The World of Program Control
&
PADS

Walt Farrell

z/OS Security Server Design

wfarrell@us.ibm.com

Trademarks

- The following are trademarks or registered trademarks of the International Business Machines Corporation in the United States, other countries, or both:
  - IBM
  - RACF
  - z/OS, OS/390

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

Agenda

- Concepts
- Simple Program Control
- PADS (Program Access to Data Sets)
- Before z/OS R4
- z/OS R4 in BASIC Program Security Mode
- EXECUTE Control
- z/OS R4 ENHANCED Program Security Mode
- Effects on PADS, EXECUTE, & UNIX
- PADS
- Migration

Basic Concepts

- Program:
  - ➢ A compiled program in load module or program object format
  - ➢ Loaded from LPA, LINKLIST, or private library (JOBLIB, STEPLIB)
  - ➢ NOT:
    - o CLIST, REXX exec, Java, PERL, etc.
    - o Program loaded from the UNIX file system (secured and processed differently)
- Environment:
  - ➢ job step in a batch job, started task, or started job
  - ➢ TSO session
  - ➢ Also, a TSO command, CLIST, or REXX exec invoked by TSOEXEC or IKJEFTSR
  - ➢ UNIX address space

Basic Concepts...

- Clean Environment
  - ➤ An environment (see preceding list) in which all programs that have run are:
    - o defined in the PROGRAM class; or
    - o loaded from LPA; or
    - o loaded from the UNIX file system and marked as program controlled by

      - extattr +p ...

- Required for some (not all) program control functions
  - ➤ PADS (program access to data sets)
  - ➤ EXECUTE
  - ➤ UNIX server/daemon functions
- Generally easier to maintain in batch or STC or UNIX
- Harder to maintain in TSO, especially for ISPF

Basic Concepts...

- PADS (Program Access to Data Sets)
  - ➤ Allows access to a data set only when a user is running a particular program

- EXECUTE
  - ➤ An access level less than READ
  - ➤ Used in DATASET or PROGRAM access lists
  - ➤ Prevents user from seeing content of a program

Basic Concepts...

Program Security (PGMSECURITY) Modes

- BASIC
  - ➢ Default mode in z/OS R4
  - ➢ Only mode before z/OS R4

- ENHANCED
  - ➢ Optional, new, higher security mode for z/OS R4
  - ➢ More resistant to malicious users and hackers
  - ➢ Enabled using new FACILITY profile IRR.PGMSECURITY
  - ➢ Supports a WARNING mode to help with migration

Simple Program Control (Basic or Enhanced)

- Functions:
  - ➤ restrict selected users / groups from running a program
  - ➤ audit use of programs
- Define program via RDEFINE PROGRAM
- Permit READ to allow; NONE to restrict
- Format:

  RDEFINE PROGRAM name +

  ADDMEM('library.name'[/[volser][/[NO]PADCHK]]

  UACC(NONE or READ)

  - ➤ volser may be omitted
    - o or an actual volser
    - o or '******' to represent the IPL volume
- WARNING will not have any effect

Simple Program Control (Basic or Enhanced)...

- Example:
  ```
  RDEFINE PROGRAM DELUSER +
    ADDMEM('SYS1.LINKLIB'//NOPADCHK) UACC(READ)

  PERMIT DELUSER CLASS(PROGRAM) +
    ID(group1) ACCESS(NONE)
  ```

- If program has an alias, protect it, too:
  ```
  RDEFINE PROGRAM DU +
    ADDMEM('SYS1.LINKLIB'//NOPADCHK) UACC(READ)

  PERMIT DU CLASS(PROGRAM) +
    ID(group1) ACCESS(NONE)
  ```

Simple Program Control (Basic or Enhanced)...

Recommendations:

➢ Omit the volser when defining programs

➢ Usually specify NOPADCHK

➢ Use READ or NONE. Avoid EXECUTE unless

- o program contains a sensitive algorithm; or
- o program contains sensitive data

➢ Restrict UPDATE access to libraries that contain protected programs, just as for APF libraries

Simple Program Control (Basic or Enhanced)...

Notes:

➤ Program name in RDEFINE can have * at end

➤ ABC* protects all programs with names beginning ABC... that are in the library named in the ADDMEM

➤ Profiles are discrete, not generic. Some differences:

     o RLIST PROGRAM ABCD will not display ABC*

     o A* may be more specific than ABC* depending on ADDMEM value

Simple Program Control (Basic or Enhanced)...

More Notes:

- ➤ ADDMEM can specify multiple members
  - o allows protection for copies of program in other libraries
- ➤ all ADDMEM operands for a PROGRAM profile should have same PADCHK or NOPADCHK specification
- ➤ Protected programs can reside in
  - o public libraries: Libraries in the system LINKLIST
  - o Private libraries
    - – Not in the system LINKLIST
    - – Accessed by JOBLIB, STEPLIB, ISPLLIB, etc. or by TSO command CALL 'library.name(program)'
  - o Cannot reside in LPA. You can define them, but system does not check user's authority to run them

Simple Program Control (Basic or Enhanced)...

More Notes:

➤ SETR CLASSACT(PROGRAM) has no effect

    o Use SETR WHEN(PROGRAM) to activate program control

    o Use SETR WHEN(PROGRAM) REFRESH after changes

➤ Consider how the library that contains the program is protected

    o Usually allow READ to the program libraries via DATASET profiles and READ to the programs via PROGRAM profiles

    o Most programs operate on data; protect the data (example: IEBGENER, AMASPZAP)

    o Programs needing APF won't work if user copies them to another library

    o Programs using PADS won't work if user copies them to another library

Simple Program Control (Basic or Enhanced)...

More Notes:

➤ ACCESS(NONE) can work for some program libraries:
  o Programs accessed via system LINKLIST
  o Programs run under TSO as commands or via
       CALL *(program)
➤ But not for:
  o STEPLIB, JOBLIB,etc
  o CALL 'library(program)'

Program Control by System ID (Basic or Enhanced)

- Function: Restricts access to programs based on user / group and system
  - Uses the (usually) 4-character SMF ID and conditional access lists
- Example:
  - Assume program ABC in library ABC.LOAD
  - You have two systems, PROD and TEST
  - Most Users should only run ABC on the TEST system

```
RDEFINE PROGRAM ABC +
     ADDMEM('ABC.LOAD'//NOPADCHK) UACC(NONE)
PERMIT ABC CLASS(PROGRAM)  +
  ID(*) ACCESS(READ) WHEN(SYSID(TEST))
```

Clean Environments (Basic or Enhanced)

- Users need a clean environment for:
  - ➢ PADS (Program Access to Data Sets: WHEN(PROGRAM(...))
  - ➢ EXECUTE access to PROGRAMs or libraries
- UNIX daemons and servers need a clean environment if you have FACILITY BPX.DAEMON defined
- Without this requirement, users could easily bypass security controls if you use PADS or EXECUTE

- To create a clean environment for a user: Ensure that all programs the user runs are:
  - ➢ Defined by PROGRAM profiles; or
  - ➢ Loaded from the LPA
- Problem: How to do that?

Clean Environments (Basic or Enhanced)…

One method: Figure out each program the user needs and
define a separate PROGRAM profile for it.

Problem:
- Very difficult to figure out
- Many profiles
- Massive administrative overhead

Clean Environments (Basic or Enhanced)...

Recommended method:

- RDEFINE PROGRAM ** UACC(READ)
- ADDMEM at least:

```
'SYS1.LINKLIB'//NOPADCHK,
'SYS1.MIGLIB'//NOPADCHK,
'SYS1.CMDLIB'//NOPADCHK,
'cee.version.SCEERUN'//NOPADCHK,
'tcpip.SEZALINK'//NOPADCHK,
'tcpip.SEZATCP'//NOPADCHK,
'ftp.userexits'//NOPADCHK,
'db2.DSNLOAD'//NOPADCHK,
'db2.DSNEXIT'//NOPADCHK
```

Clean Environments (Basic or Enhanced)...

Recommended method (continued):

RDEFINE PROGRAM ICHDSM00 +

    ADDMEM('SYS1.LINKLIB'//NOPADCHK) UACC(NONE)

RDEFINE PROGRAM IRRDPI00 +

    ADDMEM('SYS1.LINKLIB'//NOPADCHK) UACC(NONE)

- These programs check for a controlled environment as an alternative to AUDITOR or SPECIAL, so you need to ensure that only appropriate users can run them.

Clean Environments (Basic or Enhanced)...

Notes:

- UACC(READ) is recommended, and appropriate. With PROGRAM ** you are merely defining everything to keep the environment clean.
  - ➢ ID(*) ACCESS(READ) is not the same as UACC(READ)
  - ➢ In z/OS R4, or with APAR OW50327, RACF will use UACC(READ) when loading from SYS1.LINKLIB using PROGRAM ** or PROGRAM *
- For actual protection of programs, use separate PROGRAM profiles with desired UACC and access list
- Potential problems still in TSO. After users run their own programs,
  - ➢ May need to use TSOEXEC to use PADS or EXECUTE
  - ➢ Or logoff and logon again

PADS (pre-R4, R4 BASIC mode)

- PADS (Program Access to Data Sets) allows access to data only when a user is running a particular program
  - The specified program provides an extra layer of security by controlling what the user does with the data. It
    - o Restricts what the user can read, omitting some data
    - o Restricts what the user can write, validating the user's data
- Terminology: the program mediates the user's access to the data
- PADS uses WHEN(PROGRAM(xyz)) as a conditional access list entry

PADS (pre-R4, R4 BASIC mode)...

Example 1: User runs program XYZ, and you want to allow READ access to a data set:

    ADDSD 'some.data.set.profile' UACC(NONE)

    PERMIT 'some.data.set.profile' ID(*) +

        ACCESS(READ) WHEN(PROGRAM(XYZ))

Notes:

➢ Environment must be clean

➢ Specify exact program name;   PROGRAM(*) not allowed

➢ If other non-LPA programs are active in the environment they must be

   o defined with NOPADCHK; or

   o added to the conditional access list with their own WHEN(PROGRAM(...))

PADS (pre-R4, R4 BASIC mode)...

Example 2: Users run program XYZ1, and XYZ1 invokes (LINK, ATTACH) XYZ2. You want to allow READ access to a data set:

- Before z/OS R4:
  - ➤ PERMIT 'some.data.set.profile' ID(*) +

    ACCESS(READ) WHEN(PROGRAM(XYZ2))

- With z/OS R4:
  - ➤ Either the same as before z/OS R4

    or (simpler)
  - ➤ PERMIT 'some.data.set.profile' ID(*) +

    ACCESS(READ) WHEN(PROGRAM(XYZ1))

PADS (Any Release or Mode)

- PADS works for granting READ and UPDATE
- PADS generally does not work for ALTER (data set creation or deletion)
  - The user's program is not running during allocation / deallocation, which happen before / after the user's program runs
  - Exception: Dynamic Allocation

- Example: User executes program ABC and you want to allow creation of data set ABC.DATA

```
//stepname EXEC PGM=ABC
//DD1         DD    DSN=ABC.DATA,DISP=(NEW ...
```

You can not use

```
PERMIT 'ABC.DATA' GENERIC ID(user1) +
    ACCESS(ALTER) WHEN(PROGRAM(ABC))
```

EXECUTE Control (Pre-R4, and R4 BASIC mode)

- Purpose: Use when you must prevent the user from copying or viewing a program
  - ➤ Generally because the program
    - o Contains sensitive data
    - o Contains sensitive algorithms
- Use ACCESS(EXECUTE) or UACC(EXECUTE)
  - ➤ For DATASET profile protecting a private (non-LINKLIST) library; or
  - ➤ For PROGRAM profile that protects one or more programs

- User needs at least EXECUTE authority to run a program; READ is easier to setup; Avoid EXECUTE if possible

- Use of EXECUTE requires a clean environment
  - ➤ For EXECUTE with a DATASET profile, ensure all programs in the data set have PROGRAM profiles

EXECUTE Control (Pre-R4, and R4 BASIC)...

- **EXECUTE on PROGRAM profile:**
  - ➢ Useful for programs in LINKLIST

  - ➢ Specify ACCESS(NONE) on DATASET profile to prevent user from OPENing data set to copy or view the program

  - ➢ Specify ACCESS(EXECUTE) on specific PROGRAM profile for the program to
    - o prevent dumps via SYSUDUMP, SYSABEND
    - o require loading into a clean environment (to prevent viewing from an active user program)

EXECUTE Control (Pre-R4, and R4 BASIC)...

- **EXECUTE on DATASET profile:**
  - ➤ Most useful for private (non-LINKLIST) libraries

  - ➤ ACCESS(NONE)　would prevent user from accessing the programs via JOBLIB, STEPLIB, ISPLLIB, etc.

  - ➤ Allows user to OPEN the library but
    - o Prevents user from accessing the library except via LINK, LOAD, XCTL, ATTACH, or other supervisor-state processing

  - ➤ Thus user cannot copy the program, or view it via Browse, etc.

z/OS R4: ENHANCED Program Security Mode

- z/OS R4 offers two program security (PGMSECURITY) modes that affect PADS and EXECUTE
  - BASIC -- Functions like pre-R4 (but with PADS enhancement)
  - ENHANCED -- Better security and resistance to hackers and malicious users
- ENHANCED mode offers a WARNING option for migration
  - Performs all checks as though in ENHANCED mode
  - If checks fail, but would have worked in BASIC mode
  - Issues messages, creates SMF record
  - Allows function to continue successfully

z/OS R4: ENHANCED Program Security Mode...

- FACILITY profile IRR.PGMSECURITY controls the mode
  - ➢ BASIC:
    - o IRR.PGMSECURITY not defined; or
    - o RDEFINE FACILITY IRR.PGMSECURITY APPLDATA('BASIC')
  - ➢ ENHANCED:
    - o RDEFINE FACILITY IRR.PGMSECURITY APPLDATA('ENHANCED')
  - ➢ ENHANCED with WARNING
    - o RDEFINE FACILITY IRR.PGMSECURITY APPLDATA('anything else')
- RACF does not validate the APPLDATA value during RDEFINE or RALTER, but inspects it during SETR WHEN(PROGRAM) [REFRESH] processing

z/OS R4: ENHANCED Program Security Mode...

- SETROPTS LIST shows the mode:

    WHEN(PROGRAM -- BASIC)
    WHEN(PROGRAM -- ENHANCED)
    WHEN(PROGRAM -- ENHANCED WARNING)

z/OS R4: ENHANCED Program Security Mode...

ENHANCED mode supports 3 kinds of PROGRAM definitions:

- BASIC:
  - ➢ A PROGRAM profile protecting 1 program name (no * at end) with APPLDATA('BASIC')

- MAIN:
  - ➢ A PROGRAM profile protecting 1 program name (no * at end) with APPLDATA('MAIN')

- Normal:
  - ➢ Any PROGRAM profile with a * in the name, or with any other APPLDATA value

- RACF does not validate APPLDATA value during RDEFINE or RALTER

z/OS R4: ENHANCED Program Security Mode...

- ENHANCED mode affects PADS and EXECUTE processing
  - PADS and EXECUTE work only if
    - o a program with the MAIN attribute established the environment
    - o the first program executed in the current task or a parent task, has the BASIC attribute
  - Also affects UNIX server and daemon processing if you define FACILITY profile BPX.MAINCHECK
    - o Server and daemon functions work only if a MAIN program established the environment
    - o May require copying some programs from UNIX file system to load library

z/OS R4: ENHANCED Program Security Mode...

- PADS Example 1: The user executes program ABC via JCL

  ```
  //  EXEC PGM=ABC
  ```

  or via TSO

  ```
  TSOEXEC ABC
  ```

  or

  ```
  TSOEXEC CALL *(ABC)
  ```

- You want to allow the user to read data set ABC.DATA while running ABC

- You can define ABC as a MAIN program

  ```
  RDEFINE PROGRAM ABC +
      ADDMEM('load.library'//NOPADCHK) APPLDATA('MAIN')
  PERMIT 'ABC.DATA'  ID(*) ACCESS(READ) +
      WHEN(PROGRAM(ABC))
  ```

z/OS R4: ENHANCED Program Security Mode...

- PADS Example 2: The user executes program ABC in TSO:

    CALL 'load.library(ABC)'

  or

    ISPEXEC SELECT PGM(ABC)

- You want to allow the user to read data set ABC.DATA while running ABC

- ABC will not create the environment (IKJEFT01 did that) so you cannot define ABC as a MAIN program. Instead, define it as BASIC

  - ➢ RDEFINE PROGRAM ABC +

    ADDMEM('load.library'//NOPADCHK) APPLDATA('BASIC')

  - ➢ PERMIT 'ABC.DATA' ID(*) ACCESS(READ) WHEN(PROGRAM(ABC))

z/OS R4: ENHANCED Program Security Mode...

- z/OS R4 Security Administrator's Guide contains more examples and scenarios
- Avoid use of BASIC where possible; MAIN is more secure
  - For some TSO (or other) users, MAIN cannot work, so for those cases you must use BASIC.
- Most controlled programs should not have (or need) MAIN or BASIC attributes
- New ICETOOL report will show programs with MAIN or BASIC
- Auditors may want to question
  - use of BASIC PGMSECURITY mode
  - use of programs with BASIC attribute
- However, some cases require either
  - BASIC attribute or
  - Not using PADS or EXECUTE

Migration to ENHANCED PGMSECURITY Mode

First figure out which programs need MAIN or BASIC attributes

- Examine conditional access lists via IRRDBU00 0402 records
  - For each program found, determine whether users actually execute it, or some other program
  - Also determine intended method of execution (batch, TSO) and for TSO whether user can use TSOEXEC
  - This will give initial list of MAIN and BASIC candidates
- Examine IRRDBU00 records 0400, 0402, 0404 to find EXECUTE-controlled data sets
  - Examine programs in those data sets to see which need MAIN or BASIC, based on intended usage
- Similarly, examinine IRRDBU00 records 0500, 0505 to find EXECUTE-controlled programs
- Then RDEFINE new specific PROGRAM profiles or use RALTER to add APPLDATA('MAIN') or APPLDATA('BASIC') as needed

Migration to ENHANCED PGMSECURITY ...

- Next activate warning mode:
  - ➤ RDEFINE FACILITY IRR.PGMSECURITY APPLDATA('ENHWARN')
    - o Any APPLDATA value except BASIC or MAIN gives Enhanced-Warning mode
  - ➤ SETR RACLIST(FACILITY) REFRESH if needed
  - ➤ SETR WHEN(PROGRAM) REFRESH

- z/OS R4 now in ENHANCED-WARNING mode
  - ➤ has no effect on pre-R4 systems
  - ➤ has no effect on jobs, STCs, TSO sessions that are already running
- Watch messages and collect SMF records
- Refine list of PROGRAM profiles that need MAIN or BASIC
- IPL at least once to test jobs, STCs, and TSO users who were active when you enabled ENHANCED-WARNING
- Repeat as needed

Migration to ENHANCED PGMSECURITY ...

- Finally, switch to ENHANCED mode
  - ➢ RALTER FACILITY IRR.PGMSECURITY APPLDATA('ENHANCED')
  - ➢ SETR RACLIST(FACILITY) REFRESH if needed
  - ➢ SETR WHEN(PROGRAM) REFRESH

- Again, does not affect running jobs, STCs, or TSO users
- Watch for any problems
- If problems, fix by adjusting PROGRAM profiles or IRR.PGMSECURITY profile

- IPL to make change effective for all jobs, STCs, and TSO users

Migration to ENHANCED PGMSECURITY ...

- Finally, switch to ENHANCED mode
  - ➤ RALTER FACILITY IRR.PGMSECURITY APPLDATA('ENHANCED')
  - ➤ SETR RACLIST(FACILITY) REFRESH if needed
  - ➤ SETR WHEN(PROGRAM) REFRESH

- Again, does not affect running jobs, STCs, or TSO users
- Watch for any problems
- If problems, fix by adjusting PROGRAM profiles or IRR.PGMSECURITY profile

- IPL to make change effective for all jobs, STCs, and TSO users

Questions?

The World of Program Control
&
PADS

Walt Farrell

z/OS Security Server Design

wfarrell@us.ibm.com