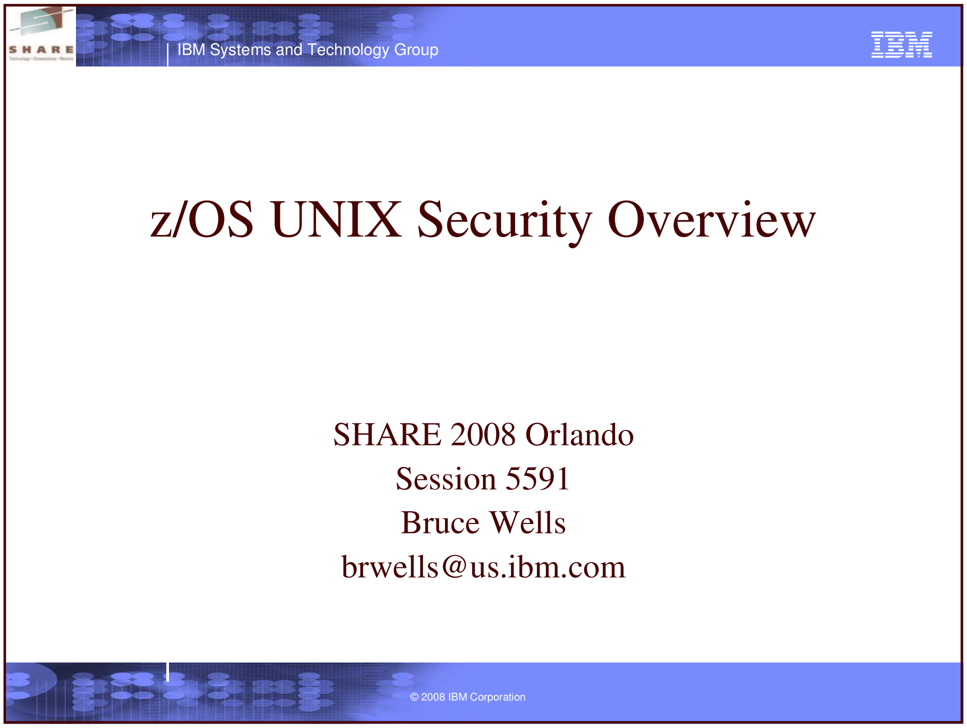
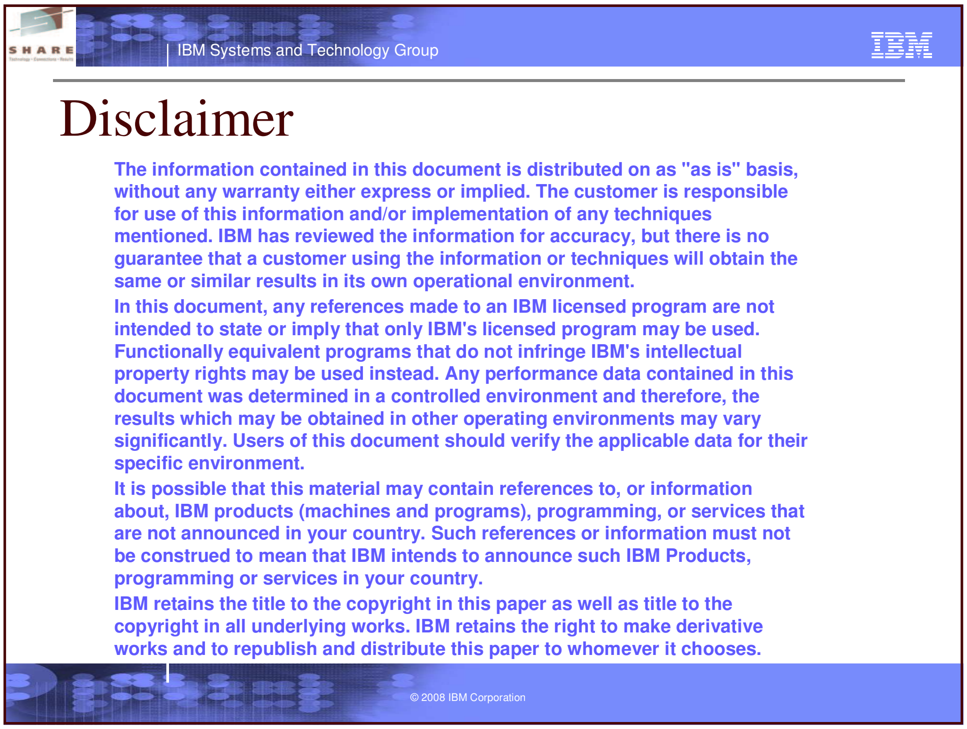


z/OS UNIX Security Overview

SHARE 2008 Orlando
Session 5591
Bruce Wells
brwells@us.ibm.com



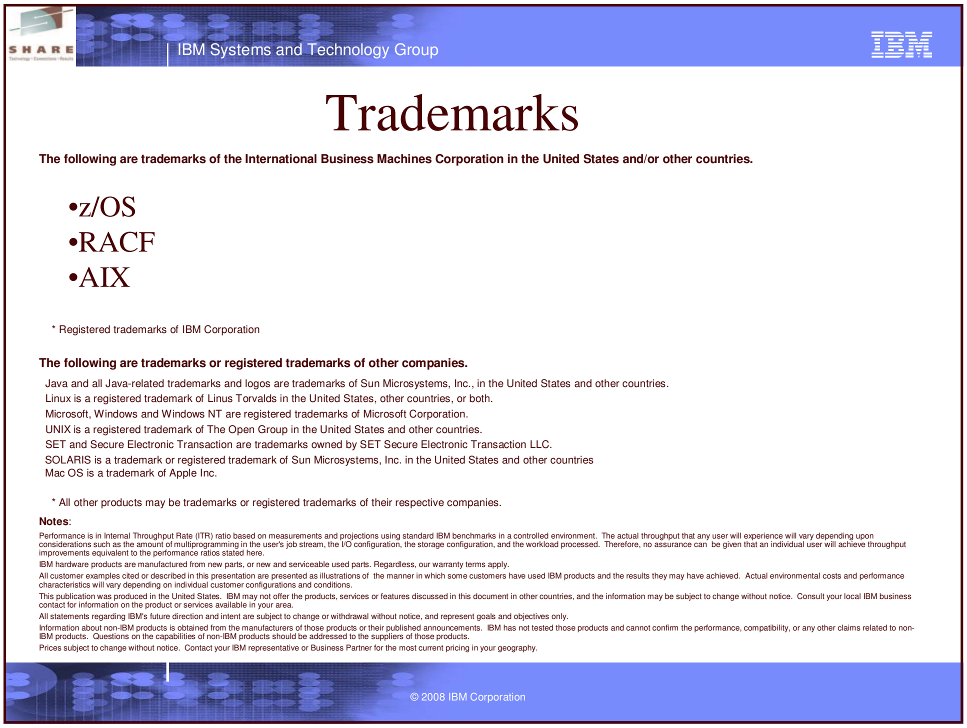
Disclaimer

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
•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.

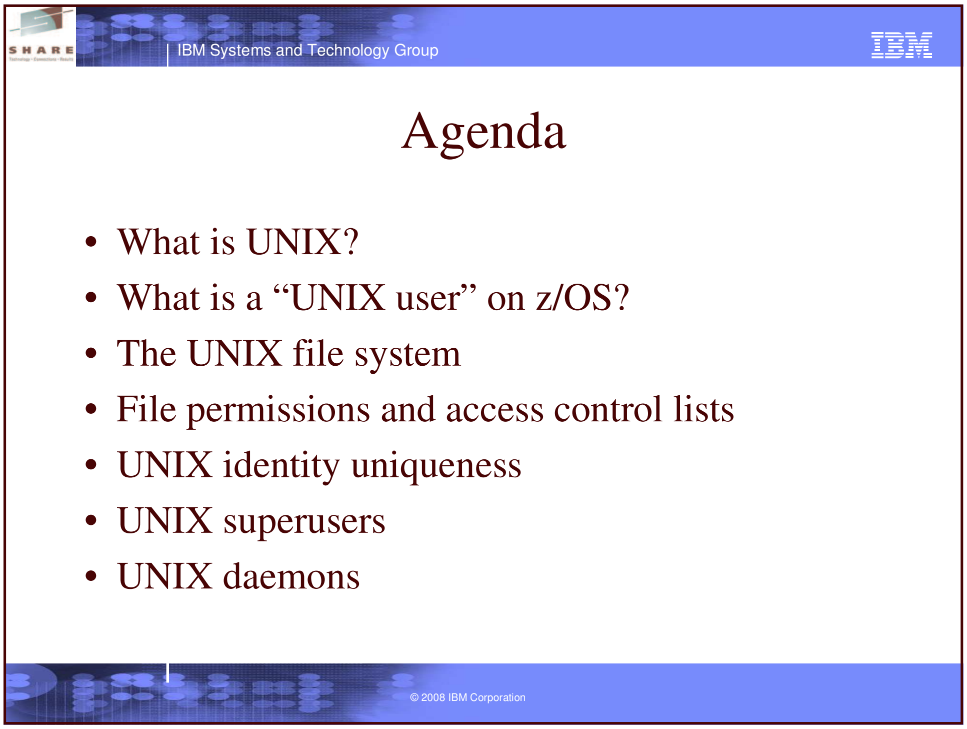
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations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

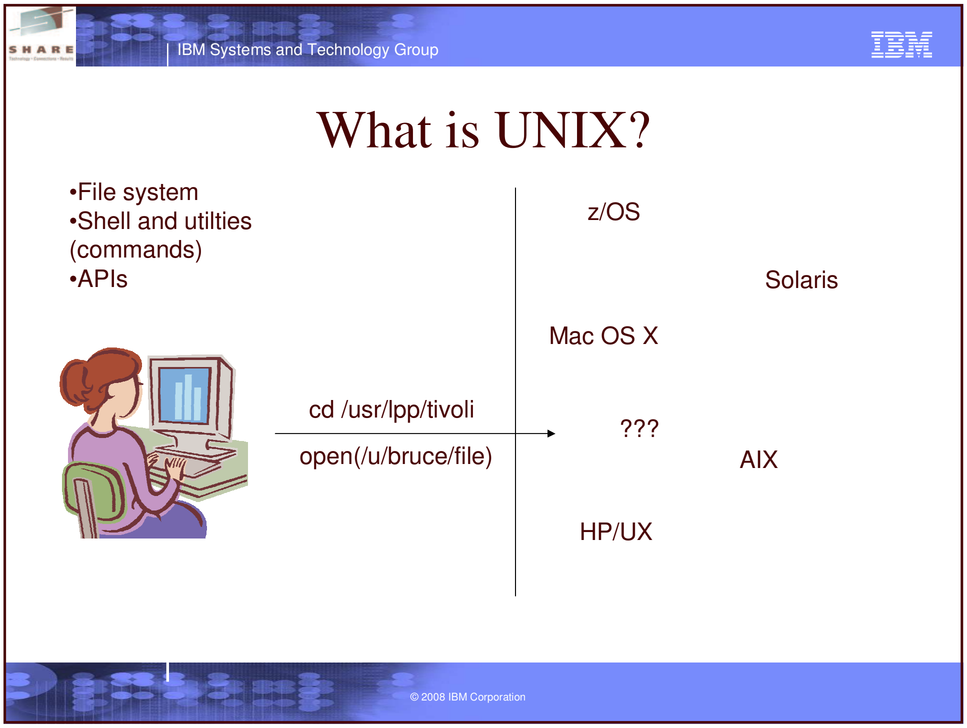
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products 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.



Agenda

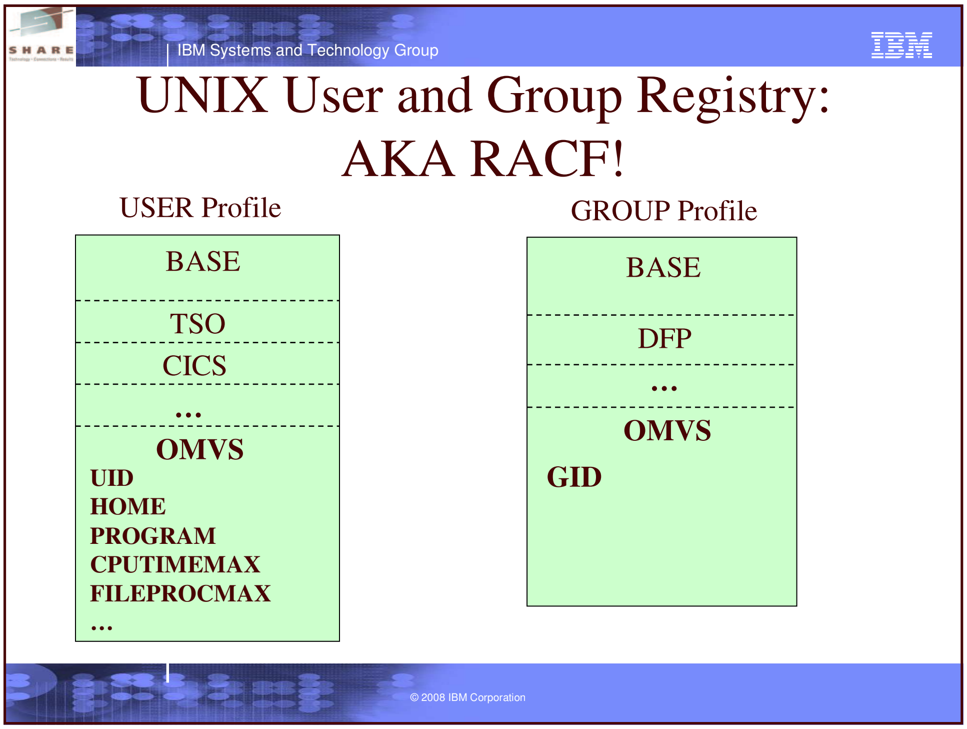
- What is UNIX?
- What is a “UNIX user” on z/OS?
- The UNIX file system
- File permissions and access control lists
- UNIX identity uniqueness
- UNIX superusers
- UNIX daemons



What is UNIX?

- •File system
- •Shell and utilties (commands)
- •APIs

cd /usr/lpp/tivoli
open(/u/bruce/file)
z/OS
Solaris
Mac OS X
???
AIX
HP/UX



UNIX User and Group Registry: AKA RACF!

USER Profile

BASE

TSO

CICS

…

OMVS

UID
HOME
PROGRAM
CPUTIMEMAX
FILEPROCMAX
…

GROUP Profile

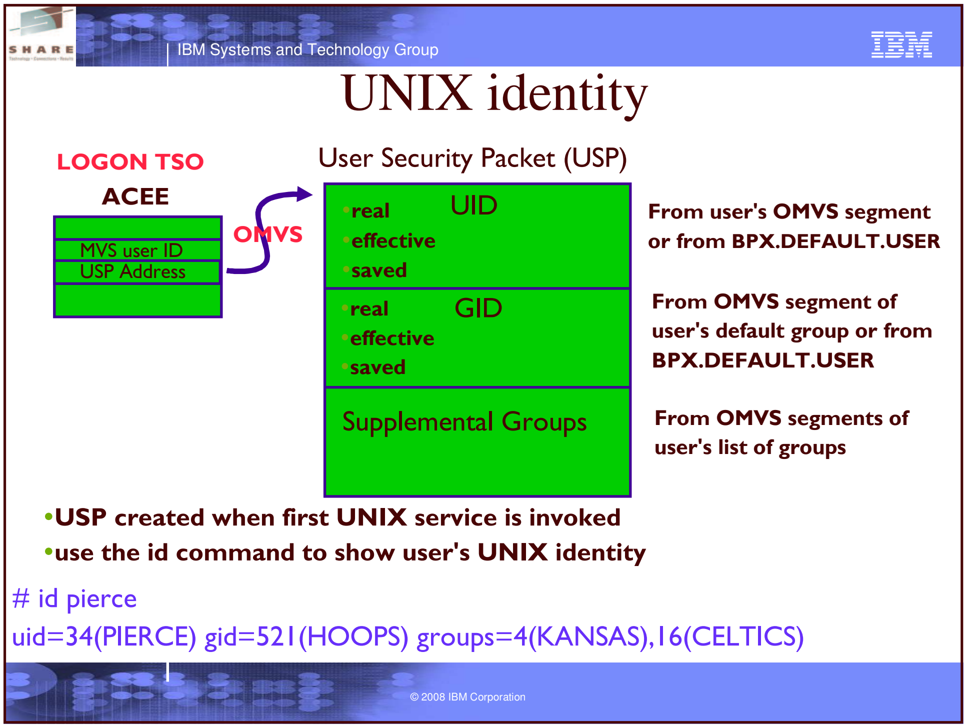
BASE

DFP

…

OMVS

GID



UNIX identity

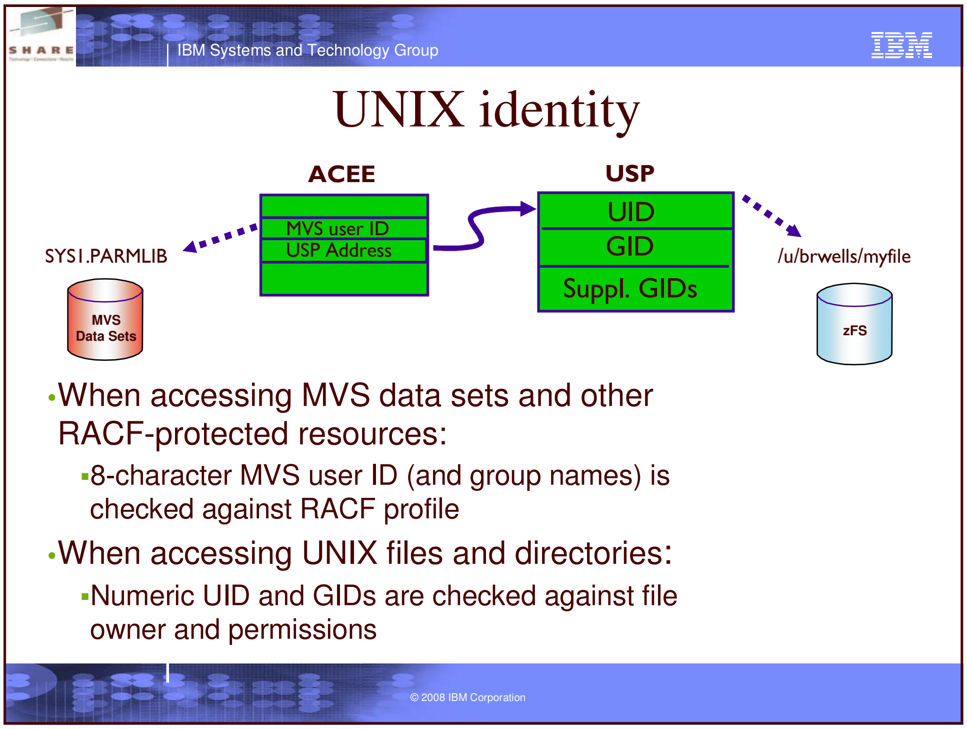
From user's OMVS segment or from BPX.DEFAULT.USER

From OMVS segment of user's default group or from BPX.DEFAULT.USER

From OMVS segments of user's list of groups

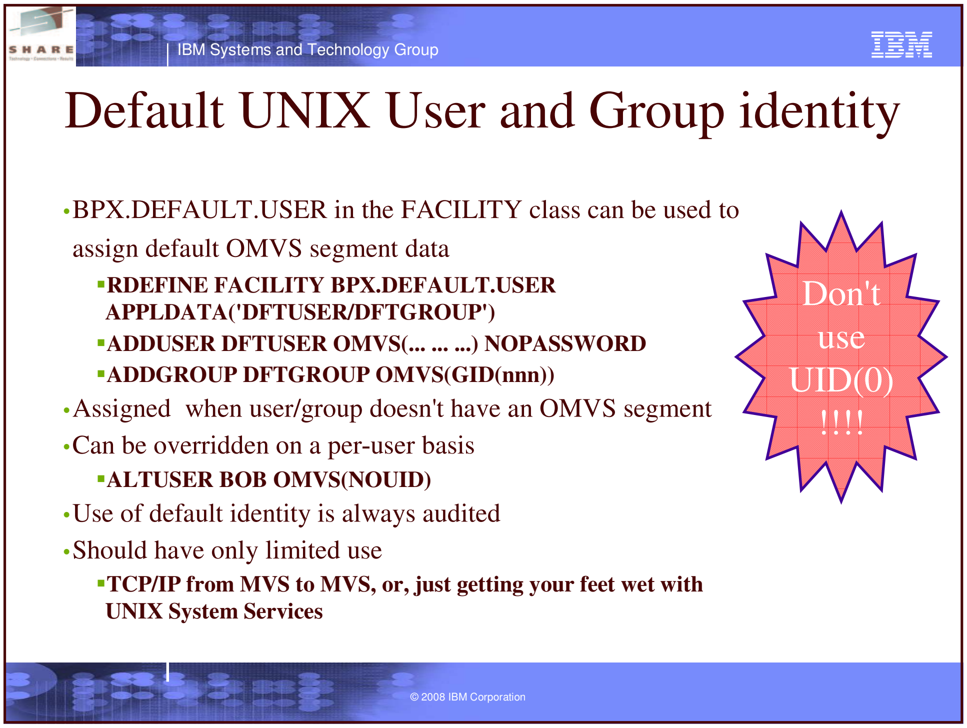
- **USP created when first UNIX service is invoked**
- **use the id command to show user's UNIX identity**

```
# id pierce
uid=34(PIERCE) gid=521(HOOPS) groups=4(KANSAS),16(CELTICS)
```



UNIX identity

- When accessing MVS data sets and other RACF-protected resources:
  - 8-character MVS user ID (and group names) is checked against RACF profile
- When accessing UNIX files and directories:
  - Numeric UID and GIDs are checked against file owner and permissions

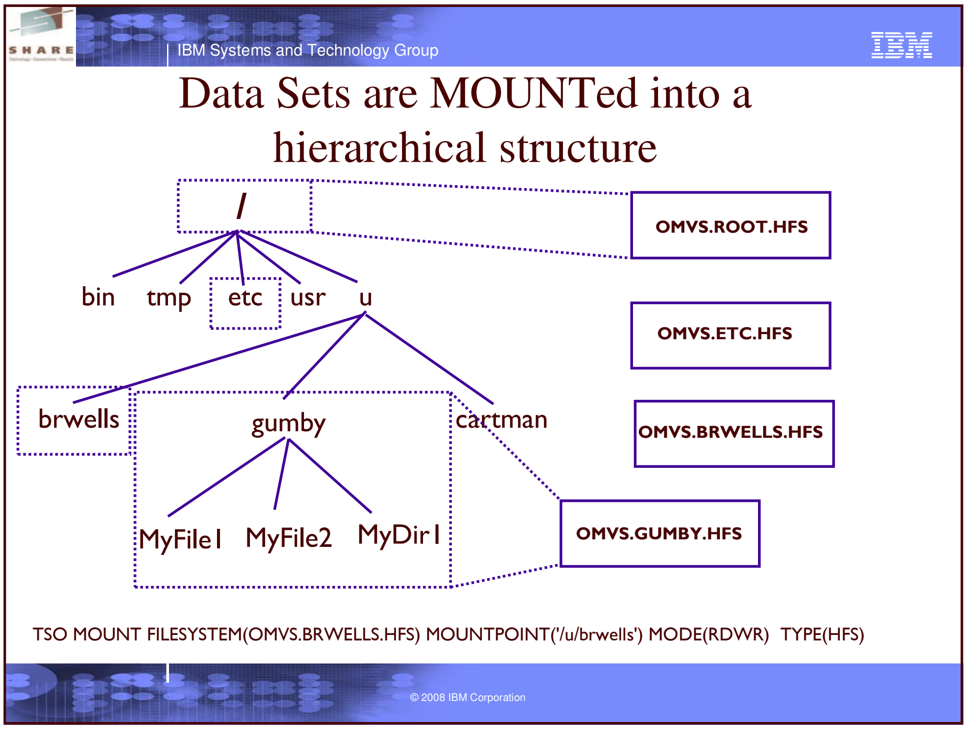


IBM Systems and Technology Group

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**

Don't use UID(0) !!!!



IBM Systems and Technology Group

Data Sets are MOUNTed into a hierarchical structure

/

bin tmp etc usr u

brwells gumby cartman

MyFile1 MyFile2 MyDir1

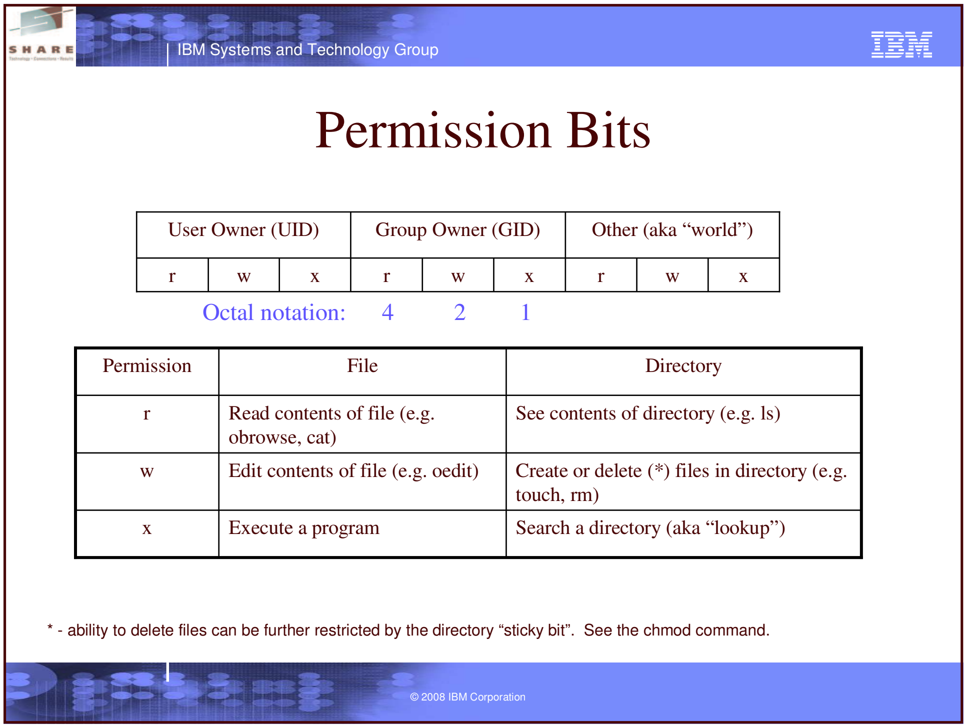
OMVS.ROOT.HFS

OMVS.ETC.HFS

OMVS.BRWELLS.HFS

OMVS.GUMBY.HFS

```
TSO MOUNT FILESYSTEM(OMVS.BRWELLS.HFS) MOUNTPOINT('/u/brwells') MODE(RDWR)  TYPE(HFS)
```



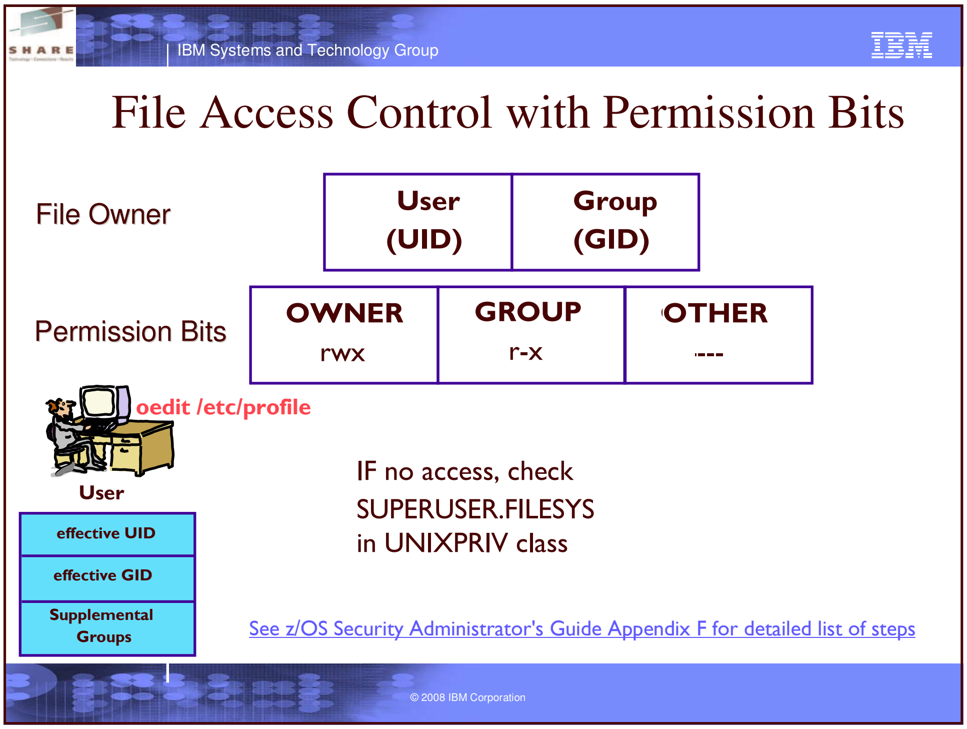
Permission Bits

User Owner (UID)			Group Owner (GID)			Other (aka “world”)		
r	w	x	r	w	x	r	w	x

Octal notation: 4 2 1

Permission	File	Directory
r	Read contents of file (e.g. obrowse, cat)	See contents of directory (e.g. ls)
w	Edit contents of file (e.g. oedit)	Create or delete (*) files in directory (e.g. touch, rm)
x	Execute a program	Search a directory (aka “lookup”)

* - ability to delete files can be further restricted by the directory “sticky bit”. See the chmod command.



File Access Control with Permission Bits

File Owner

User (UID)	Group (GID)

Permission Bits

OWNER	GROUP	OTHER
rwx	r-x	---

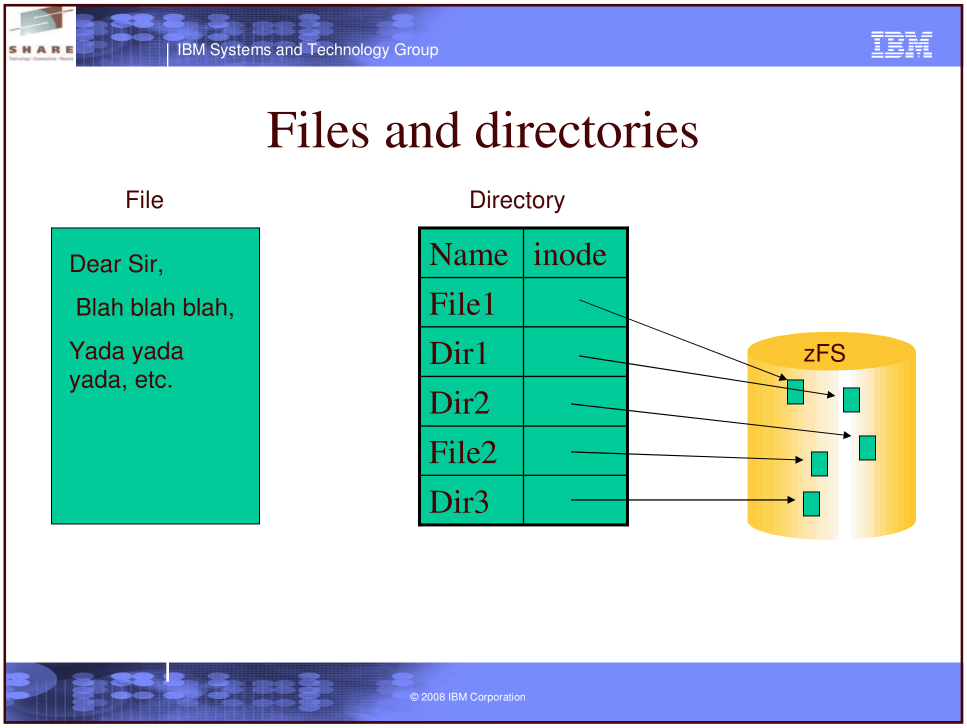
oedit /etc/profile

User

effective UID
effective GID
Supplemental Groups

IF no access, check SUPERUSER.FILESYS in UNIXPRIV class

See z/OS Security Administrator's Guide Appendix F for detailed list of steps



Files and directories

File

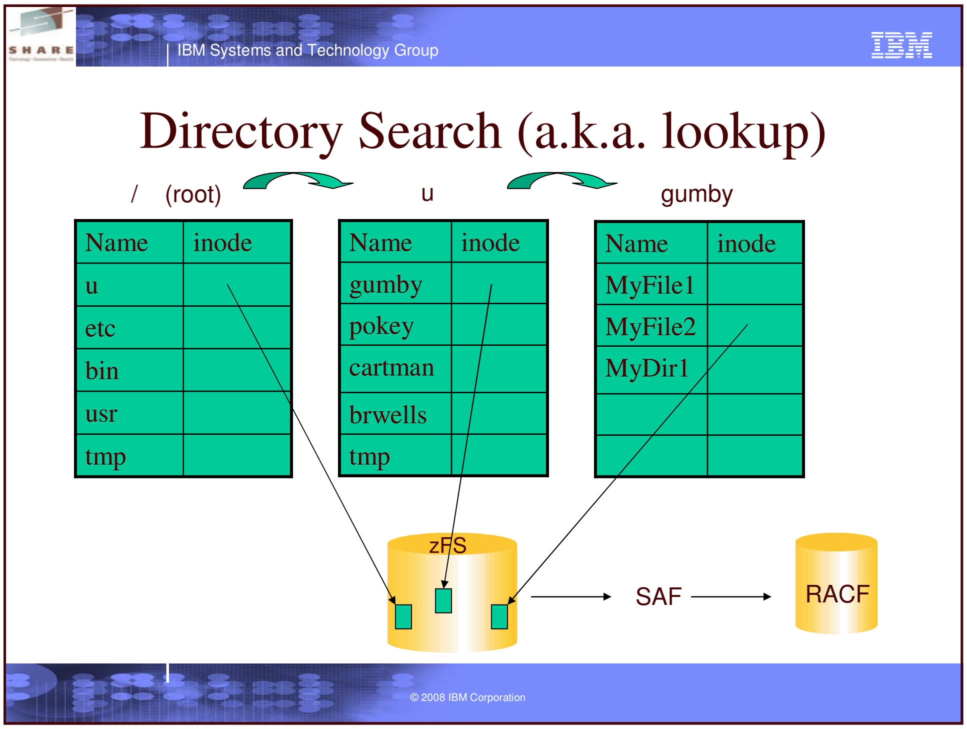
Dear Sir,

Blah blah blah,

Yada yada yada, etc.

Directory

Name	inode
File1	
Dir1	
Dir2	
File2	
Dir3	



Directory Search (a.k.a. lookup)

/ (root)

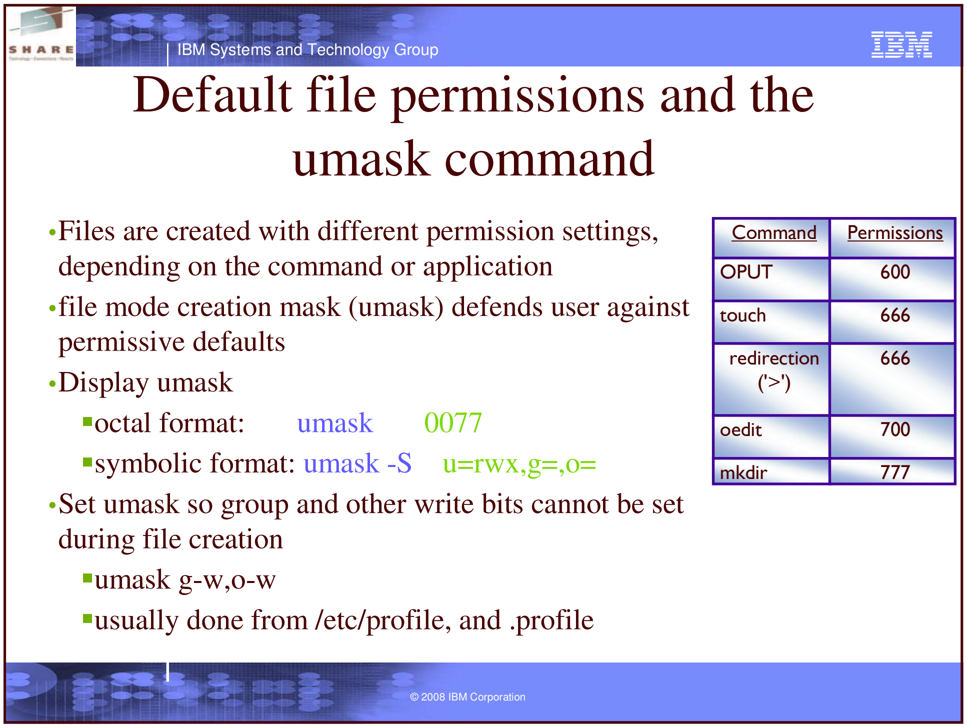
Name	inode
u	
etc	
bin	
usr	
tmp	

u

Name	inode
gumby	
pokey	
cartman	
brwells	
tmp	

gumby

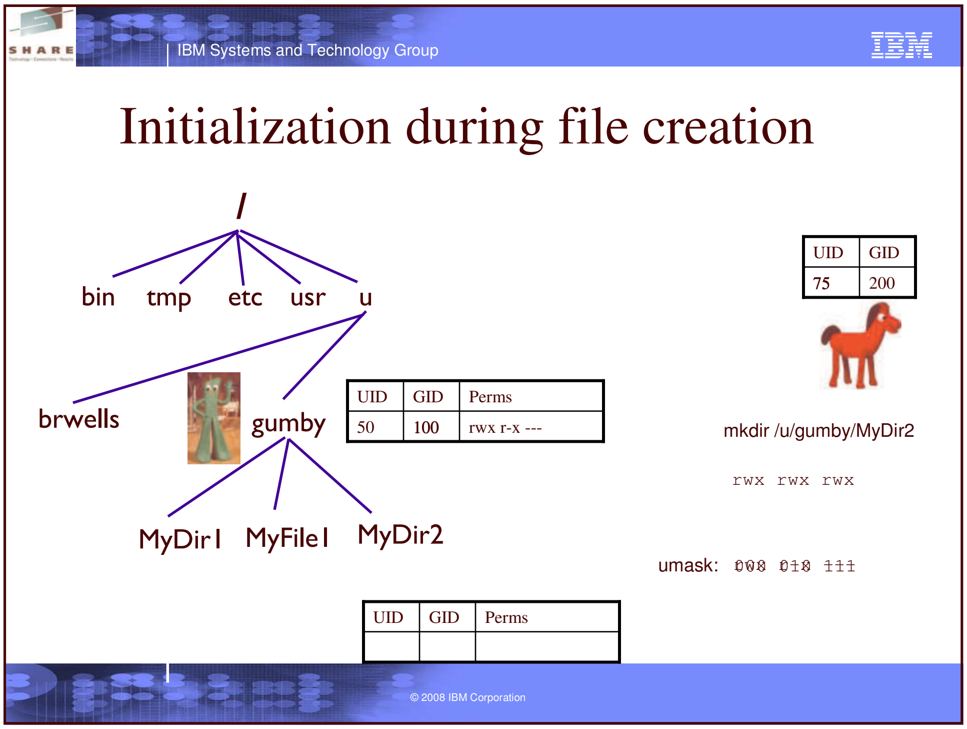
Name	inode
MyFile1	
MyFile2	
MyDir1	



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



Initialization during file creation

UID	GID	Perms
50	100	rwx r-x ---

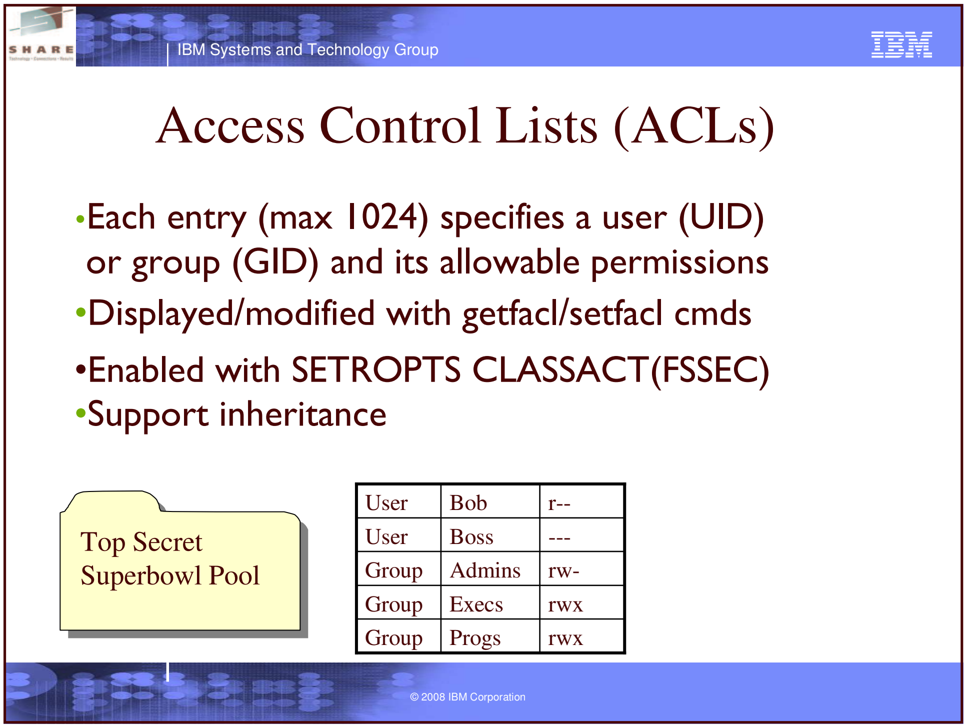
UID	GID
75	200

mkdir /u/gumby/MyDir2

```
rwx rwx rwx
```

umask: `000 010 111`

UID	GID	Perms

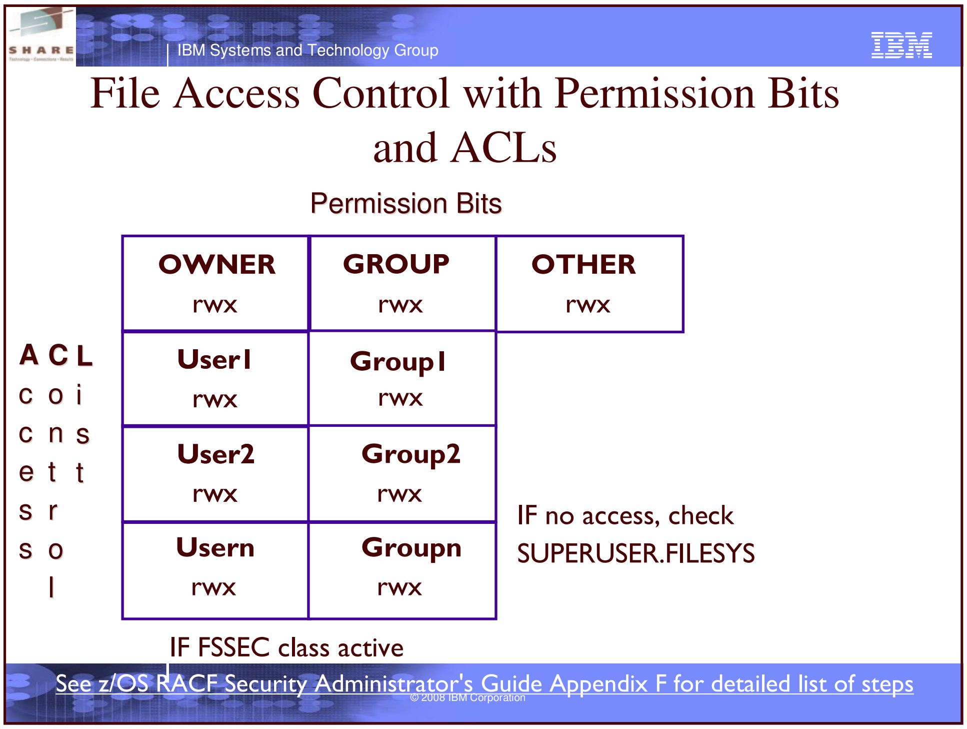


Access Control Lists (ACLs)

- 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



File Access Control with Permission Bits and ACLs

Permission Bits

	OWNER rwx	GROUP rwx	OTHER rwx
Access Control List	User1 rwx	Group1 rwx	
	User2 rwx	Group2 rwx	
	Usern rwx	Groupn rwx	IF no access, check SUPERUSER.FILESYS

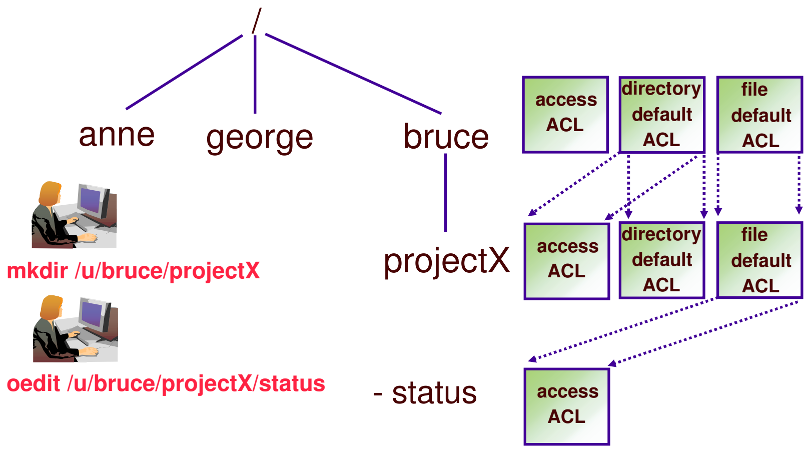
IF FSSEC class active

See z/OS RACF Security Administrator's Guide Appendix F for detailed list of steps

ACL Inheritance

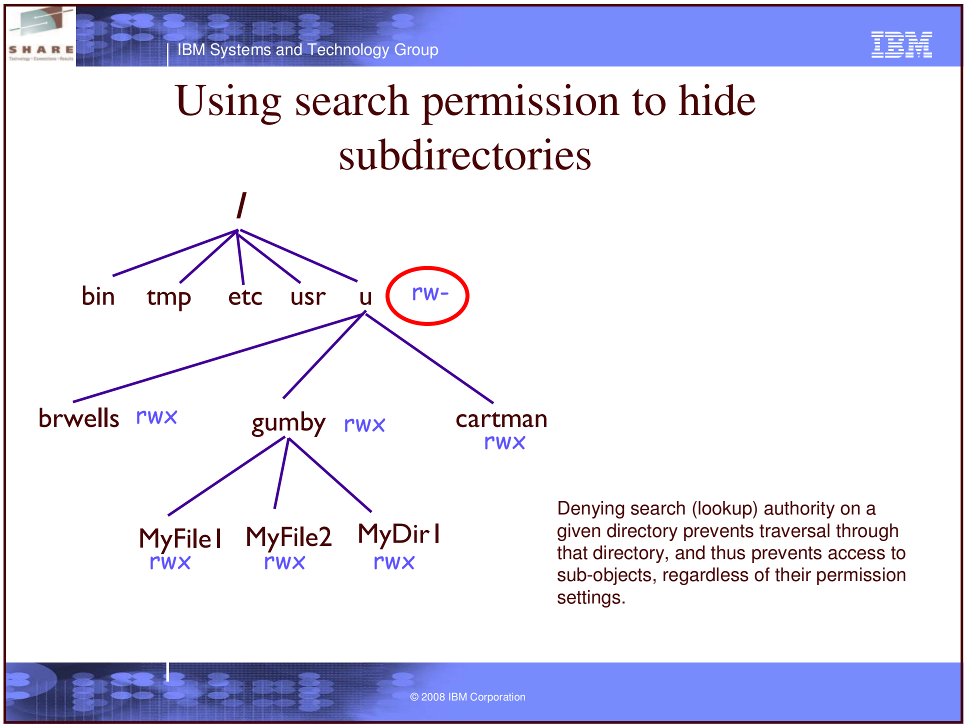
- Can establish default (or 'model') ACLs on a directory
- Get automatically applied to new files/directories created within the directory
- Separate default used for files and subdirectories
- Reduces administrative overhead

ACL Inheritance ...



mkdir /u/bruce/projectX

oedit /u/bruce/projectX/status



Using search permission to hide subdirectories

/

bin tmp etc usr u rw-

brwells rwx

gumby rwx

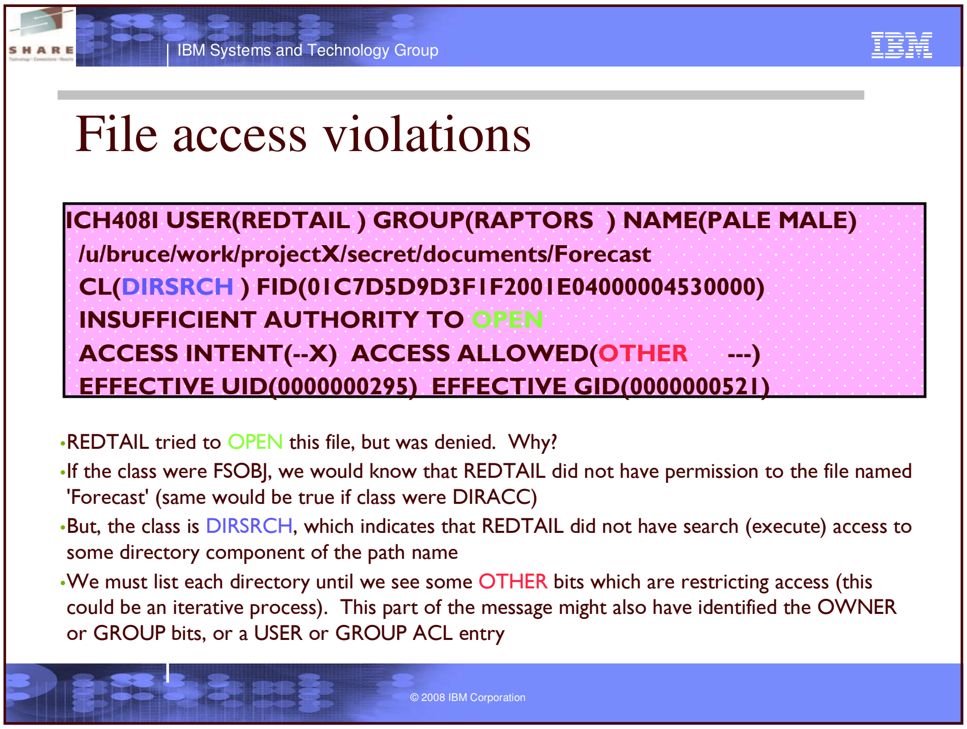
cartman rwx

MyFile1 rwx

MyFile2 rwx

MyDir1 rwx

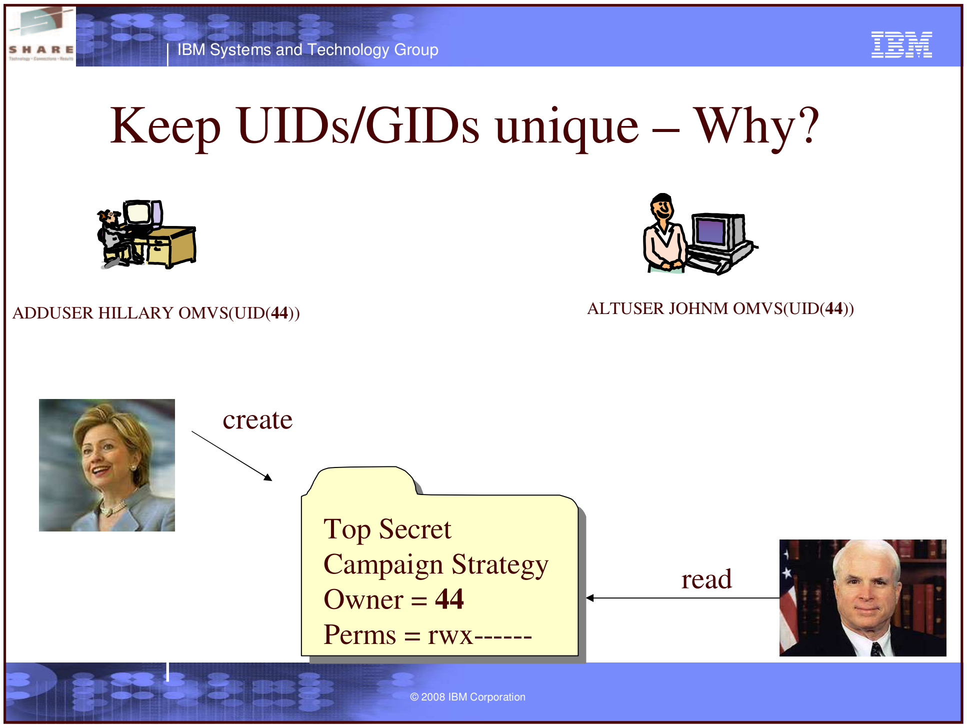
Denying search (lookup) authority on a given directory prevents traversal through that directory, and thus prevents access to sub-objects, regardless of their permission settings.



File access violations

```
ICH408I USER(REDTAIL ) GROUP(RAPTORS  ) NAME(PALE MALE)
 /u/bruce/work/projectX/secret/documents/Forecast
 CL(DIRSRCH ) FID(01C7D5D9D3F1F2001E04000004530000)
 INSUFFICIENT AUTHORITY TO OPEN
 ACCESS INTENT(--X)  ACCESS ALLOWED(OTHER     ---)
 EFFECTIVE UID(0000000295)  EFFECTIVE GID(0000000521)
```

- REDTAIL tried to OPEN this file, but was denied. Why?
- If the class were FSOBJ, we would know that REDTAIL did not have permission to the file named 'Forecast' (same would be true if class were DIRACC)
- But, the class is DIRSRCH, which indicates that REDTAIL did not have search (execute) access to some directory component of the path name
- We must list each directory until we see some OTHER bits which are restricting access (this could be an iterative process). This part of the message might also have identified the OWNER or GROUP bits, or a USER or GROUP ACL entry



Keep UIDs/GIDs unique – Why?

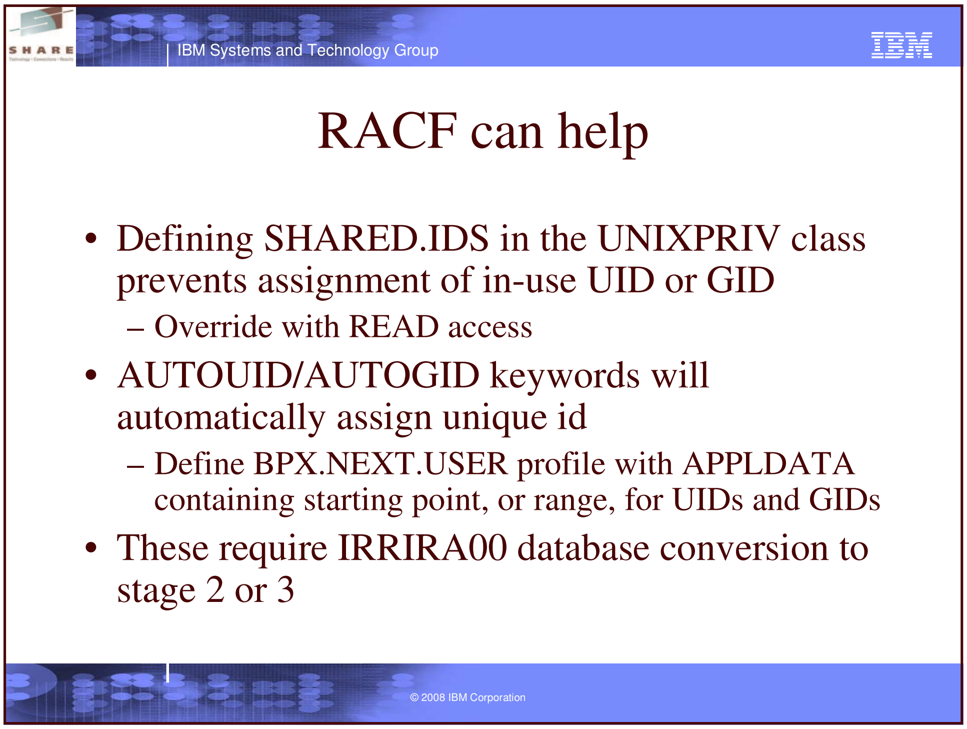
ADDUSER HILLARY OMVS(UID(**44**))

ALTUSER JOHNM OMVS(UID(**44**))

create

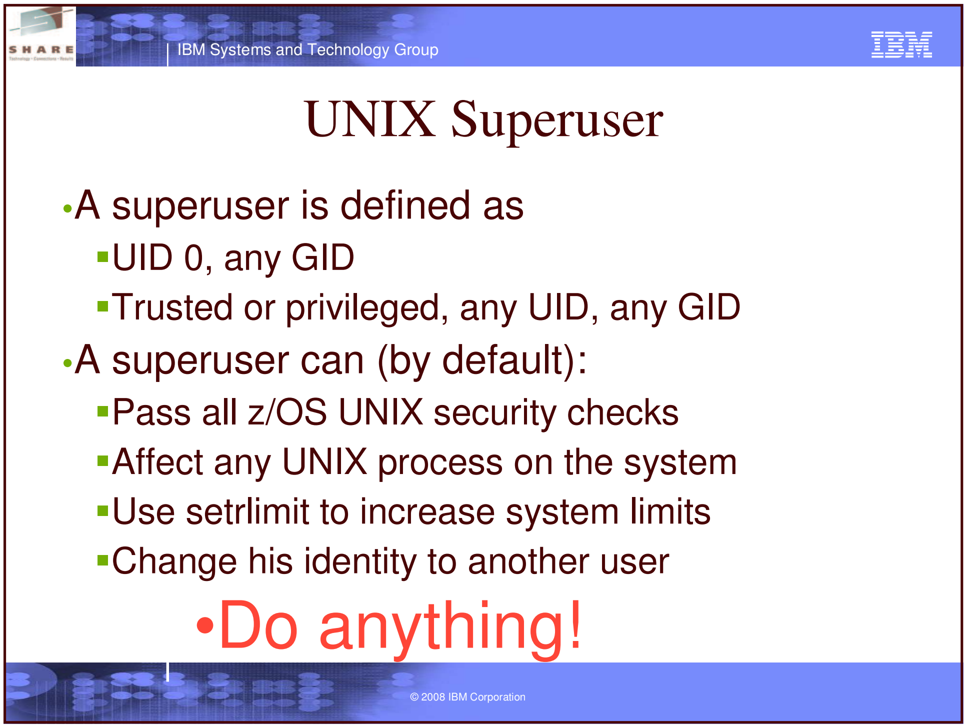
Top Secret
Campaign Strategy
Owner = **44**
Perms = rwx------

read



RACF can help

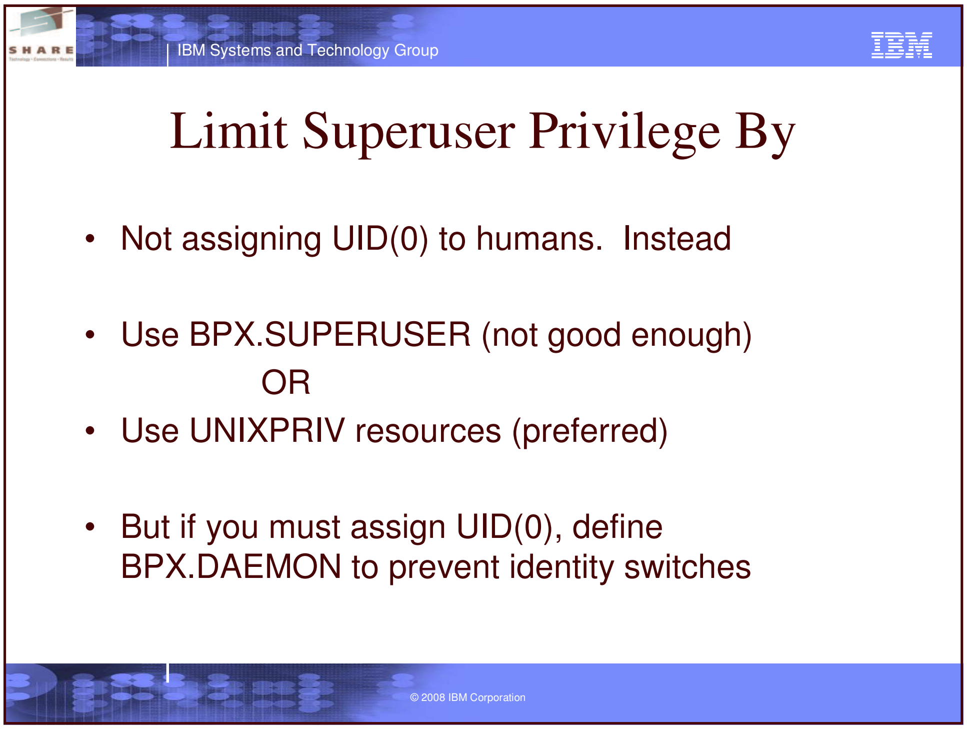
- Defining SHARED.IDS in the UNIXPRIV class prevents assignment of in-use UID or GID
  - Override with READ access
- AUTOUID/AUTOGID keywords will automatically assign unique id
  - Define BPX.NEXT.USER profile with APPLDATA containing starting point, or range, for UIDs and GIDs
- These require IRRIRA00 database conversion to stage 2 or 3



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!



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 identity switches

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

See z/OS UNIX System Services Planning for complete list of UNIXPRIV resources

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

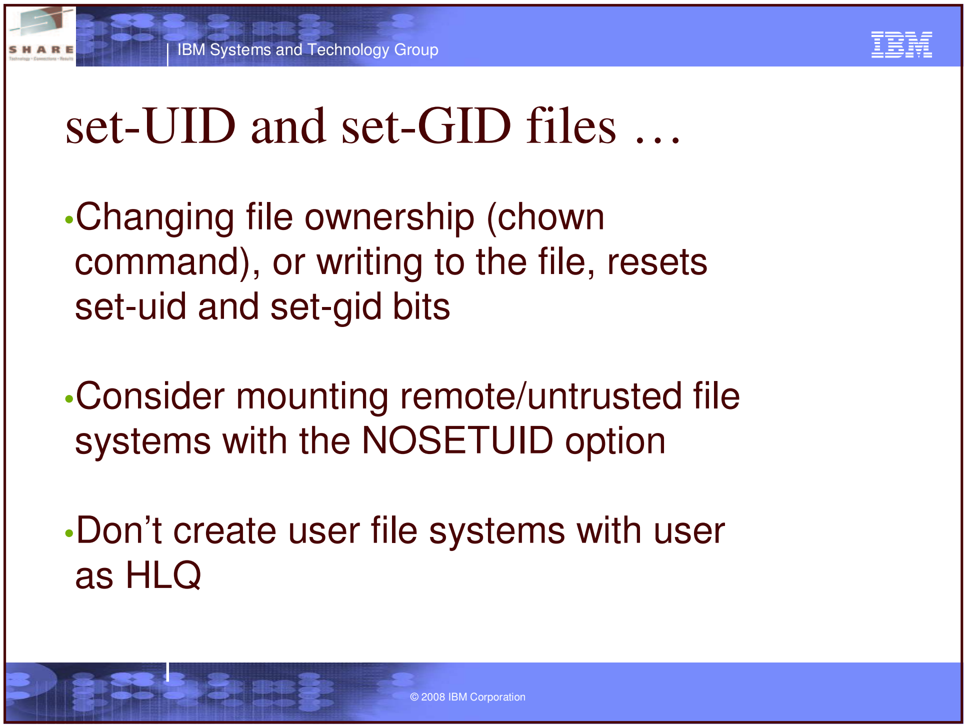
Controlling Daemons ...
z/OS UNIX-Level Security

- 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 *nosetuid* or *nosecurity*

set-UID and set-GID files

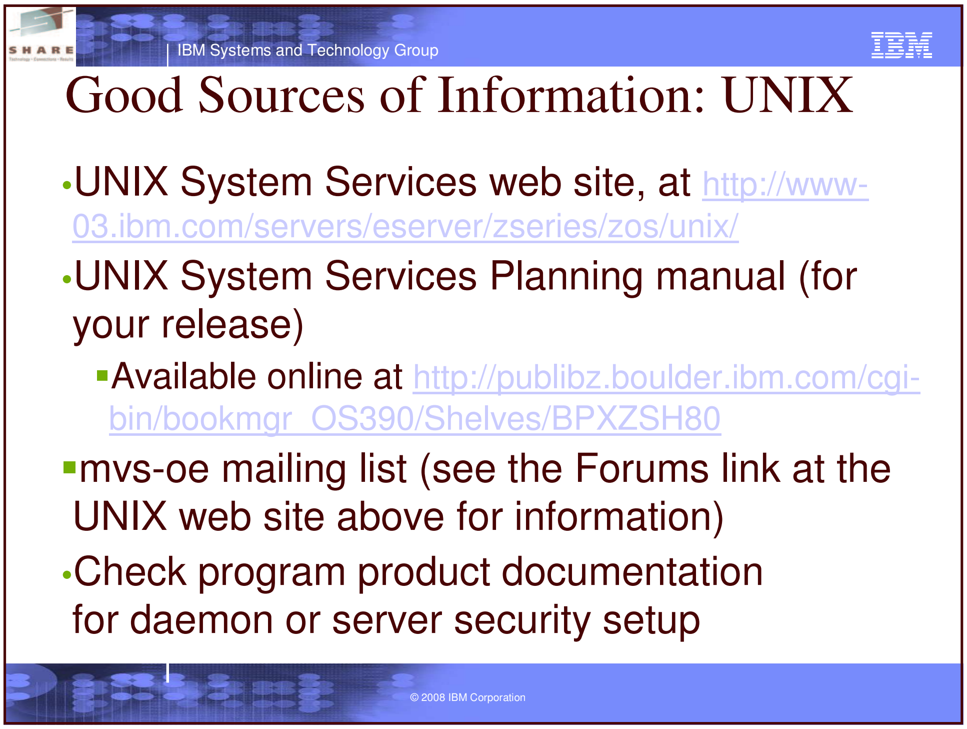
- Executable files which change the effective UID/GID to that of the file owner
  - UNIX file access now based on owner (user and/or group) of set-id file
  - does *not* change the MVS identity
  - locate your set-uid files with find / -perm -4000 or by using irrhfsu
- chmod u+s,g+s myprogram
  - must be file owner or have superuser privilege
- ls -l myprogram

```
-rwsr-s--x   2 BRWELLS  DEPTD60    8192 Feb  8 10:51 myprogram
```



set-UID and set-GID files …

- Changing file ownership (chown command), or writing to the file, resets set-uid and set-gid bits
- Consider mounting remote/untrusted file systems with the NOSETUID option
- Don't create user file systems with user as HLQ



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://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/BPXZSH80
- 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

Good Sources of Information: RACF

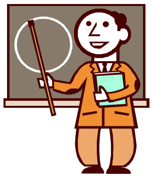
- RACF Auditor's Guide
  - UNIX auditing classes
- RACF Macros and Interfaces
  - SMF 80 formats and SMF Unload mappings
- RACF Security Administrator's Guide
  - Chapter on z/OS UNIX security

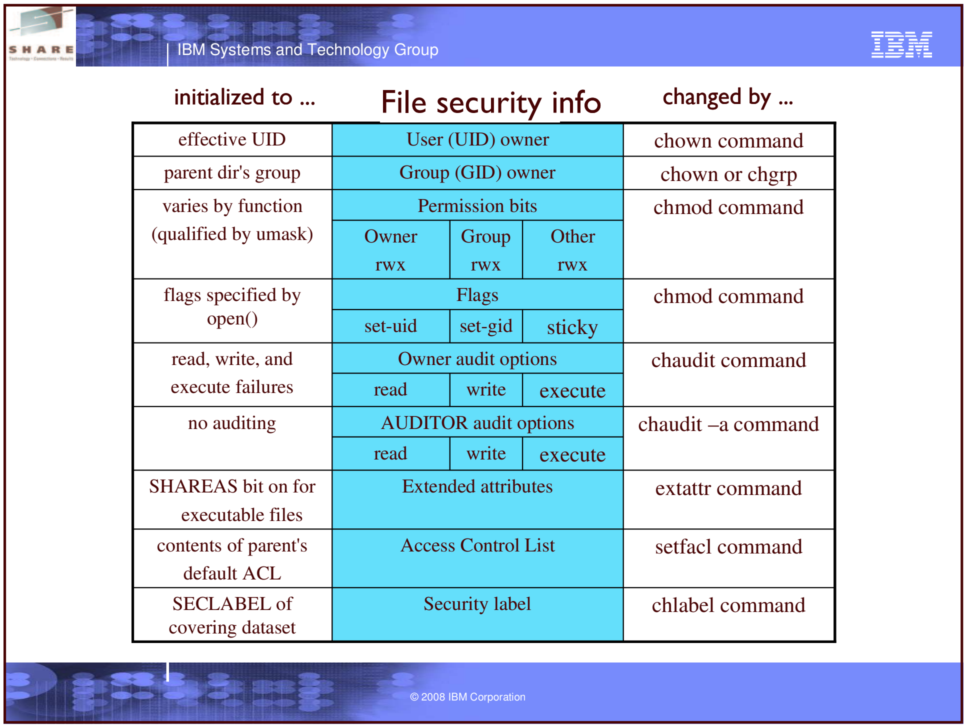
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/ICHZBK80

- RACF web page – irrhfsu and presentations

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

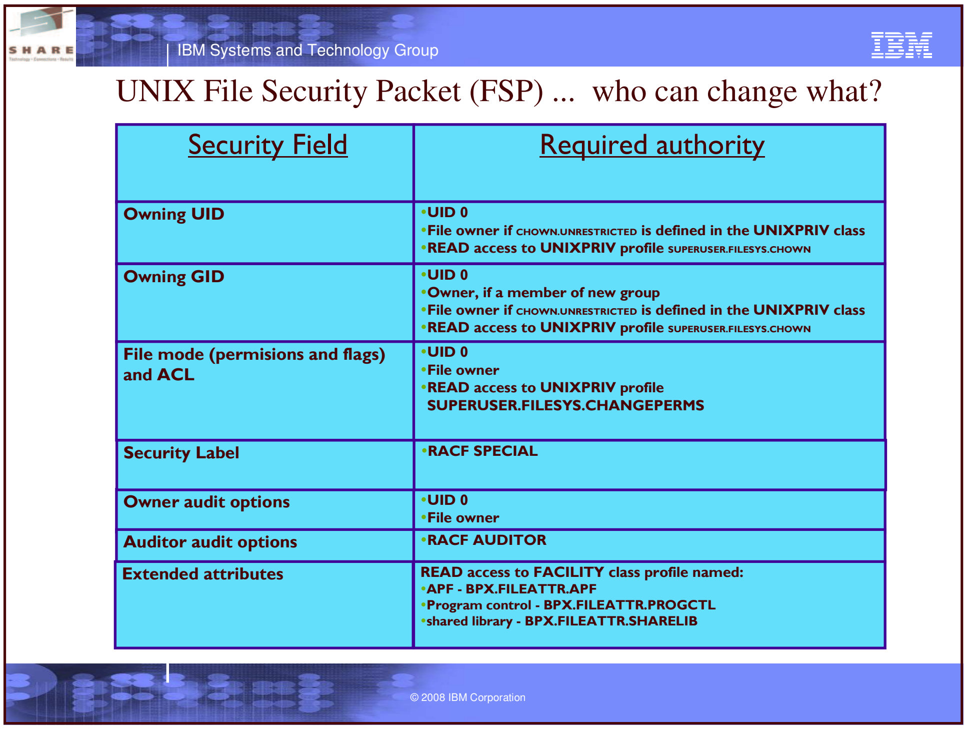
Appendix: Supplementary material





File security info

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	Group rwx	Other rwx	
flags specified by open()	Flags			chmod command
	set-uid	set-gid	sticky	
read, write, and execute failures	Owner audit options			chaudit command
	read	write	execute	
no auditing	AUDITOR audit options			chaudit –a command
	read	write	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



UNIX File Security Packet (FSP) … who can change what?

Security Field	Required authority
Owning UID	•**UID 0** •**File owner if CHOWN.UNRESTRICTED is defined in the UNIXPRIV class** •**READ access to UNIXPRIV profile SUPERUSER.FILESYS.CHOWN**
Owning GID	•**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 (permisions and flags) and ACL	•**UID 0** •**File owner** •**READ access to UNIXPRIV profile SUPERUSER.FILESYS.CHANGEPERMS**
Security Label	•**RACF SPECIAL**
Owner audit options	•**UID 0** •**File owner**
Auditor audit options	•**RACF AUDITOR**
Extended attributes	**READ access to FACILITY class profile named:** •**APF - BPX.FILEATTR.APF** •**Program control - BPX.FILEATTR.PROGCTL** •**shared library - BPX.FILEATTR.SHARELIB**