RSH CONSULTING

z/OS System Integrity Protection

February 2024

RSH Consulting - Robyn E. Gilchrist

RSH CONSULTING

RSH Consulting, Inc. is an IT security professional services firm established in 1992 and dedicated to helping clients strengthen their IBM z/OS mainframe access controls by fully exploiting all the capabilities and latest innovations in RACF. RSH's services include RACF security reviews and audits, initial implementation of new controls, enhancement and remediation of existing controls, and training.

- www.rshconsulting.com
- 617-969-9050
- www.linkedin.com/company/rsh-consulting-inc.

Robyn E. Gilchrist is a Senior RACF and CA ACF2 Consultant. She assists clients with evaluation of their z/OS security posture and works with them to enhance their access controls. As a systems programmer and network engineer, Ms. Gilchrist has installed, configured, and maintained the z/OS Communications Server and WebSphere Application Server (WAS) for z/OS in Network Deployment (ND) mode with associated ACF2 and RACF controls. She has converted CPF-connected ACF2 databases to RRSF-connected RACF databases.

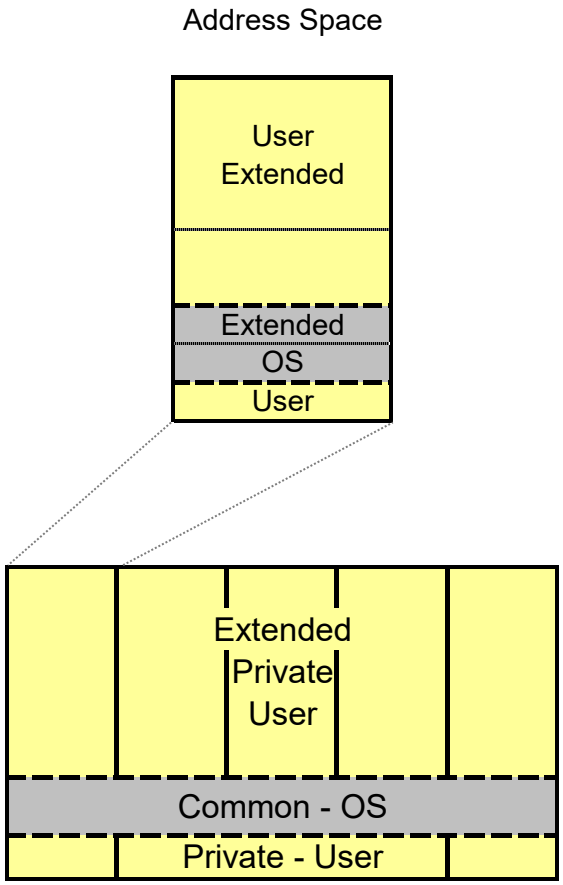
- 617-977-9090
- R.Gilchrist@rshconsulting.com
- www.linkedin.com/in/robyn-e-gilchrist/

Topics

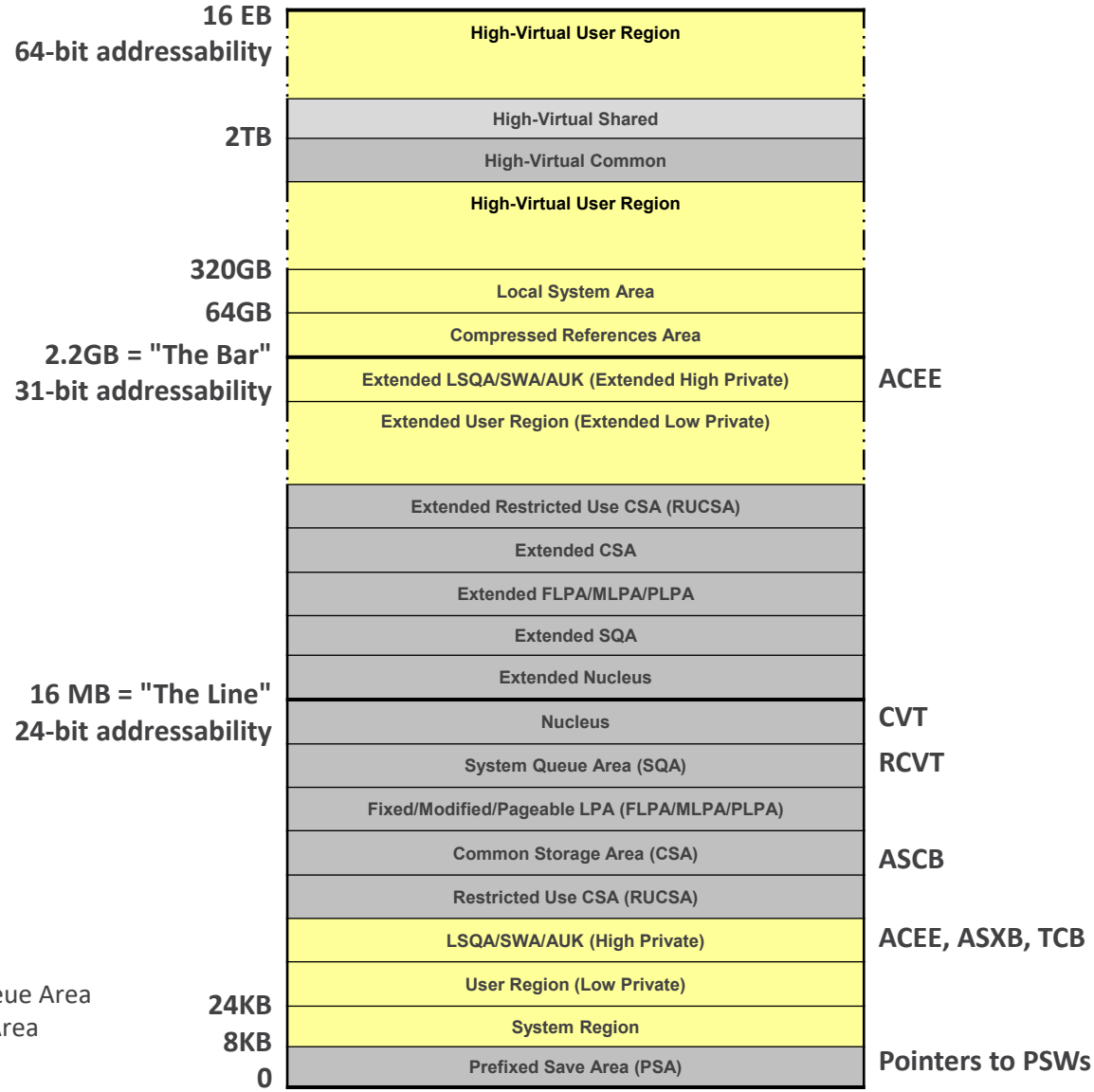
- z/OS Address Spaces
- z/OS System Configuration and Components
- IBM's Integrity Commitment
- Privileges and Authorization
- z/OS Integrity Protection

z/OS Address Spaces

- A z/OS address space is the extent of memory that is addressable by a program
  - "Trimodal addressing" can switch between 24-, 31- and 64-bit mode
- Individual address spaces are assigned to Batch jobs, Started Tasks, and TSO Users
  - z/OS maintains segregation of individual address spaces
  - Allows use of common resources like CPU and memory without disrupting other users
- User perspective is one contiguous virtual memory space
  - Part of memory is Private Area (User)
  - Part of memory is Common Area (OS)
- z/OS perspective
  - One Common Area - shared by all Address Spaces
  - Multiple Private Areas
- z/OS uses storage keys to protect memory from store (write) and fetch (read) operations by unauthorized users and programs
- Attempting to access memory that is not available to your address space usually results in a System 0C4 abend (S0C4 - protection exception)



64-bit z/OS Virtual Storage Map



LSQA - Local System Queue Area
SWA - Scheduler Work Area
AUK - Auxiliary User Key

(map not drawn to scale)

z/OS Control Blocks

- Control blocks are data areas in memory used by the operating system to execute a task
  - An address space can have many tasks running within it
  - Each address space has its own set of control blocks which point to other control blocks to create chains of control blocks
- Some control blocks reside in the common area and can be addressed by all address spaces
  - Prefixed Save Area (PSA) represents virtual storage starting at address 0 and points to the CVT
  - Communication Vector Table (CVT) is the address space interface to the nucleus of the operating system, points to the RCVT and every ASCB in the system
  - RACF Communication Vector Table (RCVT) is the communication area for information global to RACF functions (i.e. maps SETROPTS) and points to the RACF database
  - Address Space Control Block (ASCB) is the anchor for the address space, contains control information and points to the ASXB and all the TCBs running in the address space
- Some control blocks reside in the private area and can only be addressed by the address space where they reside, the operating system or authorized programs
  - Address Space Extension Block (ASXB) contains address space control information and points to an ACEE
  - Task Control Block (TCB) represents a unit of work and can point to an ACEE
  - Accessor Environment Element (ACEE) represents the authorities of a single accessor (RACF USERID) in the address space and is pointed to by an ASXB, a TCB or by a field supplied by the caller
- Browse TSO address space ACEE with ISRDDN command

  BROWSE 224.?+6C?+C8?
  - 224 is the offset in PSA that contains a pointer to the address of the ASCB
  - 6C is the offset in ASCB that contains a pointer to the address of the ASXB
  - C8 is the offset in ASXB that contains a pointer to the address of the ACEE

z/OS System Software

- Nucleus contains the basic supervisor program modules (“kernel”) of z/OS used by the master scheduler
  - Loaded with modules from SYS1.NUCLEUS
    - IEANUC0x - Control Program module
    - IEAVNIP0 - Nucleus Initialization Program
    - Supervisor Call (SVC) Types 1, 2, 6
- Link Pack Area (LPA) is a common area of virtual storage that contains modules used concurrently by multiple users
  - LPA contains
    - SVC Types 3 and 4
    - System product modules like DFP, JES, RACF and corresponding exits
  - LPA is created at IPL by the CLPA parameter specified by the operator or in IEASYSxx
    - Built from SYS1.LPALIB and installation-specified libraries in LPALSTxx PARMLIB member
    - If CLPA is not specified, z/OS tries to find a usable PLPA in existing page datasets
  - LPA search order, first to last
    - Dynamic LPA - PROGxx, dynamic changes to LPA
    - Fixed (FLPA) - IEAFIXxx, not eligible to reside in auxiliary storage (page datasets)
    - Modified (MLPA) - IEALPAxx, modify or update PLPA
    - Pageable (PLPA) - LPALSTxx or PROGxx, eligible to reside in auxiliary storage (page datasets)
- z/OS exit modules can be installed into the nucleus or into LPA

z/OS System Initial Program Load Sequence

- z/OS Initial Program Load (IPL) is a software reboot of the z/OS operating system and is performed by a z/OS operator or their functional equivalent
- Some settings endure for the life of the IPL and some settings can be modified dynamically through the use of operator commands entered at a console
- Specify configuration options in datasets that are read during the IPL sequence
  - SYSx.IPLPARM where x = 0 – 9
    - Use is optional
    - Resides on the same volume as the SYSx.IODFxx dataset, the z/OS Input-Output Definition File (IODF)
  - SYS1.PARMLIB
    - Used if SYSx.IPLPARM dataset not found
- SYSx.IPLPARM members
  - LOADxx - IPL Configuration specifications
    - IODF — IODF dataset suffix and high-level qualifier
    - SYSCAT — Master catalog name, unit and volume
    - SYSPARM — IEASYSxx member to use for initialization parameters
    - IEASYM — Suffix for system symbolics parameter member IEASYMxx
    - PARMLIB — PARMLIB dataset name concatenation
    - NUCLEUS — IEANUC0x Nucleus Specification
    - SYSPLEX — Name of the SYSPLEX in which the system participates
  - NUCLSTxx - Nucleus Customization

IPLPARM LOADxx

```
IODF     99 SYS1
SYSCAT   OS39M1113CCATALOG.ZOS25.MASTER
SYSPARM  CS
IEASYM   00
PARMLIB  USER.PARMLIB                                  OS39M1
PARMLIB  ADCD.ZOSV14S.PARMLIB                          S4RES1
PARMLIB  SYS1.PARMLIB                                  S4RES1
NUCLEUS  1
SYSPLEX  ADCDPL
```

PARMLIB Concatenation Members

IEASYSxx	Member suffixes (directors) and parameters used at IPL
IEASYMxx	System Symbolics
COMMNDxx	Initial Start Commands (SETPROG command, run JCL to execute CLIST/REXX)
CONSOLxx	Console Definitions (MCS, SMCS, HMCS, subsystem and system console)
IEAAPFxx	APF-authorized Libraries (limits system usability, replace with PROGxx)
IEAAPPxx	I/O Appendages (only used if using installation-written EXCP processing)
IEACMDxx	IBM-supplied commands
IEAFIXxx	FLPA Modules and Libraries
IEALPAxx	MLPA Modules and Libraries
IEASVCxx	Installation-defined SVC definitions
IEFSSNxx	Subsystems (e.g., RACF, JES2, OMVS)
IFAPRDxx	Product Enabling Parameters
IKJTSOxx	Authorized TSO Commands
LNKLSTxx	Linklist Library Concatenation (limits system usability, replace with PROGxx)
LPALSTxx	PLPA Concatenation Libraries
PROGxx	Dynamic Libraries (APF, LNKLST, LPA, EXIT)
SCHEDxx	Program Properties Table (PPT)
SMFPRMxx	SMF Parameters
BPXPRMxx	z/OS Unix Parameters

IEASYSxx Entries

```
CLOCK=00,                    SELECT CLOCK00
CLPA,                        CREATE LINK PACK AREA (COLD START)
CMD=CS,                      SELECT COMMNDCS
CON=(00,NOJES3),             SELECT CONSOL00
FIX=00,                      SELECT IEAFIX00, FIX MODULES SPECIFIED
LNKAUTH=LNKLST,              AUTHORIZE LNKLST00, APFTAB IS ALTERNATE
LPA=00,                      SELECT LPALST00, PLPA LIBRARY LIST
MLPA=00,                     SELECT IEALPA00, MLPA PARAMETERS
MSTRJCL=00,                  SELECT MSTJCLEX, MASTER JCL
OMVS=CS,                     SELECT BPXPRMCS
OPI=YES,                     ALLOW OPERATOR OVERRIDE TO IEASYS00
PAGE=(SYS1.PLPA.PAGE,        SPECIFY PAGE DATASET NAMES
      SYS1.COMMON.PAGE)      PLPA & COMMON REQD FOR WARM/HOT START
PROG=(00,73,J3,SI,DB,IF,MS,  SELECT PROGxx members
      SY,LA,LB,DC,MR,MC,LE,LJ,LN,SL,LQ,L9),
SCH=00,                      SELECT SCHED00
SSN=(02,00),                 SELECT IEFSSNxx, SUBSYSTEM NAMES
SYSNAME=PRDA,                SYSTEM NAME (OVERRIDES IEASYMxx)
```

PROGxx Entries

```
APF ADD                          /* Add and delete libraries to   */
 DSNAME(ISM330.SIDIAUTH)         /* APF list                      */
 VOLUME(VTISMD)
                                 /*                               */
EXIT ADD                         /* Add, replace, modify, delete, */
 EXITNAME(IEAVTABX_EXIT)         /* change attributes of system   */
 MODNAME(IDIXDCAP)               /* exits                         */
                                 /*                               */
SYSLIB LINKLIB(VENDOR.LINKLIB)   /* Chg default system datasets   */
SYSLIB MIGLIB(SYS1.MIGLIB)       /* in LNKLST, LPALST             */
SYSLIB CSSLIB(SYS1.CSSLIB)       /* concatenation                 */
SYSLIB LPALIB(VENDOR.LPALIB)
                                 /*                               */
LNKLST ADD NAME(LINKLIST.IPL)    /* Specify LNKLST concatenation  */
 DSN(ISM330.SCAZLINK      )      /* with ordered list of datasets */
 VOLUME(VTISMD)
                                 /*                               */
LPA                              /* Add libs to LPA at end of IPL */
                                 /* Add Del LPA libs after IPL    */
```

System Datasets - Standard Dataset Names

SYS1.NUCLEUS	Nucleus modules, SVCs and z/OS exits
SYS1.CMDLIB	TSO routines
SYS1.DUMPnn	System dumps
SYS1.HELP	Help information
SYS1.LINKLIB	Nonresident system routines, utilities, service aids
SYS1.LPALIB	PLPA resident system routines, SVCs and z/OS exits
SYS1.MACLIB	Macro definitions
SYS1.PARMLIB	z/OS parameters
SYS1.PROCLIB	System JCL Procedures (PROCs)
SYS1.SVCLIB	Appendage modules
SYS1.UADS	TSO logon information
SYS1.MIGLIB	IPCS Routines (LINKLSTed)
SYS1.CSSLIB	Linkage Assist Routines (LINKLSTed)

System Datasets - Installation-Defined Names

PARMLIBs	Additional System Parameter Libraries
LINKLIBs	Additional LINKLST Libraries
LPALIBs	Additional LPA Libraries
IODF	I/O Definition File (Hardware Configuration)
SMF	System Management Facility (SYS1.MANx)
Master/User Catalogs	Dataset location pointers
Page/Swap Datasets	Auxiliary storage location (DASD only)
JES Spool	Batch Input/Output; System Console Log
JES Checkpoint	JES Serialization (JES2)
RACF	Security Database
VTAMLST	VTAM Network Definitions
ZFS	Unix System Services filesystems

Common Address Spaces in z/OS

MASTER	Master Scheduler address space (loaded from SYS1.LINKLIB(MSTRJCLxx))
PCAUTH	Cross Memory Services
TRACE	Problem trace routines
DUMPSRV	Dump Services
GRS	Global Resource Serialization
CONSOLE	Console communication services
ALLOCAS	Allocation Services and data areas
SMF	System Management Facility
LLA	Linklist Lookaside
VLF	Virtual Lookaside Facility
CATALOG	Catalog
SMS	System Managed Storage
TCAS	Terminal Control Address Space (TSO/E Logon Support)
XCFAS	Cross System Coupling Facility
DLF	Data Lookaside Facility (High Performance Batch – HIPERBATCH)
JES2	Job Entry Subsystem
INIT	Initiators
IOSAS	I/O Supervisor, ESCON, I/O Recovery
OMVS	OpenEdition MVS (z/OS Unix System Services)
RACF	RACF Console and RRSF Services

z/OS System Integrity

- IBM Integrity Statement
- Program Status Word
- Program Execution State
- Storage Protection
- Supervisor Call
- Authorized Program Facility
- Program Properties Table
- Integrity Protection

IBM z/OS System Integrity Statement

IBM z/OS® System Integrity Statement

First issued in 1973, IBM's MVS™ System Integrity Statement, and subsequent statements for OS/390® and z/OS, has stood for over three decades as a symbol of IBM's confidence in and commitment to the z/OS operating system.

IBM's commitment includes design and development practices intended to prevent unauthorized application programs, subsystems, and users from bypassing z/OS security – that is, to prevent them from gaining access, circumventing, disabling, altering, or obtaining control of key z/OS system processes and resources unless allowed by the installation. Specifically, z/OS "System Integrity" is defined as the inability of any program not authorized by a mechanism under the installation's control to circumvent or disable store or fetch protection, access a resource protected by the z/OS Security Server (RACF®), or obtain control in an authorized state; that is, in supervisor state, with a protection key less than eight (8), or Authorized Program Facility (APF) authorized. In the event that an IBM System Integrity problem is reported to IBM, IBM will always take action to resolve it in the specified operating environment for releases that have not reached their announced End of Support[1] dates.

IBM's long-term commitment to System Integrity is unique in the industry[2], and forms the basis of z/OS' industry leadership in system security. z/OS is designed to help you protect your system, data, transactions, and applications from accidental or malicious modification. This is one of the many reasons IBM z Systems™ remains the industry's premier data server for mission-critical workloads.

Notes:

1. End of Support dates are the last dates on which IBM will deliver standard support services for a given version or release of a product. Information about end of support dates is available at http://www.ibm.com/software/support/lifecycle/index_z.html
2. IBM reserves the right to change, modify or withdraw its offerings, policies and practices at any time. All products and support obligations are subject to the terms of the applicable license and services agreements.

ZSL03361-USEN-02

IBM

z/OS System Integrity Statement originated in 1973 with the MVS System Integrity Statement
https://www.ibm.com/downloads/cas/OWGOKG40

IBM z/OS System Integrity Statement

- "System integrity" is defined as the inability of any program not authorized by a mechanism under the installation's control to:
  - Circumvent or disable store or fetch protection
  - Access a resource protected by the z/OS Security Server (RACF)
  - Obtain control in an authorized state:
    - Supervisor state
    - Protection key less than eight (8)
    - Authorized Program Facility (APF) authorized
- z/OS integrity and security can be trusted only if
  - z/OS programs and parameters are properly implemented
  - z/OS programs and parameters are protected from unauthorized modification
  - z/OS is not improperly modified or customized
- IBM will always resolve system integrity problems in the specified environment for releases that have not yet reached end-of-support

Program Status Word (PSW)

- The PSW is a 16-byte (128-bit) control structure that governs the status of the system in relation to the currently running program
  - Contains the address of the instruction being executed
  - Contains the condition code associated with the instruction
  - Contains additional information to control instruction sequencing and CPU state
- The PSW problem state bit is used to indicate the CPU execution state
- The PSW key is used for storage references by the CPU
- PSA control block maps the current PSW, the previous PSW and the next PSW

Program Execution State in PSW

- A program executes in one of two states
  - Supervisor state (system) is used to allow a program to access system services
    - Obtain storage or use data in another address space through cross-memory services
    - Execute privileged instructions like PROGRAM CALL (PC)
  - Problem state (user) is how programs run when executing user code and logic
    - Updating a text value with a REXX program or calculating a value with a COBOL program
    - Use a system macro or SVC to request services like I/O or obtaining storage
    - User programs enter the system and normally run in problem state
- PSW bit 15 indicates the state in which the program is executing

z/OS Storage

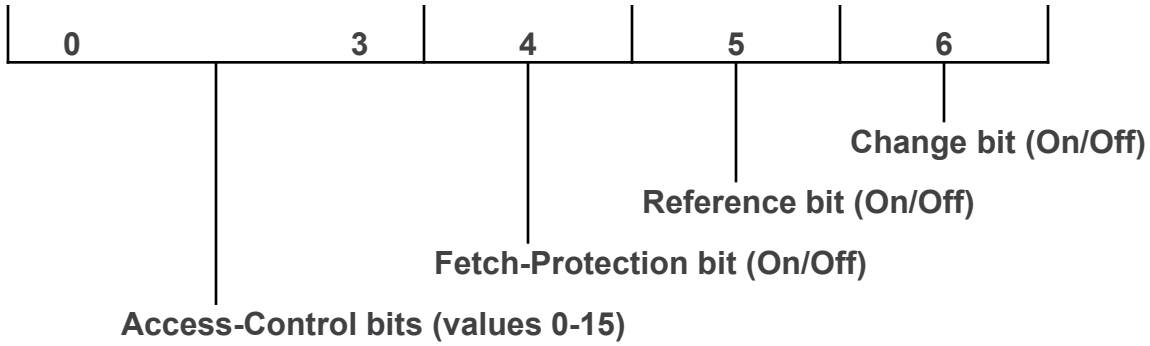
- z/OS memory is referred to as storage
- Storage is unique to a Logical Partition (LPAR)
  - Central or main storage (CSTOR) is real, physical memory
    - Both the program and the data it uses must reside in CSTOR for a program to run
  - Expanded storage is obsolete on z/OS but still used on z/VM
  - Auxiliary storage resides on an external storage device, for example a page dataset
- The smallest manageable unit of storage is a 4K block
  - When describing virtual storage, a 4KB block is called a page
  - When describing real storage, a 4KB block is called a frame
- Dynamic Address Translation (DAT) translates virtual addresses (pages) into real addresses (frames)
  - Allows program virtual storage usage to exceed real storage size
  - Programs typically address storage with virtual addresses
- The private area of virtual storage is where a user program resides and executes

z/OS Storage Protection

- Five z/OS mechanisms prevent storage misuse and destruction
- Key-controlled protection compares the PSW key and the storage key to allow or disallow access to the block
  - Can be overridden by control-register* settings
  - Provides protection against improper storing, or against improper storing or fetching, but not against improper fetching alone
- Access-list controlled protection refers to access-register* mode settings, not RACF access lists
  - A CPU in access-register mode is resolving a virtual address to a real address
  - Added in OS/390 to suppress fetch command execution of fetch protected pages
- DAT protection uses protection bit on virtual storage page-table and segment-table entries to protect against improper storing
  - Replaces page-protection facility from OS/390 and MVS
- Low-address protection protects the first 512 bytes of the first and second 4K block of virtual storage making it unavailable to a program for storing data
  - If operating in ESA/390 mode, only the first 512 bytes of the first 4K block of virtual storage is unavailable to a program for storing data
  - Protects storage used by the CPU during normal processing
- Instruction-execution protection (IEP), when installed, is a zHardware feature that marks allocated storage as non-executable
  - Helps protect from stack overflow and storage misuse
  - Supported on z/OS 2.4 and above, z/OS 2.4 and 2.5 require APAR PH39134
  - Available for CICS TS 6.1 Dynamic Storage Area (DSA) protection on supported hardware with feature toggle com.ibm.cics.sm.iep=true

*Control registers and access registers are programming mechanisms used by user programs and the operating system

Storage Key



- The storage key is a 7-bit control field that is assigned to a block when the storage is acquired
  - Four access-control bits (0 thru 3) on the storage block are matched against the PSW key in PSW bits 8 thru 11
  - Fetch-protection bit (4) indicates whether key-controlled protection applies to fetch-type references
  - Reference bit (5) is set to 1 every time the storage block is referenced for store or fetch
  - Change bit (6) is set to 1 when information is stored in the block

Storage Key Access-Control Bit Assignments

- System Keys
  - 0 Supervisor and system functions
  - 1 JES
  - 2 VSPC (Virtual Storage Personal Computing)
  - 3 Availability Manager (AVM)
  - 4 Reserved
  - 5 Data Management (OPEN, CLOSE, IOS)
  - 6 TCAM or VTAM
  - 7 IMS and DB2 (and others)
- User Keys*
  - 8 - 9 V=V (Virtual) - problem programs - batch jobs, TSO users, started tasks
  - 10 - 15 V=R** (Real)

* Use of user-key (8 - 15) common storage creates a security risk. IBM recommends the elimination of all user-key common storage. See RSH RACF Tips – January 2008
https://www.rshconsulting.com/racftips/RSH_Consulting__RACF_Tips__January_2008.pdf

** V=R was prevalent in the early days of Single Virtual Storage (SVS) and Multiple Virtual Storage (MVS) which were z/OS predecessors. Replaced by DAT, V=R use is discouraged on z/OS systems.

Key-Controlled Storage Protection

- For a program to store data at a storage location, the program must have either
  - PSW key that matches the storage key on the block
  - PSW key 0
  - If access is prohibited, storage location remains unchanged
- If the storage key fetch-protection bit is OFF (0), fetch-type references do not use key-controlled protection
- If the storage key fetch-protection bit is ON (1), fetch-type references use key-controlled protection
  - Program must have either
    - PSW key that matches the storage key on the block
    - PSW key 0
  - If access is prohibited, data is not retrieved and the instruction is suppressed
  - Fetch-protection is determined by the storage subpool used when it is allocated
    - Problem state programs usually allocate private area storage from fetch-protected subpools
    - RACF control block DEXP is the data encryption exit parameter list pointed to by ICHDEX01 and ICHDEX11 and resides in key 0, fetch-protected, high private storage from subpool 229

SVC - Supervisor Call

- A Supervisor Call (SVC) is a user program interface to z/OS system services
  - Consists of system instructions and macros
  - The program runs in supervisor state and key(0) while these functions execute
  - The program reverts to problem state when the SVC completes
- An SVC resides in the nucleus or the link pack area of virtual storage and is loaded at IPL or through the use of operator commands
- Some SVCs are not protected and are available to all users
  - SVC10 (x'0A') Obtain or release private storage below 16MB (GETMAIN/FREEMAIN)
  - SVC19 (x'13') Open a dataset (OPEN)
  - SVC26 (x'1A') Write message-to-operator (WTO)
  - SVC120 (x'78') Obtain or release private storage above 16MB (GETMAIN/FREEMAIN)
- Some SVCs are protected and require APF authorization, supervisor state or PSW key 0 - 7 to use
  - SVC107 (x'6B') Change program state to supervisor state or to PSW Key(0) (MODESET)
  - SVC123 (x'7B') Remove z/OS Service Request Block from dispatch (PURGEDQ)
  - SVC131 (x'7E') Identify a RACF-defined user (RACINIT)

Installation-defined SVCs

- Written for home-grown needs (rare) or part of a vendor package (e.g., CICS)
- Can be defined and installed via:
  - SYS1.PARMLIB(IEASVCxx) entry
    - APF(YES) operand flags the SVC as privileged
  - SVCUPDTE macro
    - Only available to APF-authorized callers

SVC Exposures

- Improperly written SVCs are a source of z/OS integrity exposures when they:
  - Return control to the user in
    - Supervisor State
    - PSW key 0 - 7
  - Accept inappropriate instructions to run for the user while in Supervisor State
  - Intentionally allow security to be bypassed

Authorized Program Facility (APF)

- APF authorization allows a program to
  - Change the PSW key to any value including 0 - 7
  - Change program to supervisor state
  - Execute protected SVCs
- Can use APF authority to change address space control blocks
  - ACEE - Set attributes like SPECIAL on a USERID
  - RCVT - Change system options like disabling auditing
- APF is intended for the operating system and vendor system products when specifically required by documentation

Acquiring APF Authorization

- Program loaded into LPA
  - SYS1.PARMLIB(LPALSTxx) or (PROGxx) specify LPA libraries
  - Libraries dynamically added using the CSVDYLPA macro or SETPROG command
  - Program loaded into MLPA during IPL from a library designated as APF-authorized and listed in SYS1.PARMLIB(IEALPAxx) or (PROGxx)
- Program loaded into memory from a library designated as APF-authorized
  - SYS1.PARMLIB(IEAAPFxx) or (PROGxx) specify APF libraries
  - Dynamically added using the CSVAPF macro or SETPROG command
  - Program must be link-edited with AC=1
- Program loaded into memory from SYS1.LINKLIB or a concatenated library if PARMLIB(IEASYSxx) specifies LNKAUTH=LNKLST
  - SYS1.PARMLIB(LNKLSTxx) or (PROGxx) specify LINKLST libraries
  - Libraries dynamically added using the CSVDYNL macro or SETPROG command
  - Program must be link-edited with AC=1
  - SYS1.CSSLIB and SYS1.MIGLIB are automatically LINKLSTed
  - APF-authorization is nullified if SYS1.PARMLIB(IEASYSxx) specifies LNKAUTH=APFTAB
- TSO command designated as authorized in SYS1.PARMLIB(IKJTSOxx) and loaded into memory from an APF library
- Unix program with the APF Extended Attribute as set in its File Security Packet

Designating APF libraries

- SYS1.PARMLIB(IEAAPFxx)
  - Up to 253 libraries
  - Change via IPL only
  - Older and still viable method but it is not preferred
- SYS1.PARMLIB(PROGxx)
  - No limit to number of libraries
  - Defined as Dynamic -or- Static
  - Dynamic - libraries designation can be changed by console SET command
  - Newer and more favored method
- Can use either or both, but most installation typically only use PROGxx

APF Dataset Entries

- Datasets defined by APF PARMLIB member entries may not exist
  - APF list entries do not indicate dataset existence
  - Entries for non-existent libraries may exist for:
    - Deleted datasets (minor housekeeping issue)
    - Future datasets
    - Datasets that are only available on another system sharing PARMLIB configuration
- Not a concern provided the dataset names are protected by strict profiles

Assigning APF Authorization to a Load Module

- Binder assigns an authorization code (AC) to a load module when it is link-edited (a program creation step that resolves external references)
  - SETCODE control statement on SYSIN DD
  - EXEC PARM= in JCL
- Binder allows AC to be set to a numeric value from 0 - 255
  - System default is AC=0
- APF recognizes load modules with the AC=1 attribute that are loaded from LPA or a library on the APF list as being APF-authorized
  - Any value not AC=1 is considered unauthorized
  - Assigning AC=1 to a module in a library that is not on the APF list does not make the module APF-authorized
- Utilities like AMASPZAP/SUPERZAP can assign AC=1 to a load module
  - Change load module contents post link-edit
  - UPDATE access to the load module dataset is required

APF Program Execution

- First program in a job step must be APF-authorized for the step to be APF-authorized
- To retain APF authorization, APF-authorized programs can only call other programs that come from APF-authorized libraries
  - When a program runs with APF authorization, z/OS prevents the program from accessing other modules that are not in LPA or an APF-authorized dataset
  - If module not found in LPA or an authorized library, system issues abend x'306'

APF Concerns

- LINKLST libraries may contain installation or application programs whose function cannot be trusted
  - LNKAUTH=APFTAB should be coded in SYS1.PARMLIB(IEASYSxx) in lieu of using the default LNKAUTH=LNKLST
- Improperly written APF programs are a source of z/OS integrity exposures when they:
  - Return control to the user in
    - Supervisor state
    - PSW Key (0 - 7)
  - Accept inappropriate instructions to run for the user while in supervisor state
  - Intentionally allow security to be bypassed

Program Properties Table (PPT)

- PPT can grant programs special privileges
  - KEY(*nn*) PSW key other than 8
  - NODSI Dataset Integrity Bypass (Bypass ENQ)
  - NOPASS Bypass Password and RACF Dataset Protection - Started Tasks only
  - NOPASS-BATCH Bypass Password and RACF Dataset Protection - Batch Jobs
- PPT entries are defined in
  - IBM-provided PPT
  - SYS1.PARMLIB(SCHEDxx)
- PPT is reserved for the operating system and vendor system products when specifically required by documentation
  - Some installations set CICS DFHSIP program to NOPASS which is not recommended
- PPT programs assigned a system key (0 - 7) can circumvent security

PPT Considerations

- Program must be loaded into memory from an APF-authorized library as the first program in the job-step
- Program can come from any APF-authorized library
  - PPT specifies program name only
  - Any program matching a PPT entry and loaded from any APF-authorized library will acquire the designated privileges
  - AC=1 is not part of selection criteria used by PPT
- May contain entries for non-existent programs
  - JES3 program IATINTK in JES2 installation
  - Not a concern for IBM entries
  - May be minor housekeeping issue for installation-defined entries
- IBM advises against giving any entries NOPASS-BATCH
- Improperly written programs in PPT can be the source of an integrity exposure

Exits and Appendages

- Exits are installation-written subroutine modules that are called (i.e., executed) by z/OS or a system software product
  - Exit modules can be dynamically added, removed, or replaced using the CSVDYNEX macro if the exit is defined to the dynamic exit facility (e.g., RACF exits IRREVX01 and IRRVAF01)
- Appendages (rare)
  - Installation-written I/O exit
  - Can be used by non-authorized programs if defined in SYS1.PARMLIB(IEAAPPxx)
  - Stored in SYS1.SVCLIB
- Exits and appendages run as extensions of whatever module calls them - same State and Storage Protect Key
- Exits and appendages can be written to enhance or circumvent access controls

TSO Commands

- PARMLIB
  - Can update TSO parameters, including authorized commands and programs
  - Does so only installing new IKJTSOxx PARMLIB member
  - Protected by TSOAUTH class resource PARMLIB
    - READ View active TSO parameters - Permit to RACF Staff
    - UPDATE Replace TSO parameters - Permit to TSO Tech Support Staff
- TESTAUTH
  - Test APF-authorized programs
  - Protected by TSOAUTH class resource TESTAUTH
  - Only permit access in isolated Systems Test environments (a.k.a., Sandbox systems)

Dynamic Storage Modification

- System Management Products
  - IBM OMEGAMON
  - Broadcom(CA) Sysview
  - Rocket Software(ASG) TMON for z/OS
- Designed to examine and modify memory
  - Allowing changes to be made with these products is a potential exposure

z/OS Integrity Protection

- Strictly limit and log UPDATE access to system datasets, especially:
  - SYSx.IPLPARM
  - PARMLIB
  - LINKLIB
  - LPA
  - APF
  - SVC
- Strictly limit and log access to functions that allow dynamic changes to the configuration of the system
  - OPERCMDS
    - MVS.SET.PROG SET command Load replacement PARMLIB member
    - MVS.SETPROG SETPROG command Update APF, LNKLST, LPA and dynamic exits
  - TSOAUTH - PARMLIB command - Load replacement IKJTSOxx member
  - FACILITY
    - System configuration macros whose resource names are prefixed CSVLLA, CSVAPF, CSVDYLPA, CSVDYNEX, and CSVDYNL
    - Unix APF extended attribute assignment - BPX.FILEATTR.APF

z/OS Integrity Protection

- Restrict and log use of memory-modifying software tools
- Strictly limit SURROGAT userid.SUBMIT access to IDs with any of the aforementioned access authorities
- Use blocking permissions to restrict UPDATE access by OPERATIONS user
- Limit Storage Admin authority to storage administrators and support staff
- Generate and review daily reports that identify improperly protected datasets related to system integrity
- Generate alerts for dynamic system changes that affect system integrity
- Examine APF and SVC programs for possible coding errors that could open integrity exposures (Rocket z/Assure Vulnerability Analysis Program (VAP))
- Implement RACF class ACEECHK to detect and prevent unauthorized privilege escalation ACEE modifications

References

- z/Architecture Principles of Operation (POP)

https://www.ibm.com/support/pages/zarchitecture-principles-operation

- z/OS MVS Data Areas Volumes 1 - 4

https://www-40.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R5ga320935/$file/iead100_v2r5.pdf

- z/OS Security Server RACF Data Areas

https://www-40.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R5ga320885/$file/ichc400_v2r5.pdf

- Introduction to the New Mainframe: z/OS Basics

https://www.redbooks.ibm.com/redbooks/pdfs/sg246366.pdf

- z/OS 2.5 MVS Initialization and Tuning Reference

https://www-40.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R5sa231380/$file/ieae200_v2r5.pdf

- z/OS 2.5 MVS Diagnosis: Reference

https://www-40.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R5ga320904/$file/ieav200_v2r5.pdf