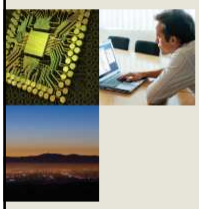

z/OS Unix Auditing

Mark S Hahn
IBM Corp

October 2, 2009
Southern California RACF User Group

Disclaimer

The information contained in this document is distributed on as "as is" basis, without any warranty either express or implied. The customer is responsible for use of this information and/or implementation of any techniques mentioned. IBM has reviewed the information for accuracy, but there is no guarantee that a customer using the information or techniques will obtain the same or similar results in its own operational environment.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used. Functionally equivalent programs that do not infringe IBM's intellectual property rights may be used instead. Any performance data contained in this document was determined in a controlled environment and therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM Products, programming or services in your country.

IBM retains the title to the copyright in this paper as well as title to the copyright in all underlying works. IBM retains the right to make derivative works and to republish and distribute this paper to whomever it chooses.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- z/OS
- RACF

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

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

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

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

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- z/OS Unix Services
  - find command
  - UNIX files vs z/OS datasets
  - chaudit command
  - Other servics
- RACF
  - Userids
  - Classes
  - Profiles
  - Datasets
  - Where to find more information

Using the UNIX 'find' command

- find can search for files using all sorts of criteria
  - file type
  - user and group ownership
  - presence of ACLs
  - presence of specific ACL entries
  - file permissions (including set-uid/set-gid bits)
  - audit settings
- use find and shell command substitution
  - **setfacl -m g:racftest:rwx $(find /u/bruce -acl_group racfdev)**
- See UNIX Command Reference

5

UNIX Audit Classes

Class	SETROPTS AUDIT	SETROPTS LOGOPTIONS
FSOBJ	Creation and deletion of all file system objects	Access to files
DIRACC	N/A	Read/write access to directories
DIRSRCH	N/A	Search access to directories
FSSEC	N/A	Changes to security data of all file system objects
PROCESS	Dubbing and undubbing of processes	Changes to process identity (UID and GID)
PROCACT	N/A	Functions that inspect (e.g. getpsent) or update (e.g. kill, ptrace) other processes
IPCOBJ	Creation and deletion of InterProcess Communication objects	Access to IPC objects, and changes to permissions and ownership

6

DSMON Status Check

```
RACF DATA SECURITY MONITOR
                              R A C F     C L A S S     D E S C R I P T O R
CLASS
NAME            STATUS        AUDITING        STATISTICS
--------------------------------------------------------------------------------
DIRAUTH         INACTIVE        YES               NO
. . .
DIRSRCH         INACTIVE        YES               NO
. . .
FSOBJ           INACTIVE        YES               NO
FSSEC           ACTIVE          YES               NO
IPCOBJ          INACTIVE        YES               NO
PROCACT         ACTIVE          YES               NO
PROCESS         INACTIVE        YES               NO
```

7

SMF Type 80 Records

- One event code per UNIX function
- Use SMF Unload (IRRADU00) to report
- Contains:
  - Audit function code of calling service
  - UID and GID (in addition to user ID and Group)
  - Indicator if authority granted due to superuser
  - Indicator if user running with default UNIX identity
  - Much, much more depending on event code

8

Auditing UNIX Files: compared with data sets

DATASET auditing	UNIX file auditing
SETROPTS LOGOPTIONS for DATASET class controls access logging	SETROPTS LOGOPTIONS for FSOBJ, DIRACC, and DIRSRCH classes contols access logging
SETROPTS AUDIT(DATASET) audits profile creation/deletion	SETROPTS AUDIT(FSOBJ) audits file creation/deletion
SETROPTS AUDIT(DATASET) audits changes to RACF profiles	SETROPTS LOGOPTIONS for FSSEC audits changes to file owner, permission bits and audit settings
Profile-level auditing can be specified by profile OWNER (AUDIT option of ALTDSD)	File-level auditing can be specified by file owner (chaudit command)
Profile-level auditing can be specified by auditor (GLOBALAUDIT option of ALTDSD)	File-level auditing can be specified by auditor (chaudit command with -a option)

9

Auditing UNIX Files: compared with data sets ...

DATASET auditing	UNIX file auditing
LOGOPTIONS with ALWAYS and NEVER overrides profile settings	same for file settings
LOGPTIONS with SUCCESSES or FAILURES merged with profile-level settings	same for file settings
LOGOPTIONS with DEFAULT uses the profile-level settings	same for file settings
Default profile setting is READ failures for owner options, and no settings for auditor options (implies UPDATE, CONTROL, and ALTER failures too)	Default is read, write, and execute failures for owner settings (note that UNIX permissions are not hierarchical - these are separate settings for each access type)
Display profile options with LISTDSD	Display file options with ls -W

10

chaudit Command: Setting File-level Auditing Options

- Audit successful write access to a file
  - chaudit w+s myfile
- Audit all access to a file
  - chaudit +sf myfile
- Set auditor audit bits to audit all attempts to execute a program
  - chaudit -a x+sf myprog
- Audit all write and execute accesses to set-uid files
  - **chaudit x+sf,w+sf $(find / -perm -4000)**

11

Output of ls (list files) Command

```
# ls -W
total 192
-rw-r--r--  --- ---   1 BPXROO  2001     ...  Odyssey
--wx--S---  --- ---   1 ACE     SYS1     ...  Program2
-r-         -aa --    1 BPXROO  KNIGH    ...  SetuidPgm
drwxr-xr-   fff ---   2 BPXROO  SYS1     ...  TestDirectory
-rwxr----t  --- --a   1 ACE     JESTER   ...  prog1
-rwxr-x--   --- ---   2 BPXROO  SYS1     ...  rac
lrwxrwxr    fff ---   1 BPXROO  SYS1     ...  racSymlink ->
-rwxr-x--   --- ---   2 BPXROO  SYS1     ...  raclink
-rwxr-x---  --- ---   1 BPXROO  SYS1     ...  racp
-rw-r--r--  -S- ---   1 1969    SYS1     ...  woodstock
```

owner audit settings

auditor audit settings

f = failures
s = successes
a = all (successes and failures)

12

File System Security Reporting - HFS Unload!!!

- Reports on HFS security data like IRRDBU00 reports on RACF profile data
- Creates Type 900 record for each file
  - currently-mounted file systems only
- Creates Type 90*n* record for each ACL entry
- Runs as UNIX command, or from batch
  - irrhfsu /etc > HfsuOutFile
  - irrhfsu -f //BRWELLS.HFSU.OUTPUT
    /u/brwells/dir1 dir2/subdir

HFS Unload *(continued)*

- UIDs mapped to user IDs and GIDs mapped to group names for your convenience

0900	file name	i-node	uid	user id	gid	group name	set uid	set gid	sticky bit	owner read	owner write	owner execute	group read	etc ...

0901	file name	i-node	Entry type	Uid or GID	user id or group name	read	write	execute

Basic file

Access ACL entry

- Type 902/903 mapped same as 901
  - 902 - File default ACL entry
  - 903 – Directory default ACL entry

Get it at: http://www-1.ibm.com/servers/eserver/zseries/zos/racf/goodies.html

User Review

- Objective: document how many humans, who and why, have uid(0)?
- Review STARTED profiles, looking for TRUSTED and PRIVILEGED
- Review default started task userid STARTED (* or **) – is it uid(0)?
  [RLIST STARTED ** STDATA NORACF
  LU userid NORACF OMVS]

```
 STARTED    SMS.* (G)
STDATA INFORMATION
------------------
USER= STRTASK
GROUP= SYS1
TRUSTED= YES
PRIVILEGED= NO
TRACE= NO
```

```
USER=IBMBMH1

OMVS INFORMATION
----------------
UID= 0000128600
HOME= /u/ibmbmh1
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
```

15

DSMON TRUSTED / PRIVILEGED Check

```
RACF DATA SECURITY MONITOR
                          R A C F    S T A R T E D    P R O C E D U R E S
FROM PROFILES IN THE STARTED CLASS:
--------------------------------------------------------------------------------
PROFILE                ASSOCIATED  ASSOCIATED
NAME                   USER        GROUP       PRIVILEGED  TRUSTED  TRACE
--------------------------------------------------------------------------------
SMFSAVW.* (G)          STRTASK     SYS1        NO          NO       NO
SMF8.* (G)             STRTASK     SYS1        NO          NO       NO
SMS.* (G)              STRTASK     SYS1        NO          YES      NO
SMSGTF*.* (G)          STRTASK     SYS1        NO          NO       NO
SMTP*.* (G)            SMTPU       SYS1        NO          YES      NO
```

16

FACILITY Class: BPX.DAEMON

- Serves two purposes
  - Upgrades z/OS Unix security to z/OS level
    - Requires PROGRAM profiles for all authorized programs
  - Grants daemon privileges to READ users
    - IBM recommendation: the only person to have BPX.DAEMON access should be systems programmer responsible for restarting daemons.
    - Daemon privileges include changing uid to any person's uid without requiring their password
- [RLIST FACILITY BPX.DAEMON ALL]

```
 OMVS      READ      000000
 PTKTCHK   READ      000000
 SYSPROG   READ      000000
 IBMBLU2   NONE      000000
```

17

FACILITY Class: BPX.SERVER

- Serves two purposes
  - Switch to z/OS security if present (should be)
  - Based on READ or UPDATE authority, authorization path to be taken (server + client, client only)
- [RLIST FACILITY BPX.SERVER ALL]

```
 USER      ACCESS
 ----      ------
 WEBSRV    UPDATE
 BPXOINIT  READ
 IBMBLU2   READ
 C2RSRV#P  READ
```

18

FACILITY Class: BPX.SUPERUSER

- First alternative to uid(0)
- Superuser status “on demand”
- Some processes (e.g. SMP/E will accept in lieu of uid(0))
- [`RLIST FACILITY BPX.SUPERUSER ALL`]

```
USER      ACCESS   ACCESS COUNT
----      ------   ------------
BPXOINIT  READ        000000
OMVS      READ        000000
SYSPROG   READ        000000
IBMBTZ1   READ        000000
IBMBMC1   READ        000000
```


FACILITY Class: BPX.DEFAULT.USER

- Default user/group for those needing uid/gid without an OMVS segment
  - Access list ignored
  - Only used if OMVS segment needed
  - Partial / broken OMVS segment blocks its use
  - Use clearly visible value ‘99999’ or such
- For users needing OMVS segment for “general” service: ftp, etc
  - Not a good idea if your users use the shell and own files
- [`RLIST FACILITY BPX.DEFAULT.USER ALL`] and inspect the APPLDATA

```
INSTALLATION DATA
-----------------
UNIX DEFAULT.USER MUST BE DISCRETE

APPLICATION DATA
----------------
CRUNIXU/CRUNIXG
```

```
USER=CRUNIXU

OMVS INFORMATION
----------------
UID= 0000099999
HOME= /nonExistingForDefaultUser
PROGRAM= exit
CPUTIMEMAX= NONE
```

```
INFORMATION FOR GROUP CRUNIXG

OMVS INFORMATION
 ----------------
 GID= 0000099999
```

FACILITY Class: BPX.SAFFASTPATH

- Trigger profile
  - If present, successful UNIX file accesses are not logged to SMF
  - Valuable during system maintenance
  - Requires SET OMVS=xx to activate, null member okay
- [`RLIST FACILITY BPX.SAFFASTPATH ALL SEARCH CLASS(FACILITY) MASK(BPX.)`]

```
BPX.DAEMON
BPX.DEFAULT.USER
BPX.DEFAULTUSER
BPX.FILEATTR.APF
BPX.FILEATTR.PROGCTL
BPX.FILEATTR.SHARELIB
BPX.SAFFASTPATH
BPX.SMF
BPX.STOR.SWAP
BPX.SUPERUSER
BPX.WLMSERVER
BPX.FILEATTR.* (G)
BPX.SERVE* (G)
```


FACILITY Class: BPX.FILEATTR.*

- Authorization to issue z/OS Unix specific command: *extattr*
- Command sets extended authorization attributes on program files including program control and APF (Authorized Program Facility)
- Review who is authorized to use command

[`RLIST FACILITY BPX.FILEATTR.APF ALL`]

[`RLIST FACILITY BPX.FILEATTR.PROGCTL ALL`]

[`RLIST FACILITY BPX.FILEATTR.SHARELIB ALL`]

UNIXPRIV Class: SUPERUSER

- Preferred means of granting superuser privileges (over BPX.SUPERUSER over uid(0))
- Failures commonly not logged
  – attempt to allow an operation, not explicit violation
- Designed to allow granular superuser privileges
  - SUPERUSER.FILESYS
  - SUPERUSER.FILESYS.**
    - CHOWN, MOUNT, ACLOVERRIDE, CHANGEPERMS
  - SUPERUSER.PROCESS.**
    - GETPSENT, KILL, PTRACE
- [`RLIST UNIXPRIV * ALL`]

```
CHOWN.UNRESTRICTED
SUPERUSER.FILESYS.CHOWN
SUPERUSER.PROCESS.GETPSENT
SUPERUSER.PROCESS.KILL
```


UNIXPRIV Class: SHARED.IDS

- Special profile
  - Triggers suppression of duplicate uid / gid
    - If RACF database restructured to IRRIRA00 Stage 2 or 3
  - Authorizes use of SHARED keyword on AU/ALU/AG/ALG command if user has READ
- Most common shared uid? 0
- Profile included in RLIST command output from previous slide
- Recommended – if only as insurance policy against accidental attempts to share uid

SURROGAT Class: BPX.SRV.userid

- Allows *su* command to switch to userid without requiring password for new userid (if issuer has READ access) – normally issuer must supply new userid's password
- Carefully review users authorized to switch without password
- Usage can be audited (APAR OA18016)
- [`RLIST SURROGAT BPX.SRV.* ALL`]

25

PROGRAM Class: **

- PROGRAM profiles help define controlled programs – needed by daemons, servers and APF users
- Can list singular programs
  - Should restrict access to: IRRDPI00, ICHDSM00, IEHINITT using separate discrete profiles
- PROGRAM ** acceptable
  - Preferred over PROGRAM * (okay if present)
- Daemons may fail if profiles not defined
- Review libraries listed –
  - Must be current / remove obsolete data set names
  - Should not be user libraries – authorized exception
- [`RLIST PROGRAM * ALL`]

26

DATASETS: parmlib

- Generally SYS1.PARMLIB, could be other dataset in parmlib sequence
- Issue [`D PARMLIB`] operator command for list of dsns
  - Sequence important, as is protection of dsn where BPXPRMxx members found
- BPXPRMxx members
  - Specified by OMVS= keyword
  - SET OMVS=xx operator command
  - SETOMVS command does NOT reference parmlib

parmlib(BPXPRMxx)

- Pairs of parameter members recommended
  - One for system limits and parameters
  - One for file system definitions
- Empty member advantageous
  - Select options require SET OMVS=xx to activate – null member works (e.g. BPX.SAFFASTPATH activation)

parmlib(BPXPRMxx)

- Review ROOT and FILESYSTEM statements
  - H/zFS (xFS) data sets should be system owned, not user owned
    - OMVS kernel need not be TRUSTED if authorized to xFS datasets
  - SMS restriction lifted
  - Consider multiple system xFS files:
    protection from runaway logging or other process
    - ROOT (mountpoint '/')
    - ETC (mountpoint '/etc')
    - TMP (mountpoint '/tmp') or better TFS
      Temporary file system – storage resident, non-persistent data
  - Consider automount for user filesystems (still system owned) – not audit requirement
  - [LD DA('xxx.yyy') ALL]
    for all datasets named in all BPXPRMxx members and for parmlib datasets housing the BPXPRMxx members

Summary

- z/OS Unix has native tools,
  RACF has controls affecting Uz/OS Unix.
- Alternative data sources include the DSMON, TSO commands, the RACF Data Base Unload (IRRDBU00) or other products
- Much of the z/OS UNIX security resides in RACF – via profiles
- Collect and review RACF protections to ensure access to sensitive features is controlled (using RACF SEARCH, RACF RLIST, zSecure Admin, zSecure Audit, or 3rd party tools)

Thank You

Thank You