

SHARE
in Orlando 2015




z/OS UNIX Security Overview

Session 17618


August 10, 2015

Eric Rosenfeld


rosenfel@us.ibm.com




#SHAREorg




SHARE is an independent volunteer-run information technology association that provides **education, professional networking and industry influence.**

Copyright (c) 2015 by SHARE Inc.  Except where otherwise noted, this work is licensed under <http://creativecommons.org/licenses/by-nc-sa/3.0/>



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.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Trademarks



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

- z/OS
- RACF
- AIX

* 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.

SOLARIS is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries

Mac OS is a trademark of Apple Inc.

* 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.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



3

Agenda



- What is UNIX?
- What is a "UNIX user" on z/OS?
- UNIX identity uniqueness
- Creating UNIX users
- The UNIX file system
- File permissions and access control lists
- UNIX superusers
- UNIX daemons

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



4

What is UNIX?



- File system
- Shell and utilities (commands)
- APIs



`cd /usr/lpp/tivoli`
`open(/u/bruce/file)`

z/OS

Solaris

Mac OS X

???

AIX

HP/UX

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando **2015**



5

UNIX User and Group Registry: AKA RACF!



USER Profile

BASE
TSO
CICS
...
OMVS
UID
HOME
PROGRAM
CPUTIMEMAX
FILEPROCMA
...

GROUP Profile

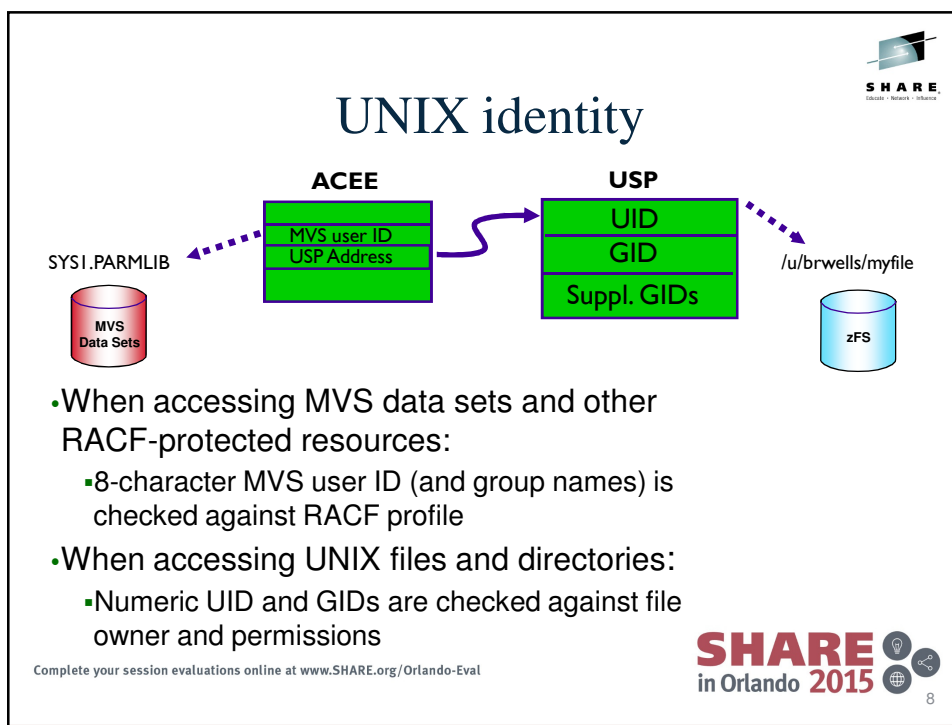
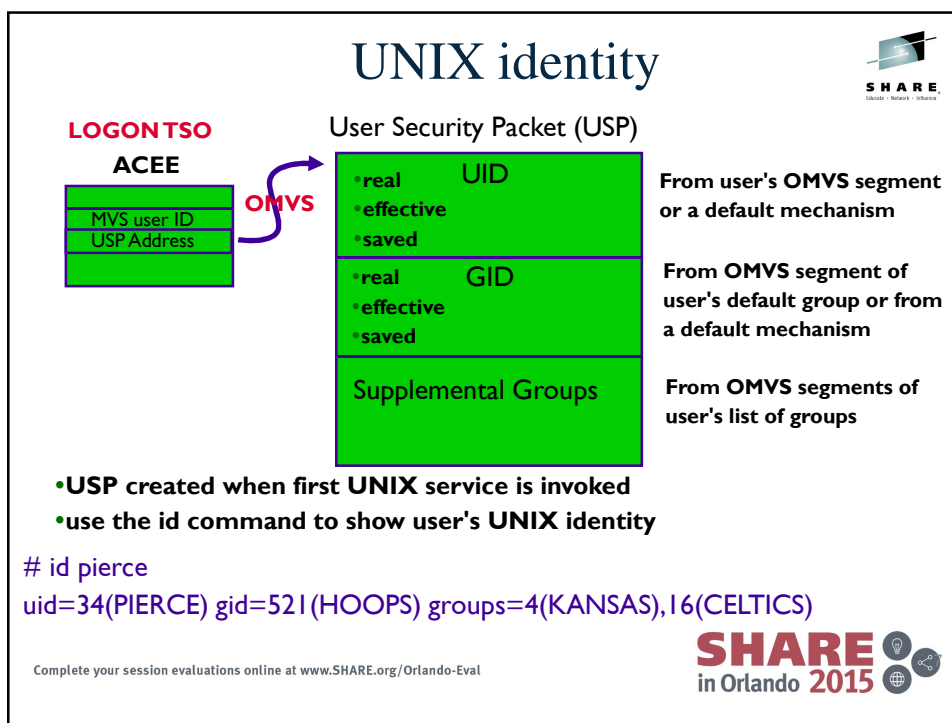
BASE
DFP
...
OMVS
GID

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando **2015**



6



How do I create a UNIX user?



- Option 0 – Do nothing. User cannot use UNIX.
- Option 1 – Using ALTUSER/ALTGROUP, assign OMVS segments and explicit UID/GID values
 - ALTUSER MARK OMVS(UID(88) HOME(...) ...)
 - ALTGROUP HISDFLT OMVS(GID(300))
- Option 2 - Use the default (BPX.DEFAULT.USER)
 - Not recommended – **No longer supported after z/OS V1R13!**
- Option 3 – Tell RACF to generate the xID
 - ALTUSER MARK OMVS(AUTOUID HOME(...) ...)
 - ALTGROUP MARKGRP OMVS(AUTOGID)
- Option 4 – Let the system generate the xID without you asking
 - BPX.UNIQUE.USER

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



9

But before we show you the mechanics



- A brief word on doing it securely
- By default, z/OS does not require or enforce that UIDs and GIDs be unique
- But you should

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



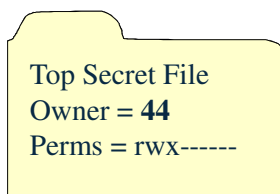
10

Keep UID/GIDs unique – Why?



ADDUSER BILLB OMVS(UID(44))

create



ALTUSER TOMC OMVS(UID(44))

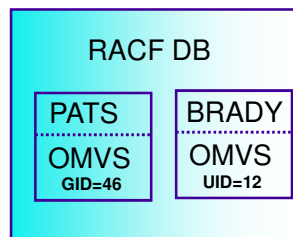
Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Prevention of shared IDs ... SHARED.IDS



- RDEFINE UNIXPRIV SHARED.IDS
UACC(NONE)
- SETROPTS RACLIST(UNIXPRIV)
REFRESH
- ADDUSER MARCY OMVS(UID(12))
IRR52174I Incorrect UID 12. This value is already in use by BRADY.
- ADDGROUP ADK OMVS(GID(46))
IRR52174I Incorrect GID 46. This value is already in use by PATS.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Back to the mechanics



- We've already shown options 0 and 1
- Now on to option 2, which you should not be using
- To be followed by some better options

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



13

Default UNIX User and Group identity



- BPX.DEFAULT.USER in the FACILITY class can be used to assign default OMVS segment data
 - **RDEFINE FACILITY BPX.DEFAULT.USER APPLDATA('DFTUSER/DFTGROUP')**
 - **ADDUSER DFTUSER OMVS(... ..) NOPASSWORD**
 - **ADDGROUP DFTGROUP OMVS(GID(nnn))**
- Assigned when user/group doesn't have an OMVS segment
- Can be overridden on a per-user basis
 - **ALTUSER BOB OMVS(NOUID)**
- Use of default identity is always audited
- Should have only limited use
 - **TCP/IP from MVS to MVS, or, just getting your feet wet with UNIX System Services**



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



14



Automatic UID/GID Assignment

- AUTOUID keyword in the OMVS keyword of the ADDUSER and ALTUSER commands
- AUTOGID keyword in the OMVS keyword of the ADDGROUP and ALTGROUP commands
- Derived values are guaranteed to be unique



ADDUSER MELVILLE OMVS(HOME(/u/melville) AUTOUID)

IRR52177I User MELVILLE was assigned an OMVS UID value of 4646.

ADDGROUP WHALES OMVS(AUTOGID)

IRR52177I Group WHALES was assigned an OMVS GID value of 105.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

13



Automatic UID/GID Assignment ... BPX.NEXT.USER

- Uses APPLDATA of **BPX.NEXT.USER** profile in the FACILITY class to derive candidate UID/GID values
- APPLDATA consists of 2 qualifiers separated by a forward slash (/)
 - left qualifier specifies starting UID value, or range
 - right qualifier specifies starting GID value, or range
 - qualifiers can be null, or specified as 'NOAUTO', to prevent automatic assignment of UIDs or GIDs

RDEFINE FACILITY BPX.NEXT.USER APPLDATA('10000-100000/500-50000')

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



16

Automatic OMVS segment assignment – BPX.UNIQUE.USER



- Define FACILITY profile BPX.UNIQUE.USER, and optionally a user profile in APPLDATA field:

```
RDEFINE FACILITY BPX.UNIQUE.USER
[APPLDATA('USER01')]
```

- If this profile exists, the BPX.DEFAULT.USER profile is not considered.
- For a user or group without an OMVS segment, the service will create one and store a unique UID or GID in it for permanent use.
- If a user name is specified in APPLDATA, its other OMVS fields are copied to the target user when the new UID is saved.
- Uses BPX.NEXT.USER along with AUTOxID (and also requires SHARED.IDS)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Guidelines



- If you only want to allow UNIX functions to users you bless
 - Use AUTOUID/AUTOGID
- If you want to open up the system so any user can use UNIX
 - Use system-assigned OMVS segments

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



And now an abrupt transition

- Into file system security
- We're not quite done with identity-related issues though
- We'll come back to superusers and daemons in a little bit



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



19

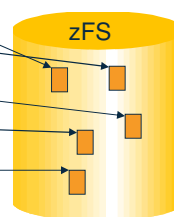
Files and directories

File

Dear Sir,
Blah blah blah,
Yada yada
yada, etc.

Directory

Name	inode
File1	
Dir1	
Dir2	
File2	
Dir3	



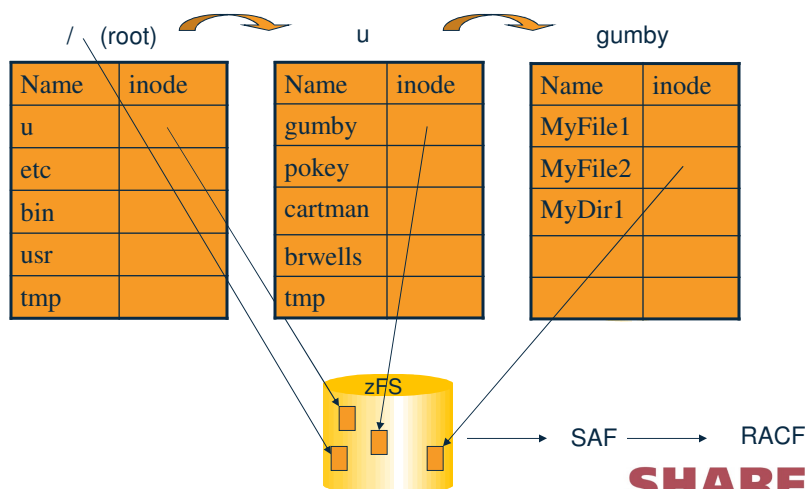
Complete your session evaluations online at www.SHARE.org/Orlando-Eval



20

Directory Search (a.k.a. lookup)

/u/gumby/MyFile2



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

21

Default file permissions and the umask command


- Files are created with different permission settings, depending on the command or application
- file mode creation mask (umask) defends user against permissive defaults
- Display umask
 - octal format: `umask 0077`
 - symbolic format: `umask -S u=rwx,g=,o=`
- Set umask so group and other write bits cannot be set during file creation
 - `umask g-w,o-w`
 - usually done from `/etc/profile`, and `.profile`

Command	Permissions
OPUT	600
touch	666
redirection ('>')	666
oedit	700
mkdir	777

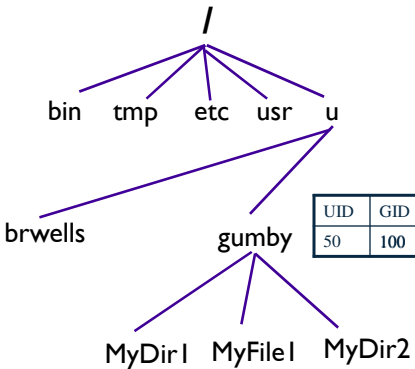
Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

22

 **SHARE**
Secure - Network - Influence

Initialization during file creation



```

graph TD
    root["/"] --- bin
    root --- tmp
    root --- etc
    root --- usr
    root --- u
    u --- brwells
    u --- gumby
    gumby --- MyDir1
    gumby --- MyFile1
    gumby --- MyDir2

```

UID	GID
75	200

UID	GID	Perms
50	100	rwX r-X ---

```

mkdir /u/gumby/MyDir2


rwx rwx rwx


rwx r-x ---
umask: 000 010 111

```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

 23

 **SHARE**
Secure - Network - Influence

Access Control Lists (ACLs)


- Each entry (max 1024) specifies a user (UID) or group (GID) and its allowable permissions
- Displayed/modified with getfacl/setfacl cmds
- Enabled with SETROPTS CLASSACT(FSSEC)
- Support inheritance

Top Secret
Superbowl Pool

User	Bob	r--
User	Boss	---
Group	Admins	rw-
Group	Execs	rwX
Group	Progs	rwX

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

 24

File Access Control with ACLs

Permission Bits

ACL
coi
cns
ett
sr
sol

OWNER	GROUP	OTHER
rwX	rwX	rwX
User1 rwX	Group1 rwX	IF no access, check SUPERUSER.FILESYS
User2 rwX	Group2 rwX	
Usern rwX	Groupn rwX	

IF FSSEC class active

See [z/OS RACF Security Administrator's Guide Appendix F](#) for detailed list of steps

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015



ACL Inheritance

- Can establish default (or 'model') ACLs on a directory
- Get automatically applied to new files/directories created within the directory
- Separate default used for files and subdirectories
- Reduces administrative overhead

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015



ACL Inheritance ...

`mkdir /u/bruce/projectX`

`oedit /u/bruce/projectX/status - status`

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

Using search permission to hide subdirectories

Denying search (lookup) authority on a given directory prevents traversal through that directory, and thus prevents access to sub-objects, regardless of their permission settings.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

Using the FSACCESS class



- Allow Security Administrator to control access to a zFS file system container (data set) using RACF profiles
- Provides very coarse-grained control without needing to use UNIX semantics
 - Works sort of like the search permission example on the previous slide, but only on file system boundaries
- Introduced in z/OS V1R13 with:
 - RACF APAR OA35973
 - SAF APAR OA35974
 - USS APAR OA35970
- Not documented in R13 publications. See APAR documentation:
 - <ftp://public.dhe.ibm.com/eserver/zseries/zos/racf/pdf/oa35973.pdf>

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



29

Using the FSACCESS class



- Mount point traversal triggers one-time check to the container
 - Access failure prevents any operation within file system, regardless of permission bits, acls, file ownership, or UID(0)
 - Successful access (or no covering profile) simply continues with existing UNIX-style checks which may or may not allow access to file system object
 - RACF AUDITOR attribute bypasses FSACCESS check
- UPDATE access required to (new) FSACCESS class resource name which equals the containing data set name
 - Only performed if FSACCESS class is active
- Example:
 - RDEFINE FSACCESS OMVS.ETC.HFS UACC(NONE)
 - PERMIT OMVS.ETC.HFS CLASS(FSACCESS) ID(SYSPROGS) ACCESS(UPDATE)
 - SETROPTS CLASSACT(FSACCESS) RACLIST(FSACCESS)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



30

Now that we've seen UNIX files



- Let's talk about highly privileged UNIX users
- They can do almost anything with those files

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



31

UNIX Superuser



- A superuser is defined as
 - UID 0, any GID
 - Trusted or privileged, any UID, any GID
- A superuser can (by default):
 - Pass all z/OS UNIX security checks
 - Affect any UNIX process on the system
 - Use setrlimit to increase system limits
 - Change his identity to another user

• **Do anything!**

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



32

Just to drive home the point



- A UID(0) user can:

```
#define _POSIX_SOURCE
#include <unistd.h>
#include <stdio.h>
int main(int argc, char *argv[])
{
    FILE *stream;
    int c;
    setuid(295); /* Known UID of powerful z/OS user */
    stream = fopen("//'PAYROLL.DATA'", "r")
    printf("The contents of 'PAYROLL.DATA' are:\n");
    while ((c=getc(stream)) != EOF)
        putc(c, stdout);
    fclose(stream);
    return;
}
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



33

Limit Superuser Privilege By



- Not assigning UID(0) to humans. Instead
- Use BPX.SUPERUSER (not good enough)
OR
- Use UNIXPRIV resources (preferred)
- But if you must assign UID(0), define BPX.DAEMON to prevent unauthenticated identity switches

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



34

UNIXPRIV Class Resources

- Used to assign subset of SUPERUSER authority to a user
- Goal: principle of least privilege
- Partial list of functions you can grant:
 - ability to read or write any HFS file
 - ability to change file ownership
 - ability to change file permissions/ACLs
 - ability to send signals to any process
 - ability to mount/unmount file systems

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



35

Controlling Daemons ... z/OS UNIX-Level Security

- Activated by defining FACILITY BPX.DAEMON
- Restricts the use of unauthenticated identity changing services
- Only trusted daemons should be given authority
- The daemon address space must be kept clean
 - If a program that is NOT a controlled program is loaded, the address space is marked dirty and cannot perform daemon activities
- Clean environment ensures daemons perform their intended function

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



36

Controlling Daemons ... z/OS UNIX-Level Security



- All programs loaded must be controlled
 - PROGRAM profiles covering all programs from MVS libraries (UACC READ is OK)
 - 'sticky' bit on file executable defers to MVS
 - Controlled attribute for programs from the HFS
 - Set with *extattr +p*
 - Issuer needs authority to BPX.FILEATTR.PROGCTL (UID 0 does not grant authorization for extattr!)
 - Turned off automatically if file is changed
 - Ignored if HFS mounted with *noisetuid* or *nosecurity*

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Good Sources of Information: UNIX



- UNIX System Services web site, at <http://www-03.ibm.com/servers/eserver/zseries/zos/unix/>
- UNIX System Services Planning manual (for your release)
 - Available online at <http://www-03.ibm.com/systems/z/os/zos/library/bkserv/v2r1pdf/#BPX>
 - mvs-oe mailing list (see the Forums link at the UNIX web site above for information)
- Check program product documentation for daemon or server security setup

Complete your session evaluations online at www.SHARE.org/Orlando-Eval





Good Sources of Information: RACF

- RACF Auditor's Guide
 - UNIX auditing classes
- RACF Macros and Interfaces
 - SMF 80 formats and SMF Unload mappings
- RACF Security Administrator's Guide
 - Chapter on z/OS UNIX security

<http://www-03.ibm.com/systems/z/os/zos/library/bkserv/v2r1pdf/#ICH>

- RACF web page – irrhfsu and presentations

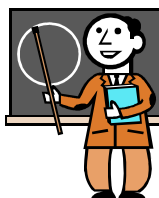
<http://www-03.ibm.com/servers/eserver/zseries/zos/racf/>

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



39

Appendix: Supplementary material



Complete your session evaluations online at www.SHARE.org/Orlando-Eval



40

initialized to ...	File security info	changed by ...
effective UID	User (UID) owner	chown command
parent dir's group	Group (GID) owner	chown or chgrp
varies by function (qualified by umask)	Permission bits	chmod command
	Owner rwx	
flags specified by open()	Group rwx	chmod command
	Other rwx	
read, write, and execute failures	Flags	chmod command
	set-uid	
no auditing	set-gid	chmod command
	sticky	
read, write, and execute failures	Owner audit options	chaudit command
	read	
no auditing	write	chaudit -a command
	execute	
no auditing	AUDITOR audit options	chaudit -a command
	read	
no auditing	write	chaudit -a command
	execute	
SHAREAS bit on for executable files	Extended attributes	extattr command
contents of parent's default ACL	Access Control List	setfacl command
SECLABEL of covering dataset	Security label	chlabel command

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

41

Security Field	Required authority
Owning UID	<ul style="list-style-type: none"> •UID 0 •File owner if CHOWN.UNRESTRICTED is defined in the UNIXPRIV class •READ access to UNIXPRIV profile SUPERUSER.FILESYS.CHOWN
Owning GID	<ul style="list-style-type: none"> •UID 0 •Owner, if a member of new group •File owner if CHOWN.UNRESTRICTED is defined in the UNIXPRIV class •READ access to UNIXPRIV profile SUPERUSER.FILESYS.CHOWN
File mode (permissions and flags) and ACL	<ul style="list-style-type: none"> •UID 0 •File owner •READ access to UNIXPRIV profile SUPERUSER.FILESYS.CHANGEPERMS
Security Label	•RACF SPECIAL
Owner audit options	<ul style="list-style-type: none"> •UID 0 •File owner
Auditor audit options	•RACF AUDITOR
Extended attributes	READ access to FACILITY class profile named: <ul style="list-style-type: none"> •APF - BPX.FILEATTR.APF •Program control - BPX.FILEATTR.PROGCTL •shared library - BPX.FILEATTR.SHARELIB

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

42

Auditing UNIX Files: compared with data sets



<u>DATASET auditing</u>	<u>UNIX file auditing</u>
SETROPTS LOGOPTIONS for DATASET class controls access logging	SETROPTS LOGOPTIONS for FSOBJ, DIRACC, and DIRSRCH classes controls 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)

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

43

Auditing UNIX Files: compared with data sets ...



<u>DATASET auditing</u>	<u>UNIX file auditing</u>
LOGOPTIONS with ALWAYS and NEVER overrides profile settings	same for file settings
LOGOPTIONS 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 (implies UPDATE, CONTROL, and ALTER failures too) failures for owner options, and no settings for auditor options	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

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SHARE
in Orlando 2015

44