificate for the RACF address space:

```
RACDCERT ID(RACF) GENCERT SUBJECTSDN(CN('RACF AddrSpace')O('ibm')C('us'))  
WITHLABEL('RASP')SIGNWITH(CERTAUTH LABEL('RACFCA')) KEYUSAGE(HANDSHAKE  
DATAENCRYPT DOCSIGN) NOTAFTER(2008-12-31))
```

4. Create a RACF keyring named IRR.PWENV.KEYRING:

```
RACDCERT ID(RACF) ADDRING(IRR.PWENV.KEYRING)
```

5. Connect the RACF's certificate to the keyring as the default certificate:

```
RACDCERT ID(RACF) CONNECT(LABEL('RASP') RING(IRR.PWENV.KEYRING) DEFAULT  
USAGE(PERSONAL))
```

6. Define the certificate as trusted:

```
RACDCERT ID(RACF) ALTER (LABEL('RASP')) TRUST
```

7. Create a certificate for IBM Tivoli Directory Integrator user. During the RACF password enveloping process, RACF encrypts data that can only be recovered by the intended recipient of that data, in our case IBM Tivoli Directory Integrator. Certificates that identify intended recipients of RACF password envelopes must be connected to the key ring IRR.PWENV.KEYRING associated with the user ID of the RACF subsystem address space.

You can generate the certificate through RACF, connect it to the keyring and export it to the IBM Tivoli Directory Integrator server or you can create certificates directly on the IBM Tivoli Directory Integrator platform and import them into RACF. Any key management system can be used to create the recipient key pair and certificate, as long as it can export certificates in an industry standard format understood by RACF's RACDCERT command.

In our configuration, we generated the certificate on the IBM Tivoli Directory Integrator platform, exported the certificate and FTP in ASCII it to the host and then import the certificate to RACF. Although the RACDCERT method would work fine, we suggest to generate the key directly at the recipient system to eliminate the need to transfer it. Because the private key must ultimately be transferred to the recipient system, there is a greater likelihood of compromise of the key, due to the need to transfer it to the recipient.

8. Import the certificates in RACF using the RACDCERT command:

```
RACDCERT ID(IDI1) ADD(CERT.IDI1.TEXT) TRUST WITHLABEL('IDI1')
```

If you created self-signed certificates, RACF warns that the certificate authority is not defined to RACF but properly imports the certificates.

9. Connect the recipient certificates to the key ring. RACF encrypts the password for up to 20 recipient certificates:

```
RACDCERT ID(RACFSUB) CONNECT(ID(IDI1) LABEL('IDI1')  
RING(IRR.PWENV.KEYRING) USAGE(PERSONAL))  
RACDCERT CERTAUTH EXPORT(LABEL('RACFCA')) DSN(CERT.RACFCA.TEXT)  
FORMAT(CERTB64)  
RACDCERT ID(RACF) EXPORT(LABEL('RASP')) DSN(CERT.RASP.TEXT) FORMAT(CERTB64)
```

10. Authorize the IBM Tivoli Directory Integrator user ID to retrieve the password envelope from RACF:

```
RDEFINE FACILITY IRR.RADMIN.EXTRACT.PWENV UACC(NONE)  
PERMIT IRR.RADMIN.EXTRACT.PWENV CLASS(FACILITY) ID(ITDI) ACCESS(READ)
```

11. Define the PASSWORD.ENVELOPE resource:

```
RDEFINE RACFEVNT PASSWORD.ENVELOPE UACC(READ) APPLDATA('MD5/WEAK')
```

Although in our sample configuration we use this level of encryption, we recommend that you use the defaults in your configuration.

5.5.8 IBM Tivoli Directory Integrator configuration

The Tivoli Directory Integrator configuration that we developed for our configuration synchronized an instance of RACF running under z/VM with an RACF system running under z/OS in another LPAR. This configuration performs the following operations:

- ▶ Creates new z/VM RACF user profiles corresponding to existing or newly created z/OS user profiles
- ▶ Updates existing z/VM RACF user profiles to match existing or changed z/OS or user profiles
- ▶ Synchronizes z/VM RACF passwords with z/OS passwords when the latter changes

The configuration file, nominally named *zVM_Security_book_TDI_Config.xml*, has three AssemblyLines:

- ▶ syncExisting

AssemblyLine *syncExisting* reads a z/OS RACF database through the LDAP SDBM backend and then updates the target z/VM RACF database, also through its LDAP SDBM backend.

- ▶ syncChanges

AssemblyLine *syncChanges* monitors changes in the z/OS RACF through its GDBM backend which drives an LDAP changelog suffix. The AssemblyLine then transforms various attributes to match the target z/VM RACF database and updates the LDAP SDBM exactly as AssemblyLine *syncExisting*.

- ▶ zz_readEnvelope

AssemblyLine *zz_readEnvelope* is test fixture that confirms RACF password enveloping, the Directory Integrator key store and associated certificates and passwords are all configured and correct.

Both AssemblyLines *syncExisting* and *syncChanges* share an identical flow section. Reusing is a Directory Integrator best practice. Encapsulating this flow section in a separate AssemblyLine and calling it from the main AssemblyLines extends the solution and eases long-term maintenance.

Here is a schematic of the common flow section: lookupInZOS: LDAP Connector; retrieves RACF user entries from z/OS LDAP.

- ▶ Config: Lookup Mode; binary attributes: racfPasswordEnvelope
- ▶ Link Criteria: \$dn = \$targetdn (which means retrieve the object in z/OS whose LDAP distinguished name matches the value in the Tivoli Directory Integrator work entry attribute targetedn)
- ▶ Input Map:

\$dn	\$dn
objectclass	objectclass
racfattributes	racfattributes
racfdefaultgroup	racfdefaultgroup
racfid	racfid
racflogondays	racflogondays
racflogontime	racflogontime
racfowner	racfowner
racfpassword	(script returning decrypted password)
racfprogrammename	racfprogrammename

- ▶ Hooks:
 - Before Execute: Sets LDAP Return Attributes to get just racpasswordenvelope or all attributes
 - Lookup Successful: Removes targetdn attribute from work entry
- mapZOSToZVM: Attribute Map; transforms selected attributes for target z/VM**

\$dn	(script returning \$dn for z/VM)
objectclass	(script; removes z/OS specific objectclasses)
racfattributes	(script; removes specified values)
racfdefaultgroup	(script; transforms z/OS defaultgroup to z/VM group)
racfowner	(script; sets racfowner to valid object in z/VM)

updZVMsdbm: LDAP Connector; updates or creates z/VM user
- ▶ Config: Update Mode
- ▶ Link Criteria: \$dn = \$\$dn (which means retrieve the object in z/VM whose LDAP distinguished name matches the Tivoli Directory Integrator work entry attribute \$dn value)
- ▶ Output Map:

\$dn	\$dn
objectclass	objectclass
racfattributes	racfattributes
racfdefaultgroup	racfdefaultgroup
racfid	racfid
racflogondays	racflogondays
racflogontime	racflogontime
racfowner	racfowner
racfpassword	racfpassword
racfprogrammername	racfprogrammername
- ▶ Hooks
 - Before Modify: selects logic to update racfpassword (only) or regular attributes
 - On No Changes: optional logging
 - After Modify: optional logging
 - Before Add: optional logging
 - After Add: optional logging
 - Update Error: selects logic to add, rather than update, a user

The *syncExisting* AssemblyLine feed section iterates the z/OS SDBM LDAP and delivers the distinguished name (\$dn) of synchronizable objects to the flow section. Here is a schematic of this section: read_zOSLDAP: LDAP Connector, reads z/OS LDAP.

- ▶ Config: Iterator Mode
- ▶ Input Map

targetdn	\$dn
----------	------
- ▶ Hooks:
 - After GetNext: skips excluded users

The *syncChanges* AssemblyLine feed section polls the z/OS changelog and sends the \$dn of synchronizable objects that have changed to the flow section. Here is a schematic of this section: read_zOSclog: zOS LDAP Changelog Connector.

- Config: Iterator Mode, Iterator State Key: zos (stored in System Store)

- Input Map:

changenumber	changenumber
changetype	changetype
racfPassword	racfPassword
targetdn	targetdn

- Hooks

- After GetNext: skips excluded users
- Iterator Error: handles exception and logs it

The code in the After GetNext hook skips the users listed in an external file and cached in a Java HashSet object called *excludedUsers* as described later. It is simple to modify this hook to only process users listed in an external file rather than exclude them.

All AssemblyLines share a common prolog in the script library called *global_Prolog*. The prolog has three sections:

- The first part sets several string constants used in various places in the AssemblyLine. This is a standard practice and ensures that deployers only have to look one place to configure the details for a particular site. A commonly-used alternative is to put that type of information in an external properties file.
- The second section of the prolog configures a slightly customized logging framework. Directory Integrator supports numerous and flexible logging options which provide logging to multiple destinations (log4j appenders) and multiple logging levels. However, to log a set of related objects as they move down the AssemblyLine requires individual logging statements in different hooks or attribute maps.

The procedure is simple, but enabling and disabling lots of statements during development and deployment testing is a bit cumbersome. This prolog sets two global variables LOGLEVEL and LOGDEPTH that control logging through any appender and the within run window in the config editor. A single JavaScript™ function *cat()*, defined in a script library object called *cat*, writes both strings and other objects to logs. The main advantage is this enables detailed, normal or no logging just by changing the single variable LOGDEPTH from 2 to 1 to 0, respectively. This small innovation is a convenience to the developer of this configuration and it might be useful to readers. It is not necessarily a best practice or the only way to provide this flexibility.

In conjunction with the log setup described above, both AssemblyLines *syncExisting* and *syncChanges* have three loggers defined, one set at the DEBUG level directing output to a file, another at the INFO level sending output to a file roller and another at the INFO level going to the console. The deployer can enable or disable these loggers as needed. What these loggers print, especially those at the INFO level, is largely controlled by the value of LOGDEPTH. Detailed logging statements throughout both AssemblyLines only write to the INFO level loggers when LOGDEPTH is set to 2.

- The third section of the prolog reads and stores a list of users to exclude from synchronization. It puts the list in a *java.util.HashSet* object called *excludedUsers*, which is a high-performance associative array. The After GetNext hook in the feed connector of both AssemblyLines searches the *excludedUsers* object and skips the rest of the iteration when the retrieved object is a member of *excludedUsers*. In other implementations, this logic could read the same information from a database or user z/OS RACF attributes to control AssemblyLine behavior.

Any Directory Integrator server instance with network access to the z/OS and z/VM LDAP servers can run the AssemblyLines in this config. The server could be on z/OS, a Linux instance running on z/VM or any distributed server platform. In the ITSO configuration, it is deployed on a mainframe Linux platform running in a z/VM **1par**. Directory Integrator runs as a daemon and is controllable with standard start and kill scripts or other scripts per local practice.

5.5.9 Summary

Tivoli Directory Integrator is an universal tool that lets you filter and mirror all available LDAP informations. For example, we used the Tivoli Directory Integrator rule that for any z/OS RACF, changes related to the user ID starting with *LIN***** are reflected to z/VM RACF side. Tivoli Directory Integrator uses the LDAP GDBM backend to capture the changes and replicate them.

Archived

Cryptography on z/VM

In this chapter, we describe how to prepare a z/VM system and Linux guests to use encryption technologies to increase the security of your environment. *Cryptographic operations* are by nature CPU intensive. Therefore, in this chapter, we focus on how to enable the environment to benefit from the hardware capabilities of IBM System z to support cryptographic operations.

Protecting information from unintended use is one key element of a secure IT environment. Basically there are two different methods to ensure privacy of information:

- ▶ Access control
- ▶ Encryption methods

With access control mechanisms it is determined who has the right to access particular information or data. It is verified who accesses the information (*authentication*) and whether has the right to access this information (*authorization*). There are cases where proper access control cannot be guaranteed in all situations. This applies especially if data is stored on movable media and also when data is transferred through a network, which might not be protected. It is not possible to guarantee that there is no unintended access to data while it is stored or transferred through a network. The the only way to protect such information is by using encryption methods.

We describe where and how to use encryption to improve the security of the environment. By its nature, encryption is a very CPU intensive task. To reduce CPU consumption and to reduce related costs, System z offers hardware support for encryption. In addition, hardware support provides improved security when secure key operations can be used instead of clear key operations. We show how to set up the environment so that z/VM guests can benefit from the hardware support that System z provides.

Note: If not stated differently, the descriptions in this chapter are assumed for an IBM System z9™ and to z/VM 5.3.

6.1 Secure communication to the z/VM System using SSL

Depending on the security policies in an enterprise, and depending on the network environment, clients might want to secure (encrypt) the communication for their connections to the z/VM guest user IDs to avoid sending passwords in clear text over a network and to protect the content of the communication.

With z/VM you can setup the TCP/IP connections to the z/VM guests to be protected by Secure Socket Layer (SSL).

Like in other areas, the implementation of SSL is done in z/VM by the usage of a virtual server to handle the work. The SSL server, which runs in the SSLSERV virtual machine, provides processing support for encrypted communication between remote clients and z/VM TCP/IP server that listen on secure ports. The SSL server manages the database in which the server authentication certificates are stored. The TCP/IP stack server routes requests for secure ports to the SSL server. The SSL server, representing the requested application server, participates in the handshake with the client in which the cryptographic parameters are established for the session. The SSL server then handles all the encryption and decryption of data.

Figure 6-1 illustrates the principal setup and the information flow.

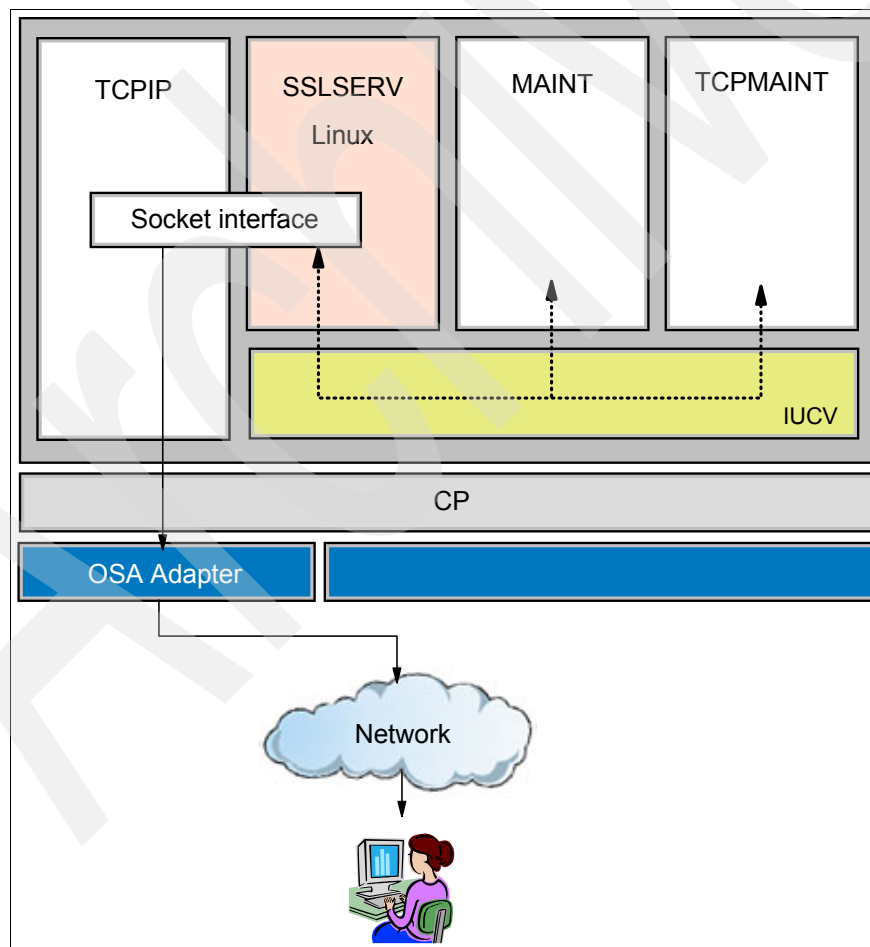


Figure 6-1 SSL implementation architecture in z/VM

The SSL Server is a virtual service machine, with a special Linux server installed and configured for exclusive use of SSL. Only specific Linux distributions and kernel levels are supported. For a list of supported Linux distributions and kernel versions, and also for detailed setup instructions, see the z/VM Web page:

<http://www.vm.ibm.com/related/tcpip/>

On this page you can find a link to SSL Server Configuration where all related information is available:

<http://www.vm.ibm.com/related/tcpip/vmsslinf.html>

Verify whether there are important service updates that might contain new information such as necessary PTFs or APARs. Check this information before you begin your implementation.

To set up secure communication to z/VM:

1. Check the documentation for a supported Linux on System z version and install such a Linux as a guest under z/VM. Later, you will configure this guest to be used as SSL server.
2. Retrieve the additional Red Hat Package Manager (RPM) packages for this Linux server from the IBM z/VM Web site. You need these RPM packages to configure the Linux server for z/VM SSL usage.
3. Now, configure the Linux guest server for SSL. You need to adapt the directory entry for the Linux system that is used as SSL Server (that is, SSLSERVE user ID).
4. Install the RPM packages to the Linux and configure this Linux to act as an SSL server.

Note: After you configure this Linux guest to act as an SSL server for z/VM TCP/IP, you can no longer use it for any other purpose, because other services and capabilities are disabled.

5. Configure TCP/IP of z/VM so that the SSL server is started automatically when TCP/IP is initialized and so that it is listening to encryption requests.
6. Set up a certificate data base that is managed by the SSL server. This database must contain the SSL server certificate.

If you have such a certificate already, you can import it and use it. If you do not have such a certificate, you can create one by yourself. Depending upon whether you have a Certification Authority (CA), the self-created certificate can be signed by the CA.

If you do not have a CA, you can request your certificate to be signed by any official CA (which might cost a small fee). For development and testing purpose, you can also sign your certificate yourself. In general, a self-signed certificate is not recommended for production purpose, especially for accessing the server from a public network (Internet).

7. When you have established an SSL connection, the client needs to verify the authenticity of the server in order to connect. Therefore, the client must know either the signature of the CA of the server certificate, or if you are using a self-signed certificate, the certificate must be imported into the key database of the client.

If you are using IBM Personal Communications for accessing the z/VM guest user IDs, then use its Certificate Management Utility to import the server certificate. Take care when you transfer the server certificate from your z/VM system to the workstation with any method (FTP, cut-and-paste, and so forth) that you do not include or insert blanks or other invisible characters at the ends of lines or at the end of the file.

In your 3270 client, you can enable the SSL security to start encrypted sessions to your z/VM guest user IDs.

During the SSL handshake the client and the SSL server negotiate which encryption algorithms will be used to secure the connection. A cipher suite is selected by them which is common for both parties. Using the `ssladmin query status` command in the TCP/IP server, you can verify which cipher suites are allowed to be used by the SSL connection. Note that not all cipher suites provide a high degree of security. We recommend that you carefully consider which suites to allow. You might want to exempt individual cipher suites, such as NULL, NULL_SHA, or NULL_MD5, or you might want to instruct the SSL server to operate in Federal Information Processing (FIPS) mode. For details, refer to the chapter on the VMSSL and SSLADMIN commands in *z/VM TCP/IP Planning and Customization*, SC24-6125.

The SSL support of z/VM is an easy method to protect the communication to the z/VM server especially for administration tasks. This increases the total security and protection of the z/VM system, as sensitive information such as passwords from administrators are protected independent from the network.

For detailed information about installation and configuration, examine the official documentation provided by z/VM (*z/VM TCP/IP Planning and Customization*, SC24-6125) and the z/VM Web page. For z/VM 5.2 there is a step-by-step description available in *SSL Server Implementation for z/VM 5.2*, REDP-4348.

6.2 Preparing System z for the hardware encryption support

System z provides two different types of hardware support for cryptographic operations: CP Assist for Cryptographic Function (CPACF) and PCI cryptographic card.

CPACF is incorporated in every central processor that is shipped with IBM System z. CPACF was introduced with z990 and z890. The CPACF feature delivers support for symmetric encryption algorithms Data Encryption Standard (DES) and Triple DES (TDES) and hashing algorithm SHA-1. The CPACF incorporated in System z BC and EC also provides support for Advanced Encryption Standard (AES) as well as for SHA-256. Because these algorithms are implemented in each central processor (CP), the potential throughput scales with the number of used CPs. The algorithms in the CPACF are executed synchronously with enhanced performance and are only for clear key operations (i.e. the encryption key is provided by the application software in clear format).

The second way is by using additional PCI cryptographic cards. The current generation for System z BC and EC is the Crypto Express2. Crypto Express2 replaces PCIXCC, PCICA and PCICC used in prior server generations. The adapters of the Crypto Express2 feature can be configured either as Accelerator (CEX2A) or as Coprocessor (CEX2C). If the feature is running as CEX2A, it can perform clear key RSA operations with high speed. If it is configured as CEX2C, it can perform symmetric as well as asymmetric operations (RSA) in clear key and also in secure key mode, i.e. all encryption operations are performed inside the CEX2C and the encryption key used is not available outside of the card in clear format.

Clear key encryption is widely used by Web servers (SSL communication), such as Apache, IBM HTTP server, and also for using a secure communication over an unprotected network, like establishing a virtual private network (VPN), or for access to a Linux system with Secure Shell (SSH). Clear key encryption is also used in Linux systems for encryption of the file system. When DES, TDES, AES, SHA-1, or SHA-256 are used, the CPACF can be used. To benefit from cryptography hardware acceleration for clear key operation for the RSA algorithm, you need to have Crypto Express2 feature installed on System z. It can be configured either as CEX2C or as CEX2A. If only clear key operations are used on your system, then configure it as a CEX2A, because this provides higher processing speed.

Secure key encryption is used when the requirements for security are higher and the encryption keys must be stored and protected by hardware (for example, when you have a requirement to fulfill Federal Information Processing Standard (FIPS) 140-2 Level 4 Certification). This is often found in financial areas or other highly sensitive environments. If you have a need for secure key encryptions and one of your applications requires it, then you need to configure the Crypto Express2 as CEX2C.

Note: The Crypto Express2 feature contains internally two cryptography engines (PCI-X adapters), which can be configured independently from each other. You can have two CEX2A, or two CEX2C, or one CEX2A and one CEX2C in one feature. In April 2007 IBM announced a Crypto Express2-1P feature which can be used in a IBM System z9 BC. It has only one PCI-X adapter and therefore a lower entry price to use hardware encryption for z9 BC customers.

To benefit from the CPACF or from the Crypto Express2 feature, you must have installed and enabled LIC internal feature 3863 (Crypto Enablement feature), which is available free of charge. By default, IBM System z is delivered to customers without this feature. You need this feature in any case to use CPACF, or before you install the optional Crypto Express2 feature. The installation of this Crypto Enablement feature is non disruptive.

Note: We recommend that you install this Crypto Enablement feature, even if you do not intend to use the Crypto Express2, because there is already a benefit from an active CPACF

You can verify whether this feature is installed on your system on the Hardware Management Console (HMC). In the CPC Details panel, you can see whether the CPACF feature is installed (see Figure 6-2).

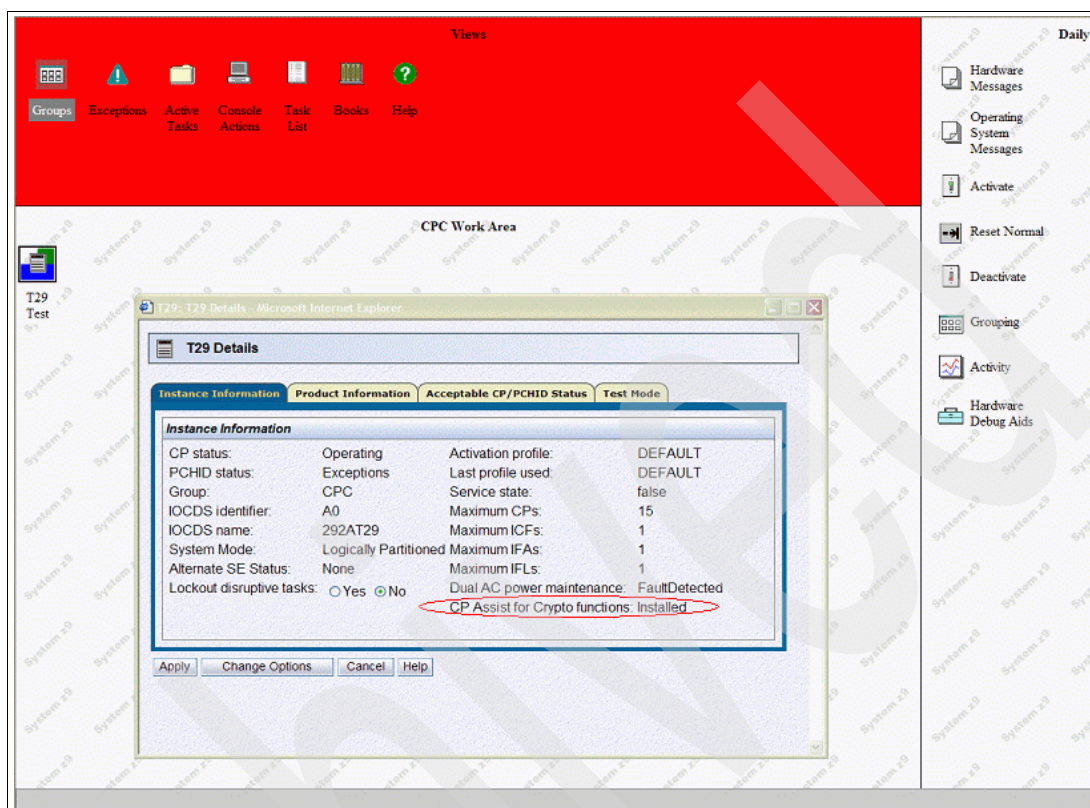


Figure 6-2 Processor status with Feature Code 3863 (CPACF) installed

For planning and configuring your System z to use hardware encryption check also *IBM System z9 109 Configuration Setup*, SG24-7203 and *z9-109 Crypto and TKE V5 Update*, SG24-7123.

6.2.1 Planning to use Crypto Express2

Each cryptographic engine of a Crypto Express2 feature has 16 physical sets of registers or queue registers, each set belonging to a domain, whether it is configured as a CEX2C or CEX2A. To newly installed cryptographic coprocessors during power-on-Reset numbers are assigned sequentially. However, when a Crypto Express2 feature is installed concurrently using the Nondisruptive Hardware Change¹ task, it is possible for the installation to select an out-of-sequence coprocessor number from the unused range. In this case, the client should communicate the desired Cryptographic numbers to the IBM installation team.

In order to make the adding of the Crypto Express2 nondisruptive, the logical partitions must be configured to contain the appropriate domain indices and Cryptographic numbers when the partitions are activated. A change to a logical partition image profile to modify its domain indices or Candidate list is disruptive to the partition. It requires a partition deactivation-activation to take effect. Therefore a careful planning how to distribute the 16 domains of a PCI-X adapter across all partitions is necessary to avoid updates to the system.

¹ The Nondisruptive Hardware Change is only available when logged on to the Support Element in Service mode.

To enable use of a new PCI-X adapter dynamically to a partition requires that:

- ▶ At least one usage domain index to be defined to the logical partition.
- ▶ The cryptographic coprocessor numbers to be defined in the partition Candidate list, which contains all adapters that are eligible for the logical partition and are configured on during partition activation, or can be configured on later dynamically.

Note that these definitions for the partition activation profile can be done before the appropriate PCI-X adapters are installed.

For planning the cryptography configuration for the system, there are some rules:

- ▶ Depending on the workload characteristic, you might need additional Crypto Express2 features.
- ▶ For availability, assignment of multiple PCI-X adapters of the same type (accelerator or coprocessor) to one logical partition should be spread across multiple features.
- ▶ The use of retained private keys on a PCI-X adapter configured as a Crypto Express2 coprocessor creates an application single point of failure, because RSA-retained private keys are not copied or backed up.
- ▶ There is an intrusion latch within the PCI-X adapter logic that is set any time the feature is removed from the system. If the feature is re-installed, and power is applied, the coprocessor keys and secrets are zeroized and the intrusion latch is reset.
- ▶ To manage a PCI-X adapter with a Trusted Key Entry (TKE) workstation, the adapter number must be included in the Control Domain Index (see Figure 6-5) and TKE commands must be permitted for this adapter in the Cryptographic configuration (see Figure 6-15).

If a Trusted Key Entry (TKE) workstation is available, the PCI-X adapter can first be disabled from the TKE workstation before removing the feature from the system. In that case, when the feature is re-installed, the coprocessor keys and secrets are not zeroized, but the intrusion latch is reset and the coprocessor remains in the disabled state. The PCI-X adapter then can be enabled from the TKE and normal operations can resume.

For more information, see *z/OS ICSF TKE Workstation User's Guide*, SA22-7524.

- ▶ The same usage domain index can be defined more than once across multiple logical partitions. However, the cryptographic coprocessor number coupled with the usage domain index specified must be unique across all active logical partitions.

The same cryptographic coprocessor number and usage domain index combination can be defined for more than one logical partition. This can be used, for example, to define a configuration for backup situations. In this case, only one of the logical partitions can be active at any one time.

Figure 6-3 illustrates how domains and cryptographic coprocessors can be assigned to logical partition.

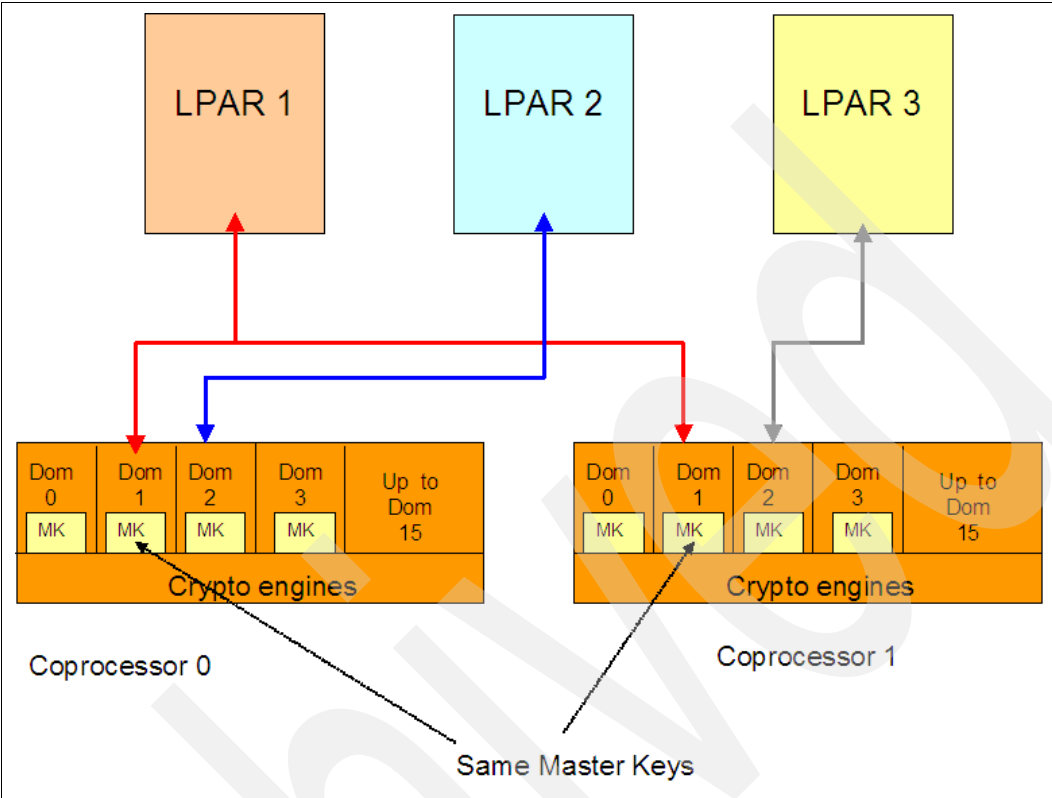


Figure 6-3 Assign Crypto Domains to LPARs

For the planning task, a configuration map as listed in Table 6-1 can help to keep an overview and to simplify the work. Each row identifies a PCI-X adapter and each column identifies a domain index number. Each cell entry indicates the logical partition to be assigned the cryptographic coprocessor number coupled with the usage domain index.

Table 6-1 Planning for logical partitions, domains, and PCI-X adapter numbers

	Adapter Type	Domain index 0	Domain index 1	Domain index 2	.../...	Domain index 14	Domain index 15
PCI-X adapter 0	CEX2C/A	LP00 LP02	LP05	LP04		LP04	
PCI-X adapter 1	CEX2C/A	LP01 LP02					
PCI-X adapter 2	CEX2C/A	LP00					
.../...							
.../...							
PCI-X adapter 13	CEX2C/A						
PCI-X adapter 14	CEX2C/A						
PCI-X adapter 15	CEX2C/A						

Any given cell should contain only one *active* logical partition because the combination of cryptographic coprocessor number and usage domain index must be unique across all *active* logical partitions.

In Table 6-1:

- ▶ Logical partition LP04 and LP05 use different domain numbers for PCI-X cryptographic 0, there is no conflict. The combination, domain number and cryptographic coprocessor number, is unique across partitions.
- ▶ Logical partitions LP00 and LP01 use domain 0, but are assigned different PCI-X adapters. There is no conflict. They can be concurrently active.
- ▶ Logical partition LP02 uses domain 0 on the set of cryptographic adapters already defined to LP00 and LP01. Therefore, partition LP02 cannot be active concurrently with either LP00 or LP01. However, the definition can be valid for backup situations.

Each PCI-X adapter provides 16 domains, and up to 60 partitions can be defined and active on the IBM System z9. To allow all 60 logical partitions to use cryptographic services, either accelerator or coprocessor, requires at a minimum two Crypto Express2 features without redundancy, or four Crypto Express2 features if redundancy is required.

6.2.2 Customize the partition image profile for cryptographic usage

As already mentioned, make sure that the Crypto Enablement feature 3863 is installed on your system to benefit from hardware support from CPACF, or from Crypto Express2 feature. In the CPC Details panel (see Figure 6-2 on page 206) you can check whether this feature is installed. If you find in this panel *CP Assist for Crypto Functions: Installed*, then 3863 feature is installed, if you find *CP Assist for Crypto Functions: Not Installed*, then the feature is still missing, but you are able to customize the partition image. In this case the cryptographic operations do not operate.

To customize the image profile of a LPAR to enable the usage of the cryptographic functions of the Crypto Express2 feature, you have to define for each partition:

- ▶ Usage Domain Index
- ▶ Control Domain Index
- ▶ PCI Cryptographic Coprocessor Candidate List
- ▶ PCI Cryptographic Coprocessor Online List

You can define these partitions through the Customize/Delete Activation Profile task. (You can perform this from the HMC Workplace™ or from the SE Workplace).

In the Customize/Delete Activation Profile List (Figure 6-4), select the name of the partition image profile that you want to configure.

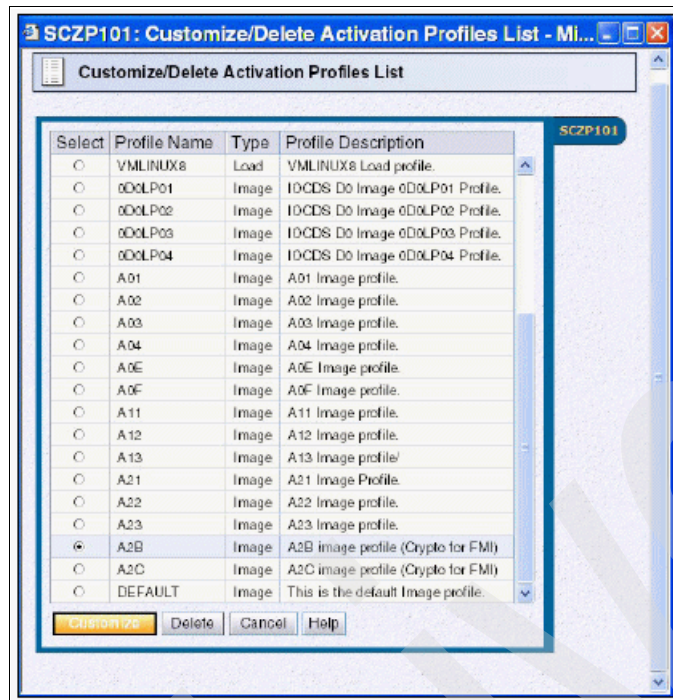


Figure 6-4 Customize/Delete Action Profile List

This brings you to the Customize image profiles notebook, where you select the Crypto option. This leads you to the Crypto panel (Figure 6-5) where you can specify the necessary values.

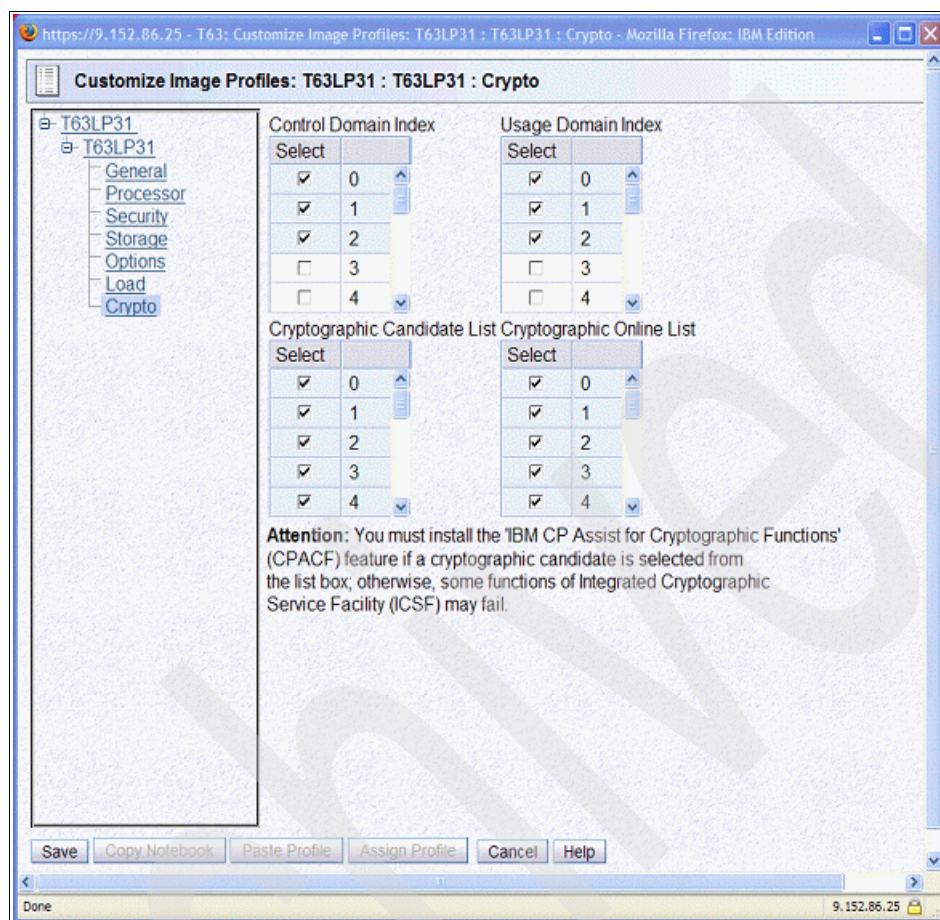


Figure 6-5 Customize Image Profiles - Crypto

Note that you can change the definitions in the image profile independent whether or not the logical partition is active. However, the new definitions will not take effect until the next time the partition is activated!

The *Usage Domain Index* identifies the cryptographic coprocessor domains assigned to the partition for all cryptographic coprocessors that are configured on the partition. The same usage domain index can be used by multiple partitions regardless of which LCSS they are defined to, but the combination PCI-X adapter number and usage domain index number must be *unique* across all partitions planned to be active at the same time.

Although it is possible to define duplicate combinations of PCI-X adapter numbers and usage domain indexes, such logical partitions cannot be concurrently active. This is a valid option, for example, for backup configurations.

The *Control Domain Index* identifies the cryptographic coprocessor domain indexes that can be administered from this logical partition being set up as the TCP/IP host for the TKE. The Control Domain Index must include the usage domain index specified for the partition. If any selected usage domain index is not part of the control domain index selection, the update is rejected.

If you are setting up the host TCP/IP in this logical partition to communicate with the TKE, the partition is used as a path to other domains' Master Keys. Indicate all the control domains you want to access (including this partition's own control domain) from this partition.

The *Cryptographic Candidate List* identifies the cryptographic coprocessor numbers that are eligible to be accessed by this logical partition. From the scrollable list, select the coprocessor numbers, from 0 to 15, that identify the PCI-X adapters to be accessed by this partition. When a cryptographic coprocessor number selected in the partition Cryptographic Candidate List is not available to the partition when the partition is activated, either because it is configured off or not installed, no error condition is reported. The cryptographic coprocessor number is ignored and the activation process continues.

When a new Crypto Express2 feature is installed and the PCI-X adapter numbers have been previously selected in a partition Candidate list, they can be dynamically configured to the partition from the Support Element using the Configure On/Off option in the Crypto Service Operations task list.

A PCI-X adapter number not in the partition Cryptographic Candidate List cannot be configured on to the partition.

The *Cryptographic Online List* identifies the cryptographic coprocessor numbers that are automatically brought online during logical partition activation. The numbers selected in the Online list must also be part of the Candidate list.

After the next partition activation, installed PCI cryptographic coprocessors that are on the partition Cryptographic Candidate List but not on the Cryptographic Online list are in a *configured off* state (Standby). They can later be configured on to the partition from the Support Element by using the Configure On/Off option in Crypto Service Operations task list (see "Config On/Off from the CPC Work Area" on page 215).

When the partition is activated, no error condition is reported if a cryptographic coprocessor number selected in the partition Cryptographic Online List is not installed. The cryptographic coprocessor is ignored and the activation process continues. When a cryptographic coprocessor number selected in the partition Cryptographic Online List has been previously configured off to the partition, it is automatically configured back on when the partition is next activated.

If the cryptographic coprocessor number and usage domain index combination for the coprocessor selected in the partition Cryptographic Online List are already in use by another active logical partition, activation of the logical partition fails.

After saving the definitions for the selected partition, you can continue to customize the activation profiles for each logical partition that needs Crypto Express2 support.

Note: A power-on reset is not necessary to activate these definitions, but they will only take effect when the partition is activated next time.

6.2.3 Configuration of the cryptography adapter as coprocessor or accelerator

You install one or more Crypto Express2 features using the IBM service persons. After the installation, the cryptographic adapters get numbers assigned either sequentially during power-on reset, or a *predefined* out-of-sequence number as specified by the installation team (see 6.2.1, "Planning to use Crypto Express2" on page 206).

Note: To have a nondisruptive environment, these assigned cryptographic coprocessor number must match the specified numbers in the Cryptographic Coprocessor Candidate list, specified in the image activation profile of the logical partition.

After the installation of a Crypto Express2 feature, the cryptography adapters are configured off, and by default they are configured as coprocessor. This is important to know for hot plugging of the Crypto Express2. To change the configuration from CEX2C to CEX2A, use the Cryptographic Configuration panel (see Figure 6-15 on page 218). When you have installed this feature and have made the previous preparations, the CEX2C or CEX2A will be available to all partitions specifying their assigned cryptographic numbers in the Cryptographic Candidate List. To bring the Cryptographic numbers online, perform either:

- ▶ Config On of the Crypto (nondisruptive)
- ▶ An Activate of the partition (disruptive)

To visualize the LPAR cryptographic controls go to the CPC Workarea and select the CPC object. Then, in the Task List work area select the CPC Operational Customization Task List (see Figure 6-6).

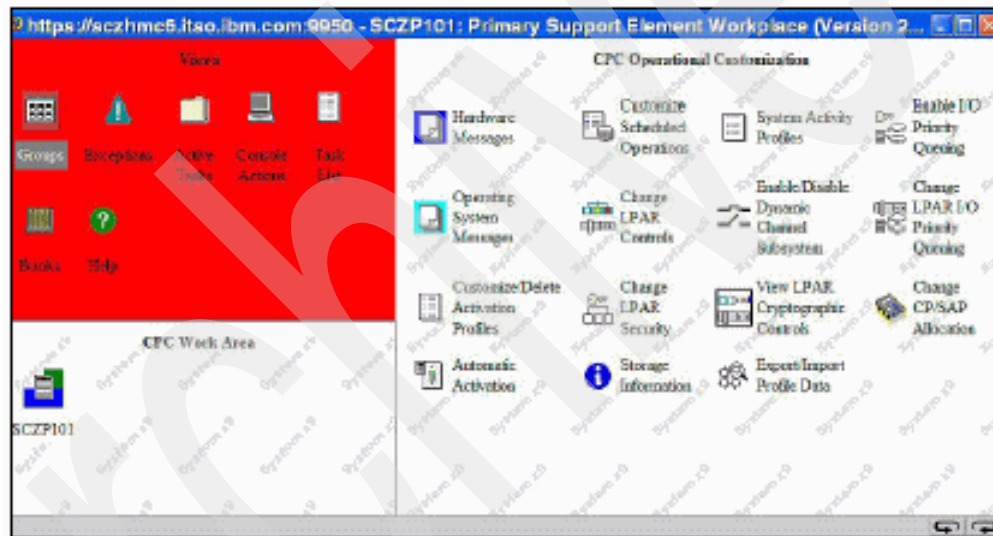


Figure 6-6 CPC Operational customization

From the task list, double-click **View LPAR Cryptographic Controls**. The View LPAR Cryptographic Controls panel opens, which contains all the definitions per LPAR (Figure 6-7). This panel is for information only. You cannot perform any change from this panel.

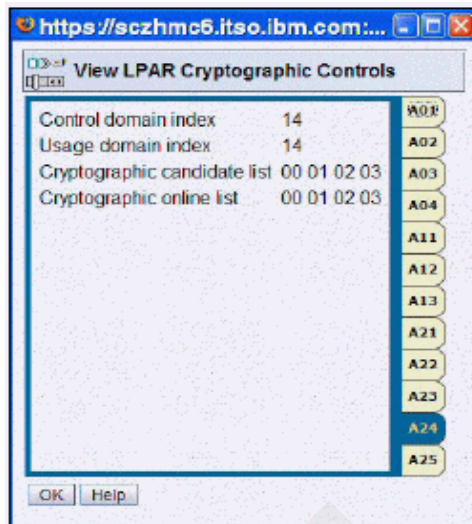


Figure 6-7 View LPAR Cryptographic Controls

To modify the cryptographic coprocessor On/Off status requires the use of the Configure On/Off task from the Crypto Service Operations Task list. You can apply the Configure On/Off task to either the CPC object or to a specific image selected from the Images Work Area.

Config On/Off from the Images Work Area

If you intend to reconfigure only one logical image, use the Images Work Area path. For this purpose start in the Image work area and right-click the selected image to open the context menu. In the context menu select **Cryptos** to get to the PCI Crypto Work Area (Figure 6-8).

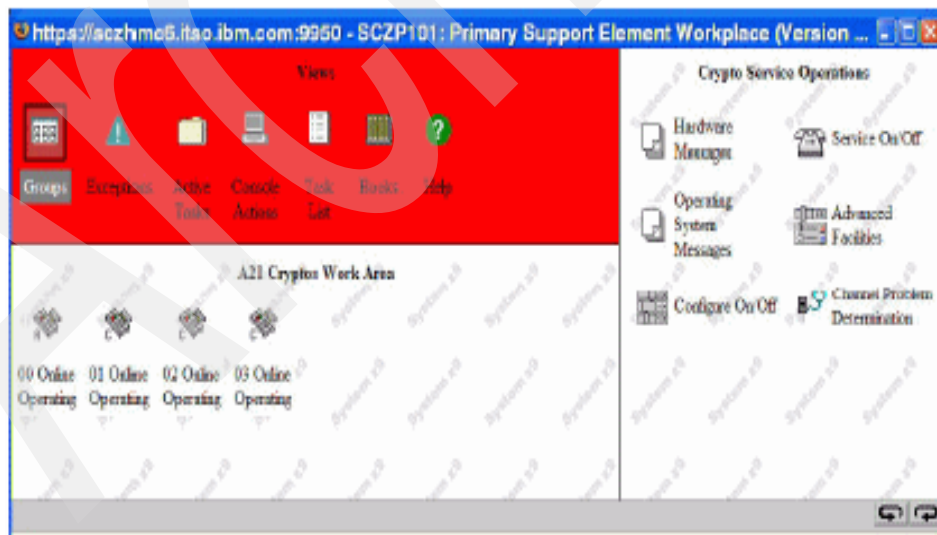


Figure 6-8 Logical Partition Cryptos work area

Each icon displays *C* or *A* to indicate whether the cryptography adapter is configured as a coprocessor or as an accelerator. From the Crypto Work Area, select the cryptographic icon and drag and drop it on the Configure On/Off Task. The Configure Channel Path On/Off panel is displayed (Figure 6-9).

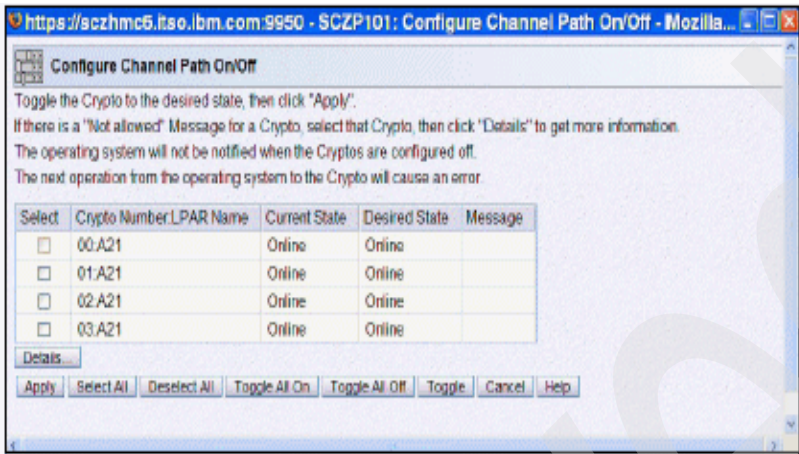


Figure 6-9 Configure Channel Path On/Off

Select the coprocessor number that you want and click **Toggle** to switch to the desired state. Then, click **Apply** to initiate the configuration change. A Configure Channel Path On/Off Progress panel pops up (Figure 6-10). When the status is complete, click **OK** to return to the previous panel where you can perform additional changes. When you have completed reconfiguration for all partitions, click **Cancel**.

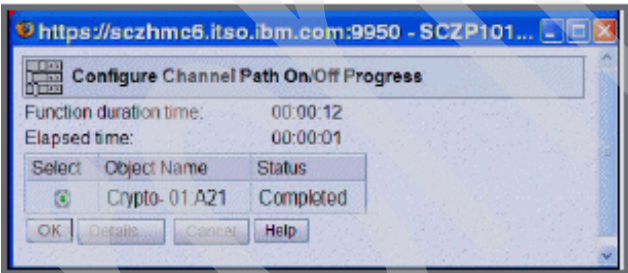


Figure 6-10 Configure Channel Path On/Off

Config On/Off from the CPC Work Area

When using the CPC Work Area, the scope of cryptographic objects displayed spans all defined logical partitions, active or not (note that you cannot issue a Config On/Off command to a cryptographic coprocessor in an inactive logical partition).

Use this path when you want to configure a cryptographic processor across multiple logical partitions.

For this purpose start in the CPC Work Area and right-click the selected CPC icon to open the context menu. In the context menu select **Cryptos** to get to the PCI Crypto Work Area. In the Crypto Work Area select the cryptographic coprocessor icons, then right-click to get the

context menu. Select **Crypto Service Operations Configure On/Off** as shown in Figure 6-11, and you get the Configure Channel Path On/Off Panel (Figure 6-12). Note, this panel is used, even you do not assign explicitly CHPIDs for the cryptography adapters.

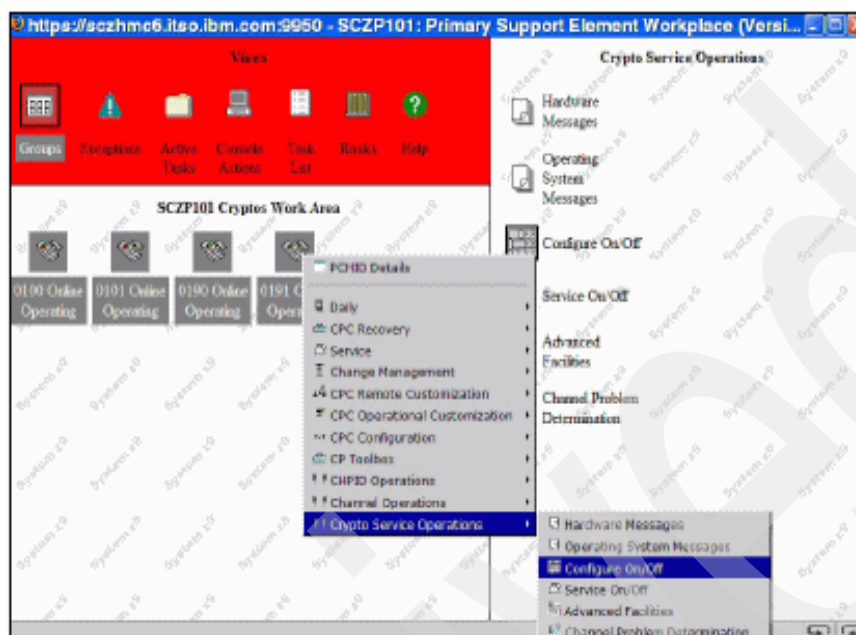


Figure 6-11 Crypto Service Operations - Configure On/Off

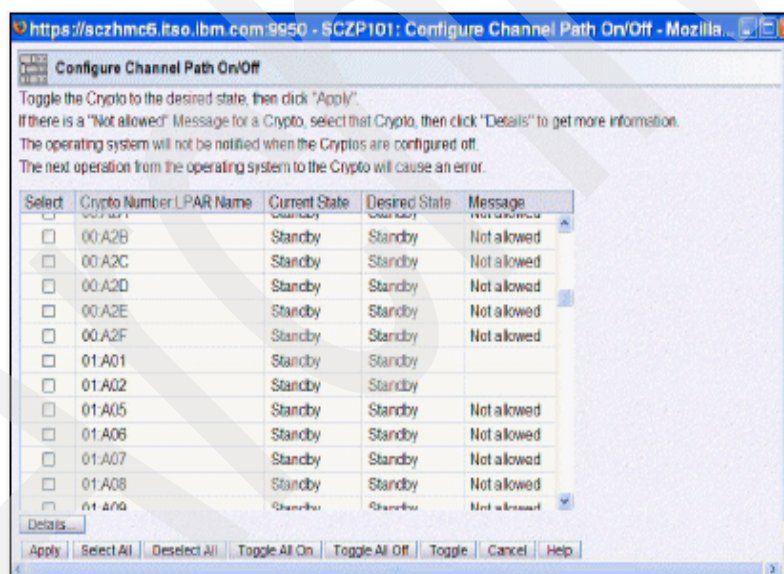


Figure 6-12 Configure Channel Path On/Off

Because we are now using the CPC object, all partitions that have the PCI-X cryptographic adapter in their Candidate list are listed, whether active or inactive.

If the partition is not active, you cannot configure its PCI-X cryptographic adapter On/Off and the indication *Not Allowed* is displayed in the message area for the partition. The list is sorted first by PCI cryptographic coprocessor number, then by partition.

Form this list, select the desired coprocessor/partition and use **Toggle** to change the status to the desired state. Then, click **Apply** to initiate the configuration change.

A Configure Channel Path On/Off Progress panel opens, as shown in Figure 6-13.

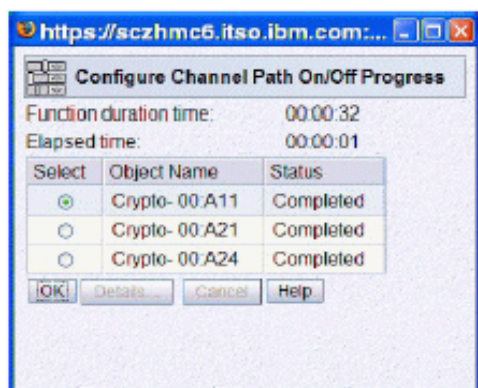


Figure 6-13 Configure Channel Path On/Off

When the status is complete, click **OK** to return to the previous panel where you can perform additional changes. When you have completed reconfiguration for all coprocessors/partitions, click **Cancel**.

Change the configuration of a PCI-X adapter

Each PCI-X adapter of a Crypto Express2 feature can be configured either as a coprocessor or as accelerator. Independent from its configuration, an adapter can be shared between up to 16 logical partitions. When the adapter is configured as accelerator it cannot be used for coprocessor functions. It can be used for clear key RSA operations, which are part of the SSL handshake. When the adapter is configured as coprocessor, it still can perform the accelerator functions, but slower. If you intend to use the adapter only for clear key operations and if you do not need secure key operations, then accelerator mode might be the better choice.

Note, if you intend to share an adapter between a z/OS partition and a Linux system, where the z/OS system uses secure key operations and the Linux system needs only clear key operation for some Web serving, then you would have to configure the adapter as coprocessor, or use separate adapters for the two systems, for z/OS configured as coprocessor and for Linux configured as accelerator.

Note, after installation, the adapters are configured by default as coprocessors.

If you need to reconfigure a cryptography adapter, perform the following steps:

1. Make sure, that the cryptographic adapter status is Off, as reconfiguration is only possible for adapter that are Off.
2. If necessary, configure the cryptographic adapter Off for all partitions, that have it in their Cryptographic Candidate List (perform as described in "Config On/Off from the CPC Work Area" on page 215).
3. From the Views, select **Groups CPC** to get in the CPC Work Area.

4. In the CPC Work Area, right-click the selected CPC icon to get the context menu. Select **CPC Configuration Cryptographic Configuration** as shown in Figure 6-14 to get to the Cryptographic configuration panel.

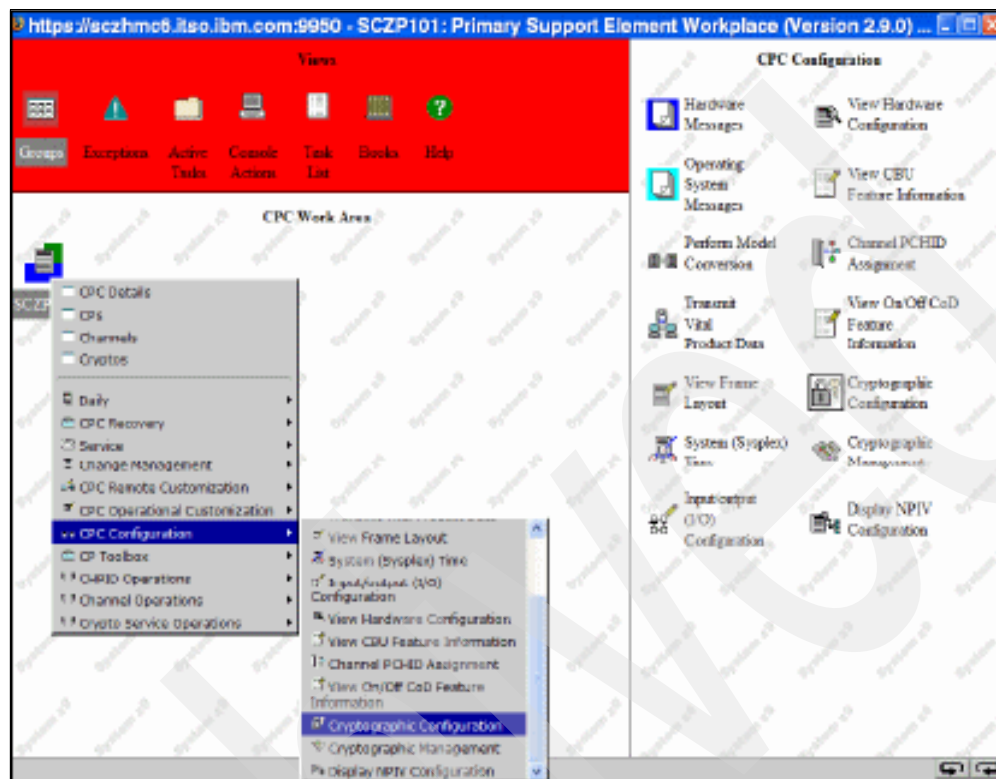


Figure 6-14 CPC Cryptos Work Area and Crypto Service Operation

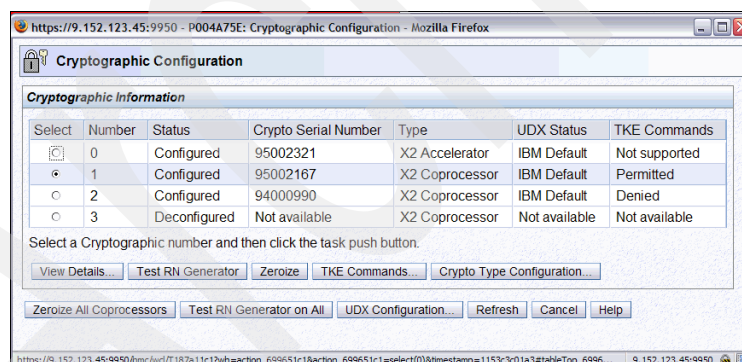


Figure 6-15 Cryptographic Configuration

- From the Cryptographic Configuration panel, select the target PCI-X adapter and click **Crypto Type Configuration** to display the Crypto Type Configuration panel shown in Figure 6-16.

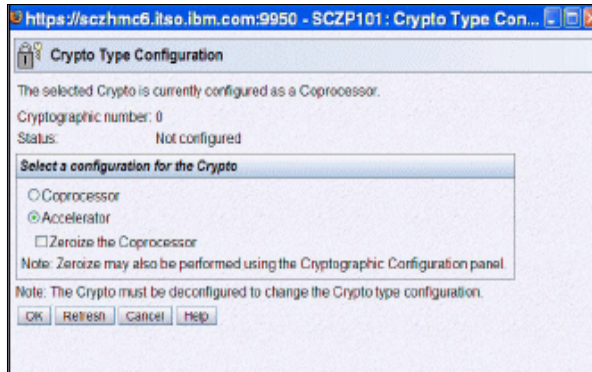


Figure 6-16 Crypto Type Configuration

- On the Crypto Type Configuration panel select a target configuration for the Crypto adapter; choose either **Coprocessor** or **Accelerator**. If you select **Accelerator**, there is also an option to zeroize the coprocessor. Click **OK**.
- A confirmation message displays (Figure 6-17).

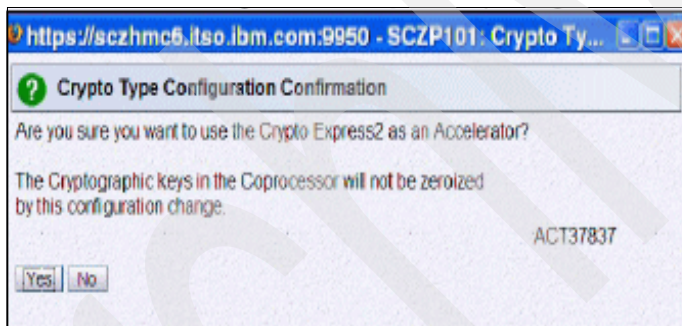


Figure 6-17 Crypto Type Configuration confirmation, message ACT37837

- After clicking **Yes**, a completion message displays (Figure 6-18). Click **OK**. You have completed the reconfiguration of the adapter.

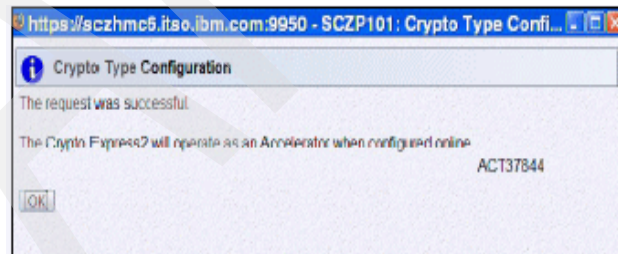


Figure 6-18 Crypto Type Configuration, message ACT37844

- Now, the last step is to configure the adapter back On, as explained in “Config On/Off from the CPC Work Area” on page 215.

The reconfigured PCI-X Adapter status might still be displayed *Initializing* before returning to *Operation*. Now the preparation of the hardware to use CryptoExpress2 is ready.

6.3 z/VM definitions

When an LPAR has been configured to benefit from hardware cryptography support, z/VM running in such an LPAR can use the hardware support for cryptographic operations to provide it to its guests. In this section, we focus on how to set up the z/VM definitions for guests running Linux on System z.

The way how z/VM provides this support is by gaining access to the Adjunct Processor (AP) queues to the guests. From a system implementation perspective, an AP of a Crypto Express2 feature is one of its internal cryptography engines (cryptography coprocessor units). Note, that AP designates to the processor, while AP ID specifies the number associated with it.

Note: While in z/VM documentation the term AP is widely used, whereas in the IBM System z9 hardware documentation, for example in *System z9 Processor Resource/Systems Manager Planning Guide*, SB10-7041, and when using the SE or HMC for configuration setup this term is usually not used. In these areas the terms *cryptography coprocessor*, *cryptography engine*, or *cryptography card* are used.

To make use of the accessible hardware by z/VM and to provide it to the guests, note the following rules:

- ▶ Each Adjunct Processor (AP) can have up to 16 usage domains assigned to it.
- ▶ Each usage domain:
 - Has a separate set of master keys for secure key operation stored in the CEX2C
 - Is associated with a separate AP queue
- ▶ The AP queues reside in the Hardware System Areas (HSA) and provide access to an AP.
- ▶ An AP queue can be identified by the AP number and the usage domain index.
- ▶ The AP numbers are assigned to the Cryptographic Candidate List or Cryptographic Online List in the LPAR activation profile.
- ▶ Each LPAR is assigned at least one usage domain which apply to all of the APs configured to this LPAR.
- ▶ An AP can be shared among 16 LPARs.
- ▶ The combination of usage domain and AP must be unique among active LPARs.

According to these rules, a z/VM system can have up to 256 AP queues, which can be used by the z/VM guests.

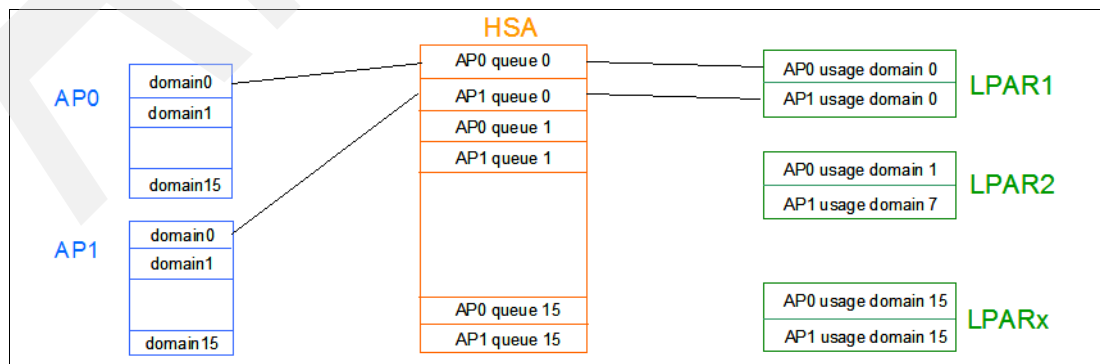


Figure 6-19 Mapping of AP queues to LPARs

In a z/VM environment it is expected, that the LPAR running z/VM will have access to multiple AP queues. There are two ways z/VM can provide access for the guests to the AP queues:

- ▶ Shared queue support
- ▶ Dedicated queue support

The shared queues support is the choice to provide for one or more Linux guests hardware encryption support for clear key operation (for example for SSL).² With shared queue support, z/VM intercepts and simulates AP instructions for all shared-queue guests, and then issues the instructions on behalf of the guest. If there are multiple queues available, z/VM decides which AP queue is actually used under the cover.

The dedicated queue support for a guest must be used, if the guest needs secure key support and relies on stored encryption keys in the hardware coprocessors. For guests using dedicated-queue support, z/VM does not intercept and instead allows the guest to execute the AP instructions under SIE. In this case, no virtualization of AP queues is done.

If secure key encryption is needed in a Linux guest, the Linux System needs at least one dedicated AP queue. This means, inside of one z/VM environment a maximum of 256 individual Linux guests could use secure encryption, as there are up to 256 AP queues, which can be dedicated to the guests.

If you dedicate the same AP queue(s) to multiple guests, only that guest can use the queues that is logged on first.

In an environment, where the Linux guests on z/VM only need clear key support, use the shared queue support for hardware encryption. Even z/VM virtualizes the AP queues for shared queue support, there must be at least one physical queue available for z/VM that is not dedicated to any guest.

The IBM System z allows hotplugging of additional cryptography hardware. Note, that for hotplugging, your system must be configured accordingly (see 6.2.2, “Customize the partition image profile for cryptographic usage” on page 209). When a PCI cryptography card is hotplugged, a Channel Subsystem Call (CHSC) event notification is received from the channel subsystem to indicate that the accessibility of one or more APs has been changed. z/VM can handle this support and determine, which AP cryptography queues are available. The hotplugging support applies only for the z/VM system itself to provide additional capacity for queue-sharing. The CHSC event notification is not simulated to the guest. Added dedicated queues to a guest will not be available for the guest until the guest is restarted! An AP that is hotplugged is recognized by CP even if it is the first AP to come up since system initialization. If a AP is hotplugged after system initialization, Linux guests that are already logged on prior to this event, are not able to have shared-queue access. Linux guests that log on after the AP is hotplugged can have access to the shared-queues.

6.3.1 Setup for a Linux guest to use cryptography cards

To enable a z/VM guest (Linux guest) to make use of the hardware cryptography support provided by the Crypto Express2 feature, there must be an entry in the user directory of the Linux guest in the VM USER statement. This is done with the CRYPTO statement for each guest (see *z/VM CP Planning and Administration*, SC24-6083).

² This applies also for z/OS and z/VSE guests, if only clear key support is necessary.

Guests with dedicated-queues support

For a Linux guest that needs access to dedicated-queues, the CRYPTO statement in the USER entry for the guest needs to contain which domain and which AP number is used. This means, on or more AP queues are identified and reserved for this guest. There is no virtualization for these dedicated-queues, no sharing will be done and the queues are not available for other guests. With dedicated-queues secure key as well as clear key operations can be performed by the Linux guest. The statement in the directory looks like:

```
CRYPTO DOMAIN x APDED y
```

where

DOMAIN x: x can be one or more domains defined for the z/VM LPAR

APDED y: y can be one or more AP (CEX2C cards) defined for the z/VM LPAR

The domain index must be selected out of the set of Usage Domain Index as specified in the Crypto Image Profile. Accordingly the APs must be selected out of the APs specified in the Cryptographic Candidate List of the Crypto Image Profile page (see Figure 6-5 on page 211). The combination of AP numbers and Domain numbers should be unique across all cryptography users in the directory. Although directory processing allows you to specify the same AP and DOMAIN combination for multiple users, these users should not be logged on at the same time. If they are, more than one user might have concurrent access to the same AP queue. Directory processing does not enforce this restriction because duplicate definitions can be useful for backup configurations.

Note that you can have multiple CRYPTO statements within one single user statement. However, if you choose different domains to different APs, all APs are available for all defined domains:

```
CRYPTO DOMAIN 10 APDED 1
CRYPTO DOMAIN 11 APDED 4
```

means that AP 1 and 4 are defined to the domains 10 and 12. This corresponds to:

```
CRYPTO DOMAIN 10 11 APDED 1 4
```

In Figure 6-20 on page 223 we illustrate a configuration in which LPAR A has access to AP 0 and 1 through the Cryptographic Candidate List, and it has also access to domain 1 and 2 through the Usage Domain Index List in the Crypto Image Profile page of LPAR A. The first guest has dedicated access to domain 1 on two different APs. The second guest has dedicated access to domain 2 on the first AP and the third guest has dedicated access to domain 2 on the second AP. The directory entry for the guests would look as shown in Example 6-1.

Example 6-1 Sample directory entries for dedicated-queues for cryptography access

```
USER GUEST1 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO DOMAIN 1 APDED 0 1
----- 3 line(s) not displayed -----
USER GUEST2 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO DOMAIN 2 APDED 0
```

```

----- 3 line(s) not displayed -----
USER GUEST3 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO DOMAIN 2 APDED 1
----- 3 line(s) not displayed -----

```

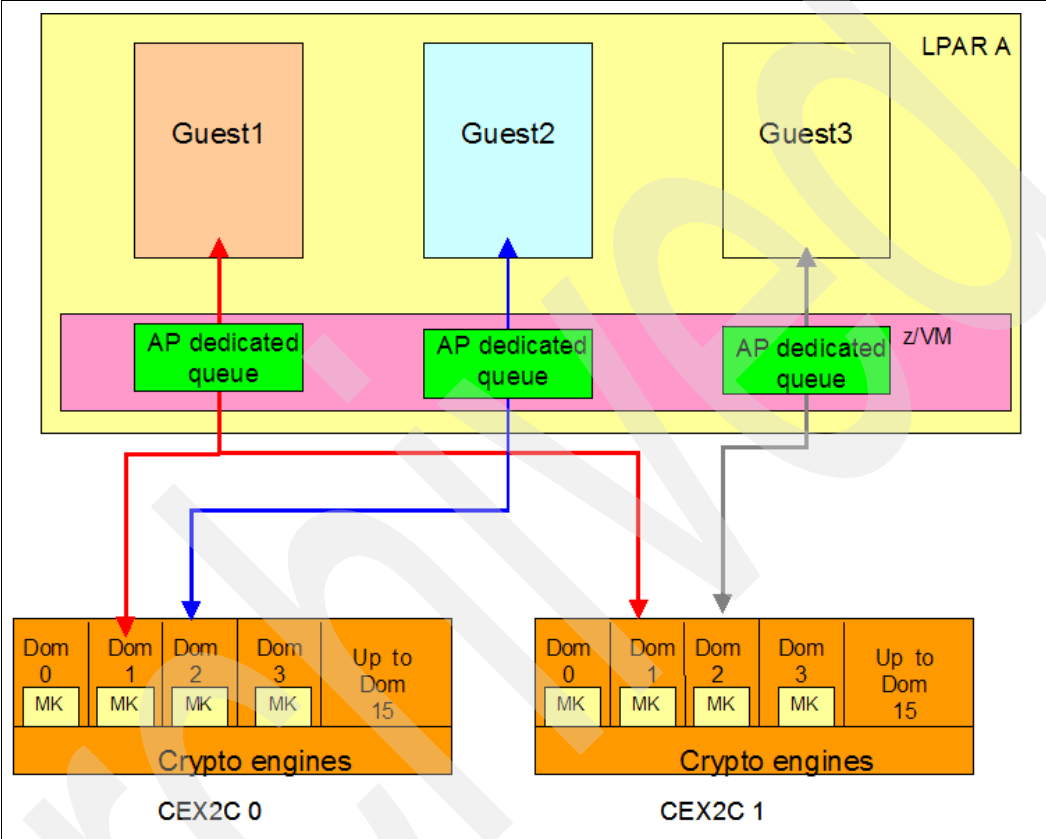


Figure 6-20 z/VM guests with dedicated-queues for cryptography access

Guests with shared-queue support

For a Linux guest that needs access to clear key cryptography operations, shared access to AP queues is the preferred way to implement. For this case the CRYPTO statement in the USER entry for the guest needs only to indicate that access to virtual queues is desired. No domain and no AP queue need to be specified. The Linux guest gets one virtualized card. One random virtual queue on one random virtual AP. The AP number and domain are chosen by z/VM and are not identical with the one for the z/VM LPAR. For this support, z/VM uses all available AP queues, which are not dedicated to other guests, and these are shared between all guests using the shared support. If there are multiple AP types available for z/VM, then z/VM will choose the best AP type for acceleration for the Linux guest. The internal order for prioritizing for z/VM is: CEX2A, PCICA, CEX2C, PCIXCC, PCICC. When a type is selected, z/VM will route all cryptography requests from the guest to however many queues/cards of that type are available. The statement in the directory looks like this:

```
CRYPTO APVIRT
```

Figure 6-21 illustrates a configuration in which the z/VM LPAR has access to coprocessor AP02, AP03, AP04, AP05, and AP07 through the Cryptographic Candidate List, and it has also access to Domain 7 and 9 through the Usage Domain Index list in the Crypto Image Profile page of the z/VM partition. The first guest has dedicated access to domain 9 on AP02 and AP03, whereas the second and third guest are using shared access to the remaining APs for clear key operations. The AP queues number and the domain number, which are provided by z/VM to these two guests are virtual numbers and do not correspond to the “real” domains and APs, which are used by z/VM to execute the cryptography requests of these guests. The directory entry for the guests would look like as shown in Example 6-2.

Example 6-2 directory entry with dedicated and shared cryptography queues

```

USER GUESTL1 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO DOMAIN 9 APDED 2 3
----- 3 line(s) not displayed -----
USER GUESTL2 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO APVIRT
----- 3 line(s) not displayed -----
USER GUESTL3 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
  CRYPTO APVIRT
----- 3 line(s) not displayed -----

```

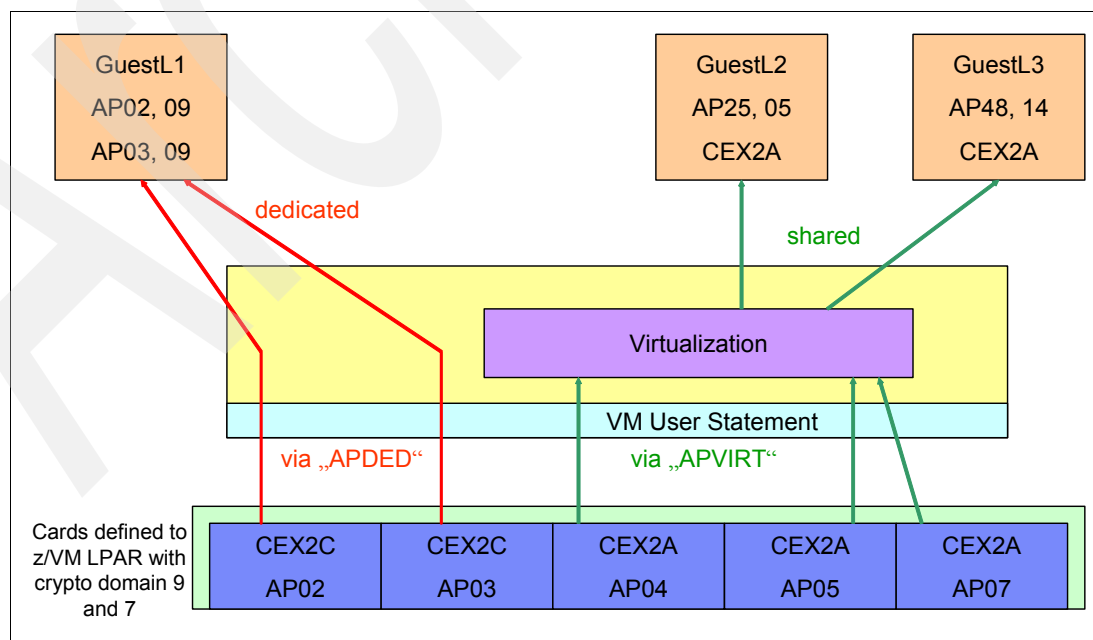


Figure 6-21 Dedicated and shared queues for Crypto access

To update the USER entry in the directory to contain the CRYPTO statement, you can use an editor, and change all necessary USER entries. In an environment with DirMaint, proceed as described below to provide shared access to a Linux guest LNXSU1 for clear key operation and dedicated access to the AP queue with domain 11 and AP number 02 to LNXSU2 for secure key operation.

To change the directory for LNXSU1 to get shared access to the cryptography hardware, issue the command:

```
dirm for LNXSU1 crypto
```

You get the panel shown in Figure 6-22. In this panel select APVIRTUAL (for the operand APVIRT in the CRYPTO statement) with any character and press F5 to submit the request.

```

-----DirMaint CRYPTO-----
Query or update the current CRYPTO statement in the user's directory entry.
Select one of the following:
_ ? (Query) _ DELETE X APVIRTUAL _ DOMAIN
For DOMAIN, Select one or more domain values (0 thru 15):
_ 0 _ 1 _ 2 _ 3 _ 4 _ 5 _ 6 _ 7
_ 8 _ 9 _ 10 _ 11 _ 12 _ 13 _ 14 _ 15
Optionally select one of the following:
CSU _ * _ 0 or _ 1
Optionally select one or more of the following:
_ APVIRTUAL _ KEYENTRY _ MODIFY _ SPECIAL _ APDEDICATED
For APDEDICATED, Select one or more ap values (0 thru 63):
_ 0 _ 1 _ 2 _ 3 _ 4 _ 5 _ 6 _ 7 _ 8 _ 9
_ 10 _ 11 _ 12 _ 13 _ 14 _ 15 _ 16 _ 17 _ 18 _ 19
_ 20 _ 21 _ 22 _ 23 _ 24 _ 25 _ 26 _ 27 _ 28 _ 29
_ 30 _ 31 _ 32 _ 33 _ 34 _ 35 _ 36 _ 37 _ 38 _ 39
_ 40 _ 41 _ 42 _ 43 _ 44 _ 45 _ 46 _ 47 _ 48 _ 49
_ 50 _ 51 _ 52 _ 53 _ 54 _ 55 _ 56 _ 57 _ 58 _ 59
_ 60 _ 61 _ 62 _ 63

5741-A05 (c) Copyright IBM Corporation 1979, 2007.
1= Help      2= Prefix Operands    3= Quit      5=Submit    12=Cursor

```

Figure 6-22 Define a Linux guest with APVIRT

The processing starts and you get the successful completion message of your request (Figure 6-23).

```
DVHZMT1191I Your CRYPTO request has been sent for processing.  
Ready; T=0.07/0.07 14:10:39  
DVHREQ2288I Your CRYPTO request for LNXSU1 at * has been accepted.  
DVHBIU3450I The source for directory entry LNXSU1 has been updated.  
DVHBIU3424I The next ONLINE will take place immediately.  
DVHDRC3451I The next ONLINE will take place via delta object directory.  
DVHBIU3428I Changes made to directory entry LNXSU1 have been placed  
DVHBIU3428I online.  
DVHREQ2289I Your CRYPTO request for LNXSU1 at * has completed; with RC =  
DVHREQ2289I 0.
```

Figure 6-23 Successful completion message of DirMaint processing

To change the directory for LNXSU2 to get dedicated AP queue access, issue the following command:

```
dirm for LNXSU2 crypto
```


You get the panel shown in Figure 6-24. In this panel, follow these steps:

1. Select DOMAIN, the DOMAIN value (domain number) APDEDICATED, and a value for APDEDICATED (AP queue number) with any character.
2. Then, press F5 to submit the request.
3. Ensure, that you select a queue number that belongs to a coprocessor (CEX2C) and not to an accelerator (CEX2A). Verify the queue number by means of the QUERY USER CRYPTO command, as shown in Example 6-5 on page 229.

```

-----DirMaint CRYPTO-----
Query or update the current CRYPTO statement in the user's directory entry.
Select one of the following:
  _ ? (Query) _ DELETE _ APVIRTUAL X DOMAIN
For DOMAIN, Select one or more domain values (0 thru 15):
  _ 0 _ 1 _ 2 _ 3 _ 4 _ 5 _ 6 _ 7
  _ 8 _ 9 _ 10 X 11 _ 12 _ 13 _ 14 _ 15
Optionally select one of the following:
  CSU _ * _ 0 or _ 1
Optionally select one or more of the following:
  _ APVIRTUAL _ KEYENTRY _ MODIFY _ SPECIAL X APDEDICATED
For APDEDICATED, Select one or more ap values (0 thru 63):
  _ 0 _ 1 X 2 _ 3 _ 4 _ 5 _ 6 _ 7 _ 8 _ 9
  _ 10 _ 11 _ 12 _ 13 _ 14 _ 15 _ 16 _ 17 _ 18 _ 19
  _ 20 _ 21 _ 22 _ 23 _ 24 _ 25 _ 26 _ 27 _ 28 _ 29
  _ 30 _ 31 _ 32 _ 33 _ 34 _ 35 _ 36 _ 37 _ 38 _ 39
  _ 40 _ 41 _ 42 _ 43 _ 44 _ 45 _ 46 _ 47 _ 48 _ 49
  _ 50 _ 51 _ 52 _ 53 _ 54 _ 55 _ 56 _ 57 _ 58 _ 59
  _ 60 _ 61 _ 62 _ 63

5741-A05 (c) Copyright IBM Corporation 1979, 2007.
1= Help      2= Prefix Operands      3= Quit      5=Submit      12=Cursor
====>

Macro-read 1 File

```

Figure 6-24 Define dedicated access to an AP queue for a Linux user

The processing starts and you get the success message shown in Figure 6-25.

```
DVHEMT1191I Your CRYPTO request has been sent for processing.
Ready; T=0.07/0.07 14:26:59
DVHREQ2288I Your CRYPTO request for LNKSU2 at * has been accepted.
DVHBIU3450I The source for directory entry LNKSU2 has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHDRC3451I The next ONLINE will take place via delta object directory.
DVHBIU3428I Changes made to directory entry LNKSU2 have been placed
DVHBIU3428I online.
DVHREQ2289I Your CRYPTO request for LNKSU2 at * has completed; with RC =
DVHREQ2289I 0.
```

RUNNING VMLINU25

Figure 6-25 Successfully processing with DirMaint

In our environment, we set up DirMaint so that it processes the directory changes immediately. As soon as the Linux guests LNKSU1 and LNKSU2 are logged on or, if these guests are already logged on, then when they log off and log on again, they can use the newly provided support.

After these DirMaint commands, the directory entries of the two Linux users are changed and now contain the CRPYTO statement (see Example 6-3).

Example 6-3 Directory entries for two Linux guests with APVIRT and APDED

```
USER LNKSU1 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  CRYPTO APVIRTUAL
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
----- 3 line(s) not displayed -----
*
*DVHOPT LNK0 LOG1 RCM1 SMS0 NPW1 LNGAMENG PWC20070710 CRCM|
USER LNKSU2 xxxxxx 256M 1G G
  INCLUDE IBMDFLT
  CRYPTO DOMAIN 11 APDEDICATED 2
  IPL CMS
  MACH XA
  NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
----- 3 line(s) not displayed -----
```

Checking the cryptographic card definitions in z/VM

To verify that hardware cryptography support is available in z/VM and can be provided to z/VM guests, you can verify the definitions in the image activation profile of the LPAR in which z/VM is running. Then, you can check the definitions in the z/VM user directory to see what is already provided to the guests.

The QUERY CRYPTO command

You can use the QUERY CRYPTO command to verify the cryptography support. Figure 6-26 shows the syntax for this command. For more information about this command, see *z/VM CP Commands and Utilities*, SC24-6081.

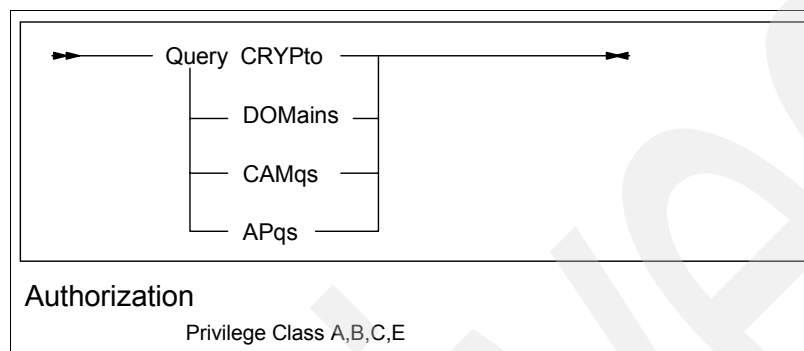


Figure 6-26 QUERY CRYPTO syntax

The QUERY CRYPTO command displays the status of the cryptographic units in the processor configuration and the status of installed domains, Crypto Asynchronous Messages (CAM) including Direct Attached Crypto (DAD) queues, and Adjunct Processor (AP) queues. This command displays only information about the subset of cryptography cards and domains as defined for the LPAR in which the z/VM system is running. The z/VM user that performs this command must have a high Privilege Class. In general, a Linux guest has Privilege Class G. Therefore, this command cannot be issued from inside such a guest machine.

Note: CAM and DAD refers to servers prior to z990, z890.

In z/VM environment, where there are cryptographic units available but no guests have already been assigned access, Example 6-4 shows the response to the QUERY CRYPTO command.

Example 6-4 Crypto units available, z/VM guests do not have access to AP queues

CP Q CRYPTO

Crypto Adjunct Processor Instructions are installed

Using also the AP operand, you get more information about the installed AP queues as shown in Example 6-5. In this example, there is one domain 11, and 4 APs are available. In addition, there is no AP queue dedicated, and for shared access (clear key) of the queues, the coprocessors are superseded because there is a mixture of AP queue types.

Example 6-5 Crypto queues are available but are not dedicated

CP Q CRYPTO AP

AP 00 CEX2A Queue 11 is installed
AP 01 CEX2A Queue 11 is installed
AP 02 CEX2C Queue 11 is superseded by CEX2A
AP 03 CEX2C Queue 11 is superseded by CEX2A

Note: In the response of QUERY CRYPTO AP, the AP 00 means crypto processor 00, equivalent to AP number, and Queue 11 means crypto domain 11 as defined in the image activation profile of the z/VM LPAR.

After enabling one or more z/VM guests to use hardware cryptography support by including an appropriate CRYPTO statement in their USER entry, the response to the QUERY CRYPTO command reflects the changed setup. Example 6-6 shows how it looks when there are two queues dedicated to one or more guests that are currently not logged on (AP00 and AP01) and how it looks when queue AP02 is dedicated to user T2912002. In this example, the non-dedicated queues can be used for shared access. In addition, with shared access, some queues are superseded as a mixture of queue types.

Example 6-6 Response to QUERY CRYPTO for dedicated and shared AP queues

CP Q CRYPTO AP

```
AP 00 CEX2C Queue 10 is reserved for dedicated use <= are dedicated for secure
key on other guests
AP 01 CEX2C Queue 10 is reserved for dedicated use <=
AP 02 CEX2C Queue 10 is dedicated to T2912002 <= is dedicated to guest you are
logged on
AP 03 CEX2A Queue 10 is installed
AP 04 CEX2C Queue 10 is superseded by CEX2A
AP 05 CEX2C Queue 10 is superseded by CEX2A
AP 06 CEX2C Queue 10 is superseded by CEX2A > are available for APVIRT
(clear key)
AP 07 CEX2C Queue 10 is superseded by CEX2A
AP 08 CEX2C Queue 10 is superseded by CEX2A
AP 09 CEX2A Queue 10 is installed
AP 10 CEX2C Queue 10 is superseded by CEX2A
AP 11 CEX2A Queue 10 is installed
```

Info: AP 00 means crypto processor 00 as shown on SE (owning AP queue 00)
Queue 10 means domain id is '10' (VM LPAR crypto domain is 10)

With the four available queues for z/VM as shown in Example 6-5 on page 229 and directory definitions for Linux guests as shown in Example 6-3 on page 228, Example 6-7 shows the response to the QUERY CRYPTO command.

Example 6-7 Response to QUERY CRYPTO with dedicated use of AP

CP Q CRYPTO AP

```
AP 00 CEX2A Queue 11 is installed
AP 01 CEX2A Queue 11 is installed
AP 02 CEX2C Queue 11 is reserved for dedicated use
AP 03 CEX2C Queue 11 is superseded by CEX2A
```

The QUERY VIRTUAL CRYPTO command

The QUERY VIRTUAL CRYPTO command displays status information of your virtual cryptographic facilities of the z/VM guest, on which you are currently logged. If the guest to which you are currently logged does not have access to cryptography queues, the response is shown in Example 6-8.

Example 6-8 Guest does not have access to cryptography queues

CP Q V CRYPTO

No AP Crypto Queues are available

If the guest has access to a dedicated AP queue, then the command looks like that shown in Example 6-9.

Example 6-9 Guest has access to a dedicated AP queue

CP Q V CRYPTO

AP 02 CEX2C Queue 10 dedicated

If the guest has access to shared AP queues for clear key operations, then the indicated AP and Queue value are virtual numbers. The response looks like that shown in Example 6-10.

Example 6-10 Guest has access to shared AP queues for clear key operations

CP Q V CRYPTO

AP 13 CEX2A Queue 15 shared

6.4 Using cryptography hardware support with Linux

In this section, we describe how to set up Linux to use hardware cryptography support. If not otherwise stated, the description and the examples refer to Novell SUSE Linux SLES 10 SP1.

Note: For the Linux setup, there is no difference whether Linux is running as a guest under z/VM or native in a LPAR.

Linux provides encryption support for different areas:

- ▶ In-kernel cryptography
- ▶ Cryptographic support for user programs or applications

In-kernel cryptography is used when the Linux kernel itself performs encryption requests by using in-kernel modules. Because the in-kernel modules are not available for user programs and applications, there are specific encryption libraries available. To make use of hardware support for encryption, the in-kernel cryptography modules or the cryptography libraries must be aware of the available hardware and be able to use it. The System z specific modules and libraries can autodetect which hardware support is available. The in-kernel modules are shipped with the kernel, whereas the support for user programs might be installed separately (using YaST, RPM).

To set up a Linux system for using hardware support for encryption, you need to have the following software installed:

- ▶ z90crypt device driver
- ▶ OpenSSL
- ▶ libica
- ▶ openCryptoki
- ▶ ibmca engine (OpenSSL engine).

When also using secure key cryptography used on the Linux system, you need the Common Cryptographic Architecture (CCA) libraries. The CCA libraries provided by IBM are not Open Source but proprietary code. Therefore, they are made available only as binaries.

Ensure, that the software is installed and configured and that the 32- and 64-bit versions (if available) are installed. For this purpose you can use YaST or the following command:

```
rpm -q <packagename>
```

You should *not* get a message similar to:

```
package <packagename> not installed.
```

Example 6-11 shows our setup.

Example 6-11 Necessary software for cryptographic with Linux

```
lnxsul:/ # uname -a  
Linux lnxsul 2.6.16.46-0.12-default #1 SMP Thu May 17 14:00:09 UTC 2007 s390x  
s390x s390x GNU/Linux
```

```
lnxsul:/ # rpm -q openssl  
openssl-0.9.8a-18.15  
lnxsul:/ # rpm -q openssl-32bit  
openssl-32bit-0.9.8a-18.15
```

```
lnxsul:/ # rpm -q openCryptoki  
openCryptoki-2.2.2-24.14  
lnxsul:/ # rpm -q openCryptoki-32bit  
openCryptoki-32bit-2.2.2-24.14  
lnxsul:/ # rpm -q openCryptoki-64bit  
openCryptoki-64bit-2.2.2-24.14
```

```
lnxsul:/ # rpm -q xcryptolinzGA  
xcryptolinzGA-3.28-rc08
```

```
lnxsul:/ # rpm -q libica  
libica-1.3.7-0.17  
lnxsul:/ # rpm -q libica-32bit  
libica-32bit-1.3.7-0.17
```

```
lnxsul:/ # rpm -q openssl-ibmca  
openssl-ibmca-1.0.0-7.11  
lnxsul:/ # rpm -q openssl-ibmca-32bit  
openssl-ibmca-32bit-1.0.0-7.11
```

```
lnxsul:/ # locate z90crypt  
/dev/z90crypt  
/etc/init.d/boot.d/K19z90crypt
```

```

/etc/init.d/boot.d/S02z90crypt
/etc/init.d/z90crypt
/etc/sysconfig/z90crypt
/lib/modules/2.6.16.46-0.12-default/kernel/drivers/s390/crypto/z90crypt.ko
/usr/sbin/rcz90crypt
/var/adm/fillup-templates/sysconfig.z90crypt

```

Note: To be able to install openCryptoki with SUSE Linux SLES 10 SP1 without error, you also need to install the RPM package *xcryptolinz* that contains CCA host libraries and tools. This package is not delivered with the distribution of SUSE SLES 10 SP1. You can get this package from the Web site:

<http://ibm.com/security/cryptocards>

Check in the PCIXCC/CEX2C section. The manual is available through the library page. The RPM file and the README.linz are available in the download section.

All these software components have to be installed. Typically an application program does not implement its own cryptographic solution but rather uses standard interfaces to handle the encryption work. These libraries can make use of available hardware support for symmetric and asymmetric encryption request, or perform the operations in software. In the simplest way, the application provides the keys to be used for the encryption requests. This is *clear key cryptography*, as the key is somewhere in the software stack in clear. In contrast to clear key cryptography, we have *secure key cryptography* when applications can make use of the capability of the underlying hardware to store keys inside the hardware and also perform all the encryption work inside and never having the key outside of the hardware in readable form. We discussed secure key aspects in 6.4.8, “Using secure key encryption with Linux: an outlook” on page 264.

Figure 6-27 shows an overview of clear key cryptography and how different applications and the cryptography libraries in a Linux environment are related to each other.

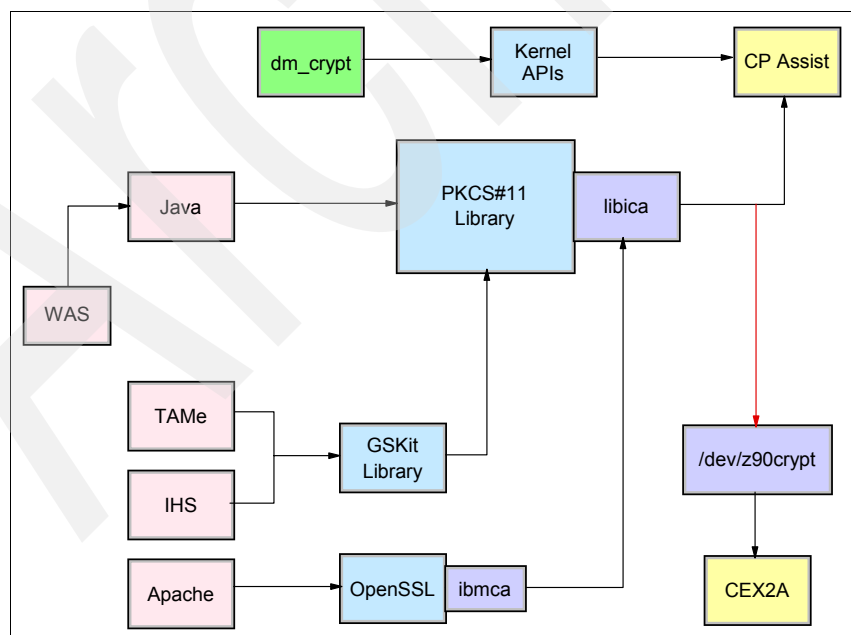


Figure 6-27 Overview of clear key cryptography exploitation in Linux

OpenSSL, PKCS#11 (openCryptoki is the Open Source implementation on Linux) and GSKiT are the APIs for accessing cryptography services on Linux for System z. OpenSSL and PKCS#11 are well established open standards. GSKiT can handle the access to the cryptography hardware of System z and provides possibilities for cryptography key handling for IBM applications. It is shipped together with the applications by IBM. In addition to the aforementioned APIs, which are available for applications running in the Linux user space, there are also cryptography kernel APIs that can make use of the CPACF (see also 6.4.7, “In-kernel cryptography with Linux kernel 2.6” on page 262).

Figure 6-27 illustrates how the different parts interact to support hardware encryption in z/VM environment running multiple Linux systems. Symmetric encryption requests which can be performed using the CPACF are directly executed using specific System z instructions by the libica library. For asymmetric encryption requests which are to be performed by the PCI-X adapters, a device driver z90crypt is used. This device driver makes use of AP queues. z/VM can virtualize the PCI-X adapters and provide virtual queues to the Linux system. A sharing of the PCI cryptography hardware is possible.

A user program uses either the OpenSSL library or the PKCS#11 interface for cryptographic operations. The OpenSSL library can pass the encryption request to the IBM engine ibmca which invokes the services of the underlying library libica. Similarly, if a user program uses the PKCS#11 interface (openCryptoki is an implementation of this interface) the request can also be passed to libica. Inside of libica it is checked, whether there is hardware support available for the requested operation. If it is a request for symmetric cryptography operation (such as DES, TDES, AES) and the CPACF feature of IBM System z is available, then libica executes the request by using the CPACF instructions. If CPACF is not available then libica performs the request in software. If it is a asymmetric request (RSA) and the a PCI feature is installed and available for Linux (APDED or APVIRT in CRYPTO statement of the Linux guest) and if the cryptography device driver (z90crypt) is loaded, then libica makes use of the PCI support for RSA. If no PCI-X adapter is available (dedicated or virtual) then libica performs the request in software.

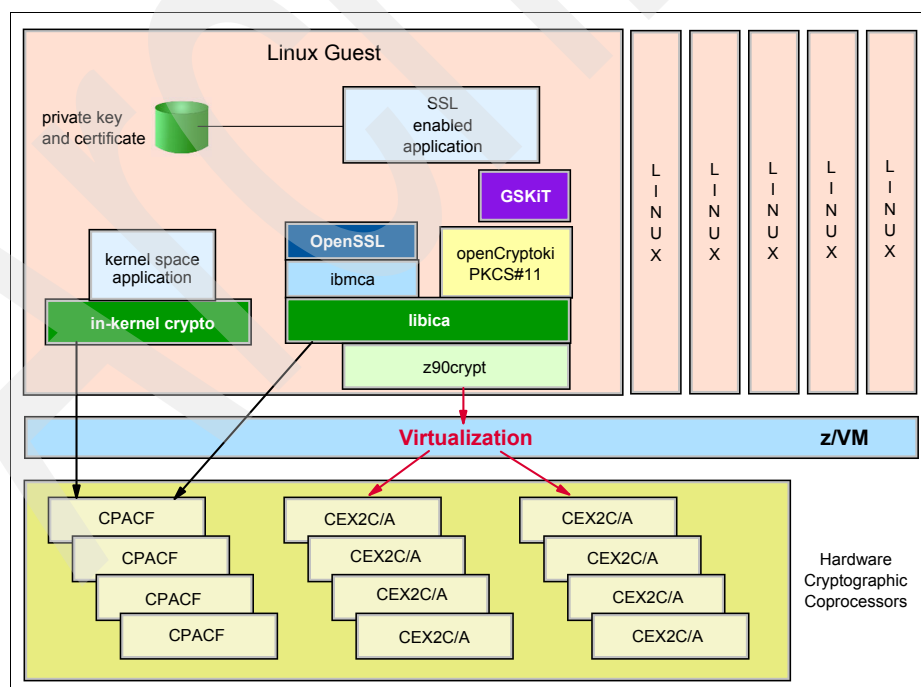


Figure 6-28 Access of hardware cryptography support by Linux clear key application

The z90crypt device driver

A new version of the z90crypt device driver is incorporated in the open source Linux kernel 2.6.19. SUSE Linux has included the new z90crypt device driver in SUSE Linux SLES10 Service Pack 1. The device driver z90crypt acts as the interface to the PCI cryptography hardware and performs asynchronous encryption operations (RSA) as used during the SSL handshake. These are clear key only operations. In addition, to the clear key encryption, now also secure key cryptography is supported. This means, that now also Linux can perform highly secure encryption tasks with the secret keys stored in special hardware and the encryption operations are performed also inside the special hardware. The z90crypt device driver together with the RPM package xcryptolinzGA allows applications to use the Common Cryptographic Architecture (CCA) interface now also on Linux for System z and submit secure key requests to the PCI-X cryptography hardware adapters. The following cryptographic features are supported: CEX2C, CEX2A, PCIXCC, PCICA, PCICC.

For IBM System z9 only the Crypto Express2 feature can be used. To perform secure key operations with z90crypt, at least one processor of the Crypto Express2 feature must be configured as cryptography coprocessor (CEX2C) and Linux needs dedicated access to one or more domains. Secure key operations can also be performed on z990 and z890 with this version of z90crypt if SUSE Linux SLES 10 SP1 is installed.

The new z90crypt device driver provides better load balancing and better performance by using a poll thread.

If there are multiple AP queues (from a Linux perspective the AP queues are treated as devices) available, the z90crypt device driver can distribute the work more efficient than in the past. Each AP device type has an internal speed rating assigned. Before submitting new requests to a device, the list of devices is sorted according their current load, where the load is determined by the number of outstanding requests on this device and its speed rating. The new requests are submitted to the device with the lowest load. This guarantees the lowest latency for new requests.

The Linux internal kernel timer interrupt resolution is 100 Hz. This means without additional polling, non-threaded applications using the cryptography device driver have a very high latency as usually the execution inside the CEX2C or CEX2A hardware is much faster than the 100 Hz resolution. To provide a way to reduce the latency for an application, the driver can run now in two modes: with or without polling thread. When running with polling thread one CPU with no outstanding workload is constantly polling the cryptographic cards for finished cryptographic requests. The polling thread will sleep when no cryptographic requests are currently being processed. This mode will utilize the cryptographic cards as much as possible at the cost of blocking one CPU during cryptographic operations. Be aware, even the poll thread is a low-prioritized kernel process to query the AP bus for outstanding requests, this process consumes CPU cycles. In a virtualized environment (z/VM) with a high number of Linux systems having the poll thread active this might lead to a higher CPU consumption.

The cryptography domain to be used by the z90crypt driver is to be specified when the driver is loaded. Domain 0 to 15 can be specified. If no domain is specified for the z90crypt the default value of -1 is used. This means that the system selects automatically the domain number, that is the domain with the highest number of AP queues is selected, if there are multiple domains with identical highest numbers of AP queues then the one with the lowest number is selected. The z90crypt driver supports only one domain to be used. For secure key cryptography, you must specify a domain number if z90crypt would not choose the domain with the desired CEX2C adapters (APs) automatically. In addition the APs used by the domain must be dedicated to the Linux system (using CRYPTO APDED in the z/VM user directory). In this case it is evident that all APs within the specified domain, must have identical secure keys. For clear key cryptography with shared APs it is not necessary to

specify a domain (using CRYPTO APVIRT in the z/VM user directory). The default -1 can be used.

To check the status of the new z90crypt driver, the following command line is still supported to examine the /proc file system:

```
cat /proc/driver/z90crypt
```

A new way to examine the file system is through the sysfs /sys/bus/ap. For more information, see “Verifying the status of the z90crypt device driver” on page 238.

Loading the z90crypt device driver

The z90crypt driver uses the domain index to access the available AP queues. If more than one domain is available in Linux (through the CRYPTO statement in the USER directory entry of z/VM), the new version of the z90crypt device driver can choose automatically the domain to be used during its loading. This is the default behavior and is identical to as though you would specify domain=-1. This is especially useful, if you work with shared AP queues, because in this case the domain number is virtual and might be different after a Logoff-Logon of the Linux system.

You can use two methods to load the device driver.

You can load the device driver manually with **modprobe** or **insmod** command, as shown in Figure 6-29.

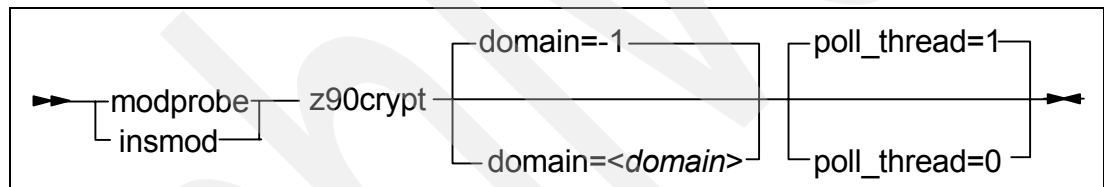


Figure 6-29 z90crypt module syntax

In this command, <domain> is of the range from 0 to 15, or -1 for autodetect, and the value for poll_thread is 1 (for enabled) or 0 (for disabled).

The other method to load the device driver is to use a script provided in SUSE Linux:

```
rcz90crypt start
```

You can insert the value for the domain and poll_thread through YaST or by editing /etc/sysconfig/z90crypt. This file is used to keep the domain value and the switch to enable the poll thread to check for outstanding encryption results on the PCI card. In an environment with multiple Linux running in one z/VM LPAR, select this value for the poll_thread carefully when loading the z90crypt driver.

Example 6-12 shows how to specify the domain and the poll thread in the /etc/sysconfig/z90crypt file.

Example 6-12 sysconfig file for z90crypt

```
## Path:      Kernel/z90Crypt
## Description: Set domain parameter for z90crypt
## Type:      integer(-1:15)
## Default:   -1
#
# This variable selects the crypto domain to be used,
# required if an LPAR owns several crypto domains.
```

```
# The value of -1 is used for autodetect.
#
Z90CRYPT_DOMAIN=-1

## Path:      Kernel/z90Crypt
## Description: Turn poll thread on/off
## Type:      list(0,1)
## Default:   1
#
# When running with polling thread one CPU without
# outstanding workload is constantly polling the
# cryptographic requests. The polling thread will
# sleep when no cryptographic requests are currently
# processed. This mode utilize the cryptographic card
# as much as possible at the cost of blocking one CPU.
# Without polling thread the cryptographic cards are
# polled at a much lower rate resulting in higher
# latency and reduced throughput for cryptographic
# requests but without a notable CPU load.
#
Z90CRYPT_POLL=1
```

To unload the device driver use either one of the following commands:

- ▶ `modprobe -r z90crypt`
- ▶ `rcz90crypt stop`

Example 6-13 shows the usage and the responses of the SUSE script `rcz90crypt`.

Example 6-13 Using SUSE `rcz90crypt`

```
lnxsul:/ # rcz90crypt start
Loading z90crypt module                               done
lnxsul:/ # rcz90crypt status
Checking for module z90crypt:                          running
lnxsul:/ # rcz90crypt stop
Unloading z90crypt module:                             done
lnxsul:/ # rcz90crypt status
Checking for module z90crypt:                          dead
lnxsul:/ # rcz90crypt start
Loading z90crypt module                               done
lnxsul:/ # rcz90crypt status
Checking for module z90crypt:                          running
```

To load the cryptography device driver automatically at Linux system boot, the `z90crypt` system service must be set to *on* for boot initialization. Use the `chkconfig` command, as shown in Example 6-14.

Example 6-14 The `chkconfig` command

```
lnxsul:/ # chkconfig z90crypt
z90crypt  off
lnxsul:/ # chkconfig z90crypt on
lnxsul:/ # chkconfig z90crypt
z90crypt  on
```

Verifying the status of the z90crypt device driver

You can check and query the status of the device driver. The first indication can be seen at the time of loading the driver.

Example 6-15 shows how a load fails if there is no AP queue (dedicated or virtual) available for the Linux. This situation might occur, if there is no CRYPTO statement for the Linux guest in the z/VM directory, or the Linux guest has not been logged off and restarted after the CRYPTO statement for this guest has been inserted in the directory.

Example 6-15 Unsuccessful load of cryptography driver as no domain available

Load via modprobe:

```
lnxsu1:~ # modprobe z90crypt domain=-1
FATAL: Error inserting z90crypt
(/lib/modules/2.6.16.46-0.12-default/kernel/drivers/s390/crypto/z90crypt.ko): No
such device
```

Load via rcz90crypt:

```
lnxsu1:~ # rcz90crypt start
Loading z90crypt moduleFATAL: Error inserting z90crypt
(/lib/modules/2.6.16.46-0.12-default/kernel/drivers/s390/crypto/z90crypt.ko): No
such device
failed
```

If everything is installed properly, the CRYPTO entry for the Linux guest is correct, and a Logoff and Logon for the Linux guest has been performed after the directory change, then the load of the cryptography driver should work, as shown with the following commands:

```
lnxsu1:~ # rcz90crypt start
Loading z90crypt module done
```

You can also verify that the driver is loaded with `lsmod`, as shown in Example 6-16.

Example 6-16 Verify that the driver is loaded with lsmod

```
lnxsu1:/ # lsmod
Module          Size  Used by
z90crypt        89976  0
...
```

After the device driver is loaded, you can query its status using either of the following methods:

- ▶ By checking the sysfs `/sys/bus/ap`
The sysfs provides various information:
 - `ap_domain` - domain used for all cards
 - `config_time` - interval for re-scanning the AP bus
 - `poll_thread` - enable/disable `poll_thread`

- device/cards<xx> - each card which is scanned
 - depth - input queue length
 - hwtype - card type number
 - modalias - internal used device bus ID
 - online - setting online/offline the card
 - request_count- number of requests processed by the card
 - type - type of the card

To check the online status of cryptographic device with bus ID 0x08 issue the following command:

```
cat /sys/bus/ap/devices/card08/online
```

The value is 1 if the device is online, and the value is 0 if the device is offline.

- By checking the /proc file system using the following command:

```
cat /proc/driver/z90crypt
```

In the meantime, the /proc file system is considered as a deprecated user interface, which disappears in the future. Therefore, the general recommendation for the future is to use the sysfs instead.

Example 6-17 shows the status of the device driver just after it has loaded and before any encryption request has been executed. In this example, the domain 15 is used, there is one CEX2A card available, and the device that is online at the ninth position is a CEX2A (the type is indicated by use of the 6). The ninth counter in last section of the output is still zero, because no encryption requests has been completed successfully up to now. The device driver status information is for up to 64 devices, which is the current architectural limit for installed cryptography cards on System z. Note that today System z9 supports up to 8 Crypto Express2 features with two cards each. When Linux is running as a guest on z/VM, the device ID is virtual, therefore be not astonished, if you would see a device ID higher than 15. The device at the ninth position corresponds to ID 0x08, as numbering starts with 0x00.

Example 6-17 Status of hardware adapter after load of z90crypt

```
lnxsul:/ # cat /proc/driver/z90crypt

zcrypt version: 2.1.0
Cryptographic domain: 15
Total device count: 1
PCICA count: 0
PCICC count: 0
PCIXCC MCL2 count: 0
PCIXCC MCL3 count: 0
CEX2C count: 0
CEX2A count: 1
requestq count: 0
pendingq count: 0
Total open handles: 0

Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
                0000000060000000 0000000000000000 0000000000000000 0000000000000000

Waiting work element counts
                0000000000000000 0000000000000000 0000000000000000 0000000000000000

Per-device successfully completed request counts
                00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

```

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

After the z90crypt driver is loaded all counters of successfully completed requests are 0. After some requests have been executed, the counters will indicate how many requests per cryptography device has been performed. Example 6-18 shows how the counter has increased (number is in hex) after an encryption request (as performed with Example 6-30 on page 246).

Example 6-18 Status of hardware adapter after execution of encryption requests

```

lnxsul:/ # cat /proc/driver/z90crypt

zcrypt version: 2.1.0
Cryptographic domain: 15
Total device count: 1
PCICA count: 0
PCICC count: 0
PCIXCC MCL2 count: 0
PCIXCC MCL3 count: 0
CEX2C count: 0
CEX2A count: 1
requestq count: 0
pendingq count: 0
Total open handles: 0

Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
00000000 00000000 0000000000000000 0000000000000000 0000000000000000

Waiting work element counts
0000000000000000 0000000000000000 0000000000000000 0000000000000000

Per-device successfully completed request counts
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00001373 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

```

6.4.1 Verify availability of cryptography hardware support

For test or troubleshooting purposes, you might want to verify the availability of the cryptography hardware and disable or enable a cryptographic device. You can do this using two methods.

The first method is by using the sysfs interface. To check the online status of cryptographic device with bus ID 0x08 issue:

```
cat /sys/bus/ap/devices/card08/online
```

To set a cryptographic device with bus device ID 0x08 online issue:

```
echo 1 > /sys/bus/ap/devices/card08/online
```

To set a cryptographic device with bus device ID 0x08 offline issue:

```
echo 0 > /sys/bus/ap/devices/card08/online
```

The second method is by using the /proc file system and editing the /proc/driver/z90crypt file using the vi editor.

To *disable* a cryptographic device:

1. To check the status, issue the commands shown in Example 6-17 on page 239.
2. Open /proc/driver/z90crypt with vi. You see several lines of code, including two lines like those shown in Example 6-19.

Example 6-19 Lines of code from /proc/driver/z90crypt

```
Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
0000000060000000 0000000000000000 0000000000000000 0000000000000000
```

The second line of code represents the list of all architecturally possible cryptographic device. Digits 1 to 6 indicate the type of the cryptographic device. In this example, one CEX2A card is in the ninth position.

3. Overwrite the digit that represents the card that you want to disable with a character *d*. In this example, to disable the card in the ninth position, overwrite the ninth digit (**6**) with a *d* (see Example 6-20).

Example 6-20 Overwrite the digit for the cryptographic device

```
Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
00000000d0000000 0000000000000000 0000000000000000 0000000000000000
```

4. Save and close /proc/driver/z90crypt. Confirm that you want to save your changes even if the content of the file has changed since you opened it.

To *enable* a disabled device:

1. Open /proc/driver/z90crypt with vi. You see several lines of code, including two lines like those shown in Example 6-21.

Example 6-21 Lines of codes from /proc/driver/z90crypt

```
Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
00000000d0000000 0000000000000000 0000000000000000 0000000000000000
```

The *d* in the second line represents the disabled device. In this example, the device at the ninth position has been disabled.

2. Overwrite the *d* that represents the device that you want to enable with an *e*, as shown in Example 6-22.

Example 6-22 Overwrite the disabled device

```
Online devices: 1=PCICA 2=PCICC 3=PCIXCC(MCL2) 4=PCIXCC(MCL3) 5=CEX2C 6=CEX2A
00000000e0000000 0000000000000000 0000000000000000 0000000000000000
```

3. Save and close /proc/driver/z90crypt. Confirm that you want to save your changes even if the content of the file has changed since you opened it. The device driver replaces automatically the *e* with the digit for the actual device type.

6.4.2 Using OpenSSL

OpenSSL implements Secure Socket Layer V2/V3—SSL and also Transport Layer Security V1 TLS—protocols. The OpenSSL library is used by applications such as Apache to perform encryption requests. The OpenSSL provides an interface to hardware engines (cryptographic devices) that perform cryptography operations in hardware instead of software. In SUSE SLES 9 and higher (also in RHEL4), OpenSSL includes an engine interface that links to the libica shared library. If hardware support for the requested encryption algorithms is available, libica used the hardware support. Otherwise, it executes the encryption in software. The engine interface can be queried to determine the supported hardware and its associated capabilities. OpenSSL supports the cryptographic adapters of System z, if the ibmca hardware engine support is reported.

The ibmca engine with OpenSSL prior to version 0.9.8

In prior SUSE SLES, where OpenSSL version was smaller than 0.9.8, an ibmca engine patch from sourceforge (<http://sourceforge.net/projects/opensslcryptoki>) was necessary to provide applications access to the cryptography hardware of System z. This patch is applied to OpenSSL before its compilation. This work has been already done by the distributors. As soon as OpenSSL is installed, ibmca engine is ready for use. This can be checked with the command `openssl engine -c`.

Example 6-23 shows the result of an OpenSSL version 0.9.7d with available support for hardware encryption support on System z (ibmca) including the supported algorithms.

Example 6-23 OpenSSL engine interface prior to version 0.9.8

```
tmcc-123-88:~ # rpm -q openssl
openssl-0.9.7d-15.21

tmcc-123-88:~ # openssl engine -c
(dynamic) Dynamic engine loading support
(cswift) CryptoSwift hardware engine support
[RSA, DSA, DH, RAND]
(chil) nCipher hardware engine support
[RSA, DH, RAND]
(atalia) Atalla hardware engine support
[RSA, DSA, DH]
(nuron) Nuron hardware engine support
[RSA, DSA, DH]
(ubsec) UBSEC hardware engine support
[RSA, DSA, DH]
(aep) Aep hardware engine support
[RSA, DSA, DH]
(ibmca) Ibmca hardware engine support
```



```
[RSA, DSA, DH, RAND, DES-ECB, DES-CBC, DES-EDE3, DES-EDE3-CBC, AES-128-ECB,
AES-128-CBC, AES-192-ECB, AES-192-CBC, AES-256-ECB, AES-256-CBC, SHA1]
(sureware) SureWare hardware engine support
[RSA, DSA, DH, RAND]
(4758cca) IBM 4758 CCA hardware engine support
[RSA, RAND]
```

The ibmca engine with OpenSSL version 0.9.8

In our SUSE SLES 10 SP1, we installed OpenSSL version 0.9.8. Since this version, the OpenSSL engine interface has been changed in a way that engines are loaded automatically from a specific directory unless they could be found to already be built in or loaded.

Because the ibmca engine support is shipped as dynamic engine in a separate RPM, you need to install the ibmca support in addition to the OpenSSL package. Use these commands to perform the installation task:

```
lnxsul:/ # rpm -qa |grep openssl-ibmca
openssl-ibmca-1.0.0-7.11
openssl-ibmca-32bit-1.0.0-7.11
```

Included in this package is a sample openssl.cnf file (see Example 6-24) that you can use to enable the ibmca engine in applications where OpenSSL config support is compiled.

Example 6-24 The openssl.cnf sample

```
lnxsul:/ # cat /usr/share/doc/packages/openssl-ibmca/openssl.cnf.sample
#
# OpenSSL example configuration file. This file will load the ibmca engine
# for all operations that the ibmca engine implements for all apps that
# have OpenSSL config support compiled into them.
#
# Adding OpenSSL config support is as simple as adding the following line to
# the app:
#
# #define OPENSSL_LOAD_CONF      1
#
openssl_conf = openssl_def

[openssl_def]
engines = engine_section

[engine_section]

foo = ibmca_section

[ibmca_section]
dynamic_path = /usr/lib64/engines/libibmca.so
engine_id = ibmca
default_algorithms = ALL
#default_algorithms = RAND,RSA
init = 1
```

Prior to the enabling of the ibmca engine, a check for the available support will not show the ibmca (see Example 6-25). To enable the ibmca engine, the content from this file should be concatenated to the existing openssl.cnf file on the host. In our environment, we append it to /etc/ssl/openssl.cnf.

Example 6-25 OpenSSL engine interface since version 0.9.8 before configuration

```
lnxsu1:/ # rpm -q openssl
openssl-0.9.8a-18.15

lnxsu1:/ # openssl engine -c
(dynamic) Dynamic engine loading support
```

After you have concatenated the content of the sample file to the configuration file, move the following line of the appended part to the top of the configuration file:

```
openssl_conf = openssl_def
```

The configuration file now looks as shown in Example 6-26.

Note: Moving the `openssl_conf = openssl_def` statement to the top of the configuration file might not be mentioned in the readme file of the `openssl-ibmca` package.

Example 6-26 Resulting OpenSSL configuration file with ibmca enabled

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#
openssl_conf = openssl_def

# This definition stops the following lines choking if HOME isn't
# defined.
HOME               = .
RANDFILE           = $ENV::HOME/.rnd

----- some lines not displayed -----

# This really needs to be in place for it to be a proxy certificate.
proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo
#
# OpenSSL example configuration file. This file will load the ibmca engine
# for all operations that the ibmca engine implements for all apps that
# have OpenSSL config support compiled into them.
#
# Adding OpenSSL config support is as simple as adding the following line to
# the app:
#
# #define OPENSSL_LOAD_CONF      1
# ----- one line moved to top of the file -----
#

[openssl_def]
engines = engine_section

[engine_section]

foo = ibmca_section

[ibmca_section]
```

```
dynamic_path = /usr/lib64/engines/libibmca.so
engine_id = ibmca
default_algorithms = ALL
#default_algorithms = RAND,RSA
init = 1
```

Note: You might have to adapt the dynamic path to your installation needs in the resulting openssl.cnf file. For example, if you need the ibmca engine for 32-bit binary:

```
dynamic_path = /usr/lib/engines/libibmca.so
```

After the preparation of the openssl.cnf the you can query OpenSSL for the availability of the ibmca engine support (see Example 6-27). Besides the information, about the availability of the ibmca hardware engine support, you can see all supported algorithms. Applications, that have OpenSSL dynamic engine loading support compiled into them will now use the ibmca engine and perform encryption requests with hardware support automatically whenever it is possible.

Example 6-27 OpenSSL engine interface since version 0.98: ibmca engine

```
lnxsul:/ # openssl engine
(dynamic) Dynamic engine loading support
(ibmca) Ibmca hardware engine support

lnxsul:/ # openssl engine -c
(dynamic) Dynamic engine loading support
(ibmca) Ibmca hardware engine support
[RSA, DSA, DH, RAND, DES-ECB, DES-CBC, DES-EDE3, DES-EDE3-CBC, AES-128-ECB,
AES-128-CBC, AES-192-ECB, AES-192-CBC, AES-256-ECB, AES-256-CBC, SHA1, SHA256]
lnxsul:/etc/ssl #
```

Testing using OpenSSL speed program

Together with the OpenSSL package a small application **speed** is provided to test several encryption algorithms. As of its nature, the OpenSSL library can execute encryption request by itself in software, or can pass the requests to an engine. To use System z9 hardware encryption support, the ibmca engine is to be used. The **openssl speed** program allows to specify through a parameter **-engine ibmca** whether or not an ibmca engine is used. If you have enabled dynamic engine loading support to use the ibmca engine, then the **openssl speed** program uses the ibmca engine even if the parameter **-engine ibmca** is not specified. You can also specify which algorithm is to be executed.

When you invoke the openssl speed program with the **-engine** parameter, and you get a message such as invalid engine ibmca (shown in Example 6-28). Then your setup is not configured correctly to use hardware encryption support with OpenSSL. (The OpenSSL speed program itself continues, but the encryption is done in software.)

Example 6-28 Hardware encryption support is not available for OpenSSL

```
lnxsu2:~ # openssl speed rsa -engine ibmca
invalid engine "ibmca"
----- some lines not displayed -----
```

Influence of a PCI card for offloading RSA

To get a first idea about RSA encryption performed by the speed application, invoke the application without the **-engine** parameter and without having enabled the dynamic engine loading support for **ibmca**. In this case the RSA is executed in software by the OpenSSL library. Example 6-29 shows the result.

Example 6-29 OpenSSL speed test program: RSA in software

```
Plnxsul:/ # openssl speed rsa
Doing 512 bit private rsa's for 10s: 10088 512 bit private RSA's in 9.99s
Doing 512 bit public rsa's for 10s: 125888 512 bit public RSA's in 9.99s
Doing 1024 bit private rsa's for 10s: 2200 1024 bit private RSA's in 9.98s
Doing 1024 bit public rsa's for 10s: 44515 1024 bit public RSA's in 10.00s
Doing 2048 bit private rsa's for 10s: 367 2048 bit private RSA's in 10.01s
Doing 2048 bit public rsa's for 10s: 13096 2048 bit public RSA's in 9.98s
----- some lines not displayed -----
              sign    verify    sign/s verify/s
rsa  512 bits 0.000990s 0.000079s   1009.8   12601.4
rsa 1024 bits 0.004536s 0.000225s    220.4    4451.5
rsa 2048 bits 0.027275s 0.000762s     36.7     1312.2
```

Then, invoke the application again with the **-engine ibmca** parameter (or enable the dynamic engine loading support for **ibmca**). In this case the RSA is executed through the **libica** in the PCI-X cryptography adapter (if available). Because the execution in the cryptography adapter means an offload from the CPU to the PCI-X adapter, for the calculation of how many requests have been performed per second, the used CPU time would produce misleading results. With offloading, the measurements should be done using the elapsed time. Therefore, you also need to specify the **-elapsed** parameter. Example 6-30 shows the result.

Example 6-30 OpenSSL speed test program: RSA with engine ibmca running one thread

```
lnxsul:/ # openssl speed rsa -engine ibmca -elapsed
engine "ibmca" set.
You have chosen to measure elapsed time instead of user CPU time.
To get the most accurate results, try to run this
program when this computer is idle.
Doing 512 bit private rsa's for 10s: 1000 512 bit private RSA's in 10.01s
Doing 512 bit public rsa's for 10s: 1000 512 bit public RSA's in 10.01s
Doing 1024 bit private rsa's for 10s: 1001 1024 bit private RSA's in 10.01s
Doing 1024 bit public rsa's for 10s: 999 1024 bit public RSA's in 10.01s
Doing 2048 bit private rsa's for 10s: 1000 2048 bit private RSA's in 10.01s
Doing 2048 bit public rsa's for 10s: 1000 2048 bit public RSA's in 10.01s
----- some lines not displayed -----
              sign    verify    sign/s verify/s
rsa  512 bits 0.010010s 0.010010s    99.9     99.9
rsa 1024 bits 0.010000s 0.010020s   100.0     99.8
rsa 2048 bits 0.010010s 0.010010s    99.9     99.9
```

You should see that the **ibmca** engine is used and that the executed numbers of requests is different to the execution in software. You should not worry if you see in this first comparison a smaller number executed in hardware than in software. This does not mean that there is no benefit from using a PCI-X adapter for cryptography support. The major difference between these two tests is that if performed in software, the CPU utilization is most likely about 100%. Whereas, the CPU utilization is very small if the RSA is performed on the PCI-X adapter.

In addition, there is another effect. The speed program serializes all the requests and does not start the next request until it got the result, and as of some internal polling mechanism of the scenario, this results in a relative few number of executed request, as the capacity of the cryptography card is not fully used. Note that the results shown in Example 6-30 are obtained having the poll thread disabled (Z90CRYPT_POLL=0).

To get an increase utilization of the capacity of the cryptography card, you can simply start simultaneously several threads. There is a **-multi** parameter that allows you to specify the number of encryption threads. Example 6-31 shows the result with eight simultaneous threads. Now, the number of executed RSA requests is much higher than in software, and still the CPU utilization is in the range of a small one digit number (in percentage). Here, you can see the benefit of offloading to a PCI-X card: for a heavy RSA load, you have a higher throughput and less CPU utilization.

Example 6-31 OpenSSL speed test program: RSA with engine ibmca running eight threads

```
lnxsul:/ # openssl speed rsa -engine ibmca -elapsed -multi 8
engine "ibmca" set.
----- some lines not displayed -----
              sign    verify    sign/s verify/s
rsa  512 bits 0.001252s 0.001251s   798.8   799.2
rsa 1024 bits 0.001251s 0.001251s   799.2   799.2
rsa 2048 bits 0.002167s 0.001251s   461.5   799.2
```

Another way to obtain high throughput with this test program is by enabling the polling (Z90CRYPT_POLL=1). Example 6-32 shows the results. We can now see, that the throughput capacity of the PCI-X is higher that RSA execution in software. We could even get more requests executed per second, if we would run multiple threads as shown in Example 6-31. In this scenario the active poll thread consumes CPU cycles during the execution of the test program. The test results of Example 6-30 up to Example 6-32 have been obtained with a PCI-X adapter configured as CEX2C.

Example 6-32 OpenSSL speed test program: RSA with engine ibmca running one thread with internal polling enabled

```
lnxsul:/ # openssl speed rsa -engine ibmca -elapsed
engine "ibmca" set.
----- some lines not displayed -----
              sign    verify    sign/s verify/s
rsa  512 bits 0.000933s 0.000805s  1071.8  1241.5
rsa 1024 bits 0.001212s 0.000857s   825.3  1166.9
rsa 2048 bits 0.003225s 0.000998s   310.1  1001.7
```

Influence of CPACF for symmetric encryption

To get a first idea about the influence of using the CPACF for symmetric encryption, invoke the speed program using the **-evp** parameter (omitting this parameter could lead to misleading interpretations in combination with the **-engine** parameter). The influence of using the CPACF for the DES-EDE3-CBC algorithm gets visible if you compare the results of Example 6-33 and Example 6-34.

Example 6-33 OpenSSL speed test program: des-ede3-cbc in software

```
lnxsul:/etc/ssl # openssl speed -evp des-ede3-cbc
----- some lines not displayed -----
The 'numbers' are in 1000s of bytes per second processed.
type           16 bytes    64 bytes   256 bytes  1024 bytes  8192 bytes
des-ede3-cbc    6980.06k    7193.43k    7242.58k    7245.14k    7255.69k
```

Example 6-34 OpenSSL speed test program: des-ede3-cbc with engine ibmca (CPACF)

```
lnxsul:/etc/ssl # openssl speed -evp des-ede3-cbc -engine ibmca
engine "ibmca" set.
----- some lines not displayed -----
The 'numbers' are in 1000s of bytes per second processed.
type           16 bytes    64 bytes   256 bytes  1024 bytes  8192 bytes
des-ede3-cbc    44735.71k   116265.11k  196768.00k  234151.25k  247256.41k
```

Note, if your Linux system runs only one CPU, then you would not get a higher number of executed cryptography requests in both cases with and without CPACF use, as the CPU utilization is already about 100% in each of these cases. Therefore, specifying the **-multi** parameter to run several threads will not change the quality of the result.

A similar effect for increasing the throughput and performance by using the CPACF can be seen also for the AES algorithm. With Example 6-34, you can see the benefit of using the CPACF for symmetric encryption. You can encrypt a lot more data when using the CPACF with a constant CPU utilization or for a given amount of encryption work, the CPU consumption is smaller.

Attention: You can use the speed program of OpenSSL to get a first idea about the supported algorithms and some potential throughput in your system. Even this small program provides some nice functions, be aware that the results are not giving you any official numbers for the performance of your System z. With this program you can observe how this program itself in combination with a lot of different factors behaves. If you would compare the results of the speed program with official performance numbers of System z, or with your application behavior in your environment, then there might be different results.

For System z encryption performance information, check *IBM System z9 Business Class, Performance of Cryptographic Operations* and *IBM System z9-109 Performance of Cryptographic Operations* for more details. You can find both documents on the IBM cryptography page, which is available at:

<http://www-03.ibm.com/systems/z/security/cryptography.html#cryptoexpress2>

Use the speed program of OpenSSL to get an idea about the influence of using hardware encryption support and for verification, whether OpenSSL is set up properly.

Considerations about CPACF

In addition to performance and throughput considerations, today there is no easy way to prove for a Linux user that the CPACF support has been really used for encryption requests

by an application. The background for this is simple. A library, such as libica, that executes encryption requests in software or in the CPACF hardware, if the CPACF is available, does usually not inform the requester by any method (for example with a message in a log file), how the request has been executed. Because there might be a very high number of individual encryption requests, log entries would probably lead to an overflow of the logfile in a very short time.

The system z9 hardware itself does not have any traces for the CPACF instructions either, so there is also no interface available on the hardware level. In the near future a new command, **icainfo**, will be made available. This command allows to query the libica library for all available encryption algorithms executed in software or in hardware on the actual running system.

6.4.3 Example: Configuring Apache 2.0 for using HW support

As an example for one important application, which benefits from hardware cryptography support we show in this section how to configure and setup Apache Web. Apache is supported on Linux for System z and can gain performance improvement for encryption by using System z9 Crypto Express2 feature during the SSL handshake and can also improve performance for the symmetric encryption of the data when using CPACF. Apache uses the OpenSSL library for encryption. Figure 6-30 shows how Apache gets access to the hardware. Apache uses mod_ssl to link to the OpenSSL library.

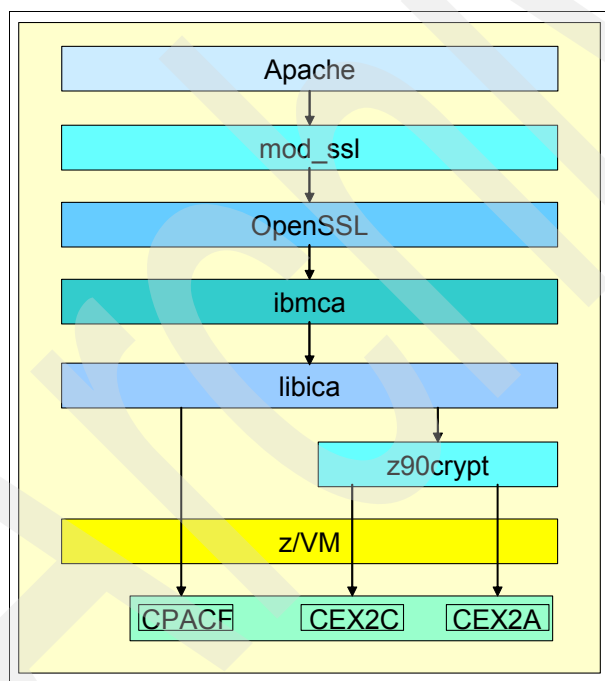


Figure 6-30 Using hardware cryptographic support with Apache

With SUSE Linux SLES 10. SP1 Apache is not enabled for SSL usage by default. To use SSL encryption some configuration has to be done up front. As soon as Apache is configured to support encryption through OpenSSL library, you can use the ibmca engine. All encryption requests are passed from OpenSSL through ibmca to the underlying libica, where all supported algorithms are executed in the hardware if available. If hardware support is not available or algorithms not supported by the installed hardware, libica performs the requests in software.

The SSL configuration for Apache 2.0 is controlled by several configuration files:

- ▶ In `/etc/sysconfig/apache2` there are general system configuration definitions
- ▶ In `/etc/apache2` directory there are server-specific configuration files and directories
 - The `httpd.conf` file is the main server configuration file
 - The `ssl-global.conf` file configures SSL for the main server and all virtual hosts
 - The `vhost.d` directory contains virtual host configuration files

With SUSE Linux, you can configure Apache 2.0 to use SSL with either of the following methods:

- ▶ With the YaST graphic interface
- ▶ Edit the configuration files manually

During SSL configuration of Apache, you need to provide a server certificate to Apache that is used to ensure the authenticity of the server and to provide the public key of the server during the SSL handshake. You can create this certificate during the SSL configuration.

Note: Depending on the requirements for security in your environment, it might be that you need to have your server certificate signed by an official CA to follow your enterprise security guidelines.

To manually configure Apache 2.0 and to enable it for SSL, follow the official installation procedures for Apache. Read also the information provided in the Apache package and check the following sources:

`/usr/share/doc/packages/apache2/README.QUICKSTART`

`/usr/share/doc/packages/apache2/README.QUICKSTART.SSL`

There are some small scripts provided that simplify the configuration tasks. Instead of using the editor, you can run these scripts. These scripts are useful for a quick configuration setup to test SSL environment. How to use these scripts is described in the file `README.QUICKSTART.SSL`.

To configure Apache 2.0 manually, follow these steps:

1. Make sure that apache starts with `mod_ssl` loaded by running:

```
a2enmod ssl
```

This script adapts `/etc/sysconfig/apache2:APACHE_MODULES`.

2. Make sure that the SSL configuration is active by running:

```
a2enflag SSL
```

This script adapts `/etc/sysconfig/apache2:APACHE_SERVER_FLAGS`.

3. For a real SSL setup for production, you might want to use TinyCA to create and manage a real SSL setup by running:

```
tinyca2
```

This script is available on SUSE Linux. Also, refer to the `mod_ssl` documentation.

4. The following steps create `_dummy_` keys:
 - a. Run `/usr/bin/gensslcert`.
 This script overwrites or writes `/etc/apache2/ssl.crt/ca.crt`,
`/etc/apache2/ssl.key/server.key`, `/etc/apache2/ssl.crt/server.crt`, and
`/etc/apache2/ssl.csr/server.csr`.
 A copy of `ca.crt` is installed as `/srv/www/htdocs/CA.crt` for download.
 - b. Issue `cp vhosts.d/vhost-ssl.template vhosts.d/vhost-ssl.conf`.
 - c. Issue `adapt vhosts.d/vhost-ssl.conf and default-server.conf al gusto`.
5. To check your vhost setup, use the following command:
`httpd2 -S -DSSL`

If you do use YaST or the scripts but want to perform all steps manually, execute the following steps to configure Apache for SSL usage:

1. Edit the `/etc/sysconf/apache2` file:
 The `APACHE_MODULES="module_list"` directive must include "ssl" in `module_list`.
 The `APACHE_SERVER_FLAGS="SSL"` directive must include "SSL".
 Execute the **SUSEconfig** command to generate the Apache configuration files.
2. Generate a server certificate or import an existing server certificate.
 SUSE provides the `/usr/share/packages/apache2/certificates.sh` command to generate a server certificate (Snake Oil).
3. Create a virtual host configuration file
 Copy a template virtual host configuration file to the virtual host configuration directory:
`# cp /etc/apache2/vhost.d/vhost-ssl.template /etc/apache2/vhost.d/vhost-ssl.conf`
 Edit the `vhost-ssl.conf` file and add the location of the server certificate and the server private key.

After having configured successfully the Apache for SSL usage you might want to start or restart the Apache Server and test it by accessing a page through SSL communication.

Enabling Apache to use ibmca OpenSSL engine interface

Up to now, no hardware support for SSL is used. To use hardware support, Apache must invoke OpenSSL with the information to use the `ibmca` engine. This can be configured with one statement in the `ssl-global.conf` file.

Edit the `/etc/apache2/ssl-global.conf` file and set the `SSLCryptoDevice` directive to `ibmca`, as shown in Example 6-35.

Example 6-35 Editing the `/etc/apache2/ssl-global.conf` file

```
# This global SSL configuration is ignored if
# "SSL" is not defined, or if "NOSSL" is defined.
<IfDefine SSL>
<IfDefine !NOSSL>
<IfModule mod_ssl.c>
    SSLCryptoDevice ibmca
    #
    # Some MIME-types for downloading Certificates and CRLs
    #
```

```
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-cr1 .cr1
```

Using encryption hardware support with Apache

Now all configuration steps are completed to run Apache with hardware support for encryption. Ensure that the z90crypt device driver is loaded. Then start or restart Apache 2.0. You can use the **rcapache2** command:

```
lnxsul:/ # rcapache2 stop
Shutting down httpd2 (waiting for all children to terminate)      done
lnxsul:/ # rcapache2 start
Starting httpd2 (prefork) httpd2-prefork: apr_sockaddr_info_get() failed for lnxsul
httpd2-prefork: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
done
```

Note: Depending on your needs, you can configure your Linux system to start Apache automatically.

Now you can check, whether Apache is using the PCI-X cryptography card support for the SSL handshake. You simply check whether the number of executed requests increases if you invoke a SSL protected page with a browser. Use `cat /proc/driver/z90crypt` before the first SSL request is performed and after some requests have been executed (see “Verifying the status of the z90crypt device driver” on page 238), or use the infos of `sysfs` (see Example 6-36).

Example 6-36 Increase of counter for executed requests after SSL protected page access

Check the current number of executed requests by the z90crypt driver:

```
lnxsul:~ # cat /sys/bus/ap/drivers/cex2a/card08/request_count
5303
```

Check the number of executed request after loading a SSL protected web page:

```
lnxsul:~ # cat /sys/bus/ap/drivers/cex2a/card08/request_count
5306
```

Note that there might be three reasons why there is no new activity visible in the device driver:

- ▶ Apache and OpenSSL are not configured correctly to use available PCI-X support.
- ▶ After the first SSL handshake is executed, using the same browser for additional SSL requests, does not increase the number of RSA request performed within a particular time frame. The SSL protocol avoids a new handshake, if possible (SSL session resume).
- ▶ During SSL handshake the client (browser) and the SSL server (Apache), negotiate a cipher suite, which is not using algorithms, for which Crypto Express2 provides hardware support.

Using hardware encryption support provides the advantage of increased performance and higher data throughput. Therefore you might want to adapt the environment to benefit from available hardware support. When the SSL client and the SSL server are negotiating for a cipher suite (i.e. which algorithm is used to exchange the keys to be used and which algorithm is used to encrypt the data) they need to find a cipher suite, which is known to both partners. If there are several common suites, the order to select the suite to be used can be determined. In a System z environment, you might want to adapt this order with respect of available hardware support. Depending on security policies in your enterprise, there might be

also additional requirement for the usage of particular cipher algorithms. Before you change the default order, ensure that you follow all necessary guidelines in your environment.

For an existing SSL session, you can simply check which algorithm has been used for data encryption. Using Firefox browser, you simply go to Tools then select Page Info and in the Page Info window select the tab Security. Figure 6-31 shows the page information of a SSL protected page. It contains the information about the used algorithm (here AES-256). Using Internet Explorer, you can obtain this information in a similar way.

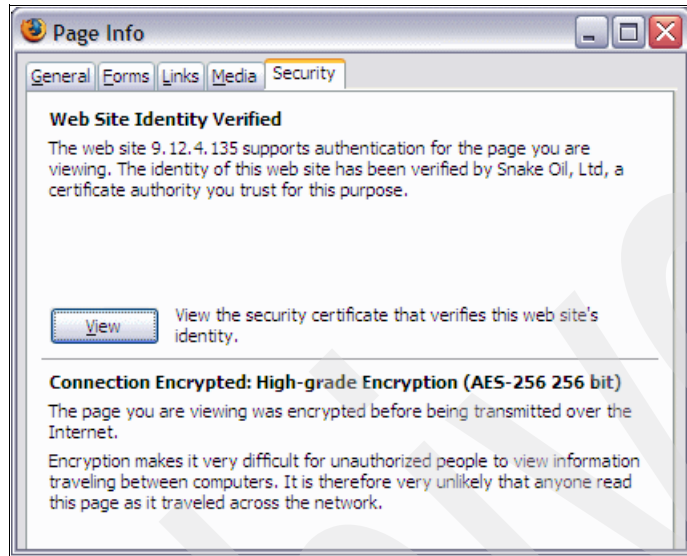


Figure 6-31 Firefox Page Info of an SSL encrypted page

The order of cipher suites to be selected during SSL handshake is determined in the virtual host configuration file `vhost-ssl.conf`. Example 6-37 shows the default order as shipped with Apache 2.0 configuration.

Example 6-37 Default order for Cipher Suites of Apache 2.0 configuration

```
<IfDefine SSL>
<IfDefine !NOSSL>

##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

    # General setup for the virtual host
    DocumentRoot "/srv/www/htdocs"
    #ServerName www.example.com:443
    #ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/error_log
    TransferLog /var/log/apache2/access_log

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # SSL Cipher Suite:
```

```
# List the ciphers that the client is permitted to negotiate.  
# See the mod_ssl documentation for a complete list.
```

```
SSLCipherSuite
```

```
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
```

When you adapt the order of cipher suites, besides putting algorithms with CPACF and PCI-X hardware support at the top of the list, consider also other security requirements in your environment. When running on a z990, there is no AES support in the CPACF available, AES is executed on a z990 in software. When you are running on IBM System z9, there is AES-128 support in CPACF available. As of August 2007 the CPACF does not support AES-256 in CPACF, but only in software. Example 6-38 of a modified `SSLCipherSuite` directive prefers a cipher suite which uses RSA (supported by Crypto Express2) and Triple DES (supported by CPACF). Using this modified directive the Security of the Page Info in Firefox browser would contain Connection Encrypted: High-grade Encryption (3DES-EDE-CBC 168bit).

Example 6-38 SSLCipher Suite directive sample

```
SSLCipherSuite EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA
```

To modify and set up successively your correct cipher-spec string, you can use the command **openssl ciphers -v <cipher-spec>**. For more details, refer to the `mod_ssl` documentation of Apache, which is available at:

http://httpd.apache.org/docs/2.0/mod/mod_ssl.html

6.4.4 Example of OpenSSH

OpenSSH is a set of programs which is used to encrypt communication sessions over a network using the ssh protocol. It replaces less secure services such as rlogin and telnet. Besides the ssh program for the client and the server sshd, there are also some other utilities contained in the OpenSSH package. During the authentication phase of a communication RSA keys can be used and for encryption of the data traffic, symmetric algorithms (such as Triple DES) are used. OpenSSH uses the OpenSSL library under the cover for multiple services. As OpenSSL can use the engine ibmca to get hardware support for cryptographic requests, there is theoretically the possibility, that also OpenSSH benefits from existing hardware encryption support. In Novell SUSE Linux SLES 10 SP1 the OpenSSH version 4.2 is used. This version is not currently taking advantage from the dynamic engine loading support and cannot request a specific engine to be used by OpenSSL. As a result, we do not support hardware encryption for this service.

Starting with version 4.4 of OpenSSH, OpenSSL dynamic engine loading is supported. For this purpose, it is only necessary to specify a flag (**--with-ssl-engine**) during the configure step when building the OpenSSH package. With this support, OpenSSH together with OpenSSL will check for an engine, and if the ibmca is the only available engine for Linux for System z, the available hardware cryptography support will be used. As soon as newer version of OpenSSH will be incorporated in the Linux for System z distribution, we will be able to take advantage from the capabilities of the System z hardware, as soon as the package would be built with dynamic engine loading support.

6.4.5 Using PKCS#11 API

In addition to OpenSSL, the PKCS#11 interface is the second method that applications request cryptographic services in a standardized manner. The `openCryptoki` package is an open source implementation of the PKCS#11 interface to provide cryptographic hardware

devices that can manage and store user keys on PKCS#11 devices. openCryptoki consists of a slot manager and an API for slot token dynamic link libraries (STDLLs). The slot manager runs as a daemon to control token slots provided to applications. Managed devices store tokens in the slot manager database. Components provided by openCryptoki include:

- ▶ Slot manager daemon (/usr/sbin/pkcs11slotd)
- ▶ Slot manager daemon service control script (/etc/init.d/pkcs11slotd)
- ▶ APIs to the STDLLs:
 - /usr/lib/opencryptoki/libopencryptoki.so
(prior to openCryptoki 2.2.2: /usr/lib/pkcs11/PKCS11_API.so)
 - /usr/lib64/opencryptoki/libopencryptoki.so
(prior to openCryptoki 2.2.2: /usr/lib/pkcs11/PKCS11_API.so64)
- ▶ Configuration utilities:
 - /usr/sbin/pkcs11_startup
 - /usr/sbin/pkcs_slot
 - /usr/sbin/pkcsconf
 - /usr/sbin/pkcsconf64
- ▶ STDLLs plugins to the cryptographic adapters:
 - /usr/lib/opencryptoki/stdll/PKCS11_ICA.so
 - /usr/lib64/opencryptoki/stdll/PKCS11_ICA.so

Configuration and PKCS#11 subsystem start up

Before configuring openCryptoki, ensure that you have installed the openCryptoki and xcryptolnz RPM packages successfully (see Example 6-11 on page 232).

You must configure openCryptoki using the pkcs11_startup script. When you execute this script, you follow these steps:

1. Create Linux group *pkcs11*.
2. Scan for an installed device (/dev/z90crypt).
3. Create the slot configuration file (/var/lib/opencryptoki/pk_config_data).

You can then start the slot manager daemon using the **pkcs11slotd** command or the openCryptoki service control script. Any application that accesses the PKCS subsystem must run as root or under a Linux user that is a member of the pkcs11 group.

Note: At the time of writing, the z90crypt driver must be loaded before pkcs11_startup script is executed to generate the IBM ICA token which enables the usage of hardware support for encryption through the PKCS#11 interface. This token is also necessary to use the CPACF support for symmetric encryption through PKCS#11. The new version of the z90crypt device driver can be loaded even if no Crypto Express2 card is available, but in this case today still the token is not built to use hardware encryption.

To start the PKCS#11 subsystem manually with SUSE SLES 10 SP1, load the z90crypt driver (if not already done), use the startup script, and start the pkcsslotd daemon using the rcpkcsslotd service control script:

```
lnxsul:~ # rcz90crypt start
Loading z90crypt module done
lnxsul:~ # pkcs11_startup
usermod: 'root' is primary group name.
lnxsul:~ # rcpkcsslotd start
Starting pkcsslotd daemon:usermod: 'root' is primary group name.
done
```

After you configure openCryptoki using pkcs11_startup, you can start the PKCS subsystem automatically at system boot. Ensure that the cryptography device driver is loaded at system initialization:

```
lnxsul:~ # chkconfig z90crypt
z90crypt on
```

Use the **chkconfig** command to start the pkcsslotd daemon at system initialization for runlevel 3 and 5:

```
lnxsul:~ # chkconfig --list | grep pkcsslotd
pkcsslotd 0:off 1:off 2:off 3:off 4:off 5:off 6:off
lnxsul:~ # chkconfig pkcsslotd on
inserv: can not symlink(../TKEcat, rc5.d/S06TKEcat): File exists
lnxsul:~ # chkconfig --list | grep pkcsslotd
pkcsslotd 0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

Managing and configuring the PKCS#11 device

Together, the PKCS#11 layer API and underlying cryptographic device is referred to as a *PKCS#11 device*. Tokens are associated with devices and are used to manage and store users keys. openCryptoki provides the **pkcsconf** command to:

- ▶ Initialize tokens.
- ▶ Set Security Officer (SO) PINs.
- ▶ Initialize, set, and change user PINs.
- ▶ Display informations about tokens.

Note: In addition to the 32-bit version of **pkcsconf**, there is also a 64-bit version (**pkcsconf64**).

Manage the PKCS#11 device

To execute **pkcsconf** commands, the slot manager daemon (**pkcsslotd** must be already running). To get syntax help of the **pkcsconf** command use the **-h** option:

```
lnxsul:~ # pkcsconf -h
pkcsconf: invalid option -- h
usage: pkcsconf [-itsmMIupP] [-c slotnumber -U userPIN -S S0Pin -n newpin]
-i display PKCS11 info
-t display token info
-s display slot info
-m display mechanism list
-I initialize token
-u initialize user PIN
-p set the user PIN
-P set the SO PIN
```

To display token information, use the **-t** option:

```
lnxsul:~ # pkcsconf -t
Token #0 Info:
    Label: IBM ICA PKCS #11
    Manufacturer: IBM Corp.
    Model: IBM ICA
    Serial Number: 123
    Flags: 0x880045
(RNG|LOGIN_REQUIRED|CLOCK_ON_TOKEN|USER_PIN_TO_BE_CHANGED|SO_PIN_TO_BE_CHANGED)
    Sessions: -1/-1
    R/W Sessions: -1/-1
    PIN Length: 4-8
    Public Memory: 0xFFFFFFFF/0xFFFFFFFF
    Private Memory: 0xFFFFFFFF/0xFFFFFFFF
    Hardware Version: 1.0
    Firmware Version: 1.0
    Time: 18:02:28
```

Note: *IBM ICA PKCS#11* is the default token label that is changed at token initialization.

To display PKCS#11 information, use the **-i** option:

```
lnxsul:~ # pkcsconf -i
PKCS#11 Info
    Version 2.11
    Manufacturer: IBM
    Flags: 0x0
    Library Description: Meta PKCS11 LIBRARY
    Library Version 2.2
```

To display slot information, use the **-s** option:

```
lnxsul:~ # pkcsconf -s
Slot #0 Info
    Description: Linux 2.6.16.46-0.12-default Linux (ICA)
    Manufacturer: Linux 2.6.16.46-0.12-default
    Flags: 0x5 (TOKEN_PRESENT|HW_SLOT)
    Hardware Version: 0.0
    Firmware Version: 1.1
```

Configuring the PKCS#11 device

To configure a PKCS#11 device

1. Initialize the token.

A token must be initialized before it can be used. To initialize the token label (replacing the default label), specify the slot number using the **-c** option and the **-I** option:

```
lnxsul:~ # pkcsconf -c 0 -I
Enter the SO PIN: *****
Enter a unique token label: MGCrypto
```

When prompted, provide the default SO PIN (87654321). To check that the label has changed, use the **-t** option:

```
lnxsul:~ # pkcsconf -t
Token #0 Info:
    Label: MGCRYPTO
    Manufacturer: IBM Corp.
    Model: IBM ICA
    Serial Number: 123
    Flags: 0x880445
(RNG|LOGIN_REQUIRED|CLOCK_ON_TOKEN|TOKEN_INITIALIZED|USER_PIN_TO_BE_CHANGED|
SO_PIN_TO_BE_CHANGED)
    Sessions: -1/-1
    R/W Sessions: -1/-1
    PIN Length: 4-8
    Public Memory: 0xFFFFFFFF/0xFFFFFFFF
    Private Memory: 0xFFFFFFFF/0xFFFFFFFF
    Hardware Version: 1.0
    Firmware Version: 1.0
    Time: 19:24:01
```

The token label (MGCRYPTO in this example) identifies the cryptographic token for storing the HTTP SSL server certificate.

2. Set the SO PIN.

It is a good security practice to set the SO PIN. The SO PIN secures access to the administrative functions for the PKCS#11 device. Use the **-P** option:

```
lnxsul:~ # pkcsconf -c 0 -P
Enter the SO PIN: *****
Enter the new SO PIN: *****
Re-enter the new SO PIN: *****
```

3. Set the user PIN.

You set the user PIN by the security officer. The user PIN secures access to the token stored in the PKCS#11 device slot database. To access the token holding the SSL certificate, users (or applications such as IBM HTTP Server) must provide the user PIN. Specify the **-u** option:

```
lnxsul:~ # pkcsconf -c 0 -u
Enter the SO PIN: *****
Enter the new user PIN: *****
Re-enter the new user PIN: *****
```

Important: Avoid the user PIN *12345678*. There is a hard coded check in openCryptoki 2.2 that will fail requests with that PIN.

4. Change the user PIN.

Use the **-p** option:

```
lnxsul:~ # pkcsconf -c 0 -p
Enter user PIN: *****
Enter the new user PIN: *****
Re-enter the new user PIN: *****
```

The user PIN is required to access the token. The length of the PIN for SO and the user is between 4 to 8 characters.

The PKCS#11 device is now configured to store and manage the keys for an application such as Tivoli Access Manager, WebSphere MQ, or IBM HTTP Server.

6.4.6 Example: configure IBM HTTP server for using HW support

As an example for an application that benefits from cryptographic hardware support, this section shows how to configure and set up IBM HTTP Server. IBM HTTP Server uses the openCryptoki PKCS#11 interface to perform SSL requests (see Figure 6-32).

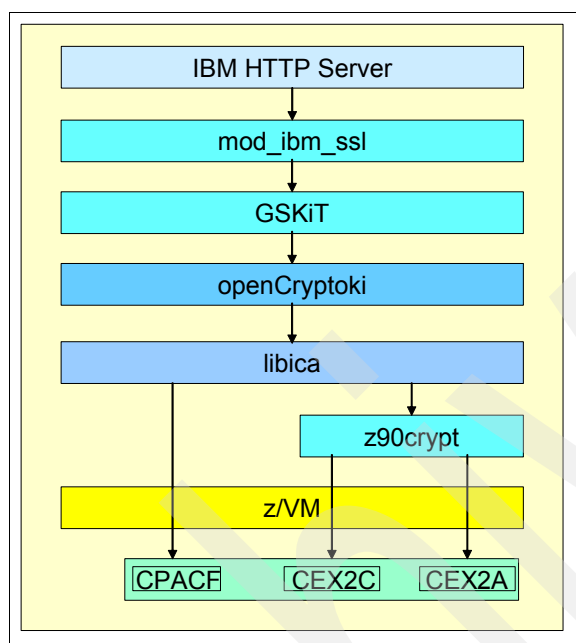


Figure 6-32 Using hardware cryptographic support with IBM HTTP Server

To run IBM HTTP Server make sure, that

- ▶ z90crypt driver is loaded (as described in “Loading the z90crypt device driver” on page 236).
- ▶ PKCS#11 subsystem is configured and started (as described in “Configuration and PKCS#11 subsystem start up” on page 255).
- ▶ GSKiT is installed with IBM HTTP Server (check the installation log files in the IBM HTTP Server installation directory).

To enable SSL with IBM HTTP Server, you need to perform the following three steps:

1. Create the server certificate.
2. Configure IBM HTTP Server for SSL.
3. Start the IBM HTTP Server server.

Creating the server certificate

IBM HTTP Server server needs a certificate to be authenticated in a SSL connection and for encryption of the cipher keys during the SSL handshake between a client and the server. To create such a certificate and store it in the PKCS#11 device, use the **keyman** command, which is part of the GSKIT tools. A detailed description how to proceed is available in *Using Cryptographic Adapters for Web Servers with Linux on IBM System z9 and zSeries*, REDP-4131 and in the IBM HTTP User Guide.

While creating a server certificate, you will be prompted to choose a Cryptographic Token Label of the PKCS#11 device. Use the token label assigned to the PKCS#11, as described “Configuring the PKCS#11 device” on page 257. In our example, we chose MGCRYPTO. As Cryptographic token Password for the PKCS#11 device, use the user PIN assigned to the PKCS#11 token (see also “Configuring the PKCS#11 device” on page 257). You also create a secondary key database file of the type CMS. Then, you can create a certificate and have it self-signed.

Note: Self-signed certificates are basically for internal test purposes. They are not intended to be used in a highly secure environment.

When you have created the certificate, the certificate label displays for your information. This label, as well as the name of the secondary key database, will have to be specified when you set up an application (such as IBM HTTP Server) to use PKCS#11.

Configuring the IBM HTTP Server server to use SSL

To configure IBM HTTP Server to use SSL, edit the `/opt/IBMIHS/conf/httpd.conf` server configuration file:

1. Define the server name.

Set the `ServerName` directive to the server host name (`ServerName hostname`).

2. Enable loading of the `mod_ibm_ssl` module.

Add a `LoadModule` directive to conditionally load the `mod_ibm_ssl` module:

```
*** ADDED FOR SSL ***
<IfDefine SSL>
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
</IfDefine>
```

3. Set the user and group used to run the server. By default, IBM HTTP Server runs under user and group *nobody*:

```
##User nobody
##Group nobody
User wwwrun
Group www
```

Note: Because IBM HTTP Server uses the PKCS subsystem, the selected user (in this case, user *wwwrun*) must be a member of group *pkcs11*. To add the user to the *pkcs11* group, use the **usermod** command:

```
# usermod -G pkcs11 wwwrun
```

4. Create a stash file to hold the user PIN of the PKCS#11 token.

To create a stash file to hold the user PIN for the token, execute the **sslstash** command:

```
# /opt/IBMIHS/bin/sslstash -h
Usage: sslstash [-c] <file> <function> <password>
-c = Create a new stash file. If not specified, sslstash will
attempt to update an existing file.
file = Fully qualified name of the file to be created or updated.
function = Function for which the password will be used. Valid
values are "crl" or "crypto".
password = The password to be stashed
```

To create a stash file for the token, execute the **sslstash** command:

```
# /opt/IBMIHS/bin/sslstash -c /opt/IBMIHS/security/pkcs11.passwd crypto
24681357
```

In this example, 24681357 is the user PIN for the token (the value assigned when the PKCS#11 is configured as described in “Managing and configuring the PKCS#11 device” on page 256).

5. Define a virtual host for the SSL connections.

VirtualHost directives can be inserted at the end of section 3 in configuration file:

```
#</VirtualHost>
<IfDefine SSL>
Listen 443
<VirtualHost 0.0.0.0:443>
SSLEnable
SSLClientAuth none
SSLServerCert MGCrypto:ntc-keyA
SSLStashfile /opt/IBMIHS/security/pkcs11.passwd
SSLPKCSDriver /usr/lib/libopencryptoki.so
</VirtualHost>
SSLDisable
Keyfile /opt/IBMIHS/security/ntckey.kdb
</IfDefine>
<IfDefine SSL>
Addtype application/x-x509-ca-cert .crt
Addtype application/x-pkcs7-crl .crl
</IfDefine>
```

Important configuration directives values include:

- **SSLServerCert**
Use the certificate label defined in “Creating the server certificate” on page 259.
- **SSLStashfile**
Use the previously created stash file name.
- **SSLPKCSDriver**
Specify the PKCS#11 API module (for 31-bit /usr/lib/libopencryptoki.so is the link to /usr/lib/opencryptoki/libopencryptoki.so).
Because openCryptoki 2.2.2 the paths has been changed. For compatibility reasons, the old paths are still working through links. Prior to openCryptoki 2.2.2 the API module was /usr/lib/pkcs11/PKCS11_API.so.
- **Keyfile**
Use the secondary key database file name defined when creating the server certificate.

Starting the IBM HTTP Server server

The server can be started in SSL mode. Use the **apachectl** command with the **startssl** option:

```
# /opt/IBMIHS/bin/apachectl startssl
```

Note: Similarly as mentioned with Apache, you might want to determine the order of Cipher Suites to be selected during the SSL handshake according the availability of your hardware encryption support to increase performance or throughput. This can be achieved by using the SSLCipherSpec directive in each virtual host stanza in the IBM HTTP Server configuration file. Multiple instances of the directive are possible to create a specific order. On a System z9, you might want to start the list with:

- ▶ SSLCipherSpec SSL_RSA_WITH_3DES_EDE_CBC_SHA
- ▶ SSLCipherSpec TLS_RSA_WITH_AES_128_CBC_SHA

Viewing the cipher specifications for secure transactions and for a specific HTTP request is possible by means of adapting the LogFormat directive to include the cipher specifications as part of the information logged for each request. For more information, see the IBM HTTP Server User's Guide.

6.4.7 In-kernel cryptography with Linux kernel 2.6

Linux kernel version 2.6 provides a set of modules which execute encryption functions by the kernel instead of the user space programs, software or libraries. These functions are built into the kernel as loadable modules, based on kernel compilation options and installed with the kernel. IBM provides modules for specific support of System z9 for in-kernel cryptography:

- ▶ des_s390
- ▶ sha1_s390
- ▶ sha256_s390
- ▶ aes_s390
- ▶ prng

These System z9 specific modules benefit from the CPACF support for SHA-1, SHA256, AES, DES, TDES, PRNG. It is evident, that the CPACF must be enabled through Crypto Enablement feature 3863 to support in-kernel cryptography modules (see 6.2, "Preparing System z for the hardware encryption support" on page 204).

Note: In-kernel cryptography with CPACF encryption support can be used, even without having Crypto Express2 feature installed or without having provided access to AP queues (through APVIRT or APDED in CRYPTO statement of z/VM Linux USER entry).

You can check, which cryptography algorithms are supported by the Linux system and for which algorithms IBM provides specific support. Check the content of the following directories:

/lib/modules/<kernelversion>/kernel/crypto

/lib/modules/<kernelversion>/kernel/arch/s390/crypto

Example 6-39 shows the in-kernel cryptography modules that were installed in our system.

Example 6-39 In-kernel modules in SUSE Linux SLES 10 SP1

```
lnxsu1:/ # ls /lib/modules/2.6.16.46-0.12-default/kernel/crypto/
aes.ko      cast5.ko      deflate.ko    michael_mic.ko sha512.ko    twofish.ko
anubis.ko   cast6.ko      des.ko        serpent.ko      tcrypt.ko    wp512.ko
arc4.ko     crc32c.ko     khazad.ko     sha1.ko         tea.ko
blowfish.ko crypto_null.ko md4.ko        sha256.ko       tgr192.ko

lnxsu1:/ # ls /lib/modules/2.6.16.46-0.12-default/kernel/arch/s390/crypto/
aes_s390.ko      des_check_key.ko  prng.ko      sha256_s390.ko
crypt_s390_query.ko  des_s390.ko      sha1_s390.ko
lnxsu1:/ #
```

To use the System z9 specific in-kernel cryptography modules, you need to configure your system to load the architectural dependent modules and to use them instead of the default modules. In SUSE Linux SLES 10 SP1, you need to add alias statements in `/etc/modprobe.conf.local` as shown in Example 6-40.

Example 6-40 Alias statements in `/etc/modprobe.conf.local`

alias	des	des_s390
alias	sha1	shs1d_s390
alias	sha256	sha256_s390
alias	aes	aes_s390

After adding the alias definitions, issue the following command to resolve dependencies and to update the definitions:

```
depmod -a
```

These in-kernel cryptography modules are loaded on request. If a cryptography request is issued after these definitions have been made, the System z specific modules will be loaded and used automatically. If there are already some of the general cryptography modules loaded while you have configured to use the System z specific modules, you can unload them by using the `rmmod` command. Especially the new setup will be used after shutdown and reboot of your Linux system.

In-kernel cryptography modules are used by the kernel itself. For example with

- ▶ IPSEC for secure communication
- ▶ Disk encryption, when using file system encryption with dm-crypt in combination with LUKS.

This provides a transparent access to encrypted data on the disk: dm-crypt is a device-mapper that provides transparent encryption of block devices and LUKS (Linux Unified Keys Setup) is used for the administration of appropriate keys.

Internal performance and throughput tests with IBM showed the benefit of CPACF also for using encrypted files systems with Linux: There is only a very small performance impact for the user, when reading or writing to a disk with encrypted file system in comparison with reading or writing of unencrypted data. The throughput to and from disk when using CPACF for encryption is much better than when the encryption is performed by software. There is only a relative small decrease when DES is used and a very small decrease when AES is used for reading and writing to and from the encrypted file system if CPACF is used compared with no encryption. Note, any kind of encryption is CPU intensive, that means here: Although

the throughput with encryption is nearly the same as with no encryption, the CPU utilization is higher.

Note: For in-kernel cryptographic functions the library libica is not used.

6.4.8 Using secure key encryption with Linux: an outlook

In 2007 support for secure key cryptographic for Linux for System z is now made available. Linux can benefit from the capability that are part of the Crypto Express2 feature. Secure key capability for Linux allows now also to use this platform to run applications where a high degree of security is required such as in banking and finance environment.

The solution for secure key consists of:

- ▶ Crypto Express2 configured as coprocessor (CEX2C)
- ▶ device driver (z90crypt)
- ▶ Common Cryptographic Architecture (CCA) libraries

CCA is a standard introduced by IBM and used across server platforms. In the newly provided CCA libraries for Linux for System z the full standard is implemented (see *Linux for System z, Secure Key Solution with the CCA*, SC33-8294). In addition, PKCS#11 and Java/JCE library support using of secure keys and using encryption with DES, TDES and AES as well as key generation.

Using secure key applications always leads to the question of management of cryptography keys and cryptography hardware. The cryptography cards must be configured with master keys for symmetric and for asymmetric functions. This can be done with different methods:

- ▶ Using z/OS ICSF
- ▶ Using a Trusted Key Entry (TKE) console with connection to a z/OS
- ▶ Using a TKE with connection to a new Linux CCA utility
- ▶ Using a new Linux CCA utility

Any of these methods leads to the same result. The decision, how to manage and maintain the keys depends on the clients security policies and the environment.

Figure 6-33 gives an overview of the involved libraries and APIs.

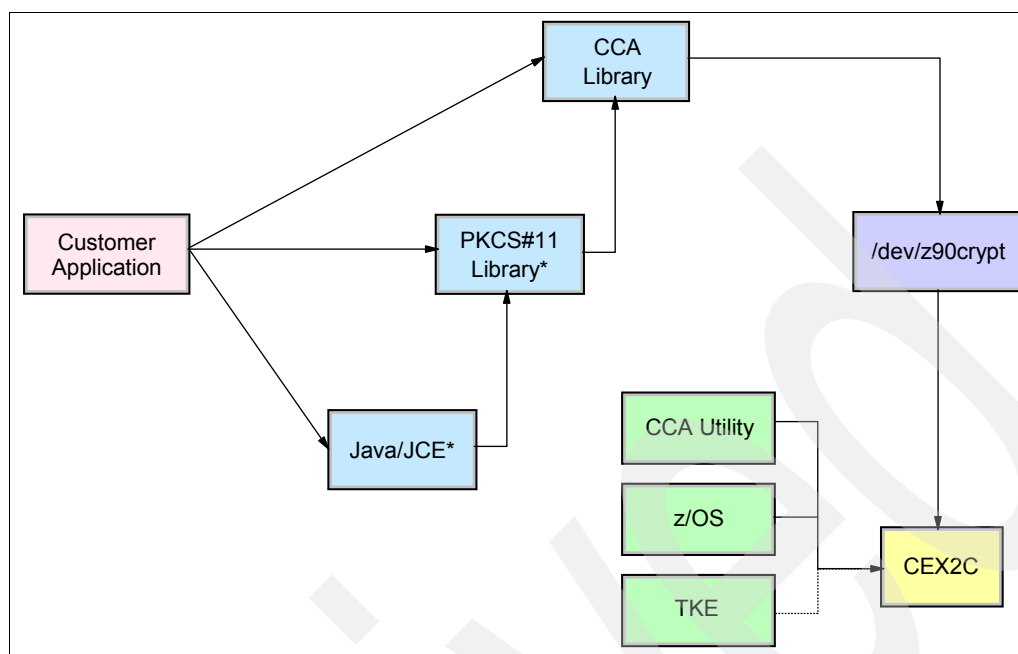


Figure 6-33 Overview of secure key cryptography in Linux

Installation

You need to have all software installed as described at the beginning of 6.4, “Using cryptography hardware support with Linux” on page 231. Ensure, that you have the CCA library that is available in the `xcryptolinzGA-3.28-rc08.s390x.rpm` package. This package is not included in SUSE Linux SLES 10 SP1, but it is available from the IBM Web site:

<http://www.ibm.com/security/cryptocards/pcixcc/ordersoftware.shtml>

There is a readme file that is available on the Web site. The readme file is also available after you install the package.

`/opt/IBM/4764/doc/README.linz`

In the `README.linz` file all relevant information about the package is contained. It contains installation notes as well as clear descriptions of syntax and usage of the delivery. The package contains:

- ▶ CCA libraries
- ▶ Installation verification program (`ivp.e`)
- ▶ TKE Catcher (TKEC) which responds to commands from a remote TKE workstation
- ▶ Panel CLI (`panel.exe`), a command-line utility that can be used to manage keys

CCA libraries

Note the CCA libraries are 64-bit only. That means, that in a 64-bit library, only 64-bit applications can use these libraries.

Installation verification

After the installation of the RPM file and after reading the `README.linz` file, you can use the installation verification program `ivp.e` for a quick verification of your environment for secure cryptography setup. The installation of the libraries as well as the availability of the cryptography cards is verified (installed, configured and device driver loaded). The `ivp.e` is located `/opt/IBM/4764/bin` directory.

Example 6-41 shows an unsuccessful verification, as we have no access to any cryptography hardware (CEX2C) in this case from this Linux system. This can be the case, if no APDED statement is provided in the User statement of the z/VM directory for this Linux, respectively the Linux has not yet been loggoff/logon since the directory entry was made.

Example 6-41 Installation verification: No cryptography queue available

```
lnxsu2:/opt/IBM/4764/bin # ./ivp.e
ivp - Installation Verification Program
RC=80400009 Status=0 errno=9 ThreadID=28f020
An error occurred trying to query the adapter.
CFQ return code = 8 (0x0008), reason code = 1100 (0x044C)
```

The ivp.e program also detects whether the cryptography device driver has not been loaded. Example 6-42 shows the response of the ivp.e when access to CEX2C is available, but the driver has not been loaded.

Example 6-42 Installation verification: z90crypt device not loaded

```
lnxsu2:/opt/IBM/4764/bin # ./ivp.e
ivp - Installation Verification Program
An error occurred trying to query the adapter.
CFQ return code = 12 (0x000C), reason code = 338 (0x0152)
```

When the cryptography hardware is available and the device driver is loaded successfully you get a positive status reported by ivp.e program. Example 6-43 shows that in the available cryptography hardware, there are up to now no keys stored.

Example 6-43 Installation verification: successful - no keys in card

```
lnxsu2:/opt/IBM/4764/bin # ./ivp.e
ivp - Installation Verification Program
Adapter query STATCCAE completed successfully.
  CCA version:      z3.25.00
  CCA build date:   20060511
  No symmetric masterkey is loaded!
  No asymmetric masterkey is loaded!
CFQ return code = 0 (0x0000), reason code = 0 (0x0000)
```

Be aware the status shown applies only to the AP with the lowest number, even if more than one is available. Whereas, Example 6-44 shows a situation where there are already some keys stored in the hardware.

Example 6-44 Installation verification: successful - keys already stored in the card

```
lnxsu2:/opt/IBM/4764/bin # ./ivp.e
ivp - Installation Verification Program
Adapter query STATCCAE completed successfully.
  CCA version:      z3.22.05
  CCA build date:   20061005
  Current symmetric masterkey register contains a key.
  Current asymmetric masterkey register contains a key.
CFQ return code = 0 (0x0000), reason code = 0 (0x0000)
```

The CLI panel

Within the CCA package there is a panel program, that provides administration functions of the active cryptography cards, like loading keys. To use this command line utility, the user must be member of the group `cca_admin`. This group is created during installation of the RPM file.

In Example 6-45, the various possibilities of this tool is shown (this tool provides similar administration possibilities as provided in z/OS TSO with ICSF panels).

Example 6-45 Administration functions provided by panel.exe

```
lnxsu2:/opt/IBM/4764/bin # ./panel.exe -?
Panel usage (-k,-a,-g,-x,-l,-s,-c,-q,-t,-o,-?):
    [CC] To determine if a TKE is allowed to administer a card:
        -k
    [CC] To specify a card other than the 0-th instance:
        (only useful combined/preceding other args)
        -a <card number>
    To list the current cards available (and basic status):
        -x

>>>Master Key (MK) Manipulation<<<

NOTE: -l,-s,-c,-q are mutually exclusive and must be
      (along with sub-options) the last option specified

To LOAD a Master Key (MK) PART:
    -l (for interactive)
OR====>
    -l -t [A|S] -p [F|M|L] KEYPART
    where: -t [A|S] is which MK: A=ASYM, S=SYM
    where: -p [F|M|L] is the part: F=FIRST, M=MIDDLE, L=LAST
    where: KEYPART is a 48 character hex string
           (2 text chars = 1 binary Byte)

To SET a Master Key:
    -s (for interactive)
OR====>
    -s -t [A|S]
    where: -t [A|S] is which MK: A=ASYM, S=SYM

To CLEAR a Master Key (clears prior key parts in 'New' Register):
    -c (for interactive)
OR====>
    -c -t [A|S]
    where: -t [A|S] is which MK: A=ASYM, S=SYM

To QUERY a Master Key Verification Pattern:
    -q (for interactive)
OR====>
    -q -t [A|S] -r [N|C|O]
    where: -t [A|S] is which MK: A=ASYM, S=SYM
    where: -r [N|C|O] is which register: N=NEW, C=CURRENT, O=OLD

To initialize a key storage file:
    -t <type> -f <file> -i
To reencipher key storage:
    -t <type> -f <file> -r
For version information:
    -v
```

[CC] To disable output to stdout:
(only useful combined/preceding other args)
-o

[CC] To set the log level:
(only useful combined/preceding other args)
-g <level>

For this usage information:
-? or -h

where:

[CC] by an arg description means it can be combined with other

args

NOTE: [CC] args MUST ALL precede non-[CC] args
<type> can be DES or PKA
<file> is the fully qualified name of a key storage file
<level> can be NONE, TRANSACTIONS, NONZERO, ALL, DEBUG, and

FUNCTIONS

This utility is mainly intended to be used in Linux only environments and where an access to a TKE workstation is not available. Note, using a TKE increases the administrative security and provides also a central place from where to administrate the Crypto infrastructure and the key management.

TKE catcher

The TKE catcher is a program running on Linux for System z that allows remote access from the TKE workstation to administrate the cryptography cards and the according keys.

To make use of the TKE catcher, the TKE workstation must be enabled to access the system through s390 SE panel and using the port 50003. The PCI-X adapter number must be included on the Control Domain Index (see Figure 6-5 on page 211) and TKE commands must be permitted for this adapter in the Cryptographic configuration (see Figure 6-15 on page 218). It is evident, that the cryptography AP queues are set dedicated (using APDED) to the Linux system.

There are three different cases to consider when using the TKE for Linux for System z:

- ▶ Environment with Linux for System z and z/OS LPARs sharing a Crypto Express2 card.
This is a difficult environment, if you intend to use the TKE catcher for administration of the Linux accessible cryptography queues and the z/OS TKE for the cryptography queues accessible for z/OS. The TKE catcher does not have the capability to figure out, whether there is a z/OS partition and whether the cryptography adapter is being configured through the z/OS TKE. In such an environment, we recommend to use the z/OS TKE to avoid conflicts.
- ▶ Environment with Linux for System z and z/OS LPARs with each exclusive use of Crypto Express2 cards
In a mixed environment, where no card sharing occurs, you can use the TKE Catcher for administration of the Linux cryptography environment. Note, the situation gets difficult if the environment is reconfigured to sharing the Crypto Express2 cards.
- ▶ Linux for System z exclusive environment.
In a Linux only environment using the TKE with TKE catcher is the most secure way to administrate the Crypto infrastructure.

In any of these cases, you have to do careful planning up front which LPAR is to be administrated with the TKE. This is then reflected in the configuration of the Image activation profile in the cryptography control domains. Control Domain Index and TKE commands must be permitted for the used cryptography adapters in the Cryptographic configuration (See 6.2.2, “Customize the partition image profile for cryptographic usage” on page 209). For information about setting up and using the TKE, see the TKE documentation.

Considerations

Be aware that using secure key with Crypto Express2 means, that the keys are stored inside the coprocessor card, which is tamper proofed. A loss of the card for whatever reason leads automatically to the loss of the keys stored in this card. If the failing card is the only *storage* of your keys, then you have automatically a loss of all your existing data which has been encrypted by these keys. Using only one cryptography card that is the only storage for your keys is a single point of failure. It is a good practice to have at least two APs in one domain dedicated to one Linux system. The two APs should reside on different Crypto Express2 features to minimize the risk. Having more than one AP helps also for load balancing if the cryptography application allows to use more than one AP. In addition, you might consider how to backup your keys for other disaster situations. For example, if you enter any key manually into the Crypto Express2 card, you might want to document the key on paper or another movable storage device and deposit it in your enterprise's safe for disaster recovery.

Linux for System z now provides a complete infrastructure to use secure key cryptography according the CCA and to run 64-bit secure key applications in addition to the established clear key encryption capabilities. With this extension, Linux for System z is a highly flexible environment to run additional business critical applications with a high demand to security, such as requested by banking and finance industries. New generations of applications, implementing secure technologies like PIN verification, single sign-on and service oriented architecture can benefit from the cryptographic support of Linux for System z and protect the enterprise from end to end.

Archived

IBM Tivoli zSecure for z/VM RACF

This chapter presents IBM Tivoli zSecure for z/VM RACF. It describes the benefits of using the product and how it can assist you in providing effective security for z/VM. We guide you through the installation and configuration process of the product.

Additionally, this chapter shows the benefits of using Tivoli zSecure in your z/VM environment. This product provides security managers, system programmers, RACF administrators, security auditors, and help desk managers with a set of easy to use utilities for daily RACF and security procedures. It can also be used to prepare for audits and to verify compliance with government and corporate security requirements.

7.1 Consul InSight Suite benefits

Consul Risk Management International based in Delft, Netherlands, with an office in Herndon, Virginia, is a recently acquired IBM company (January 2007) with over 20 years of experience in the security environment. The Consul InSight Suite is the flagship enterprise wide solution for security monitoring. This product provides automated security auditing, activity monitoring, log management, and audit reporting for your corporate security team. The Consul zSecure Suite is the mainframe product that provides for easier and more effective management and analysis of your mainframe.

Consul zSecure Pro Suite was designed to enhance security controls by the creation of audit reports, alert reports, and enhancing administrative reports that allow the RACF administrator to have better control of the complex system environment. The RACF administrator can use a series of preformatted or customized reports that show unauthorized access to system resources (data base access, for example) as well as reports that help manage user resources by providing detailed reports for userclass, groups, audit levels, and many others. The Consul zSecure suite of products allows a RACF administrator to generate reports that are concise and easy to interrupt.

The utilities supplied with this product are executed using a series of comprehensive panels with ISPF or with supplied REXX execs. You can find detailed information about the use of the utilities, commands, report generation, and how to customize reports in *IBM Tivoli zSecure Manager for z/VM RACF: Installation and Configuration Manual (SC23-6574)*.

7.2 Tivoli zSecure Pro Suite

According to recent, statistics, approximately 70% of all critical corporate data resides in the mainframe environment. This environment provides security administrators and auditors with the large and challenging task of protecting corporate data. Consul zSecure Pro Suite is a suite of utilities for the System z environment to aid in the protection of mainframe data. This set of tools provides interfaces to assist you in detecting threat management to your z/OS or z/VM system from external as well as internal attacks. These utilities were developed to allow security managers, system auditors, RACF administrators, system programmers, and help desk managers to determine whether corporate security policies are being met.

By generating predefined or customized reports with these tools it is easy to:

- ▶ Determine if your system has been a target for unauthorized access
- ▶ Check whether users have performed unauthorized functions
- ▶ Improve operational security
- ▶ Provide audit records in a concise and easy to digest manner
- ▶ Meet corporate or government regulatory audits and compliance guidelines
- ▶ Verify change management processes

Some new features of this product include:

- ▶ Enhanced XML report generation for use with Web browsers and spreadsheet utilities
- ▶ Multiple profile comparisons for adjustments of authorities
- ▶ Granular access to SMF record data
 - TCP/IP network
 - DB2 audit records
 - Tape integrity

- Enhanced controls to prevent changes to public resources access list
- Improved Tivoli Directory Integrator LDAP support
- New RACF administration and audit functionality

Figure 7-1 illustrates the manner in which this product accepts input and creates output.

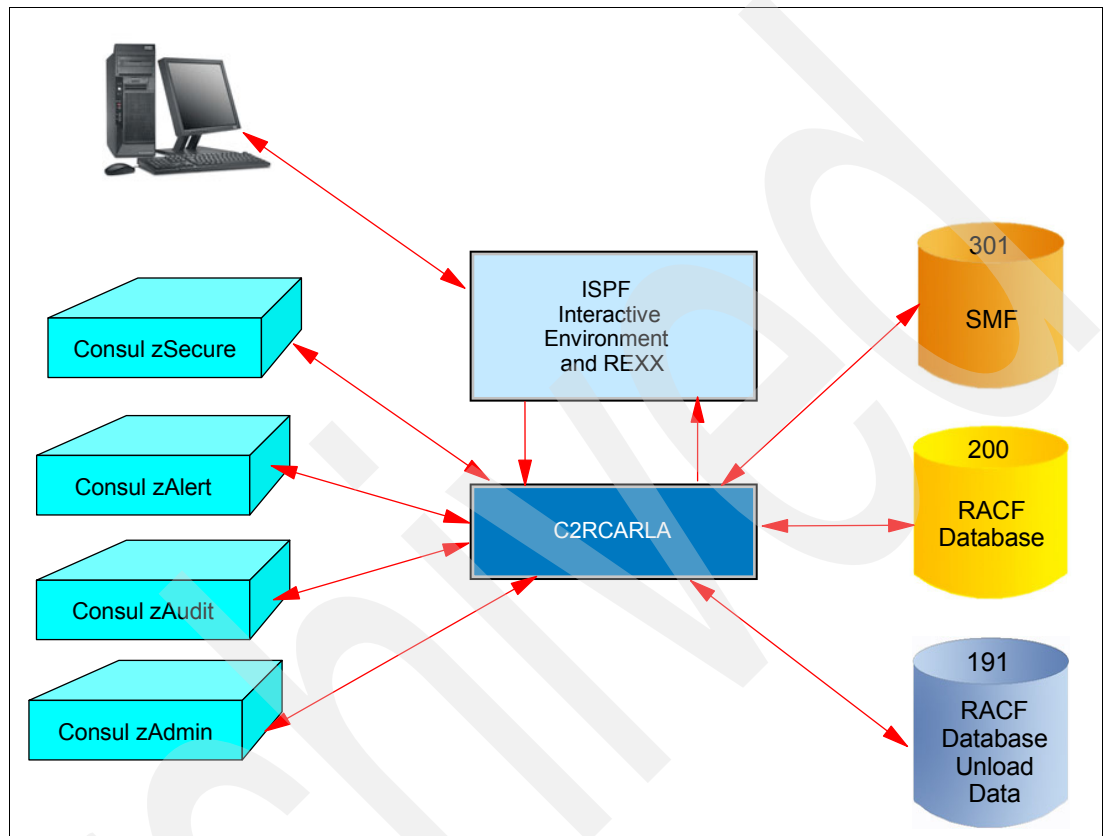


Figure 7-1 Consul zSecure Pro Suite input and output

Using ISPF is an optional feature, but it improves the usability of the product. If you do not use ISPF, then reports are created through the execution of REXX execs. Consul zSecure Pro Suite is command driven with the Consul Auditing and Reporting Language (CARLa). You can use the live or backup database disks (200 or 300), data created with the new RACF Database Unload Utility, IOCONFIG data, SMF records (301 or 302 disk), or XML/HTTP information as input for the generation of reports.

Batch use of Consul zSecure Pro Suite is attractive as part of a security monitoring function. For example, daily, weekly, monthly, or quarterly a selected set of Consul zSecure Pro Suite checks and reports might be run automatically using a tool such as Performance Toolkit for z/VM to scheduled batch generation of reports.

7.3 Introducing Tivoli zSecure

Tivoli zSecure for RACF enhances mainframe security and provides better insight into what is happening on the mainframe, mainly by increasing the functionality of security system and by reducing processing time. The clear structure and the possibility to make a more granular division between administration authority levels enable you to divert the workload without losing security. Consul zSecure RACF uses RACF information and functionality.

The product has the following major components:

- ▶ CARLa (by using CARLa - the Consul Auditing and Reporting Language): Used to create customized report
- ▶ Consul zAudit: Collects the data from three main sources and generates reports
- ▶ Consul zAdmin: Works together with zAudit to help with administrative task
- ▶ Consul zAlert: Works together with zAlert to generate security alerts based upon your security threshold defined.
- ▶ Consul zCollect: Is part of zAudit and is used to collect system information needed for alert generation

Note: SPF for z/VM is a desirable prerequisite, but it is not mandatory. If your installation does not have ISPF, you can still install and execute Consul zSecure Pro Suite using the REXX execs that are provided with the product.

Also, you will see some references to ISPF Panel and Batch execution. For ISPF panel you can consider we are referring to REXX execs. For batch execution you should consider building your own timely fashion exec.

7.3.1 CARLa

You do not have to worry too much about this programming language interface. All programming is done using a series of execs or ISPF panels provided with the product.

The commands are explained, in considerable detail, in the *Consul zAudit Reference Guide*. A normal Consul zAudit user, working through REXX exec, is not concerned with CARLa. The CARLa code is generated automatically by the execs, and sent to the appropriate application program.

The command language is generally used for two reasons:

- ▶ To generate customized reports
- ▶ Call from the Tivoli zAudit execs

7.3.2 Tivoli zAudit

Consul zAudit is a component of Consul zSecure Pro Suite, it provides the security administrator the flexibility to track all data or any subset of the data accessed at the virtual machine level. These reports can then easily be customized to show only the required data for a security audit. This tool allows you to take live SMF data (301 and 302 disk), or the output from the IFASMFDP utility (this is a z/OS utility not a z/VM utility) as input to this function. Therefore, you would have to use live SMF data or a backup copy of your SMF data. The primary processing programs are large modules that can be used in batch or interactive mode. Interactive mode is most common, Consul zAudit has a full and sophisticated interactive interface, implemented completely with various REXX execs supplied with Consul zAudit.

The standard Consul zAudit reports are comprehensive and thus customized reports might never be required. Nevertheless, the functions for producing customized reports are included with the Consul zAudit feature. There is a comprehensive set of sample reports available in a data set referred to as SCKRSAMP library.

Figure 7-2 illustrates the general flow.

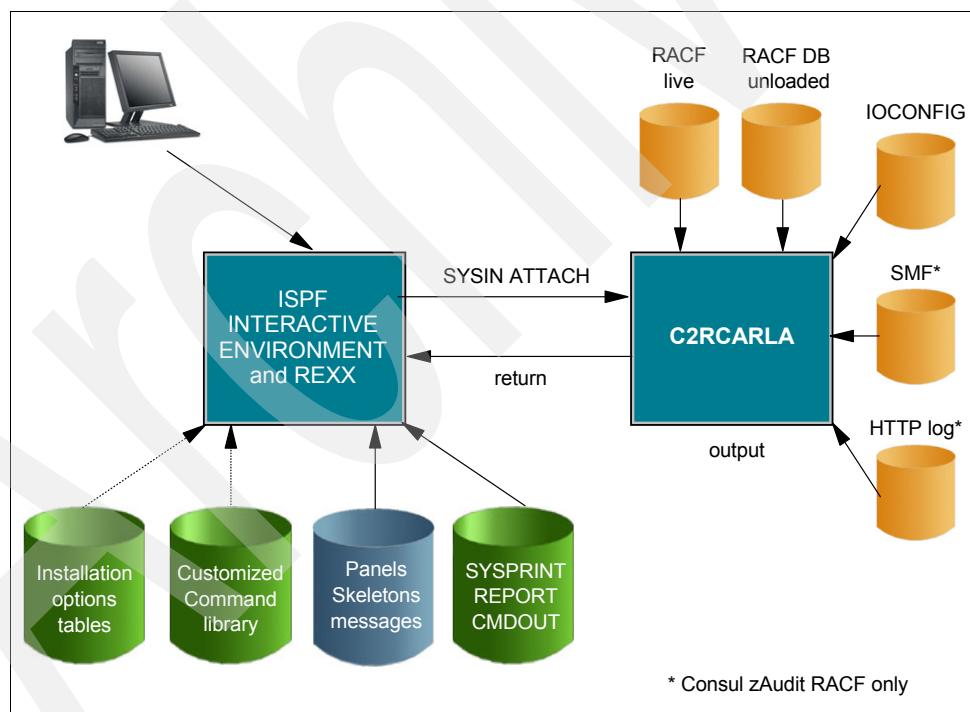


Figure 7-2 Conceptual data flow

The general design of the product, with separate interactive and non-interactive components, has a number of practical advantages:

- ▶ It separates interactive interfaces from the application program. This allows more flexibility in designing and using the interfaces and programs.
- ▶ Any functions that can be run interactively can also be run in batch mode.
- ▶ An installation can create customized reports (by using CARLa) and run these reports from the REXX exec.
- ▶ Consul zAudit can be used remotely, in cases where a TSO or VM connection is not possible or practical; for example, in NJE networks.

Consul zAudit is command driven by means of CARLa. The commands are explained, in considerable detail, in the reference sections in the *zAudit Reference Guide*. A normal Consul zAudit virtual machine, working through ISPF panels, is not concerned with CARLa. CARLa code is generated automatically by the panels, and sent to the application program.

Data Source for zAudit

Consul zAudit uses several different types of data. A quick overview of the data, and sources of the data, will help you understand Consul zAudit. First of all it is good to know that Consul zAudit can use (Figure 7-3):

- ▶ Data from your security system
- ▶ Data describing your system configuration (control blocks and DASD)
- ▶ Data describing events on your system (SMF)

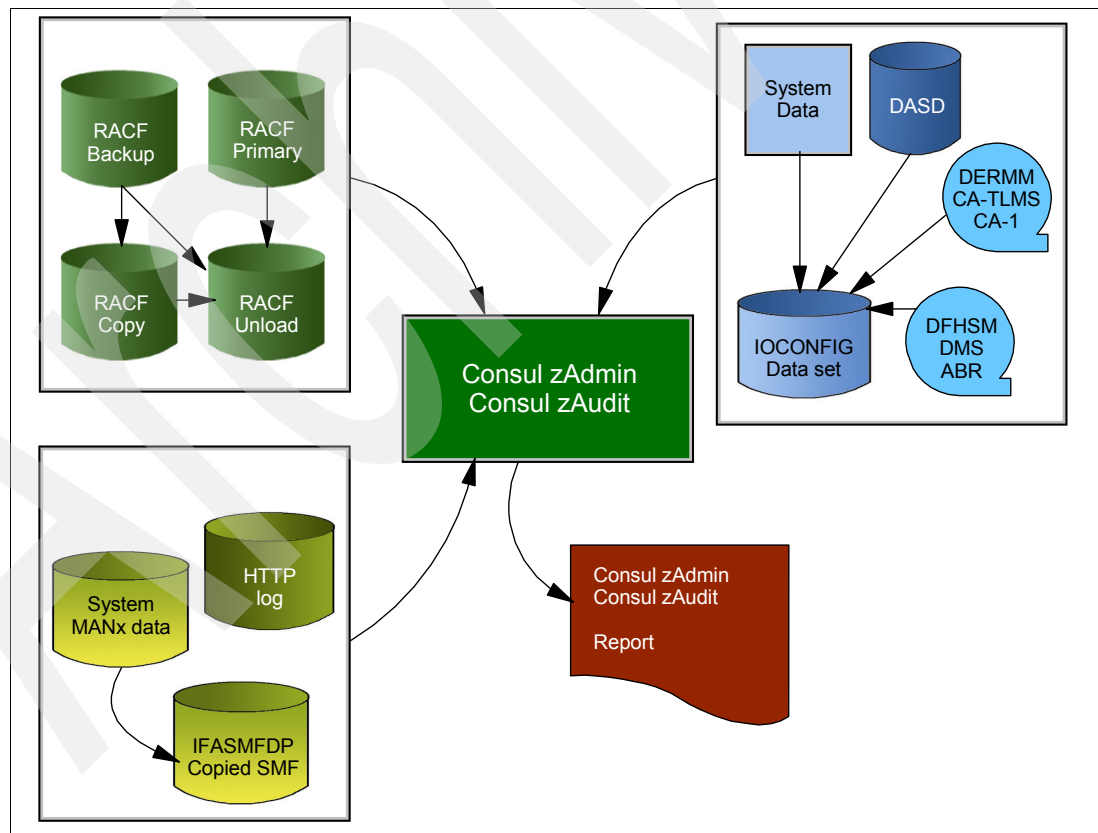


Figure 7-3 Consul zAudit Input Sources

You will define input sets for Consul zAudit by using “SETUP - Input files. For example, one set might consist only of the live security database. Another set can use the live security

database combined with a file containing system configuration data. Another set might use the RACF Unloaded Database Utility data, system configuration data, and SMF record files.

Data from your security system

Consul zAudit usually requires RACF data. It can come from one of the following sources:

- ▶ The primary live RACF database
- ▶ The backup live RACF database
- ▶ Unloaded RACF data
- ▶ A copy of a RACF database
- ▶ An active RACF database from another system.

Consul zAudit can produce unloaded RACF data by reading the live RACF database and creating a copy in a proprietary format suitable for high-speed searches, we discuss this process in 7.5.2, “Configuring the parm file” on page 288.

Note: This process does not use the IBM-provided RACF database unload program. It is much faster and uses much less space.

7.3.3 Tivoli zAlert

Consul zAlert is another component of Consul zSecure Pro Suite. This feature heightens your mainframe security and gives you a better perception in what is happening on your mainframe. It is integrated with Consul zAudit as the real-time security monitor that uses RACF Secure Server SMF records to issue alerts of important events regarding system security at the time of occurrence. This utility can aid the help desk management to receive immediate notification when unauthorized access of data is being attempted.

The processing of SMF records with the tools that ships with the RACF Security Server have a major disadvantage, this deals with the amount of time between when the SMF record is cut and when the traditional utility is run to generate the report. This might be acceptable in certain instances, but for the purpose of trying to detect potential unauthorized data access in your environment this is undesirable. This is where Consul zAlert can be most beneficial to your system. Armed with the immediate knowledge of a system security breach the help desk could take the appropriate action to disable the access. This action could be one of the following actions;

- ▶ Revoke and force the virtual machine
- ▶ Disable the network connection
- ▶ Deactivate the application

7.3.4 Tivoli zAdmin

The Tivoli zAdmin feature provides a user-friendly interface using ISPF panels on top of RACF and extends the functionality. zAdmin allows you to enter and process administrative commands quicker, generate custom reports and thoroughly clean-up databases. Additionally it provides administrative authority in a more granular fashion so people receive the specific amount of administrative authority they need to perform their job.

Tivoli zAdmin RACF provide a complete and mature interactive interface, implemented completely in ISPF functions. ISPF when installed on the z/VM system is called during most of an interactive session. ISPF uses panel, skeleton, and message libraries supplied with Tivoli zAdmin. The ISPF panels call the Tivoli zAdmin as needed.

The following functions can be performed by REXX exec under z/VM without ISPF:

- ▶ Maintain group profiles
- ▶ Maintain user profiles
- ▶ Query group structures
- ▶ Query users

7.4 Tivoli zSecure installation

The new version of IBM Tivoli zSecure Manager for z/VM RACF is fully SES enabled and is installable as any z/VM Licensed Program.

Before installing Tivoli zSecure, check with the IBM Support Center or use IBMLink™ (Service Link) to see whether there is additional Preventive Service Planning (PSP) information because no APARS have been incorporated into installation tapes.

IBM Tivoli zSecure requires ISPF Version 3.2, but if you want to add some capabilities such as ISPF browse and edit, you should have ISPF/PDF installed.

You no longer install and service Tivoli zSecure Manager for z/VM RACF strictly using the MAINT user ID but use a new user ID, 5655T13A. This is the IBM suggested user ID name. You are free to change this to any user ID name you want; however, a PPF override must be created.

Note: We do not recommend that you change this user ID but if you want to change, it can be easier to make the above PPF override change during the installation procedure according to item 6.2, “Plan Your Installation For Tivoli zSecure Manager for z/VM RACF” step 6 on page 17 in “Program Directory for IBM Tivoli zSecure for z/VM RACF” (GI11-7865-00), rather than after you have installed this product.

Follow these steps to create the virtual machine 5655T13A and install the product

1. All information necessary to create the this user ID is contained in the tape. Execute the command in Figure 7-4 to create the VMFINS PRODLIST.

```
vmfins install info (nomemo  
  
VMFINS2760I VMFINS processing started  
VMFINS1909I VMFINS PRODLIST created on your A-disk  
VMFINS2760I VMFINS processing completed successfully  
Ready;
```

Figure 7-4 VMFINS Install Info command

2. Now get the resource planning information as shown in Figure 7-5.

```
vmfins install ppf 5655T13A ZSECURE (plan nomemo
```

```
VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2601R Do you want to create an override for :PPF 5655T13A ZSECURE
:PRODID 5655T13A%ZSECURE?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINS2603I Processing product :PPF 5655T13A ZSECURE :PRODID
5655T13A%ZSECURE
VMFREQ1909I 5655T13A PLANINFO created on your A-disk
VMFREQ2805I Product :PPF 5655T13A ZSECURE :PRODID 5655T13A%ZSECURE
has passed requisite checking
VMFINT2603I Planning for the installation of product :PPF 5655T13A ZSECURE
:PRODID 5655T13A%ZSECURE
VMFRMT2760I VMFRMT processing started
VMFRMT2760I VMFRMT processing completed successfully
VMFINS2760I VMFINS processing completed successfully
```

Figure 7-5 VMFINS Install PPF

3. Obtain the user directory from 5655T13A PLANINFO file created in Maints A disk and insert the directory entry and minidisks. Your user directory should look as shown in Figure 7-6.

```
USER 5655T13A xxxxxxxxx 32M 100M G
IPL CMS
MACH XA
CONSOLE 0022 3215 T MAINT
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019E 019E RR
LINK MAINT 019D 019D RR
LINK MAINT 5E5 5E5 MR
LINK MAINT 51D 51D MR
MDISK 0191 3390 211 30 LX5U1R MR
MDISK 02B2 3390 241 85 LX5U1R MR
MDISK 02C2 3390 326 2 LX5U1R MR
MDISK 02D2 3390 328 40 LX5U1R MR
MDISK 02A6 3390 368 10 LX5U1R MR
MDISK 02A2 3390 378 10 LX5U1R MR
MDISK 100 3390 388 85 LX5U1R MR
MDISK 200 3390 473 85 LX5U1R MR
```

Figure 7-6 Directory example for ZSECURE user

4. After creating the 5655T13A virtual machine, logon and have a tape drive attached to your virtual machine. Mount the Tivoli zSecure Product tape.
5. Create a profile exec to accomplish the following tasks:
 - a. Access your Maint 5E5 (VM/SES code) mini disk as B (R/W).
 - b. Access your Maint 51D (VM/SES SW Inventory) mini disk as D (R/W).
 - c. Create profile exec and execute as shown in Figure 7-7.

```
xedit profile exec a
==> input /**/
==> input 'access 5e5 b'
==> input 'access 51d d'
==> file
profile
```

Figure 7-7 Profile exec

6. Install the product as shown in Figure 7-8 and Figure 7-9. (The installation screen was divided in two parts to make easier for reading.)

```
vmfins install ppf 5655T13A ZSECURE (nomemo nolinek

VMFINS2767I Reading VMFINS DEFAULTS B for additional options
VMFINS2760I VMFINS processing started
VMFINS2601R Do you want to create an override for :PPF 5655T13A ZSECURE
:PRODID 5655T13A%ZSECURE?
Enter 0 (No), 1 (Yes) or 2 (Exit)
0
VMFINS2603I Processing product :PPF 5655T13A ZSECURE :PRODID
5655T13A%ZSECURE
VMFREQ2805I Product :PPF 5655T13A ZSECURE :PRODID 5655T13A%ZSECURE
has passed requisite checking
VMFINT2603I Installing product :PPF 5655T13A ZSECURE :PRODID
5655T13A%ZSECURE
VMFSET2760I VMFSETUP processing started for 5655T13A ZSECURE
VMFUTL2205I Minidisk|Directory Assignments:
String Mode Stat Vdev Label/Directory
ST:VMFUTL2205I LOCALSAM E R/W 2C2 LOCALSAM
ST:VMFUTL2205I APPLY F R/W 2A6 APPLY1
ST:VMFUTL2205I G R/W 2A2 APPLY
ST:VMFUTL2205I DELTA H R/W 2D2 DELTA
ST:VMFUTL2205I BUILD0 I R/W 300 BUILD0
ST:VMFUTL2205I BASE J R/W 2B2 BASE
ST:VMFUTL2205I ----- A R/W 3BA NEI3BA
ST:VMFUTL2205I ----- B R/O 5E5 MNT5E5
ST:VMFUTL2205I ----- D R/W 51D SESINV
ST:VMFUTL2205I ----- S R/O 190 CMS22
ST:VMFUTL2205I ----- Y/S R/O 19E YDISK
```

Figure 7-8 Installation Part 1

```

ST:VMFSET2760I VMFSETUP processing completed successfully
ST:VMFREC2760I VMFREC processing started
ST:VMFREC1852I Volume 1 of 1 of INS ENVELOPE 0600
ST:VMFREC1851I (1 of 7) VMFRCAXL processing AXLIST
ST:VMFRCX2159I Loading 0 part(s) to DELTA 2D2 (H)
ST:VMFREC1851I (2 of 7) VMFRCPTF processing PARTLIST
ST:VMFRCP2159I Loading 0 part(s) to DELTA 2D2 (H)
ST:VMFREC1851I (3 of 7) VMFRCCOM processing DELTA
ST:VMFRCC2159I Loading 1 part(s) to DELTA 2D2 (H)
ST:VMFREC1851I (4 of 7) VMFRCALL processing APPLY
ST:VMFRCA2159I Loading part(s) to APPLY 2A6 (F)
ST:VMFRCA2159I Loaded 1 part(s) to APPLY 2A6 (F)
ST:VMFREC1851I (5 of 7) VMFRCALL processing BASE
ST:VMFRCA2159I Loading part(s) to BASE 2B2 (J)
ST:VMFRCA2159I Loaded 5782 part(s) to BASE 2B2 (J)
ST:VMFREC1851I (6 of 7) VMFRCALL processing BUILD
ST:VMFRCA2159I Loading part(s) to BUILD0 100 (I)
ST:VMFRCA2159I Loaded 5505 part(s) to BUILD0 100 (I)
ST:VMFREC1851I (7 of 7) VMFRCALL processing SYSSAMP
ST:VMFREC2760I VMFREC processing completed successfully
ST:VMFINT2603I Product installed
VMFINS2760I VMFINS processing completed successfully

```

Figure 7-9 Installation Part 2

7. After you finish population product disk, you have to update software inventory. Just execute the command shown in Figure 7-10.

```
vmfins build ppf 5655T13A {ZSECURE | ZSECURESFS} (serviced nolink
```

Figure 7-10 Update Software Inventory and build missing parts

7.4.1 Creating the load library

You need to create a load library containing a link-edited version of zSecure (CKRCARLA). There are two versions of the load libraries shipped on the tape. One is the ISPF version and the other is the Non-ISPF version. The ISPF version of the loadlib will work in both environments so we suggest that you chose that version, but if do not have ISPF then use the Non-ISPF version.

To create the CKRCARLA loadlib, you must perform the following steps:

1. Clean up your virtual machine environment to remove any un-necessary links and access of disk that might cause a problem. The best solution would be to logoff, then log on.
2. Make sure that there are no other program products installed on the Consul zSecure minidisk. We found it unlikely that you would have done something like this but we feel the product developer was being cautious.
3. Verify that you have linked and accessed the Consul zSecure target minidisk in R/W mode. We had installed the code on ZSECURE's 192 disk.

4. Use VMFBUILD to build CKRCARLA Loadlib as shown in Figure 7-11.

```
vmfsetup 5655T13A {ZSECURE | ZSECURESFS}  
vmfbld ppf 5655T13A {ZSECURE | ZSECURESFS} CKVBLCLL (ALL
```

Figure 7-11 Build CKRCARLA loadlib

7.4.2 Place Tivoli zSecure in production

Next, you update VMSES Partcat files on 200 disk (Figure 7-12).

```
access 100 e  
access 200 f  
vmfcopy * * e = f2 (prodid 5655T13A%ZSECURE olddate replace
```

Figure 7-12 Placing in production

Tivoli zSecure is now ready to use. Log off 5655T13A user. You next need to apply maintenance. If you do not have any maintenance to apply, move ahead to the next section (7.4.4, “Installation verification” on page 283).

7.4.3 Applying maintenance

Applying maintenance is a VMSES procedure. You have to follow the steps to set up the environment, merge libraries, receive and apply service, build new level, and then put zSecure in production.

Follow these steps:

1. Log on to 5655T13A user ID, attach and mount service tape or link to the work disk where the envelope files resides. In Figure 7-13, we used the envelope files method.

```
vmfrec info (env 5551829
```

Figure 7-13 Receiving envelop file

2. Set up the VMSES environment, as shown in Figure 7-14.

```
vmfsetup 5655T13A ZSECURE
```

Figure 7-14 Environment setup

3. Merge production files to back up DASD (Figure 7-15).

```
vmfmrdsk 5655T13A ZSECURE apply
```

Figure 7-15 Merging disks

4. If you are receiving more than one envelope file, you can receive all them at once. Figure 7-16 shows one envelope file.

```
vmfrec ppf 5655T13A {ZSECURE | ZSECURESFS} (env 5551829
```

Figure 7-16 Receiving envelope file

5. Apply the service (Figure 7-17).

```
vmfapply ppf 5655T13A ZSECURE
```

Figure 7-17 Apply service

6. Update Build Status Table, as shown in Figure 7-18.

```
vmfbld ppf 5655T13A ZSECURE (status
```

Figure 7-18

7. Build serviced objects (Figure 7-19).

```
vmfbld ppf 5655T13A ZSECURE (serviced
```

Figure 7-19 Build objects

8. Finally, put in production (Figure 7-20).

```
access 100 e  
access 200 f  
vmfcopy * * e = = f2 (prodid 5655T13A%ZSECURE olddate replace
```

Figure 7-20 Put in production

Figure 7-21 shows the message that you receive when you complete the installation successfully.

```
You have finished servicing Tivoli zSecure Manager for z/VM RACF
```

Figure 7-21 Finished installation

7.4.4 Installation verification

This section explains how to check the installation.

Checking base UI functions

Under the ISPF/PDF command option, invoke the REXX that you created in 7.5.1, “Making the software (CKREVM EXEC) available to VM/CMS users” on page 284. This check should result in a display of the Tivoli zSecure Manager for z/VM RACF primary menu, as shown in the User Reference Manual. Note that the primary menu can deviate from the one shown in the Introduction.

Checking for zCollect function

To run IBM Tivoli zSecure Collect, you need to be logged on under a user ID that has privilege classes B, E, and G and that has at least read access to the minidisk where the USER* DIRECT file resides.

From the Tivoli zSecure Manager for z/VM RACF primary menu, enter SE.1 (Setup files) and deselect all selected input sets. Next, enter SE.2 (Setup - New files) to create new CKFREEZE and UNLOAD files. You will be prompted for allocation parameters. As a raw approximation, specify:

- ▶ For CKFREEZE: 2MB per online DASD-volume
- ▶ For UNLOAD: same size as your security database

Use the REFRESH command to fill these files.

7.5 Configuring Consul zSecure

In this section, we show the process to perform basic configuration of Consul zSecure. Upon completion, you will be able to create Lists, Shows, Displays, Summaries, and Reports using the ISPF panels or REXX exec. The configuration that we show here is a basic default configuration that allows any user to execute commands, assuming that the user has correct RACF permission and z/VM class of commands to allow its execution.

Note: For complete list of commands, refer to *IBM Tivoli zSecure Manager for z/VM RACF: Installation and Configuration Manual* (SC23-6574).

7.5.1 Making the software (CKREVM EXEC) available to VM/CMS users

If you have a previous version of Consul zSecure on your z/VM system, then you already have a customized copy of the CKREVM EXEC on a minidisk that is accessible globally to your CMS user community. You should replace your old CKREVM EXEC with the one shipped with version 1.8 (CKREVM) and modify the new file with your previous information.

We found it best to place this file on MAINT's 190 or 19E disk. If you place it on one of these two disk make sure that you copy this file as a filemode 2 file. If you do not, then the file would not be accessible to your CMS user community. By placing this file on the 190 or 19E disk any virtual machine that IPLs the CMS saved system will have accessibility to this exec.

Note: Any user can use CKREVM, but only those users with correct RACF permission and z/VM class of commands will be able to execute commands, run reports, and so forth.

CKREVM EXEC

Modify this file and update the parameters listed in Figure 7-22.

```
codeMiniDisk = ' 5655T13A 200'  
codeVirtualAddress = ' ACF '  
codeFileMode = ' E '  
configurationFile = ' CKR$PARV CKRPARM * '
```

Figure 7-22 CKREVM EXEC parameters

Here is a brief description of the variables in the CKREVM EXEC:

► **codeMiniDisk**

Identifies the VM user ID who owns the minidisk where the installed software resides, and the device number for that minidisk for that user. CodeMiniDisk should correspond to the VM Directory. Because we defined the Consul zSecure Virtual Machine as 5655T13A and minidisk 200 to receive the installation files, we changed this parameter to reflect that.

► **codeVirtualAddress**

Is the virtual address that is used in the LINK command as the locally known minidisk where the software resides. You should code a virtual address that is not used by any other applications or virtual machines that would run the CKREVM EXEC.

► **codeFileMode**

This parameter defines the filemode used by the access command when running this exec. In our test system we used filemode “E”

► **configurationFile**

The file (file name file type filemode) that is to be used to configure zSecure. See Section 7.5.2, “Configuring the parm file” on page 288 to see how to configure this file

The CKREVM EXEC based upon Figure 7-22 defines the configuration file that is used by the product. Issue the following commands prior to starting the application:

```
LINK 5655T13A 192  ACF  RR
ACCESS  ACF  E
```

After the modifications, our CKREVM EXEC looked like Figure 7-23.

```

CKREVM  EXEC D2  F 80  Trunc=80 Size=29 Line=0 Col=1 Alt=0
====>
0 * * * Top of File * * *
1 /**REXX*****BeginModule*****
2 * Copyright:
3 * (C) 2004-2006 Consul Risk Management B.V. All Rights Reserved
4 * File-stamp: <041209 MR 10:59:38 CKREVM.SCKRCLIB>
5 * FMID: RC2R180 RMID: RC2R180 Consul zSecure 1.8.0
6 *
7 * Purpose: Start user interface
8 *
9 * History:
10 * 040825 1.6.0 MR  RC2R160: Invoke CKREMAIV
11 *****EndModule*****/
12 arg fname . '(' parms ')'
13
14 configurationFile      = 'C2R$PARV CKRPARM *'
15 codeMiniDisk           = '5655T13A 200'
16 codeVirtualAddress     = 'ACF'
17 codeFileMode           = 'E'
18
19 'CP LINK' codeMiniDisk codeVirtualAddress 'RR'
20 'SET CMSTYPE HT'
21 'ACCESS' codeVirtualAddress codeFileMode
22 'SET CMSTYPE RT'
23
24 if index(parms,'CONFIG')=0 then
25   parms=parms 'CONFIG=' configurationFile
26   parms=parms 'CODEFILEMODE=' codeFileMode
27
28 call CKREMAIV fname '(' parms
29 'DETACH' codeVirtualAddress
30 'SET CMSTYPE HT'
31 'ACCESS' codeVirtualAddress codeFileMode
32 'SET CMSTYPE RT'
33
34 if index(parms,'CONFIG')=0 then
35   parms=parms 'CONFIG=' configurationFile
36   parms=parms 'CODEFILEMODE=' codeFileMode
37
38 call CKREMAIV fname '(' parms
39 'DETACH' codeVirtualAddress
40 * * * End of File * * *

```

Figure 7-23 CKRVM EXEC

Notice that we created the file on the E disk. Here is the process that you can use to move a copy of this exec and then save the CMS saved system:

1. Log on into MAINT.
2. Access 19E K.
3. Access 49E L.
4. Link ZSECURE 192 992 rr.
5. Access 992 M.
6. Copy CKREVM EXEC M1 CKREVM EXEC K2.
7. Copy CKREVM EXEC M1 CKREVM EXEC L2.
8. Access 193 N.
9. SAMPNSS CMS (See Figure 7-24).
10. IPL 190 CLEAR PARM SAVESYS CMS (See Figure 7-25).

In this process, we performed these tasks from the virtual machine MAINT because MAINT is the owner of the 19E and 49E disk and, by default, has write access to these disks.

In step 6 and 7, we copied the CKREVM EXEC to both the 19E and 49E disk as filemode 2 files because of the way that VMSES/E performs service. When service for z/VM is applied (RSU or COR service), the VMFBLD process creates the new files to the BUILD0 disk. (BUILD0 disk is the target of several buildlist for the components of z/VM. See *VMSES/E Introduction and Reference Guide*, GC24-6130, for additional information.) When the PUT2PROD EXEC is run, the 49E disk is copied to the 19E disk. If we had not placed this file on both disks, then we would have lost our copy of the CKREVM EXEC for our CMS user community.

The SAMPNSS EXEC is an IBM-provided REXX exec that issues the CP DEFSYS command to define a *skeleton system data file* (system data files are special spool files). See Figure 7-24. You must issue this command before you can re-save a saved segment or saved system. This exec issues the DEFSYS command with all the required parameters for the CMS saved system. It was created in the early releases of VM/ESA to make the system programmers job easier.

```
sampnss cms
HCPNSD440I The Named Saved System (NSS) CMS was successfully defined in fileid
0043.
Ready; T=0.01/0.01 14:18:49
```

Figure 7-24 Result of SAMPNSS run

The final step is to reload the code from the 190 disk into the skeleton system data file with the IPL command using the SAVESYS parameter, as shown in Figure 7-25.

```
ipl 190 parm savesys cms
HCPNSS440I Named Saved System (NSS) CMS was successfully saved in fileid 0043.
z/VM V5.3.0 2007-06-14 11:51

Ready; T=0.01/0.01 14:19:05
```

Figure 7-25 Result of generating CMS segment

Now any user can run CKREVM from any CMS user.

7.5.2 Configuring the parm file

Tivoli zSecure ships a sample configuration file for z/VM. The file is C2R\$PARV SCKRSAMP. You can use this file as a model to create the configuration for your environment. Copy this file as C2R\$PARV CKRPARM and place it on the disk with the other zSecure code. The file needs to reside on a minidisk that is linked and accessed before the application CKREMAIV is invoked.

Note: If your configuration file does not reside on the same minidisk as the code, see *IBM Tivoli zSecure Manager for z/VM RACF: Installation and Configuration Manual* (SC23-6574) or 7.5.1, “Making the software (CKREVM EXEC) available to VM/CMS users” on page 284 for more information.

We chose to place the configuration file on the 200 disk, because the CKREMAIV EXEC is called from the CKREVM EXEC which performs the required link and access to allow the application to execute.

We suggest that you copy C2R\$PARV SCKRSAMP E1 as C2R\$PARV CKRPARM E1 and change the parameters to match Figure 7-26.

```
C2R$PARV CKRPARM D1 F 80 Trunc=80 Size=15 Line=0 Col=1 Alt=0
====>
 0 * * * Top of File * * *
 1 workFileMode      = 'A'                /* Mode for new (scratch) fi
 2 customizationFileMode = 'D'            /* Filemode with CKRPROF fil
 3 license           = 'C2R#LIC C2RPARM D' /* LICENSE fn ft fm
 4 description       = 'Default files'     /* Description for input set
 5 unload            = ''                  /* Unloaded RACF database
 6 ioconfig          = ''                  /* fn ft fm of IOCONFIG
 7 smf               = ''                  /* fn ft fm of SMF
 8 init              = 'NO'                /* initialize every time
 9 loadlib           = 'YES'               /* use loadlib
10 userLibFileMode   = ''                  /* file mode user ISPF libra
11 racfDBCMS         = 'DBCOPY RACF A'     /* RACF db in CMS minidisk f
12 racfDbDataset     = 'RACF.DATASET'     /* OS dsname of RACF databas
13 racfMiniDisk      = 'RACFVM 200'       /* Minidisk with RACF databa
14 racfVirtualAddress = 'F00'              /* Virt addr to access RACF
15 racfFileMode      = 'G'                /* Mode to access RACF db
16 * * * End of File * * *
```

Figure 7-26 Suggested zSecure parm file

Note: License parameter is not used in the new version of Tivoli zSecure, so you will not see this line in the actual file.

When CKREMAIV runs it needs to link and access the RACF database disk. The configuration file defines the virtual machine name and address of the primary RACF database. This disk is also required when the CKREVCOP EXEC runs, it will use the CMS MOVEFILE utility create a copy of the RACF database which is on an OS-simulated disk address 200 as the file DBCOPY RACF A. The DBCOPY RACF file is not a sequential file copy of the database but this file is used by many of the Consul zSecure execs as input. The file is blocked with a logical record length of 4096 and allows for extremely fast searches by the Tivoli zSecure Pro Suite of products.

7.5.3 IOCONFIG file

For all functions of Tivoli zAudit, and for many functions of Tivoli zAdmin, an IOCONFIG data set is required. For several functions, an UNLOAD is recommended.

Most z/VM users do not use IOCONFIG file because they use Dynamic I/O Configuration. When Consul zSecure functions that requires an IOCONFIG to execute do not find the file, it reads the IO Control Block from memory to get I/O information.

So, if you do not have an IOCONFIG file, do not worry. Consul zSecure can still execute. Alternative, your reports will not contain Complex or System name, but instead will include underlined question marks (???) to enhance visibility, as shown in Example 7-27.

```
CNRFNAME SYSPRINT A1 V 133                1BLK 07/07/27                1/42
====>                                     BROWSE
1CKRCARLA 1.8.0 09/22/06 15.41 RC2R180   C o n s u l   z S e c u r e 27 Jul 200
(C) Copyright 1989-2006, Consul Risk Management B.V., Olof Palmestraat 10, 261

Input:  SYSIN

      1 |SELECT CLASS=USER UAUDIT
      2 |LIST CLASS KEY

CNR0009 00 Licensed to IBM Global Services (USA)
          , -, USA
          CPU-id          , source file LICENSE C2R#LIC CKRPARM D1
          Product feature codes AUDIT,RACF
CNR1092 00 $C2R.READALL in class FACILITY not defined. Using defaults.
CNR0017 00 Processing started for DB 1 SYSRAC01          DBCOPY RACF D1
CNR0031 00 Unrestricted mode active
CNR0017 00 File SYSRAC01 has restructured database format RACF release HRF5030

CNR0615 00 Input system structure overview (default system ? complex ?):
          Complex Func Prod System Timestamp      Filename Volser
Dsnam
          ? _____ Main RACF ? _____ 27 Jul 2007 10:06 SYSRAC01
DBCOP
          ???? ? _____ current settings **CMS**

CNR0661 00 Warning: ???? system ? now processed as if protected by RACF databas
CNR1302 00 Complex ? uses database templates of complex ?
CNR0038 00 Warning: RACF Range Table for complex ? unknown, SUPPRESS ICHRRNG im
```

Figure 7-27 Example of missing Complex and System name without a valid IOCONFIG

7.5.4 Installing the license file

Due to a change in the way IBM distributes Tivoli zSecure for z/VM RACF, a license file is not necessary in new versions of the product.

7.5.5 Changing the default setup

Before proceeding with changing the default setup, you must obtain a CMS-readable copy of RACF database, because the RACF database is in on OS-formatted DASD and Consul zSecure under z/VM cannot read this directly. It is recommend that you do this at regular intervals, for instance daily if your RACF/VM is changed very frequently.

Creating a CMS-readable copy of RACF database

A CMS-readable copy of RACF database is required when your RACF-database is on OS-formatted DASD (for instance, when your database is shared between z/VM and z/OS). Tivoli zSecure Manager for z/VM RACF cannot directly read this, so you will need to create a CMS-readable copy. It is recommended that you do this at regular intervals, for instance daily. You can use EXEC CKREVCOPY or be call it from the ISPF application (option SE.VM).

The exec CkREVCOP is provided for this purpose. If you just call the exec without passing any parameter, the exec will get the information from C2R\$PARV CKRPARM D (Example 7-28).

```
ckrevcop
DASD 0F00 LINKED R/O; R/W BY RACFVM      ; R/O BY ZSECURE
DMSACC723I G (0F00) R/O - OS
CKREVCOP: start of movefile
CKREVCOP: elapsed time = 0 min(s) and 4.18 seconds, returncode = 0
DASD 0F00 DETACHED
```

Figure 7-28 CKREVCOP execution example

Changing the Default

Note: If you are just starting using Tivoli zSecure, we recommend that you do not change the defaults settings. Stay with the settings as discussed in 7.5.2, “Configuring the parm file” on page 288.

The default setup is the recommended way to customize options for groups of users. It updates the files on the customizationFileMode minidisk (disk D for instance), so that you can create different default settings for separate groups of users. You can run Setup default against a new data set for testing purposes, and rename or copy the data sets when you are done, so your users will not be affected by an incomplete change. If you use only a single PROFDSN data set (C2R\$PARV CKRPARM D1, for instance), Setup default will set system-wide options. If no default settings are present in the PROFDSN data set, standard (Consul zSecure shipped) settings will be used.

To run Setup default, you need to run CKREMAIV exec. We show just an example of how to use CKREMAIV exec to copy the RACF Database to the A disk of a user that will use zAudit later. Here are the steps:

1. Type CKREMAIV and press Enter.

The screen is cleared and the message shown in Example 7-29 displays.

```
Enter CARLa commands below, empty line to terminate
```

Figure 7-29 CARLa execution interface fro z/VM CMS environment

2. Enter the CARLa commands.
 - a. Type V and press Enter (V is the CARLa command to copy RACF/VM database).
 - b. Press Enter again and you see a panel similar to the one shown in Example 7-30.

```
Enter CARLa commands below, empty line to terminate
v

CNRO017 00 Processing started for DB 1 SYSRAC01          DBCOPY  RACF    A1
CNRO033 00 ? DB 1 DBCOPY  RACF    A1 has 1,160 segments (of 256 byte) in use, 47
,416 segments free (2% used)
CNRO224 00 771 profiles and 27 segments read, 771 profiles and 27 segments selec
ted (100%) for ?
CNRO039 00 CKRCARLA used 0.00 CPU seconds, 576KB, and took 33.3 wall clock secon
ds
Press Enter to continue
```

Figure 7-30 Result of the RACF/VM database copy

- c. After you press Enter again, you see the sysprint of the execution (Figure 7-31). Note that this is a 132 columns file and for readability we show only 80 columns.

```

CNRFNAME SYSPRINT A1 V 133                      1BLK 07/07/25          1/35
====>                                           BROWSE
1CKRCARLA 1.8.0 09/22/06 15.41 RC2R180   C o n s u l   z S e c u r e 25 Jul 200
(C) Copyright 1989-2006, Consul Risk Management B.V., 01of Palmestraat 10, 261

Input:  SYSIN

      1 |v

CNR0009 00 Licensed to IBM Global Services (USA)
          , -, USA
          CPU-id           , source file LICENSE C2R#LIC C2RPARM D1
          Product feature codes AUDIT,RACF

CNR1092 00 $C2R.READALL in class FACILITY not defined. Using defaults.
CNR0017 00 Processing started for DB 1 SYSRAC01          DBCOPY RACF A1
CNR0031 00 Unrestricted mode active
CNR0017 00 File SYSRAC01 has restructured database format RACF release HRF5030

CNR0615 00 Input system structure overview (default system ? complex ?):
          Complex  Func  Prod System  Timestamp          Filename Volser Dsnam
          ?        Main  RACF ?      25 Jul 2007 15:52 SYSRAC01          DBCOP
                               ???? ?      current settings **CMS**

CNR0661 00 Warning: ???? system ? now processed as if protected by RACF databas
CNR1302 00 Complex ? uses database templates of complex ?
CNR0038 00 Warning: RACF Range Table for complex ? unknown, SUPPRESS ICHRRNG im
CNR0171 08 Class not in descriptor table, default properties assumed - VMLAN
CNR0036 00          at SYSRAC01 block 44 segment offset 0 DB seq 1 RBA 00000002AF0
CNR0033 00 ? DB 1 DBCOPY RACF A1 has 1,160 segments (of 256 byte) in use, 4
          Index uses 0%. Space beyond 3% never used. Using BDAMQSAM.
CNR0168 00 Maximum profile length on ? is 2,119 bytes for GROUP SYS1
CNR0224 00 771 profiles and 27 segments read, 771 profiles and 27 segments sele
CNR0039 00 CKRCARLA used 0.00 CPU seconds, 576KB, and took 33.3 wall clock seco

```

Figure 7-31 Sysprint of RACF database copy

7.5.6 Checking the zCollect function

To run IBM Tivoli zSecure Collect, you need to be logged on under a user ID that has privilege classes B, E, and G, and that has at least read access to the minidisk where the USER DIRECT file resides. You can run this from MAINT user ID but if you want to run from 5655T13A user ID, do not forget to add class B and E in the user ID direct.

From the Tivoli zSecure Manager for z/VM RACF primary menu, enter SE.1 (Setup files) and deselect all selected input sets. Next, enter SE.2 (Setup - New files) to create new CKFREEZE and UNLOAD files. You will be prompted for allocation parameters. As a raw approximation, specify:

- For CKFREEZE: 2 MB per online DASD-volume
- For UNLOAD: same size as your security database

Use the REFRESH command to fill these files.

7.5.7 Fresh CKFREEZE and UNLOAD

For all Audit functions, and for many administrative functions, a CKFREEZE data set is required. For several functions, an UNLOAD is recommended. You should run the REFRESH transaction under Setup files (SE.1) at regular intervals.

7.5.8 TCP/IP domain name resolution

Consul zSecure can report in various formats, including Simple Network Management Protocol (SNMP) and Simple Mail Transport Protocol (SMTP), that is e-mail. In this respect, Consul zSecure acts as a user of TCP/IP services, therefore the environment where Consul zSecure runs (be it a TSO or CMS user, or a batch job, or for instance the Consul zAlert or Consul zVisual RACF started tasks) might need domain name resolution. Depending on the level of your IP stack, you might need to set up a *userid.TCPIP.DATA*, or a SYSTCPD DD-statement, or some other method that points to the TCP stack that provides the DNS function.

Consult the IP Configuration Guide for the z/OS or z/VM release you are using. Also, make sure that the processes that need domain name resolution have READ access to all relevant files, like TCPIP.DATA, /etc/resolv.conf, and /etc/hosts.

7.6 Examples of some reports generated by Consul zSecure

As stated previously, you usually work through the interactive component of Consul zSecure. The interactive component is an ISPF application available under TSO on z/OS and CMS on z/VM. Depending on how your system administrator has set up the system you can start Consul zSecure by selecting the Consul zSecure option from a menu, or by a custom command. Ask your system administrator for your local details.

Consul zSecure is invoked in CMS on z/VM under ISPF option 6 (CMS command) with:

```
CKREVM
```

After this, the primary option menu displays.

If you do not have ISPF, the following message displays:

```
Enter CARLa commands below, empty line to terminate
```

Other methods to invoke Consul zSecure, including in batch and through TSO and CMS line mode commands. You can run multiple copies of zSecure in parallel in split screen or different CMS users. But they must be exactly the same version of zSecure, and they must use the same license file.

7.7 Sample UAUDIT list

One of the features of zAudit is get a list of RACF UAUDIT authority. We will show the initial UAUDIT status then will add some more user with UAUDIT authority then we will show the new list. Important fact is that whenever you change RACF database, you need to get a new CMS readable RACF file using the **ckrevcop** command.

We ran this command using an ordinary CMS user because at installation time we let CKREVM exec available for all users. To get a list of the RACF users with UAUDIT authority, run the following commands in CMS command line:

```
CKREVM
SELECT CLASS=USER UAUDIT
LIST CLASS KEY
```

Note: In the following examples, the CARLa commands used to perform the functions are highlighted in red.

Figure 7-32 shows the resulting file that is generated by CARLa.

```
CNRFFNAME SYSPRINT A1 V 133          1BLK 07/07/31          1/42
====>
1C2RCARLA 1.8.0 09/22/06 15.41 RC2R180  C o n s u l  z S e c u r e 31 Jul 200
(C) Copyright 1989-2006, Consul Risk Management B.V., Olof Palmestraat 10, 261
Input:  SYSIN
       1 | SELECT CLASS=USER UAUDIT
       2 | LIST CLASS KEY

CNR1092 00 $C2R.READALL in class FACILITY not defined. Using defaults.
CNR0017 00 Processing started for DB 1 SYSRAC01          DBCOPY  RACF  D1
CNR0031 00 Unrestricted mode active
CNR0017 00 File SYSRAC01 has restructured database format RACF release HRF5030

CNR0615 00 Input system structure overview (default system ? complex ?):
          Complex  Func  Prod System  Timestamp          Filename Volser Dsnam
          ?        Main  RACF ?      31 Jul 2007 10:52 SYSRAC01          DBCOP
          ???? ?      current settings  **CMS**

CNR0661 00 Warning: ???? system ? now processed as if protected by RACF databas
CNR1302 00 Complex ? uses database templates of complex ?
CNR0038 00 Warning: RACF Range Table for complex ? unknown, SUPPRESS ICHRRNG im
CNR0171 08 Class not in descriptor table, default properties assumed - VMLAN
CNR0036 00      at SYSRAC01 block 44 segment offset 0 DB seq  1 RBA 00000002AF0
USER      ZSECURE

CNR0033 00 ? DB 1 DBCOPY  RACF  D1 has 1,161 segments (of 256 byte) in use, 4
          Index uses 0%. Space beyond 3% never used. Using BDAMQSAM.
CNR0168 00 Maximum profile length on ? is 2,134 bytes for GROUP SYS1
CNR0224 00 772 profiles and 27 segments read, 1 profiles and 0 segments selecte
P R I N T   S U M M A R Y  31 Jul 2007 10:52
Source file/member/lineno Type          Records Output file/page Name          Title
SYSIN              1 racf              1 SYSPRINT              1

CNR0039 00 C2RCARLA used 0.01 CPU seconds, 576KB, and took 7.0 wall clock secon
```

Figure 7-32 Users with uaudit authority

ZSECURE is the only user with this authority. So, let us log on a user with RACF admin authority and give some other users UAUDIT authority, as shown in Figure 7-33.

```
rac alu ibmuser uaudit
Ready; T=0.01/0.01 10:57:57
rac alu maint uaudit
Ready; T=0.01/0.01 10:58:12
rac alu raicher uaudit
Ready; T=0.01/0.01 10:58:46
rac alu gnirss uaudit
Ready; T=0.01/0.01 10:58:58
```

Figure 7-33 Granting uaudit authority

Because we changed the RACF file, we should get a new copy or RACF database in CMS readable format so that Consul zSecure can be aware of the changes. Log on to zSecure VM again and issue **c2revcop** to perform this function (Figure 7-34).

```
c2revcop
DASD 0F00 LINKED R/O; R/W BY RACFVM ; R/O BY ZSECURE
DMSACC723I G (0F00) R/O - OS
C2REVCOP: start of movefile
C2REVCOP: elapsed time = 0 min(s) and 7.58 seconds,returncode=0
DASD 0F00 DETACHED
Ready; T=0.03/0.09 11:05:57
```

Figure 7-34 Get a new racf database in CMS format

After granting UAUDIT authority and getting a new RACF database copy, we can run CKREVM again and see the result (Figure 7-35).

```

Input:  SYSIN

      1 | SELECT CLASS=USER UAUDIT
      2 | LIST CLASS KEY

CNR01092 00 $C2R.READALL in class FACILITY not defined. Using defaults.
CNR00017 00 Processing started for DB 1 SYSRAC01          DEBCOPY RACF
CNR00031 00 Unrestricted mode active
CNR00017 00 File SYSRAC01 has restructured database format RACF release
CNR0615 00 Input system structure overview (default system ? complex ?)
          Complex  Func  Prod System  Timestamp          Filename Vols
          ?        Main  RACF ?      31 Jul 2007 11:07 SYSRAC01
                                ???? ?      current settings **CMS**

CNR0661 00 Warning: ???? system ? now processed as if protected by RACF
CNR1302 00 Complex ? uses database templates of complex ?
CNR0038 00 Warning: RACF Range Table for complex ? unknown, SUPPRESS IC
USER      IBMUSER
USER      MAINT
CNR0171 08 Class not in descriptor table, default properties assumed -
CNR0036 00      at SYSRAC01 block 44 segment offset 0 DB seq  1 RBA 000
USER      RAICHER
USER      GNIIRSS
USER      ESECURE
CNR0033 00 ? DB 1 DEBCOPY RACF      D1 has 1,161 segments (of 256 byte) i
          Index uses 0%. Space beyond 3% never used. Using BDAMQSAM.
CNR0168 00 Maximum profile length on ? is 2,134 bytes for GROUP SYS1

```

Figure 7-35 new users with uaudit authority

7.8 Personalized reports for RACF users with special and operations authority

This sections discusses an example of how you can personalize your reports as well as execute more than one type of report in a single run.

In this report, we want to know which users have SPECIAL and Operations authority. We choose to personalize the report to have a title of "ITSO POUGHKEEPSIE - RESIDENCE 7471" and for the first selection, special authority, we choose this subtitle "USERS WITH SYSTEM-WIDE SPECIAL ATTRIBUTE". For the second selection, operations authority, we choose this subtitle "USERS WITH SYSTEM-WIDE OPERATIONS ATTRIBUTE".

To achieve, this we first executed CKREVM and passed the following parameters (Figure 7-36).

```
Enter CARLa commands below, empty line to terminate
print title='ITSO Poughkeepsie - Residence 7471'
select class=user
newlist subtitle='Users with system-wide SPECIAL attribute'
select special
sortlist key(8) pgmrname dfltgrp instdata
newlist subtitle='Users with system-wide OPERATIONS attribute'
select operations
sortlist key(8) pgmrname dfltgrp instdata
```

Figure 7-36 Executing more than one command

After entering the commands, the report in Figure 7-37, Figure 7-38, and Figure 7-39 is generated. For readability, we show only 80 columns of the 133 columns and divide the output into three parts: Commands, Special, and Operations.

Figure 7-37 shows the command entered (sysin).

```

1C2RCARLA 1.8.0 09/22/06 15.41 RC2R180   C o n s u l   z S e c u r e   2 Aug 200

(C) Copyright 1989-2006, Consul Risk Management B.V., Olof Palmestraat 10, 261

Input:  SYSIN

      1 |PRINT TITLE='ITSO POUGHKEEPSIE - RESIDENCE 7471'
      2 |SELECT CLASS=USER
      3 |NEWLIST SUBTITLE='USERS WITH SYSTEM-WIDE SPECIAL ATTRIBUTE'
      4 |SELECT SPECIAL
      5 |SORTLIST KEY(8)  PGMNAME DFLTGRP INSTDATA
      6 |NEWLIST SUBTITLE='USERS WITH SYSTEM-WIDE OPERATIONS ATTRIBUTE'
      7 |SELECT OPERATIONS
      8 |SORTLIST KEY(8)  PGMNAME DFLTGRP INSTDATA

CNR0009 00 Licensed to IBM Global Services (USA)
           , -, USA
           CPU-id           , source file LICENSE  C2R#LIC C2RPARM D1
           Product feature codes AUDIT,RACF
CNR1092 00 $C2R.READALL in class FACILITY not defined. Using defaults.
CNR0017 00 Processing started for DB 1 SYSRAC01          DBCOPY  RACF    D1
CNR0031 00 Unrestricted mode active
CNR0017 00 File SYSRAC01 has restructured database format RACF release HRF5030
CNR0615 00 Input system structure overview (default system ? complex ?):
           Complex  Func  Prod System  Timestamp          Filename Volser Dsnam
           ?        Main  RACF ?      2 Aug 2007 10:56 SYSRAC01      DBCOP
           ???? ?      ???? ?      current settings  **CMS**
CNR0661 00 Warning: ??? system ? now processed as if protected by RACF databas

```

Figure 7-37 Commands entered

Figure 7-38 shows the SPECIAL attribute user listing.

```

CNRI302 00 Complex ? uses database templates of complex ?
CNR0038 00 Warning: RACF Range Table for complex ? unknown, SUPPRESS ICHRRNG im
CNR0171 08 Class not in descriptor table, default properties assumed - VMLAN
CNR0036 00      at SYSRAC01 block 44 segment offset 0 DB seq  1 RBA 00000002AF0
CNR0033 00 ? DB 1 DBCOPY  RACF      D1 has 1,161 segments (of 256 byte) in use, 4
          Index uses 0%. Space beyond 3% never used. Using BDAMQSAM.
CNR0168 00 Maximum profile length on ? is 2,134 bytes for GROUP SYS1
CNR0224 00 772 profiles and 27 segments read, 132 profiles and 19 segments sele

I P R O F I L E   L I S T I N G       2 Aug 2007 10:56
ITSO POUGHKEEPSIE - RESIDENCE 7471
USERS WITH SYSTEM-WIDE SPECIAL ATTRIBUTE
Profile  Name                      DfltGrp  InstData
COSTA                      SYS1
DATAMOVE                  SYS1
DETRO                    SYS1
DIRMAINT                 SYS1
DRUKER                   SYS1
FTPSERVE                 SYS1
GNIRSS                   SYS1
JIGUET                   SYS1
MAINT                    SYS1
OPERATOR                 SYS1
RAICHER                  SYS1
VELLOSO                  SYS1
VMRACF                   SYS1
ZSECURE                  SYS1
5VMRAC30                 SYS1

```

Figure 7-38 Special authority users

Figure 7-39 shows the OPERATIONS attribute user listing.

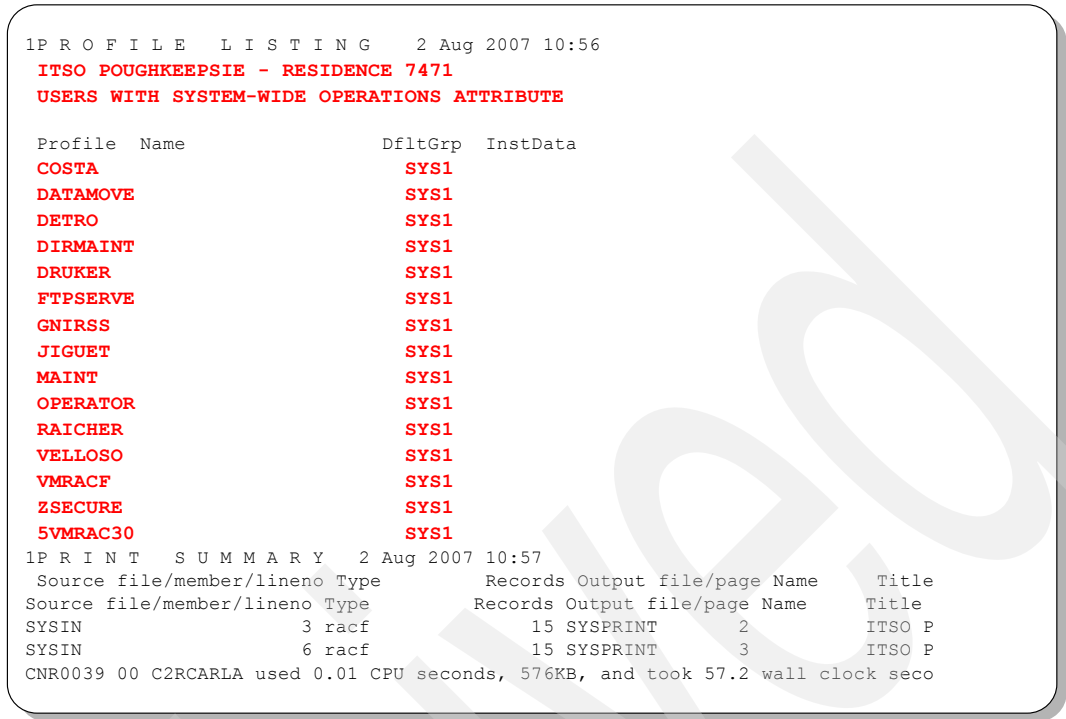


Figure 7-39 Operations attribute users

DirMaint implementation

This appendix provides the steps required to implement the IBM Directory Maintenance for z/VM (DirMaint) on a z/VM 5.3.0 system.

Note: We understand that not all installations will implement DirMaint on their system. However, we wrote this book using the DirMaint processes and felt that a discussion of how to implement the product is appropriate.

A.1 DirMaint implementation and configuration

DirMaint is a priced product shipped in disabled state with the z/VM 5.3.0 system deliverable using VMSES/E. It must be enabled and configured by the system programmer prior to using.

The program directory for the product describes the installation process and can be downloaded from the following Web site:

<http://www.vm.ibm.com/progdir>

DirMaint program directory describes the step-by-step process on how to enable and configure the product.

To implement and configure DirMaint, you need to:

1. Set DirMaint to the ENABLED state.
2. Perform post installation steps.
3. Tailor the DirMaint virtual machine.
4. Place DirMaint into production.

A.2 DirMaint installation

The installation process is performed from the virtual machine MAINT for the most part, although a couple of steps are executed from the product owning virtual machine for DirMaint.

Log on to the MAINT virtual machine. When the PROFILE EXEC executes, the appropriate disk will be accessed automatically.

Before you perform the steps in the program directory, you need to perform the following steps:

1. Update the USER DIRECT file on the 2CC disk (Figure A-1). Add passwords for DIRMAINT and DATAMOVE, and put the updated directory online. Both are set as NOLOG before your edit.

```
USER DIRECT L1 F 80 Trunc=72 Size=2007
====> ch /NOLOG/xyzzz/*
1274 USER DIRMAINT NOLOG 32M 64M BDG
1275 IPL CMS PARM AUTO CR
1276 MACHINE ESA
1277 ACCOUNT SYSTEM SYS PROG
1278 D8ONECMD FAIL LOCK
1279 OPTION CONCEAL DIAG88 D84NOPAS IGNMAXU
1280 IUCV ANY PRIORITY MSGLIMIT 100
1281 CONSOLE 009 3215 T
1282 ----- 20 line(s) not displayed -----
1302 *
1303 USER DATAMOVE NOLOG 32M 128M BG
1304 IPL CMS PARM AUTO CR
1305 MACHINE ESA
1306 ACCOUNT SYSTEM SYS PROG
1307 D8ONECMD FAIL LOCK
1308 OPTION CONCEAL IGNMAXU D84NOPAS LNKEXCLU LNKSTABL
1309 IUCV ANY PRIORITY MSGLIMIT 100
1310 CONSOLE 009 3215 T
```

Figure A-1 Updating the passwords for DIRMAINT and DATAMOVE

2. Update the SYSTEM NETID file on the 190 and 490 minidisks owned by MAINT. If you fail to update this file on the 490 disk, your data will be lost during the post installation processing for DirMaint.

A.2.1 Completing your installation of DirMaint

Your first task is to execute the service exec with the enable option for DirMaint, as shown in Figure A-2. The command is:

```
service dirm enable
```

```
service dirm enable
VMFSRV2760I SERVICE PROCESSING STARTED
VMFINS2767I READING VMFINS DEFAULTS B FOR ADDITIONAL OPTIONS
VMFINS2760I VMFINS PROCESSING STARTED
VMFINS2602R THE FOLLOWING COMPONENTS CAN BE ENABLED FOR PROD 5VMDIR30 DIR
ENTER THE NUMBER OF YOUR CHOICE
(0) BYPASS THIS PRODUCT
(1) :PPF 5VMDIR30 DIRM :PRODID 5VMDIR30%DIRM
      :DESC INSTALL/SERVICE DIRMAINT USING MINIDISK
(2) :PPF SERVP2P DIRM :PRODID 5VMDIR30%DIRM
      :DESC DIRECTORY MAINTENANCE FACILITY FUNCTION LEVEL 530
(3) EXIT
VMFINS2603I PROCESSING PRODUCT :PPF 5VMDIR30 DIRM :PRODID 5VMDIR30%DIRM
VMFINS2603I ENABLING PRODUCT 5VMDIR30%DIRM
VMFINS2771I THE CP SET PRODUCT COMMAND COMPLETED SUCCESSFULLY FOR PRODUCT
5VMDIR30
      ●
      ●
      ●
VMFBLD2180I THERE ARE 0 BUILD REQUIREMENTS REMAINING
VMFBLD2760I VMFBLD PROCESSING COMPLETED SUCCESSFULLY
VMFSUI2760I VMFSUFIN PROCESSING COMPLETED SUCCESSFULLY FOR PRODUCT 5VMDIR30%DIRM
VMFSUI2760I VMFSUFIN PROCESSING COMPLETED SUCCESSFULLY
VMFSUT2760I VMFSUFTB PROCESSING STARTED
VMFSUT2760I VMFSUFTB PROCESSING COMPLETED SUCCESSFULLY
VMFSRV2760I SERVICE PROCESSING COMPLETED SUCCESSFULLY
```

Figure A-2 Enabling DirMaint

The service exec uses the SERVP2P PPF file located on the 51D disk with the component name of DIRM to update the SYSTEM CONFIG file on the CF1 disk and issue the CP command **set product enable** for 5VMDIR30. When this completes, log off from the MAINT virtual machine.

You can then use the **query product** command to display the status of the pre-installed priced features of z/VM 5.3.0 and verify that the DirMaint is enabled, as shown in Figure A-3.

```
Q PRODUCT
Product State Description
5VMDIR30 Enabled 07/10/07.08:52:33.MAINT Install/service
DirMaint using minidisk
5VMPTK30 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT
FOR z/VM
5VMRAC30 Disabled 00/00/00.00:00:00.$BASEDDR RACF for VM
5VMRSC30 Enabled 00/00/00.00:00:00.$BASEDDR RSCS Networking
Version 5 Release 3 Modification 0
Ready; T=0.01/0.01 10:34:50
```

Figure A-3 QUERY PRODUCT command

A.2.2 Tailoring the DirMaint installation

This step must be completed from the product owning virtual machine (5VMDIR30), which executes the DIR2PROD EXEC located on the 492 disk (Figure A-4). This process accesses all the required disks so that the user input file can be created.

```
dir2prod access_new 5vmdir30 dirm
DMSACC724I 1DF replaces J (1DF)
DMSACP726I 492 E released
DMSACC724I 41F replaces L (41F)
DIR2PROD: Normal Termination.

q disk
LABEL VDEV M STAT CYL TYPE BLKSZ FILES BLKS USED-(%) BLKS LEFT
DRM191 191 A R/W 9 3390 4096 2 12-01 1608
MNT5E5 5E5 B R/O 9 3390 4096 131 1288-80 332
MNT51D 51D D R/O 26 3390 4096 273 1245-27 3435
DIR1DF 1DF J R/W 9 3390 4096 7 14-01 1606
DRM492 492 K R/W 15 3390 4096 260 1399-52 1301
DRM41F 41F L R/W 8 3390 4096 43 585-41 855
MNT190 190 S R/O 100 3390 4096 687 14592-81 3408
MNT19E 19E Y/S R/O 250 3390 4096 1010 26738-59 18262
Ready; T=0.01/0.01 09:22:38

link maint 2cc 2cc rr read
DASD 02CC LINKED R/O; R/W BY MAINT

ac 2cc m
DMSACP723I M (2CC) R/O
Ready; T=0.01/0.01 09:22:53

copy user direct m user input j
Ready; T=0.01/0.01 09:36:10
```

Figure A-4 Creating the USER INPUT file

Program directory describes the process of creating the USER INPUT file (Example A-4) on the 1DF disk owned by the DirMaint product owning virtual machine. The terminology that is used, *copy your current monolithic directory*, is referring to the USER DIRECT file on MAINT's 2CC disk. The file must have a record format of fixed and a logical record length of 80. That is how the USER DIRECT file was shipped so a simple copy command will create the required file. When you have completed this task logoff from the 5VMDIR30 virtual machine.

A.2.3 Placing DirMaint into production

This step is performed by the virtual machine MAINT. This product was developed in the VMSES/E format and the PUT2PROD exec is used to relocate the files that are necessary to use DirMaint by system administrators and end users. It requires that the 51D disk be in write mode so that the system level software inventory can be updated.

The **put2prod** exec actually calls several other exec that will copy the following disk:

- ▶ 492 ==> 491
- ▶ 41f ==> 11F
- ▶ 29E ==> 19E
- ▶ 29D ==> 19D

Because the copy of the 29E disk to the 19E disk will corrupt the Shared-Y Stat and the copy of 29D to 19D will change the HELPSEG saved segment, PUT2PROD generates a new saved segment and the CMS saved system.

Issue the following command (Figure A-5):

```
put2prod dirm
```

```
put2prod dirm
VMFP2P2760I PUT2PROD processing started
VMFP2P2760I PUT2PROD processing started for DIRM
VMFSET2760I VMFSETUP processing started for SERVP2P DIRMP2P
----- 37 line(s) not displayed -----
VMFSET2760I VMFSETUP processing completed successfully
----- 39 line(s) not displayed -----
DIR2PROD: Leaving LINDFLT DIRECT Q unchanged.
DIR2PROD: Normal Termination.
DIR2PROD: Copy of 492 disk to 491 disk has started.
----- 8 line(s) not displayed -----
DIR2PROD: Copy of 492 disk to 491 disk has completed.
DIR2PROD: Copy of 41F disk to 11F disk has started.
----- 10 line(s) not displayed -----
DIR2PROD: Copy of 41F disk to 11F disk has completed.
DIR2PROD: Normal Termination.
DMSACP726I 19E Y/S released
VMFP2P1231I Copying files from the DIRMAINT Product disk to the System P
disk
VMFP2P1231I Copying files from the DIRMAINT Help disk to the System Help
----- 126 line(s) not displayed -----
VMFP2P2760I PUT2PROD processing completed successfully
```

Figure A-5 Running the PUT2PROD Exec

Your DirMaint product is now ready for configuration. The program directory recommends you should xautolog the DIRMAINT and DATAMOVE virtual machines at this time. We also recommend to update the PROFILE EXEC for the AUTOLOG1 virtual machine so that necessary virtual machines are started automatically by the system during the IPL process (Figure A-6).

```
PROFILE EXEC Z1 V 130 Trunc=130
====>
0 * * * Top of File * * *
1 /*****/
2 /* Autolog1 Profile Exec */
3 /*****/
4
5 Address Command
6 'CP XAUTOLOG GCS'
7 'CP XAUTOLOG VMSESVS'
8 'CP XAUTOLOG VMSESVU'
9 'CP XAUTOLOG VMSESVR'
10 'CP XAUTOLOG DTCVSW1'
11 'CP XAUTOLOG DTCVSW2'
12 'CP XAUTOLOG TCPIP'
13 'CP XAUTOLOG PVM'
14 'CP SLEEP 5 SEC'
15 'CP XAUTOLOG RSCS'
16 'CP XAUTOLOG DIRMAINT'
17 'CP XAUTOLOG DATAMOVE'
18 'CP LOGOFF'
19 * * * End of File * * *
```

Figure A-6 AUTOLOG1's profile exec

A.3 DirMaint tailoring

The Program directory for DirMaint suggests that if you are migrating your system, you should copy several files to the appropriate disk for DIRMAINT. Because this is a new implementation of the product, you need to *modify* or *create* the following files:

- ▶ CONFIG DATADVH (shipped with product)
- ▶ CONFIGnn DATADVH (must be created)
- ▶ AUTHFOR CONTROL (must be created)
- ▶ RPWLIST DATA (copied from MAINT's 2CC disk)
- ▶ EXTENT CONTROL (must be updated)

CONFIG DATADVH

The CONFIG DATADVH file is the most important file for the DirMaint configuration. It contains all of the tailorable parameters for the product. You should *never* modify this IBM supplied file, because it could be updated through the service process and overwrite your changes. You create an override file instead where your installation specific parameters are defined. The override file is named CONFIGnn DATADVH, where the *nn* can be any two digits you want to assign. The CONFIG DATADVH file is self documented and additional information can be found in the *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6135.

CONFIGnn DATADVH

Note: The CONFIGnn DATADVH file must reside on the 11F disk when created. It is updated by using the **dirm send** and **file** commands. After changes are made, the DirMaint administrator must issue the **dirm rldata** command. This command is used to instruct the DIRMAINT virtual machine to reload all DATADVH files into memory.

The statements shown in Example A-1 are a good starting point for your CONFIGnn DATADVH file.

Example: A-1 CONFIG00 DATADVH

```
RUNMODE= OPERATIONAL
ONLINE= IMMED
DISK_CLEANUP=YES
CYLO_BLK0_CLEANUP=YES
EXTENT_CHECK= ON
DATAMOVE_MACHINE=DATAMOVE * *
DVHDXD_FLASHCOPY_BEHAVIOR= 0
```

We have found it to be easier to log on to the DIRMAINT virtual machine and copy the CONFIG DATADVH file to CONFIG00 DATADVH and then delete all the statements that are not required. This file *must* reside on the same disk as the CONFIG DATADVH file.

AUTHFOR CONTROL

The next step is that you must define authorized administrators for your system in DIRMAINT. You can have several administrators define and they are defined in the AUTHFOR CONTROL file (Figure A-7). The file is not shipped with the product and must be created manually. The file *must* reside on DIRMAINT's 1DF disk. This file is also case sensitive. It *must* be in upper case.

```

AUTHFOR  CONTROL  A1  F 80  Trunc=80  Size=8
====>
0 * * * Top of File * * *
1 ALL  MAINT      *   140A ADGHOMPS
2 ALL  MAINT      *   150A ADGHOMPS
3 ALL  DETRO      *   140A ADGHOMPS
4 ALL  DETRO      *   150A ADGHOMPS
5 ALL  RAICHER    *   140A ADGHOMPS
6 ALL  RAICHER    *   150A ADGHOMPS
7 ALL  VELLOSO    *   140A ADGHOMPS
8 ALL  VELLOSO    *   150A ADGHOMPS
9 * * * End of File * * *

```

Figure A-7 AUTHFOR CONTROL File

The AUTHFOR CONTROL file specifies which virtual machines have authority to modify the characteristics of other virtual machines on the system. This authority can be limited to specific target virtual machines or to specific attributes of a target virtual machine. This is implemented with DirMaint command sets. The format of this file is column specific. See Table A-1 for details.

Table A-1 Column layout for AUTHFOR CONTROL

Column	Description
1 - 8	Virtual machine being authorized
9	blank
10 - 17	DirMaint Administrator virtual machine
18	blank
19 - 26	System Netid
27	blank
28 - 31	Command Level
32	blank
33 - 68	Code Set - see DirMaint Admin Guide for description of the code

RPWLIST DATA

The RPWLIST DATA file is located on MAINT's 2CC minidisk (Figure A-8). You should link and access this disk and copy the file to the 11F disk owned by DIRMAINT. It can be used to disable passwords that you have deemed to be trivial. The syntax of the file is documented in the file itself. Syntax is case sensitive and *must* be in upper case.

```
RPWLIST DATA      D1  F 80  Trunc=80 Size=34 Line=0 Col=1 Alt=0
===>
0 * * * Top of File * * *
1 ACNT *****
2 AUTOLOG * Copyright -
3 BATCH *
4 CE * THIS MODULE IS "RESTRICTED MATERIALS OF IBM"
5 CMSUSER * 5654-A17 (C) COPYRIGHT IBM CORP. - 1988, 2001
6 CMS2 * LICENSED MATERIALS - PROPERTY OF IBM
7 CMS3 *
8 CPCMS * ALL RIGHTS RESERVED.
9 DIRM *
10 ECMODE * Status - z/VM Version 4, Release 2.0
11 IBMCE *****
12 IPCS
13 ISMAINT *****
14 ITPS *
15 IVPASS * Restricted Password List
16 LEV2VM * Format Rules:
17 MASTER *
18 MDVR * 1) The RPWLIST DATA file must be fixed record length
19 OPASS * format, with a record length of at least 8.
20 OSVS1 * 2) Each password must start in column 1.
21 PASSWORD * 3) Columns 1-8 must contain restricted passwords only.
```

Figure A-8 RPWLIST DATA

EXTENT CONTROL

The final configuration file we need to discuss is the EXTENT CONTROL file. Now that you have added your virtual machine to the AUTHFOR CONTROL file, you can use the DirMaint **send** and **file** commands to get a fresh copy of this file (we will discuss this process in the next section).

The EXTENT CONTROL file is used by the DIRMAINT virtual machine for DASD management (Figure A-9). It defines what DASD on the system and what cylinder ranges are available for creating minidisks for virtual machines. The DirMaint **amdisk** (add minidisk) command will specify the DASD name or a group name that is listed in this file to determine where the minidisk should be created.

```

EXTENT  CONTROL  E2  V 80  Trunc=80 Size=36 Line=0 Col=1 Alt=6
====>
0 * * * Top of File * * *
1 *****
2 ----- 9 line(s) not displayed -----
11 :REGIONS.
12  *RegionId  VolSer      RegStart      RegEnd  Dev-Type  Comments
13 LX5W02      LX5W02      0001          END     3390-03   DETRO 7/10/7
14 LX5W03      LX5W03      0001          END     3390-03   DETRO 7/10/7
15 LX5W04      LX5W04      0001          END     3390-03   DETRO 7/10/7
16 LX5W05      LX5W05      0001          END     3390-03   DETRO 7/10/7
17 :END.
18 :GROUPS.
19  *GroupName RegionList
20 :END.
21 :EXCLUDE.
22  * UserId Address
23   MAINT    012*
24 :END.
25 :AUTOBLOCK.
26  * IBM supplied defaults are contained in the AUTOBLK DATADVH file.
27  * The following are customer overrides and supplements.
28  *
29  *DASDType BlockSize Blocks/Unit Alloc_Unit Architecture

```

Figure A-9 EXTENT CONTROL

This file is shipped as an empty shell. You must update the REGIONS and EXCLUDE sections at a minimum. This file must reside on the 1DF. It was placed there when the PUT2PROD exec was run for DirMaint.

The entries in the REGIONS section are positional:

regionid volser starting-cylinder ending-cylinder dasd-type

The *dasd-type* of these entries are defined in the DEFAULTS DATADVH file. See Figure A-10.

The starting-cylinder parameter should specify the lowest cylinder where a minidisk could be created. This entry *must* never be zero(0). If you code zero and the disk is formatted with CMS format, CPFMTXA, or DASDFMT, it would change the label on the real DASD volume and then the minidisk would not be accessible.

If you code END as the ending-cylinder, DirMaint determines the last cylinder from the entry in the DEFAULTS DATADVH file for the DASD-type listed (Figure A-10).

```

DEFAULTS DATADVH  A1  V 80  Trunc=80 Size=87 Line=34 Col=1
====>
34 :DEFAULTS.
35 ----- 27 line(s) not displayed -----
62 3390-01      1113
63 3390-02      2226
64 3390-03      3339
65 3390-09     10017
66 3390-084     1084
67 3390-151     2226
68 3390-153     4365
69 3390-455      455
70 3390-568     1568
71 3390-32K     32760
72 3390-64K     65520
73 3390         1113
74 ----- 14 line(s) not displayed -----
88 * * * End of File * * *

```

Figure A-10 DEFAULTS DATADVH

As a final note about updating these files, you can log on to the DIRMAINT virtual machine and use CMS XEDIT *filename filetype* or you can perform the changes with the DirMaint commands **send** and **file**. However, you must tell DIRMAINT to reload the files after making modifications. There are three commands that reload the various files - **rldata**, **rlextent**, and **rlcode**. We have found it much easier to write a REXX exec to perform all three task every time we change a file (Figure A-11). This way, you do not have to remember which command is for which file.

```

FIXDIRM EXEC      A1  V 130  Trunc=130 Size=5
====>
0 * * * Top of File * * *
1 /**/
2 dirm RLDCode
3 dirm RLDData
4 dirm RLDExtn
5
6 * * * End of File * * *

```

Figure A-11 FIXDIRM Exec

A.4 DirMaint testing and operations

The daily operations of adding and updating virtual machines is performed by people who were added to the AUTHFOR CONTROL file after DirMaint has been successfully enabled and configured (Figure A-7 on page 308). The methods of doing this work are documented in the *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6135. It describes several methods of adding and modifying virtual machines. We will describe here only one of the methods.

A.4.1 Adding virtual machines

The *z/VM CP Planning and Administration*, SC24-6083 manual describes how to create virtual machine and all the statements that are available. DirMaint also has a panel to add every Control Program directory statement for minidisk, password, virtual NIC, and so forth after you have defined the virtual machine.

Note: All z/VM manuals are available at the following Web site:

<http://www.vm.ibm.com/library>

Our first step is to define a new virtual machine.

We suggest that you create the file SAMPLE DIRECT file shown in Figure A-12 on your A-disk.

```
SAMPLE    DIRECT    A1  F 80  Trunc=72  Size=5
===>
0 * * * Top of File * * *
1  USER  SAMPLE mypasswd    32M    128M    G
2  INCLUDE IBMDFLT
3  IBM CMS PARM AUTO CR
4  MACHINE XA
5  LINK TCPMAINT 0592 0592 RR
6 * * * End of File * * *
```

Figure A-12 SAMPLE DIRECT

To create a new virtual machine you will need to copy this sample file with a new file name. The file name *must* be the name of the virtual machine being defined.

For example, use the command **copy sample direct a gumby direct a** (if you want to define a virtual machine named *gumby*). Edit your new file and change the virtual machine name

(the parameter following the USER statement) to match the file name of the file and save your changes.

After completing that process, issue the command **DirMaint command dirm add**, which opens a window (Figure A-13).

-----DirMaint ADD-----

Add an entry to the directory for a new USERID or Profile.

Fill in the USERID or PROFILE being added:

==> gumby

Optionally fill in the following when using a prototype:

LIKE ==> (file name of prototype)

PW ==> (password for new user)

VPW ==> (password again for verification)

ACCT ==> (account value for new user - optional)

Notes:

- If a value is given for any one of PW, VPW, or ACCT, then a value is required for LIKE.
- If a value is given for either PW or VPW, then a value is required for both of them.

741-A05 (c) Copyright IBM Corporation 1979, 2007.

1= Help 2= Prefix Operands 3= Quit 5=Submit

Figure A-13 DirMaint ADD

Fill in the blanks and press PF5. You should receive confirmation that you successfully added the new virtual machine. It is indicated by zero return code. If you receive a non-zero return code you must correct the problem and issue the **dirm add** again.

If all parameters were correct, you should see the messages as shown in Example A-2.

Example: A-2 Output from the DIRM ADD

```
DVHREQ2288I Your ADD request for GUMBY at * has been accepted.
DVHBIU3450I The source for directory entry GUMBY has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHDRC3451I The next ONLINE will take place via delta object directory.
DVHBIU3428I Changes made to directory entry GUMBY have been placed
DVHBIU3428I online.
DVHBIU3450I The source for directory entry GUMBY has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHDRC3451I The next ONLINE will take place via delta object directory.
DVHBIU3428I Changes made to directory entry GUMBY have been placed
DVHBIU3428I online.
DVHREQ2289I Your ADD request for GUMBY at * has completed; with RC = 0.
```

A.4.2 Updating EXTENT CONTROL

Your next step would be to add a minidisk to this virtual machine. Before we can perform this task we need to inform the DIRMAINT virtual machine what DASD is available to create the minidisk on. There are two methods of doing this and we will discuss both. The first method uses the **dirm send** command (to retrieve a copy of the file) and then after it has been modified, uses the **dirm file** command to return it to the DIRMAINT virtual machine (Example A-3). We find this method useful when we have several volumes we want to add to the EXTENT CONTROL file. Issue the following command:

```
dirm send extent control
```

Example: A-3 Using DIRM SEND

dirm send extent control

DVHXMT1191I Your SEND request has been sent for processing.

Ready; T=0.03/0.03 13:42:52

DVHREQ2288I Your SEND request for DETRO at * has been accepted.

RDR FILE 0008 SENT FROM DIRMAINT PUN WAS 0011 RECS 0026 CPY 001 A NOHOLD NOKEEP

DVHREQ2289I Your SEND request for DETRO at * has completed; with RC = 0.

receive 8 (replace

File EXTENT CONTROL A2 replaced by EXTENT CONTROL E2 received from DIRMAINT

Ready; T=0.01/0.01 13:43:22

You can now edit this file and add your new regions (Figure A-9 on page 310) and then save your changes. We normally suggest that you make the region name match the volser of the DASD volume as a best practice. The DASD has to be formatted with CPFMTXA or ICKDSF, minidisk reside on cylinders that have been allocated as PERM.

When you are satisfied with your changes you need to return the file to the DIRMAINT virtual machine. This is accomplished with the *DirMaint* command **dirm file extent control**. See Example A-4.

Example: A-4 DIRM FILE command

dirm file extent control

PUN FILE 0009 SENT TO DIRMAINT RDR AS 0012 RECS 0023 CPY 001 0 NOHOLD NOKEEP

DVHXMT1191I Your FILE request has been sent for processing.

Ready; T=0.03/0.03 13:52:47

DVHREQ2288I Your FILE request for DETRO at * has been accepted.

DVHRCV3821I File EXTENT CONTROL E2 has been received; RC = 0.

DVHREQ2289I Your FILE request for DETRO at * has completed; with RC = 0.

We suggest that if you are going to use this process you always get a new copy of the file. After you have successfully returned the file, you need to have the DIRMAINT virtual machine reload the new information into memory. This is accomplished with the *DirMaint* command **dirm rldextn**. But to keep from having to remember which file requires which reload it is easier to run the exec we created earlier (Figure A-11 on page 311).

Second method uses the **dirmdasd** command to add entries to the EXTENT CONTROL file. When you use this command you will receive another DirMaint window (Figure A-14). From this window you can query, add, or delete regions and groups from the EXTENT CONTROL file and it will automatically issue the **dirmrldextn** command on your behalf.

```

-----DirMaint DASD-----
Add, delete or query DASD statement associated with Group, Region & Volume.
Select one:  x Add  _ Delete      _ Query or _ FREEExt or _ USEDext
Select one of the following for Add:
Group  GroupName ==>      _ (LINEAR) (or) _ (ROTATING) (Optional)
Region1 ==>              Region2 ==>              Region3 ==>
Region RegionName ==> USE001  Valid ==> USE001 Device Type ==> 3390-03
  Optionally you can fill one or all of the following:
    Size => 3339    Start => 1    Comments => detro 7/12/7
Volume  Valid =>    Device Type =>
  Optionally you can fill one or all of the following:
    Size =>          Start =>          Comments =>
Select one of the following for Delete:
Group  GroupName =>      Region1 =>      Region2 =>      or _ *
Region Region1 =>      Region2 =>      Region3 =>
Volume Valid1  =>      Valid2 =>      Valid3  =>
Select one of the following for Query:
Group  GrpName1 =>      GrpName2 =>      GrpName3 =>      or _ *
Region Region1 =>      Region2 =>      Region3  =>      or _ *
Volume Valid1  =>      Valid2  =>      Valid3  =>      or _ *

5741-A05 (c) Copyright IBM Corporation 1979, 2007.
1= Help      2= Prefix Operands      3= Quit      5=Submit      12=Cursor

```

Figure A-14 DirMaint DASD

To add volumes to the EXTENT CONTROL file, select the ADD function with a non-blank character. Then enter the region-name, valid, device type, size, starting cylinder, and a comment. When all the information is correct, press the PF5 key. You should receive messages similar to Example A-5.

Example: A-5 Output from DIRMDASD

```

DVHXMT1191I Your DASD request has been sent for processing.
Ready; T=0.07/0.08 14:19:58

DVHREQ2288I Your DASD request for DETRO at * has been accepted.
DVHREQ2289I Your DASD request for DETRO at * has completed; with RC = 0.

```

You can also group multiple regions into groups with this window. When creating groups, we recommend to use the rotating option over the linear option to get a better distribution of minidisks on DASDs. The rotating option will define minidisks in a round robin fashion while the linear option will fill up the entire DASD before going to the next DASD.

A.4.3 Adding MDISK to a Virtual Machine

Now it is time to create a MDISK for a new virtual machine. This is accomplished using another of the DirMaint windows. Before we go to that window, we need to discuss the syntax of the **dirmaint** command. When we updated the AUTHFOR CONTROL file, we added duplicate entries for each virtual machine, authorizing the 140A and 150A command sets. The command sets deal with the execution of the **dirmaint** commands. When issuing commands, you should use the new syntax format **dirmd for userid get** (150A command

set). The old syntax used by command set 140A, **dirm get userid** is still required because several of the panels used by the DirMaint product still use the old syntax within the program. If you did not have the command set 140A authorization, many of the DirMaint panels would fail. You can get additional information about command syntax from the Chapter 2 of Directory Maintenance Facility Commands Reference, SC24-6133.

When issuing DirMaint commands that will be directed at a specific virtual machine, use the format:

`dirm for userid command (optional parameters)`

For example: **dirm for gumby clonedisk 191 linuxmst 191 autor use001**

If you enter all the required parameters for a specific dirmaint command it will execute and not open a DirMaint window. If you only issue **dirm for gumby clonedisk**, then it opens the window. This becomes very helpful if you want to write REXX exec's that will automate processes of creating new virtual machines.

You can now add a minidisk to the newly created virtual machine. We execute the **dirm amdisk** command to open the window (Figure A-15).

```
-----DirMaint AMDISK-----
To add a new minidisk to a user definition, fill in the following:
  Minidisk Address ==> 191      Device Type ==> 3390
Fill in one of the following rows:
  Explicit Start ==>          Size ==>          Volser ==>
  AUTOV          Size ==>          Volser ==>
  VBLK  Blksize ==>          Blocks ==>          Volser ==>
  AUTOG          Size ==>          Grpname ==>
  GBLK  Blksize ==>          Blocks ==>          Grpname ==>
  AUTOR          Size ==> 5          Region ==> Lx5W02
  RBLK  Blksize ==>          Blocks ==>          Region ==>
  T-DISK          Size ==>
  TBLK  Blksize ==>          Blocks ==>
  V-DISK          Size ==>
  VDBS  Blksize ==>          Blocks ==>
  DEVNO          Real Device Number ==>
Optionally fill in:
  Link Mode ==> mr
  BLKSIZE  ==>          LABEL ==> gumby
  PWS  Read ==> xxx  Write ==> yyyyy  Multi ==> zzzzzz  (passwords)
741-A05 (c) Copyright IBM Corporation 1979, 2007.
  1= Help      2= Prefix Operands      3= Quit      5=Submit      12=Cursor
```

Figure A-15 DIRMAINT AMDISK

The fields that need to be completed are:

- ▶ Minidisk Address
- ▶ Device Type
- ▶ AUTOR (region name and size) or AUTOG (group name and size)
- ▶ Link Mode
- ▶ Label (if you want the disk CMS formatted by the DATAMOVE virtual machine)
- ▶ Passwords for read, write, and multi links (this is optional)

To verify the changes to this virtual machine, you can issue **dirm for gumby get noload** command, then **peek** the file that is sent to your reader as shown in Example A-6.

Example: A-6 DIRM GET command

dirm for gumby get noload

DVHXMT1191I Your GET request has been sent for processing.
Ready; T=0.03/0.03 15:11:45

DVHREQ2288I Your GET request for GUMBY at * has been accepted.
DVHGET3305I Entry GUMBY sent, no lock attempt was made.

RDR FILE 0010 SENT FROM DIRMAINT PUN WAS 0015 RECS 0011 CPY 001 A NOHOLD NOKEEP
DVHREQ2289I Your GET request for GUMBY at * has completed; with RC = 0.

peek 10

0010 PEEK A0 V 80 Trunc=80 Size=7 Line=0 Col=1 Alt=0
File GUMBY DIRECT from DIRMAINT at VMLINUX5 Format is NETDATA.
* * * Top of File * * *
USER GUMBY POKEY 32M 128M G
INCLUDE IBMDFLT
IPL CMS PARM AUTO CR
MACHINE XA
LINK TCPMAINT 0592 0592 RR
MDISK 0191 3390 1827 5 LX5W02 MR ALL G04IT G04IT
*DVHOPT LNKO LOG1 RCM1 SMSO NPW1 LNGAMENG PWC20070712 CRCJÖ
* * * End of File * * *

If you have several parameters you need to change for a virtual machine, you can use the **dirm for gumby get** command, which will lock the virtual machine from changes by any other administrator, and then receive the file and make the changes. When you have saved your changes you can issue **dirm for gumby replace**, which will modify the entry for this virtual machine in the cluster file and then automatically put the changes online based upon the entry in the CONFIG00 DATADVH file (Example A-6).

A.5 Conclusion

We covered some of the DirMaint basics in this chapter because these tasks are used during the RACF for z/VM configuration, management, and user management that we discussed in Chapter 2, “RACF feature of z/VM” on page 21.

Archived



RACF procedural checklist

This appendix is an excerpt from the *RACF Security Server for z/VM Program Directory*, GI10-0788. You can use it to track the activities that are required to install RACF on the z/VM system.

B.1 RACF installation steps

This section describes how to complete the installation of RACF, function level 530. RACF is pre-installed on the z/VM System. So, the steps that we describe here are the steps that you must follow to complete the installation. Make sure to follow this checklist to track the installation steps for RACF as you complete them.

The tasks are:

- ☐ Task 1. Review information about resources for RACF, especially if you plan on sharing the RACF databases.
- ☐ Task 2. Skip this step, unless you are sharing or migrating an existing RACF database, as you might have to convert the database templates.
- ☐ Task 3. Create an RPIDIRCT SYSUT1 file of RACF commands.
- ☐ Task 4. (Optional) Customize the RACFSMF user ID.
- ☐ Task 5. (Optional) Change the message routing table.
- ☐ Task 6. Delete or replace the ICHDEX01 Exit to Select Password Protection Algorithm.
- ☐ Task 7. (Optional) Delete or replace the ICHDEX02 Exit.
- ☐ Task 8. (Optional) Customize RACF within CP.
- ☐ Task 9. Enable and Install the CP part of RACF for VM.
- ☐ Task 10. (Customers sharing RACF databases) Change RACF database names. Considerations need to be made if sharing the database with z/OS.
- ☐ Task 11. IPL the CP system with RACF.
- ☐ Task 12. Initialize or update the RACF database.
- ☐ Task 13. (Optional) Create the global access table.
- ☐ Task 14. (Optional) Set RACF options.
- ☐ Task 15. (Optional) Determine audit and control options for VM events.
- ☐ Task 16. (Optional) Split the RACF database.
- ☐ Task 17. (Optional) Set up dual registration.
- ☐ Task 18. (Optional) Install the RACF ISPF panels.
- ☐ Task 19. Place RACF into production.

Additional material

The appendix describes how to download addition material to which this book refers from the Internet.

Locating the Web material

The Web material that is associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

<ftp://www.redbooks.ibm.com/redbooks/SG247471>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG24-7471.

Using the Web material

The additional Web material that accompanies this book includes the following files:

<i>File name</i>	<i>Description</i>
sg247471.zip	Zipped XML file with AssemblyLines for Directory Integrator

How to use the Web material

Create a subdirectory (folder) on your workstation, and decompress the contents of the Web material zipped file into this folder.

Archived

Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 324. Note that some of the documents referenced here might be available in softcopy only.

- ▶ *IBM System z9 109 Configuration Setup*, SG24-7203
- ▶ *Running Linux Guest in less than CP Privilege Class G*, REDP-3870
- ▶ *Introduction to the New Mainframe: Security*, SG24-6776
- ▶ *Using Cryptographic Adapters for Web Servers with Linux on IBM System z9 and zSeries*, REDP-4131
- ▶ *Linux on IBM zSeries and S/390: Securing Linux for zSeries with a Central z/OS LDAP Server (RACF)*, REDP-0221
- ▶ *z9-109 Crypto and TKE V5 Update*, SG24-7123
- ▶ *SSL Server Implementation for z/VM 5.2*, REDP-4348

Other publications

These publications are also relevant as further information sources:

- ▶ *Program Directory for Directory Maintenance Facility for z/VM*, GI10-0786
- ▶ *Program Directory for RACF Security Server for z/VM*, GI10-0788
- ▶ *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6135
- ▶ *Directory Maintenance Facility Messages*, SC24-6134
- ▶ *Directory Maintenance Facility Commands Reference*, SC24-6133
- ▶ *RACF Security Server Auditor's Guide 2.08*, SC24-6143
- ▶ *RACF Security Server Command Language Reference 4.70*, SC24-6144
- ▶ *RACF Security Server Diagnosis Guide 2.19*, GC24-6145
- ▶ *RACF Security Server General User's Guide 1.27*, SC24-6146
- ▶ *RACF Security Server Macros and Interfaces 8.55*, SC24-6147
- ▶ *RACF Security Server Messages and Codes 4.03*, GC24-6148
- ▶ *RACF Security Server Security Administrator's Guide 4.71*, SC24-6142
- ▶ *RACF Security Server System Programmer's Guide*, SC24-6149
- ▶ *Secure Configuration Guide for z/VM*, SC24-6138
- ▶ *System z9 Processor Resource/Systems Manager Planning Guide*, SB10-7041
- ▶ *z/VM CP Planning and Administration*, SC24-6083

- ▶ *z/VM CP Commands and Utilities*, SC24-6081
- ▶ *Linux for System z, Secure Key Solution with the Common Cryptographic Architecture, Application Programmer's Guide*, SC33-8294
- ▶ *z/OS ICSF TKE Workstation User's Guide*, SA22-7524
- ▶ *z/VM TCP/IP Planning and Customization*, SC24-6125
- ▶ *z/VM TCP/IP LDAP Administration Guide*, SC24-6140
- ▶ *Linux on System z, Device Drivers, Features, and Commands, February, 2007*, SC33-8289
- ▶ *z9-109 Crypto and TKE V5 Update*, SG24-7123

How to get IBM Redbooks publications

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Numerics

5VMRAC30 file 31

A

Adjunct Processor (AP) 220
Advanced Encryption Standard (AES) 204
AP queue 220
API 254
APs 220
 domain 1 222
AssemblyLine 185–186

B

B200 disc 97
BFS file
 pool 126
BFS file pool
 definition 127
Byte File System (BFS) 126

C

Candidate list 206
CCA library 265
central processor (CP) 204
CEX2A cards 241
CEX2C 239
changelog 151, 184
Channel Subsystem Call (CHSC) 221
cipher suite 204
clear key
 cryptography 223, 233
 cryptography exploitation 233
 only operation 235
 operation 201, 204
 RSA operation 204, 217
 support 204, 221
CMS 52, 305
 secondary key database file 260
CMS command 77
CMS user 117, 129, 144
Common Cryptographic Architecture (CCA) 232, 235
common name (CN) 114, 151
configuration file 243
Control Program (CP) 2–3, 21, 128
Controlled Access Protection Profile (CAPP) 8
coprocessor 206–207
coprocessor number 207, 212
CP directory
 entry 11
 file 13
 statement 312
CP privilege

 class 11
 class G 11
CPACF 204
CPC object 213
CPLOAD Module 22
Crypto Express2 204, 212
 aware, using secure key 269
 card 204, 255
 coprocessor 207
 feature 204
 nondisruptive 206
 support 212
CRYPTO statement 221–222
 operand APVIRT 225
Crypto Type Configuration
 panel 219
cryptography card 220–221
cryptography engine 205, 220
cryptography hardware
 acceleration 204
 support 231

D

DASD Dump Restore (DDR) 98
DASD request 315
data encryption standard (DES) 161, 204
database LDBM 130, 132
database LDBM GLDBLD31 148, 150
Debug setting (DS) 148
dedicated and shared AP queue
 QUERY Crypto 230
Default Porttype 19, 65
direct access storage device (DASD) 11, 157
directory entry 13, 158, 203
 BILLYBOB 56
 GUMBY 313
Directory Information Tree (DIT) 113
Directory Integrator 145, 184, 321
DirMaint administrator 55, 123, 307
DirMaint command
 dirm 123, 313
 file 47
 set 308
DirMaint command dirm
 file extent control 314
 rdextn 314
DirMaint exit 44
DirMaint implementation 61, 302
DirMaint product 21, 305
DIRMAINT Pun 23
DirMaint virtual machine 23, 302
DirMaint window 315
Discretionary Access Control (DAC) 8
distinguished name (DN) 113, 153

domain number 209

DS ENVVARS

D1 131, 149

file 131

E

encryption request 203

Evaluation Assurance Level (EAL) 9

EXTENT CONTROL

file 309

External Security Manager (ESM) 13, 21–22

F

Federal Information Processing Standard (FIPS) 205

Fibre Channel Protocol (FCP) 2–3

filelist command 46

filemode 27

G

GDBM backend 116, 188–189

Guest LAN 64

H

Hardware Management Console (HMC) 206

Hardware Security Module (HSM) 6

hardware support

different types 204

I

IBM HTTP Server

configuration file 262

User 262

IBM HTTP server 17, 204, 259

ibm objectclass 151

IBM Redbooks

Web server 321

Web site 321

IBM System

z processor 64

IBM Tivoli

Directory Integrator 184

Directory Server 145

IBM Tivoli Directory Server 111

ibmca 234

ibmca engine 232, 242

IBM-defined (ID) 148

ICHCONN Class 131, 149

in-kernel cryptography

module 263

modules 231

Input/Output Configuration Data Set (IOCDS) 161

installed cryptography card

current architectural limit 239

Integrated Cryptographic Services Facility (ICSF) 6

Interactive System Productivity Facility (ISPF) 44

International Standards Organization (ISO) 7

Interpretive Execution Facility

System z 5

Interpretive Execution Facility (IEF) 2–3

IP address 112, 153

IP packet

destination IP address 18

IPL CMS 222

PARM AUTOOCR 168

IPL process 306

IRR.PWENV.KEYR ING 195

IRRPHEX SAMPLE

file 48

ISPF 45

IUCV session 131, 149

L

Labeled Security Protection Profile (LSPP) 9

LDAP administrator 134, 152

LDAP Client

Configuration 140

Configuration panel 140–141

LDAP client 112, 139, 146

general interaction 112

LDAP Data Interchange Format (LDIF) 114

LDAP server 111, 138, 146–147

backend 129

control 179

definition 123

export 115

output file 114

parameter 126

specific BFS 126

start 131

store 116

test user 152

ldapsearch command 144, 181

LDAPSRV2 DTCPARMS 126, 147

ESM_Enable parameter 130

LDIF file 114, 151, 181

Lightweight Directory Access Protocol (LDAP) 1, 17, 111, 145

Linux distribution 138, 146

Linux guest 18

CRYPTO entry 238

CRYPTO statement 234

directory definitions 230

Setup 221

user directory 221

Linux server 6, 111, 119, 137, 146–147

System z Linux 137

Linux system 138, 145

boot 237

directory entry 236

high number 235

service 146

Linux user 111, 227

AP queue 227

directory entries 228

local area network (LAN) 10

local modification 21

logical partition 177, 206

- activation profiles 212
- host TCP/IP 212
- image activation profile 213
- LOGONBY 61

M

- Managing VSWITCH 64
- mandatory access control (MAC) 8
- Master Key
 - separate set 220
- Master Key (MK) 267
- Maxconn 19
- MDISK s 157
- MDISK statement 14
- multiple Linux
 - image 111, 154
 - instance 184
 - system 145, 234
 - user 178

N

- Name Service Switch (NSS) 138
- native authentication 111, 147, 178
- Network Interface
 - Card 10, 64
- Network Interface Card (NIC) 10
- NOAUTOLOG parameter 44, 162

O

- OpenSSL 232
- OpenSSL speed
 - program 245
- operating system
 - hardware cryptographic resources 6
 - System z hardware 6

P

- password phrase 29, 172
 - built-in rules 48
 - current year 50
 - single quotes 53
- PCICA 2 239
- PCICA count 239
- PCICC 3 239
- PCI-X adapter 205, 207–208
 - logic 207
 - number 208
- PKCS 188, 234
- Print Services Facility (PSF) 72
- Privilege class
 - B 11
 - C 11
 - D 12
 - E 12
 - F 12
 - G 12, 229
- privilege class 11, 51, 229
 - C user 12

- command 12
- G user 12
- PROFILE Exec 27, 121

Q

- QueueStorage 19, 65

R

- RACDCERT 195
- RACF 22
 - RACF 1.9.2 75
 - audit records 75
 - RACF administrator 23, 121, 149
 - attribute 71
 - authority 127
 - following commands 131
 - RACF authorization
 - concept 48
 - mechanism 19
 - RACF data 23, 171, 186
 - RACF database 11, 22, 36, 54, 104, 111, 145, 320
 - exact copy 96
 - internal organization 96
 - LDAPADM2 user 149
 - LINUX1 user 152
 - sequential output 109
 - sharing 157
 - Split/Merge/Extend utility program 97
 - verification utility program 96
 - RACF Permit 47
 - RACF Report Writer 75
 - RACF rule 50
 - RACF Security
 - Server 22
 - RACF Security Server
 - System Programmer 51
 - RACF user
 - entry 196
 - Id 23
 - information 189
 - RACFADU Message 85
 - RACFADU OUTPUT
 - file 86
 - RACVERFY File 97
 - informational screens 101
 - RAICHER DIRECT
 - file 59
 - Filemode 59
 - RDEV 65, 174
 - Red Hat Package Manager (RPM) 203
 - Redbooks Web site 324
 - Contact us xi
 - Remote Spooling Communications Subsystem (RSCS) 67
 - Resource Access Control Facility (RACF) 1, 21, 145
 - REXX exec 50, 169, 311
 - RPIBLPA EXCL0001
 - file 31
 - RPIDIRECT 160, 172

- RPIDIRECT SYSUT1 27
 - file 27, 320
- RPM package
 - xcryptolinz 233
 - xcryptolinzGA 235
- RSCS node 67

S

- SECDATA class 73
 - CATEGORY profile 73
 - SECLEVEL profile 73
- SECLABEL 7, 69
- SECLABEL class 73
- SECLABEL profile 73
- SECLEVEL profile 73
- Secure Socket Layer (SSL) 147, 202
- Secure Sockets Layer (SSL) 6
- security label 7, 72–73
- Security Officer (SO) 256, 258
- security-relevant event 8, 74
- self-signed certificate 195, 203
- server certificate 203
- SETROPTS command 61
- shared file system (SFS) 16, 22
- shared user Id
 - RACF authority 61
- SMF CONTROL
 - Card 81
 - file 27
- SMF Control 27
- SMF data 21
- SMF Record 23
- SSL connection 203
- SSL session 252
- Start Interpretive Execution (SIE) 2–3
- Storage Area Network (SAN) 2–3
- SUSE Linux 138, 231
 - Enterprise Server 10 138
 - SLES 10 233
 - SLES 10 SP1 235
 - SLES10 Service Pack 1 235
- synchronizable object 197
- System Configuration
 - file 15
 - File statement 15
- System z 137
 - adapter 2–3
 - cryptographic adapters 242
 - cryptographic solution 5
 - cryptographic solutions 6
 - cryptography hardware 234
 - distribution 138
 - encryption performance information 248
 - environment 252
 - exclusive environment 268
 - feature 1–2
 - guest 1–2, 6
 - hardware 1–2
 - hardware cryptography environment 6
 - instruction 234

- instruction set 5
- Linux 137
- Linux farm 177
- LPAR capabilities 1–2
- multiple Linux 145
- platform 6
- server 1–2, 146
- specific module 231, 263
- user administration 146
- version 235

T

- Tamper Resistant Security Module (TSRM) 6
- TCPIPUSERID 149
- tdisk space 76
- Tivoli Directory Integrator 184
 - key concept 186
- TKE catcher 268
- TKE workstation 207
 - remote access 268
- Transport Layer Security (TLS) 6, 242
- Trusted Computer System Evaluation Criteria (TCSEC) 8
- Trusted Key Entry
 - PCI-X adapter 207
- Trusted Key Entry (TKE) 207

U

- usage domain
 - index 207
 - index combination 207, 212
 - index number 211
- user Id
 - accounting information 14
- user profile 73
- USER request 23
- USER WITHPASS
 - A0 23
 - A0 rec 23
 - file 23
 - Filemode 27

V

- Virtual Channel-to-Channel Adapter (VCTCA) 9
- virtual machine
 - directory entry 105
 - duplicate entries 315
 - following tasks 37
 - IPL CMS 107
 - processor architecture 14
 - user ID 13
 - virtual environments 4
- virtual machine (VM) 1–2, 21, 36, 155
- virtual network 5, 9
- virtual private network (VPN) 204
- VLAN IDs 15
- VM user 3, 154
- VSWITCH statement 15

vswitch sw00001
 detail 19
 grant Inxsu1 19
 rdev 2e28 19

X

XML file 86
XML-binary Optimized Packaging (XOP) 117, 178

Z

z/OS server
 central IBM Tivoli Directory Server 145
z/VM 5.3.0 21
 system 22
z/VM environment
 hardware encryption 234
 SMF records 75
z/VM System
 Operation Guide 155
z/VM system 4
 central LDAP server 146
 LDAP exchanges 146
 printer 74
 programmer 76
 RACF database 145
z/VM VSWITCH
 operation 19
z90crypt device driver 232, 235
 new version 235
z90crypt driver 235

Archived



Security on z/VM

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



Redbooks®

Security on z/VM

**z/VM in the enterprise
security solution -
Sample scenarios**

**z/VM security
features, LDAP, RACF**

**Cryptography with
Linux guests on z/VM**

Discussions about server sprawl, rising software costs, going green, or moving data centers to reduce the cost of business are held in many meetings or conference calls in many organizations throughout the world. And many organizations are starting to turn toward System z and z/VM after such discussions. The VM operating system has over 40 years of experience as a hosting platform for servers, from the days of VM/SP, VM/XA, VM/ESA and especially now with z/VM. With the consolidation of servers and conservative estimates that approximately seventy percent of all critical corporate data reside on System z, we find ourselves needing a highly secure environment for the support of this infrastructure. This document was written to assist z/VM support and security personnel in providing the enterprise with a safe, secure and manageable environment.

This IBM Redbooks publication provides an overview of security and integrity provided by z/VM and the processes for the implementation and configuration of z/VM Security Server, z/VM LDAP Server, z/OS IBM Tivoli Directory Server and Linux on System z with PAM for LDAP authentication.

Sample scenarios with RACF database sharing between z/VM and z/OS, or through Tivoli Directory Integrator to synchronize LDAP databases, are also discussed in this book.

**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**

SG24-7471-00

ISBN 0738488542