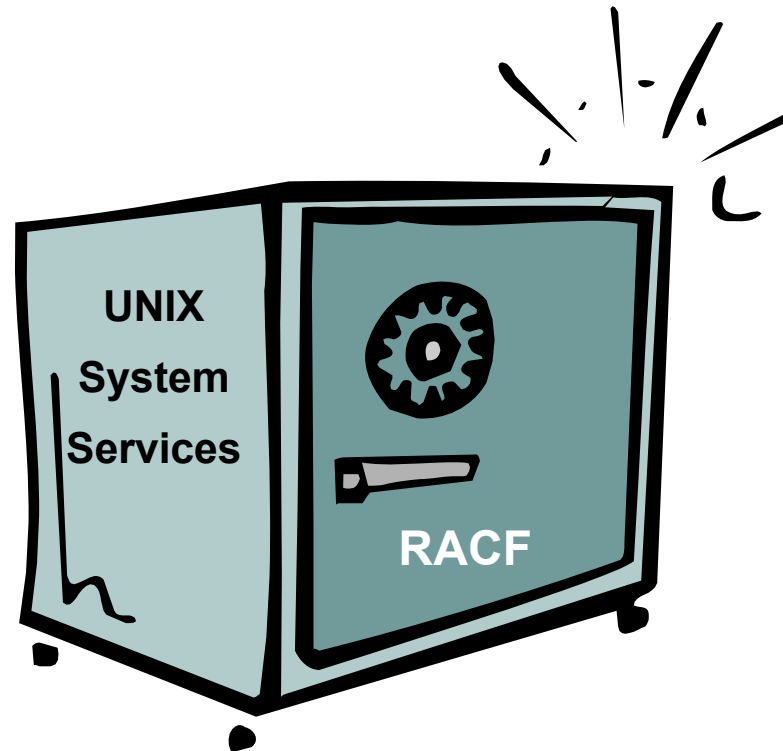


z/OS Unix - File System Security

Chicagoland RACF Users Group - October 2009



Robert S. Hansel

Lead RACF Specialist - RSH Consulting, Inc.

R.Hansel@rshconsulting.com - 617-969-9050 - www.rshconsulting.com

TOPICS

Unix File System

File System Security

File Security Packet (FSP)

Owner, Group, & Other Permissions

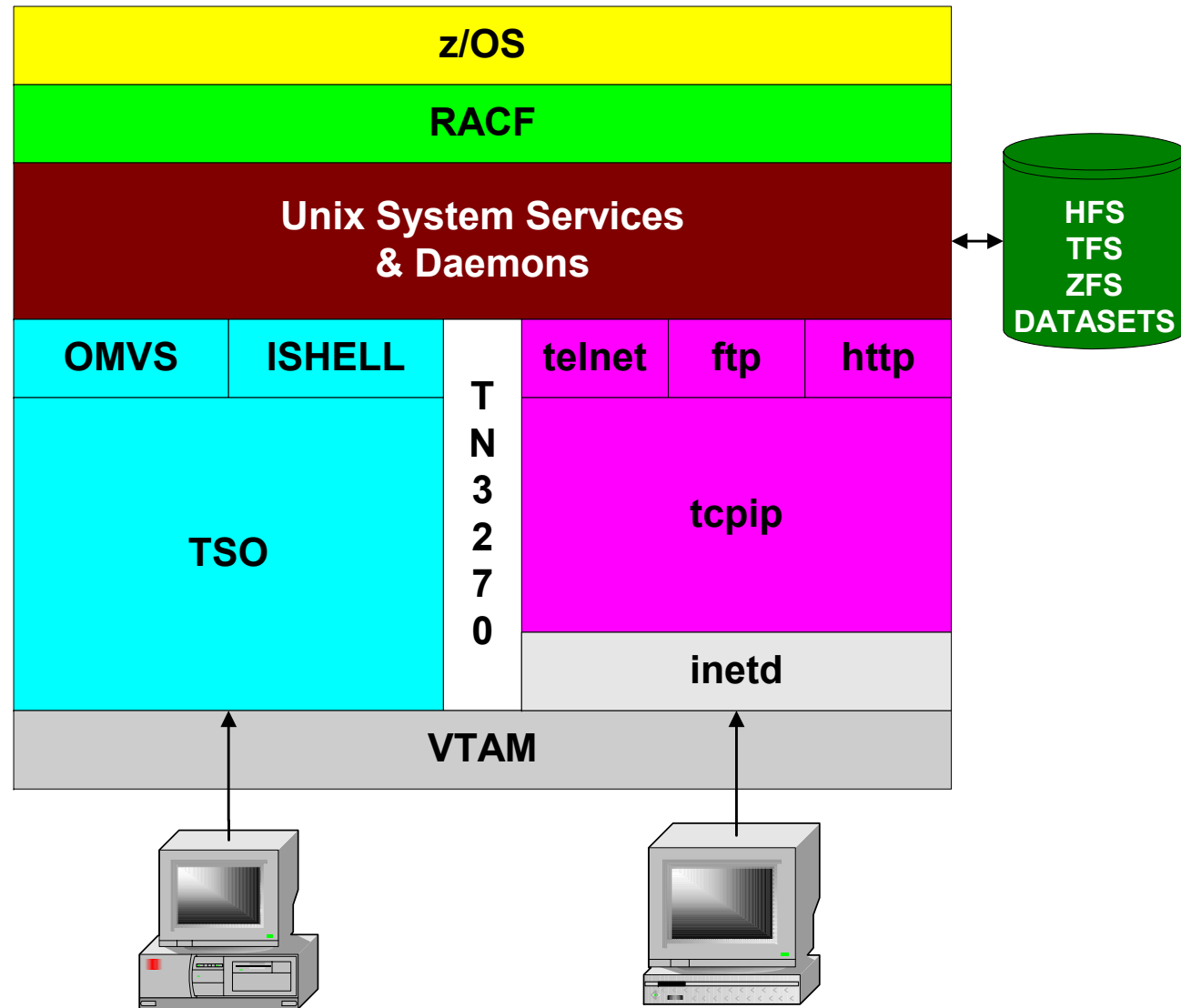
Access Control List (ACL)

UNIXPRIV Authorities

Monitoring

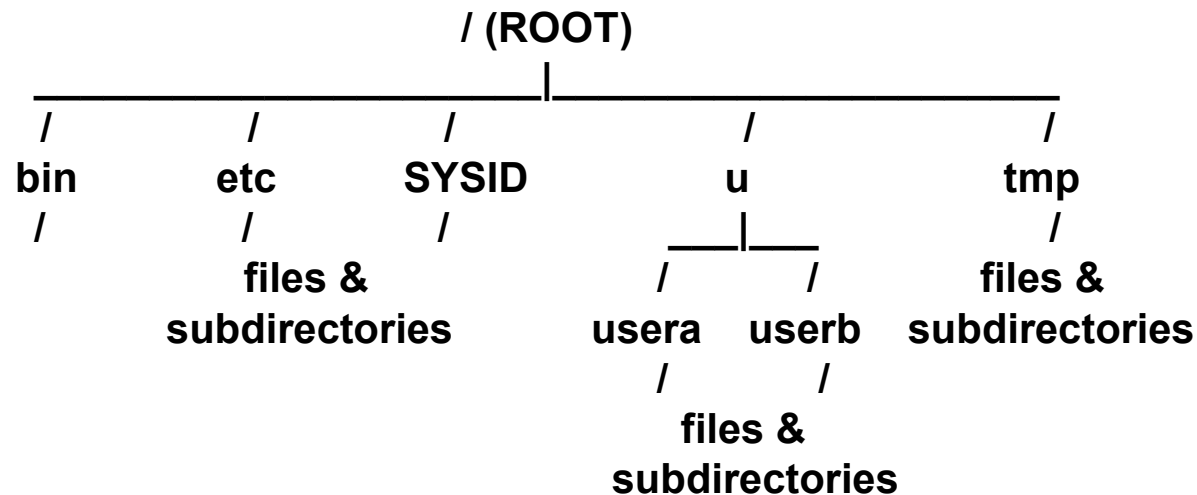
RACF and z/OS are Trademarks of the International Business Machines Corporation

UNIX SYSTEM SERVICES



LOGICAL FILE SYSTEM

Unix directory structure and files



Common directories

- **/etc** - conf files
- **/bin** - programs
- **/tmp** - temporary files

Names are mixed-case

PHYSICAL FILE SYSTEM

HFS **Hierarchical File System**

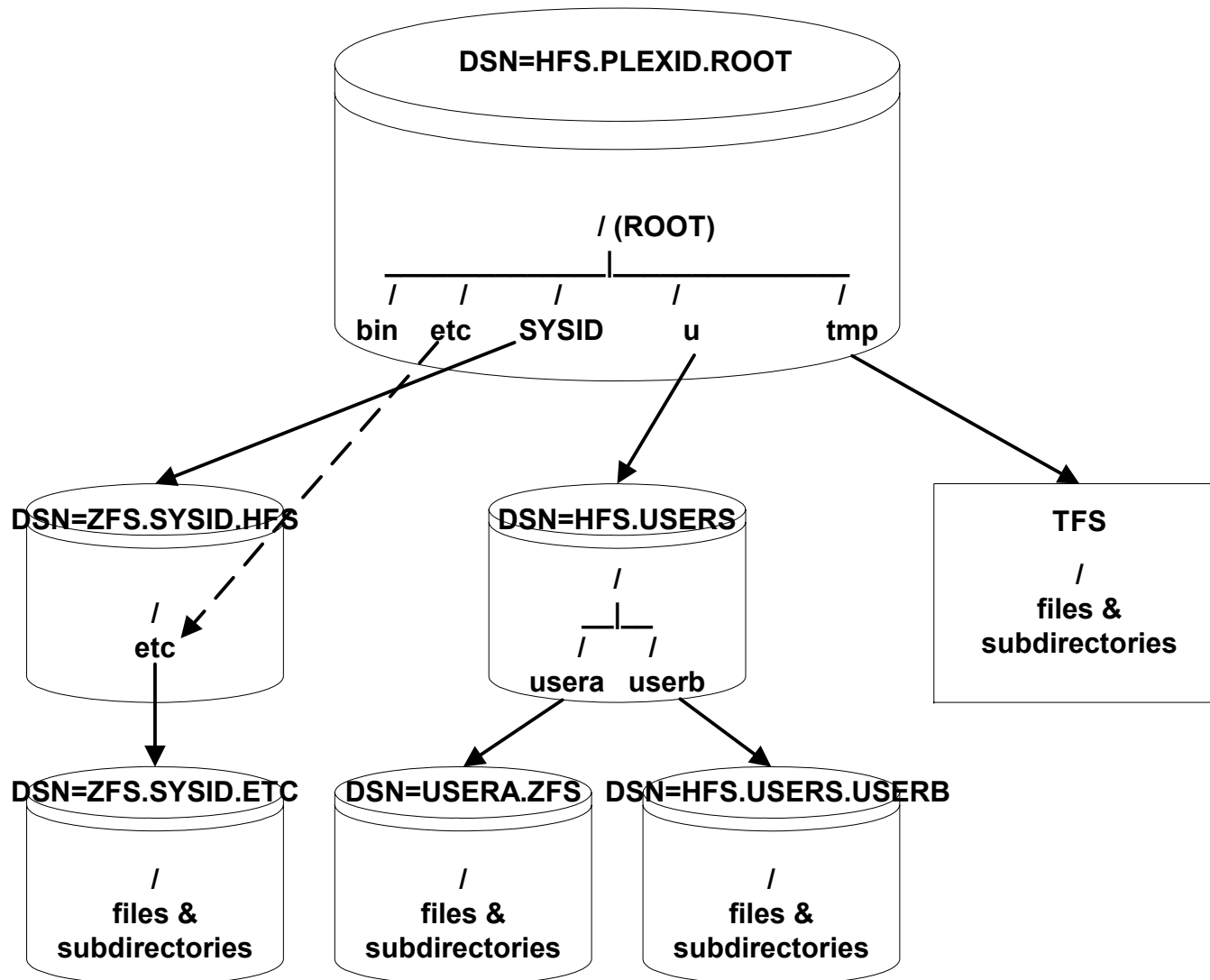
TFS **Temporary File System**

zFS **zSeries File System**

DFS **Distributed File System**

NFS **Network File System**

PHYSICAL FILE SYSTEM



OMVS FILE SYSTEM CONFIGURATION

PARMLIB(BPXPRMxx)

| | | |
|-------|------------------------------|--|
| ROOT | FILESYSTEM('fs-dsname') | Identifies Root File System DSN |
| | TYPE(HFS ZFS) | Specifies File System type |
| | MODE(<u>RDWR</u> READ) | Access allowed (RDWR = read/write) |
| | <u>SETUID</u> NOSETUID | Support setuid() & setgid() mode bit |
| MOUNT | FILESYSTEM('fs-dsname') | Identifies File System DSN to mount |
| | TYPE(HFS ZFS TFS) | Specifies File System type |
| | MOUNTPOINT('pathname') | Directory where to be mounted |
| | MODE(<u>RDWR</u> READ) | Access allowed (RDWR = read/write) |
| | <u>SETUID</u> NOSETUID | Support setuid, setgid, APF, program control |
| | <u>SECURITY</u> NOSECURITY | Perform security checks |

Note: NOSETUID - APF and Program Control extended attributes not honored

/etc/auto.master - Automatic Mount configuration

BPXPRMxx - SAMPLE

```
ROOT      FILESYSTEM( 'HFS.PLEXID.ROOT' )
          TYPE(HFS) MODE(RDWR)

MOUNT     FILESYSTEM( 'ZFS.SYSID.HFS' )
          TYPE(ZFS) MODE(RDWR) NOAUTOMOVE
          MOUNTPOINT( '/SYSID' )

MOUNT     FILESYSTEM( 'ZFS.SYSID.ETC' )
          TYPE(ZFS) MODE(RDWR) NOAUTOMOVE
          MOUNTPOINT( '/SYSID/etc' )

MOUNT     FILESYSTEM( '/TMP&SYSNAME.' )
          TYPE(TFS) MODE(RDWR) NOAUTOMOVE
          MOUNTPOINT( '/tmp' ) PARM( '-p 0755' )

MOUNT     FILESYSTEM( 'HFS.USERS' )
          TYPE(HFS) MODE(RDWR)
          MOUNTPOINT( '/u' )
```

NAVIGATING FILE SYSTEM - OMVS

| | |
|-----------------------|--|
| cd | Change working directory |
| cp | Copy a file |
| exit | Exit from OMVS |
| find | Find a file meeting specific criteria |
| id | Display user's identity |
| ls | List a file or directory |
| mkdir | Make a directory |
| more | List a file's contents |
| mv | Rename or move a file or directory |
| obrowse | Browse a file |
| oedit | Create and edit a file |
| pwd | Display working directory name |
| rmdir | Remove a directory |
| tso <i>cmd</i> | Execute TSO command [cannot contain '('] |
| who | Displays information about current users |

FILE SYSTEM SECURITY

Each directory and file has its own File Security Packet (FSP)

- Kept in the file system with directory or file
- Created and deleted along with the directory or file

FSP contains

- Owner & Group
- Permission bits - Owner, Group, & Other
- Access Control List (ACL)
- Audit bits

RACF performs security authorization checking for UNIX

UNIX invokes RACF through SAF

Access checking is based on user's UNIX identity

USER IDENTIFICATION

OMVS uses UNIX uid and gid for access control internally

OMVS Profile Segments - assign uid and gid

- **User Profile - associates USERID with uid**
- **Group Profile - associates Group with gid**
- **id range - 0-2147483647 uid 0 = root / superuser**

Can assign default uid and gid with BPX.DEFAULT.USER

Carefully consider use of FIELD class to delegate OMVS id assignment

Recommend users and groups be assigned unique values, and make consistent across all systems (RACF databases, Linux, AIX, etc.)

Use UNIXPRIV SHARED.IDS to ensure unique ids assigned

SUPERUSER

Superuser - a.k.a. root - system administrator

Authority assigned by ...

- **OMVS(UID(0))** **OMVS STCs & Daemons**
- **FACILITY BPX.SUPERUSER** **USS Tech Support staff**
- **PRIVILEGED / TRUSTED Started Tasks**

With READ access to BPX.SUPERUSER, user can execute 'su' (switch user) or issue syscalls such as seteuid (0)

Superuser authority

- **With only Unix Level Security - can assume other user's identities**
- **Full access to all USS directories and files (like TRUSTED)**
- **Can change directory and file security bits (like SPECIAL)**

Alternative - UNIXPRIV profiles - limited / tailored authority

USER SECURITY PACKET (USP)

USP is created when Unix service is invoked

User's uid/gid are obtained from either the OMVS segment or BPX.DEFAULT.USER

- Assigning an OMVS segment with no uid blocks default uid assignment & UNIX access

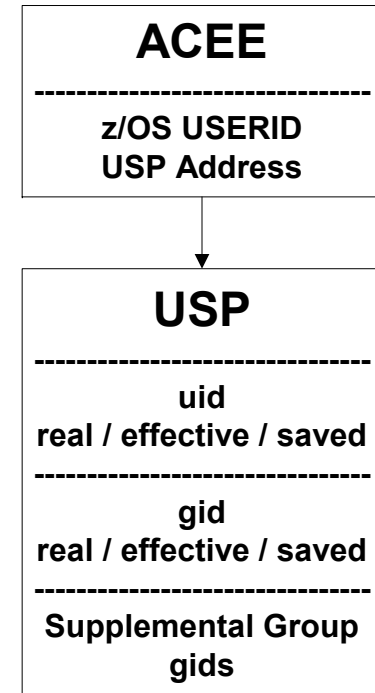
Processes using setuid & setgid type functions can change the uid/gid, becoming the 'effective' uid/gid

'Effective' uid/gid and supplemental gids are used for access authorization

Maximum of 300 supplemental groups

USS 'id' command shows user's identity and groups

```
id ibmuser
uid=0(OMVSKERN) gid=0(SYS1) groups=8(RACFTECH)
```



SETUID & SETGID BITS

Causes a program to run under authority of the Owner and/or Group (as identified in the FSP) rather than that of the invoker

- **During execution, temporarily sets invoking user's 'effective' uid/gid (but not the RACF USERID) to that of the Owner and/or Group**
- **Programs with bit turned on appear in 'ls' output with the execute bit displaying 's' for owner and/or group (instead of 'x')**
- **Example - /bin/login**

Bit set using 'chmod' command

- **Must be file Owner or have SUPERUSER privilege to set**
- **Changing the file, its Owner, or its Group turns bits off**

Mounting a file system with NOSETUID causes the bits to be ignored

- **Appropriate for remote or untrusted file systems (e.g., user HFS file)**
- **Recommended for automatic mount policy**

FILE SECURITY PACKET

File Security Packet - Base Access Control List (ACL) entries

| Owner uid | Group gid | Extended Attributes | | | | | | s | e | t | Permissions | | | | | | Auditing | | | Access Control List |
|-----------|-----------|---------------------|---|---|---|-------|---|---|---|---|-------------|---------|-------|-----|--------|-----|----------|-------|---------|---------------------|
| | | A | P | R | L | s | e | | | | Owner | | Group | | Other | | Owner | | Auditor | |
| | | | | | | | | | | | PF | ro | un | oad | uid | gid | Read | Write | Execute | |
| chown | chgrp | extattr | | | | chmod | | | | | | chaudit | | | setfac | | | | | |

Permissions

r read
w write
x execute (dir = search)
T sticky bit
t sticky bit + execute
S set uid / gid
s set uid / gid + execute

(sticky - load from MVS)

Audit

f failures
s successes
a all

Extended Attributes (only applies to programs)

a APF authorized
p bypass program control
s run shared address space
l load from shared library region

All

- null

LIST FILE & DIRECTORY FSP - ls

ls [-AadEgIMnoRW] [*pathname*]

- A** List all entries include starting with periods, exclude . and ..
- a** List all entries
- d** Display information for directory itself, not contents
- E** Display extended attribute
- g** Display group but not owner
- l** Display permissions, links, owner, group, size, time, name
- M** Display Multilevel Security seclabel
- n** Display uid and gid number instead of owner and group
- o** Display owner but not group
- R** Display subdirectories recursively
- W** Display audit bits

**Authority required: r to directory and x to upper directories
(note: -R requires r-x access to all subdirectories)**

LIST FILE & DIRECTORY FSP - ls

First character indicates file type

File types

- **Regular file**
- b Block special file (not supported)**
- c Character special file**
- d Directory**
- e External link**
- l Symbolic link**
- p FIFO**
- s Socket file type**

LIST FILE & DIRECTORY FSP - ls

ls -alEW /etc

```
-rw-r--r--  fff--- --s-  1 OMVSKERN 2611      10 Sep 27 11:13 inetd.pid
-----  fff--- --s-  1 OMVSKERN 2611      2587 Oct 21  1999 init.options
lrwxrwxrwx  fff---          1 OMVSKERN OMVSGRP      22 Oct 23  1999 ioepdcf -> ../
etc/dfs/etc/ioepdcf
-rw-r--r--  fff--- --s-  1 OMVSKERN 2611     13573 May  3  2000 javelin.conf
drwxr-xr-x  fff---          2 2134      SYS1      8192 Jan 19  1999 ldap
-rw-r--r--  fff--- --s-  1 OMVSKERN 2611      2042 May  3  2000 lgw_fcgi.conf
-rw-r--r--  fff--- --s-  1 OMVSKERN 2611      2914 Sep 27 11:13 log
-rw-r--r--  fff--- --s-  1 OMVSKERN 2611      5144 May  3  2000 mvsds.conf
-----  fff--- --s-  1 OMVSKERN 2611     19683 Dec 11  2001 profile
-----  fff--- --s-  1 OMVSKERN 2611      2093 Apr 27  2000 rc
drwxrwxrwx  fff---          2 OMVSKERN 2611      8192 Oct 21  1999 recover
-rwxr-x---  fff--- --s-  1 OMVSKERN 2611       168 Apr 27  2000 resolve.conf
drwxr-xr-x  fff---          2 2134      SYS1      8192 Jan 19  1999 security
-rwxr-xr-x  fff--- --s-  1 OMVSKERN 2611      4703 Apr 27  2000 services
-rw-r--r--  fff--- --s-  1 OMVSKERN IMWEB      4189 May  3  2000 socks.conf
-rw-r--r--  fff--- --s-  1 WEBADM  IMWEB      4189 May  2  2000 socks.conf.exp
-rw-r--r--  fff--- --s-  1 OMVSKERN 2611       396 Oct  7 16:35 utmpx
drwxr-xr-x  fff---          2 2134      SYS1      8192 Jan 19  1999 zoneinfo
$
===>
```

INPUT

| | | | | | | |
|-------|-----------|----------|------------|------------|------------|-------------|
| ESC=¢ | 1=Help | 2=SubCmd | 3=HlpRetrn | 4=Top | 5=Bottom | 6=TSO |
| | 7=BackScr | 8=Scroll | 9=NextSess | 10=Refresh | 11=FwdRetr | 12=Retrieve |

ACCESS AUTHORIZATION - PERMISSIONS

File

- r** **Read** - read file
 - w** **Write** - update file
 - x** **Execute** - execute file (scripts and programs)
- Superuser cannot execute without 'x' authority

Directory

- r** **Read** - list files/subdirectories & their attributes (e.g., ACLs)
- w** **Write** - create, rename, and delete files/subdirectories
- x** **Search** - set as current (cd) and traverse directory

ACCESS AUTHORIZATION - PERMISSIONS

User's authority checked based on

- Effective UID (eUID)
- Effective GID (eGID)
- Supplemental Group GIDs (sGID)

Access to each object in requested directory path is checked

Execution of - `fopen("/web/httpd1/httpd.conf","rw")`

Results in the following permission checks

| | |
|---------------------------------|-----------------------------|
| <code>"/</code> directory | Search (x) directory access |
| <code>"web"</code> directory | Search (x) directory access |
| <code>"httpd1"</code> directory | Search (x) directory access |
| <code>"httpd.conf"</code> file | Read/Write (rw) file access |

AUDITOR authority grants r-x to all directories

UNIXPRIV CLASS - MODIFY AUTHORITY

RESTRICTED.FILESYS.ACCESS

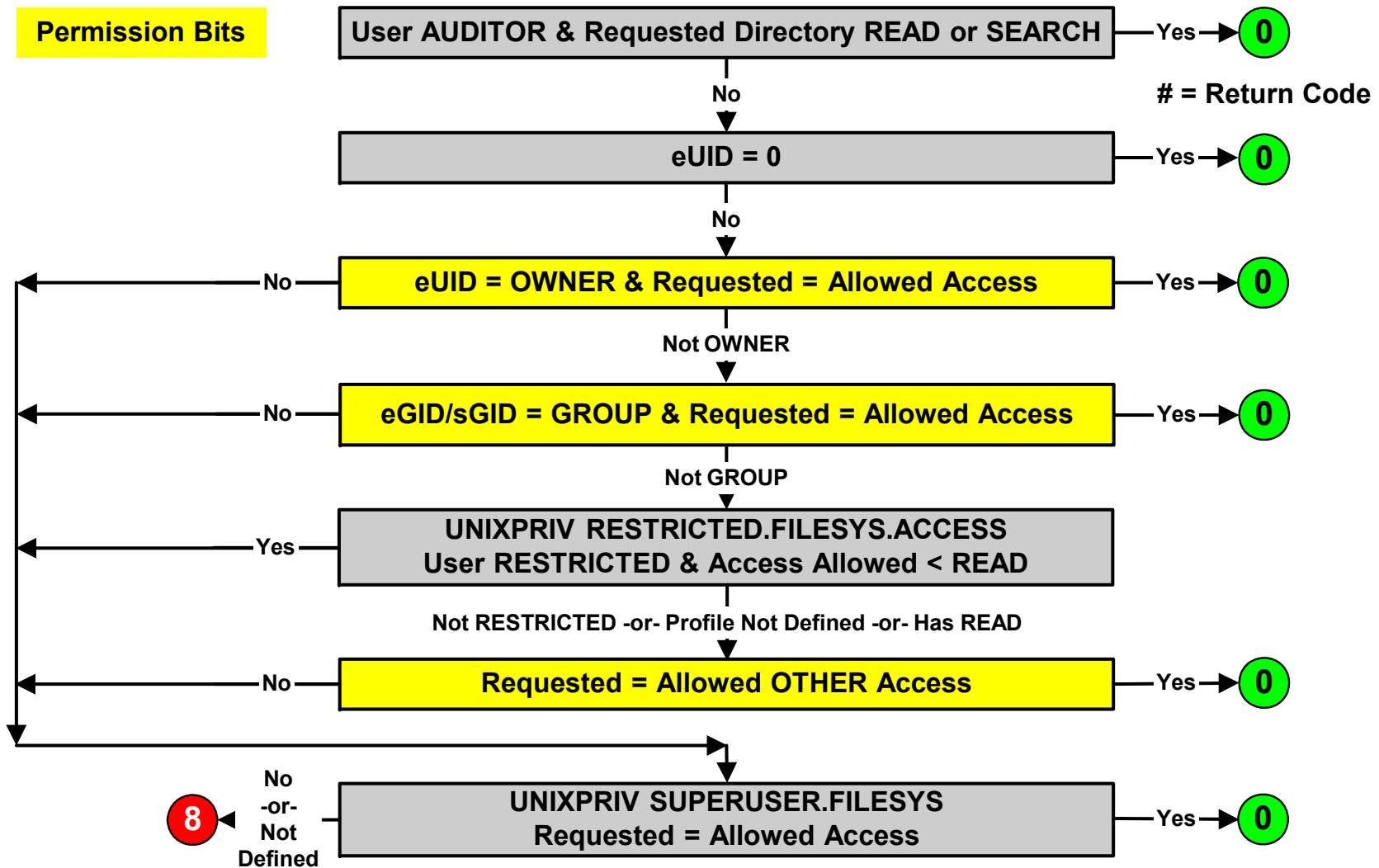
- Prohibits RESTRICTED users from obtaining access via OTHER permission bits
- Permitting READ (or greater) access bypasses the restriction

SUPERUSER.FILESYS

- Grants global access, even if denied access by FSP permission bits
- **READ** Read all files and search all directories
- **UPDATE** Write to any file
- **CONTROL** Write to any directory

ACCESS AUTHORIZATION - PERMISSIONS

Permission Bits



(c) 2008 RSH Consulting, Inc.

CHANGE OWNER - *chown*

chown* [-fhR] *owner[:group]* *pathname

- **f** Do not issue error message if cannot change
- **h** Do not follow symbolic link, change link itself
- **R** If directory, change all files and subdirectories as well,
starting at the top-most directory

***owner* can be ...**

- **USERID**
 - Must have a uid
 - If no uid, but BPX.DEFAULT.USER is defined, will assign default uid
- **uid**
 - Can be any number - need not be assigned to any USERID

***group* changes the GROUP (substitute for *chgrp*)**

CHANGE OWNER - chown

Authority required:

- **UID 0**
- **READ access to UNIXPRIV SUPERUSER.FILESYS.CHOWN and x access to the upper directories of the target file or directory**
- **File OWNER if UNIXPRIV CHOWN.UNRESTRICTED defined**

Turns off set-uid-bit and set-gid-bit

CHANGE OWNER - *chown*

| | |
|---|--|
| Change owner of file to rodger | chown rodger //lp/data/webprod.conf |
| Change owner of directory to scott | chown scott //lp/data |
| Change owner of all files & subdirectories to guy | chown guy //lp/data/* |
| Change owner of directory and all underlying files & subdirectories to scott | chown -R scott //lp/data |
| Change owner & group of directory to scott & rsh | chown scott:rsh //lp/sample |
| Change owner of file to uid 12 | chown 12 //lp/sample/dfil1.txt |

Note: -R requires r-x access to all subdirectories

CHANGE GROUP - *chgrp*

chgrp* [-fhR] *group pathname

- f Do not issue error message if cannot change
- h Do not follow symbolic link, change link itself
- R If directory, change all files and subdirectories as well,
starting at the top-most directory

***group* can be ...**

- Group name
 - Must have a gid
- gid
 - Can be any number - need not be assigned to any group

CHANGE GROUP - *chgrp*

Authority required:

- UID 0
- READ access to UNIXPRIV SUPERUSER.FILESYS.CHOWN and x access to the upper directories of the target file or directory
- File OWNER if either ...
 - Connected to target group
 - UNIXPRIV CHOWN.UNRESTRICTED defined

Turns off set-uid-bit and set-gid-bit

CHANGE GROUP - *chgrp*

| | |
|---|--|
| Change group for file to rgh | <code>chgrp rgh //l1p/data/webprod.conf</code> |
| Change group for directory to rsh | <code>chgrp rsh //l1p/data</code> |
| Change group for all files and subdirectories to rgh | <code>chgrp rgh //l1p/data/*</code> |
| Change group for directory and all underlying files & subdirectories to rsh | <code>chgrp -R rsh //l1p/data</code> |
| Change group for file to gid 1008 | <code>chgrp 1008 //l1p/sample/dfil1.txt</code> |

Note: `-R` requires `r-x` access to all subdirectories

CHANGE PERMISSIONS - chmod

Authority required:

- **UID 0**
- **READ access to UNIXPRIV SUPERUSER.FILESYS.CHANGEPERMS and x access to the upper directories of the target file or directory**
- **File OWNER**

Never changes permission on symbolic links as permissions are never used

CHANGE PERMISSIONS - *chmod*

Symbolic form: [*who*] *op permission ...*

who

- u** **Owner (User)**
- g** **Group**
- o** **Other**
- a** **All (default)**

op (operator)

- +** **Turn on specified permission(s)**
- **Turn off specified permission(s)**
- =** **Replace existing permissions with specified permissions**

CHANGE PERMISSIONS - *chmod*

permission

| | |
|------------------|--|
| r | Read |
| x | Execute file or search on directory |
| w | Write |
| s | For executable file, 'u' - set-uid-bit; 'g' - set-gid-bit |
| t | sticky bit - |
| file | Search for program in STEPLIB, LPA, Linklist |
| directory | Restrict delete & rename of file or subdirectory to file owner, directory owner, or superuser |

Can combine multiple symbolic forms by separating them with commas (,)

CHANGE PERMISSIONS - chmod

| | |
|---|--|
| Permit write to group | <code>chmod g+w /xml/msg</code> |
| Permit execute to owner & group | <code>chmod u+x,g+x /xml/xxg</code> |
| Delete write to owner | <code>chmod u-w /u/rodger</code> |
| Replace existing permits with ones specified | <code>chmod u=rwx,g=rx,o=x /appl/test</code> |
| Permit write to owner to all files and subdirectories | <code>chmod u+w /appl/test/*</code> |
| Permit read to group to directory and all underlying files & subdirectories | <code>chmod -R g+r /appl/test</code> |

Note: -R requires r-x access to all subdirectories

CHANGE PERMISSIONS - chmod

Octal - positional - [aug]o (attribute,owner,group,other)

- 0** No permissions
- 1** Execute/Search (or Sticky bit)
- 2** Write (or set-gid-bit)
- 3** Execute & Write [1 + 2]
- 4** Read (or set-uid-bit)
- 5** Execute & Read [1 + 4]
- 6** Read & Write [2 + 4]
- 7** Read, Write, & Execute [1 + 2 + 4]

chmod 2754 /llp/data/pgmx

- Attribute - Set-gid-bit
- Owner - Read, Write, & Execute
- Group - Read & Execute
- Other - Read

chmod 710 /tmp/logdir

- Owner - Read, Write, & Search
- Group - Search
- Other - No access

CHANGE EXTENDED ATTRIBUTES - *extattr*

extattr [+alps] [-alps] [-F *format*] *file ...*

Attributes

- a** **APF-authorized**
- l** **Load from share library region**
- p** **Bypass program-control**
- s** **Run shared address space**

Authority required:

- **FACILITY Class profile - READ**
 - **BPX.FILEATTR.APF** **Set APF authorization on HFS file**
 - **BPX.FILEATTR.SHARELIB** **Set shared library extended attribute**
 - **BPX.FILEATTR.PROGCTL** **Set program control attribute on HFS file**
- **x access to the parent directory of the target file**
- **w access to target file**

FSP FOR NEW FILE OR DIRECTORY

OWNER

- **Creator's effective UID**

GROUP (next slide)

Permissions based on umask

- **Specified**
 - */etc/profile* **System-wide options z/OS shell**
 - */etc/csh.cshrc* **System-wide options tcsh shell**
 - */users-home/.profile* **User's personal options**
 - **_BPX_BATCH_UMASK** **BPXBATCH execution**
 - **umask command** **User specified options (& display current umask)**
- **OR'ed to create permissions (subtract from 777)**
- **Ex: umask(022) equivalent to chmod 755**
- **Recommended: umask(077) equivalent to chmod 700**

FSP GROUP INHERITANCE

Standard Unix behavior - GROUP for FSP taken from parent Directory in which new subdirectory or file is created

New optional behavior - GROUP for FSP taken from effective gid in USP of the creating process

UNIXPRIV FILE.GROUPOWNER.SETGID

- **Existence of profile acts as switch to activate**
- **Behavior depends on set-gid bit for the parent directory**
 - **If bit OFF (default) - GROUP taken from USP**
 - **If bit ON - GROUP taken from Directory as before**
 - **Must use 'chmod' command to turn on set-gid bit for directory in order for it to revert to original behavior**
 - **'ls' display shows 's' ('x' on) or 'S' ('x' off) in x-bit place for GROUP**

Currently running processes do not recognize the change

EXTENDED ACCESS CONTROL LIST (ACL)

Extension to base (original) file and directory permissions

Activated by SETR CLASSACT(FSSEC)

Max entries - 1024

Supports inheritance of access controls - default ACLs

Commands

- **getfac1** **List base permissions & ACLs**
- **setfac1** **Administer base permissions & extended ACLs**

EXTENDED ACCESS CONTROL LIST (ACL)

```
$ ls -al rshtest
```

```
total 352
```

```
drwxr----x    4 RSH      SYS1          8192 Oct 29 23:13 .
drwxrwxrwt    3 OMVSKERN SYS1        24576 Oct 29 23:17 ..
drwxr-xr-x+   2 RSH      SYS1          8192 Oct 29 23:13 rshdirx
drwxrwxrwx    2 OMVSKERN SYS1          8192 Oct 29 02:47 rshtest2
-rwxr-xr-x+   1 RSH      SYS1       127910 Oct 29 22:48 sampfile
```

"+" indicates presences of extended ACL

EXTENDED ACL - getfacI

getfacI [-acdfhos] [-e *user*] *file-directory ...*

- a **Displays access ACL (default)**
- c **Displays ACL on single line with commas between entries**
- d **Displays directory default ACL**
- f **Displays file default ACL**
- h **Does not resolve symbolic links**
- o **Displays only extended ACL entries (not base ACL)**
- s **Skips files that only have a base ACL**
- e **Displays only the ACL entries which affect *user* access**

Authority required: r to directory and x to upper directories

EXTENDED ACL - getfacl

```
$ getfacl sampfile
#file:  sampfile
#owner:  RSH
#group:  SYS1
user::rwx
group::r-x
other::r-x
user:RLW:r-x
group:LEVEL1:--x
```

EXTENDED ACL - getfacl

```
$ getfacl -adf rshdirx
#file: rshdirx/
#owner: RSH
#group: SYS1
user::rwx
group::r-x
other::r-x
user:RLW:rwx
user:$OEDFLU:--x
group:LEVEL1:r-x
fdefault:group:LEVEL1:--x
default:user:RLW:r-x
```

EXTENDED ACL - setfacI

setfacI [-ahqv] -s|m|x *entries* [*path*]

-S|M|X *file* [*path*]

-D *type* [*path*]

- a Abort if error occurs during processing
- h Do not follow symbolic links
- q Quiet mode, suppress superfluous error messages
- v Verbose

***path* assumes current directory unless full path is specified**

Authority required:

- UID 0
- READ access to UNIXPRIV SUPERUSER.FILESYS.CHANGEPERMS and x access to the upper directories of the target file or directory
- File OWNER

EXTENDED ACL - setfacl

Options

- **s entries** Sets (replaces) all ACLs with entries
- **S file** Sets (replaces) all ACLs with entries in file
- **m entries** Modifies ACL with specified entries
- **M file** Modifies ACL with entries in file
- **x entries** Deletes ACL entries
- **X file** Deletes ACL entries in file
- **D type** Deletes all extended ACL of specified *type*
 - a Access ACL
 - d Directory default ACL
 - f File default ACL
 - e All ACLs

EXTENDED ACL - setfacl

Base ACL permission bit entries (must specify all 3 for -s or -S)

| | |
|------------------------------------|---------------|
| u[ser]::<i>permissions</i> | u::rwx |
| g[roup]::<i>permissions</i> | g::r-x |
| o[ther]::<i>permissions</i> | o::--- |

Extended ACL entries

| | |
|---|----------------------|
| u[ser]:<i>userid/uid[:permissions]</i> | u:rsh:r-x |
| g[roup]:<i>group/gid[:permissions]</i> | g:racfadm:rwx |

Default Extended ACL entries - Directory & File

| | |
|---|------------------------|
| d[efault]:<i>directory-ACL-entry</i> | d:g:racfadm:rwx |
| f[efault]:<i>file-ACL-entry</i> | f:u:rsh:r-x |

Permissions specified either absolute (r-x) or relative (+rx or ^rx)

EXTENDED ACL - setfac1

| | |
|---|--|
| Replace ACL for file FX | <code>setfac1 -s u::rwx,g::r-x,o::- ,u:bob:r-x FX</code> |
| Add permission to FX | <code>setfac1 -m g:dev9:r-x FX</code> |
| Deny sam access to directory pay | <code>setfac1 -m u:sam:--- /appl/pay</code> |
| Set default directory ACL for pay | <code>setfac1 -m d:u:sue:rwx /appl/pay</code> |
| Set default file ACL for files in pay | <code>setfac1 -m f:g:acct:r-x /appl/pay</code> |
| Delete ACL entry from file FX | <code>setfac1 -x u:sarahn /prod/rsh/FX</code> |
| Copy ACL from file Z to file Y (lone "-" denotes stdin) | <code>getfac1 Z setfac1 -S - Y</code> |

EXTENDED ACL - setfac1

| | |
|---|---|
| Delete Default Directory ACL for pay | setfac1 -D d /appl/pay |
| Delete all ACLs for all subdirectories and files in directory pay | setfac1 -D e /appl/pay/* |
| Add entries to directory pay specified in file payacl | setfac1 -M payacl /appl/pay |
| Add ACL entry to all subdirectories beneath projx | setfac1 -m u:deb:r-x \$(find /projx/* -type d) |
| Add ACL entry to directory pay and all underlying files and subdirectories | setfac1 -m u:dave:r-- \$(find /appl/pay) |

Note: \$(find) requires r-x access to all subdirectories

UNIXPRIV CLASS - MODIFY AUTHORITY

RESTRICTED.FILESYS.ACCESS

- Prohibits RESTRICTED users from obtaining access via OTHER permission bits
- Permitting READ (or greater) access bypasses the restriction

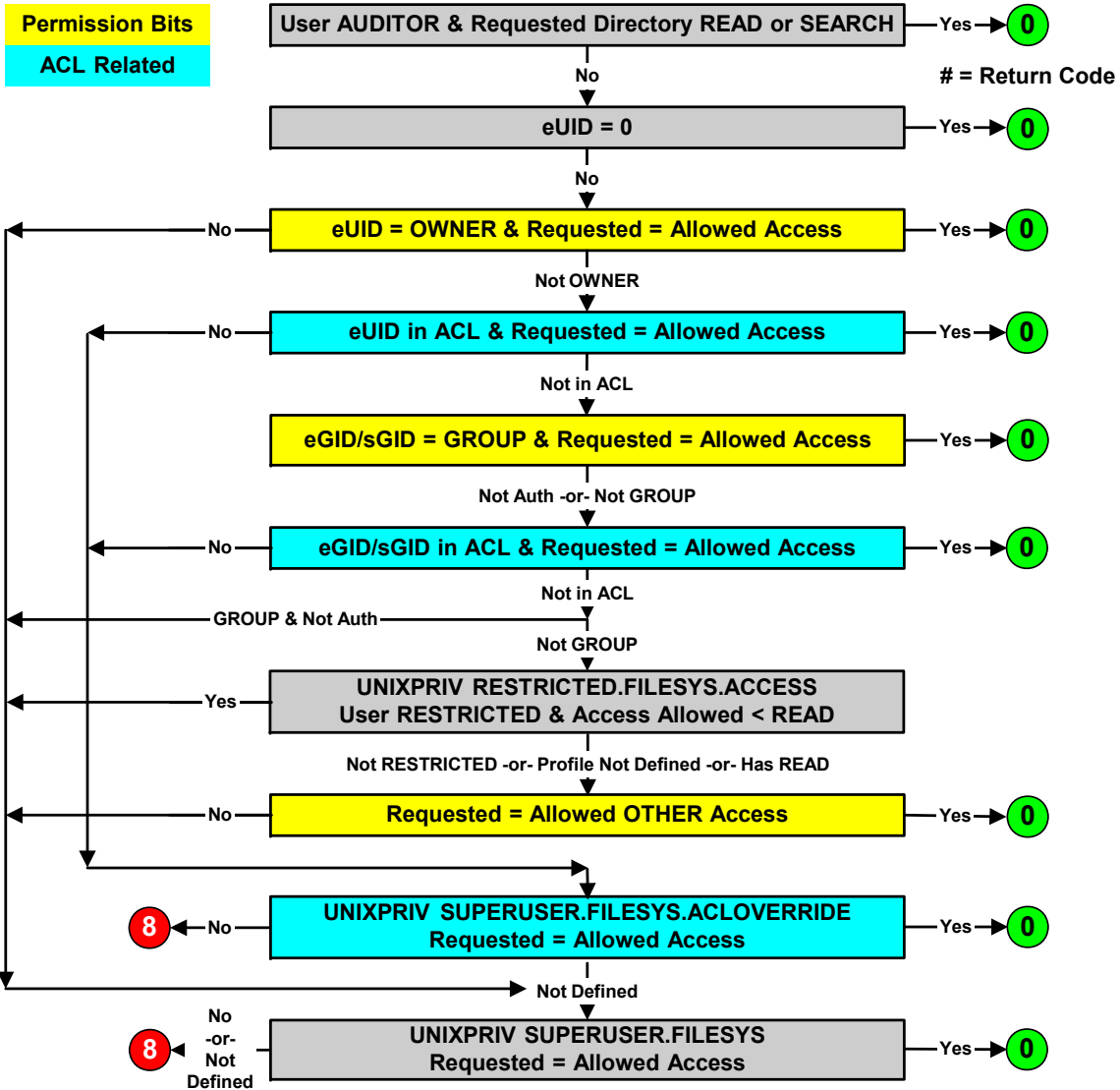
SUPERUSER.FILESYS.ACLOVERRIDE

- Causes User or Group ACL permissions to override SUPERUSER.FILESYS
- Used to explicitly deny access SUPERUSER.FILESYS would otherwise grant
- Permitting access at the level otherwise specified for SUPERUSER.FILESYS overrides the override

SUPERUSER.FILESYS

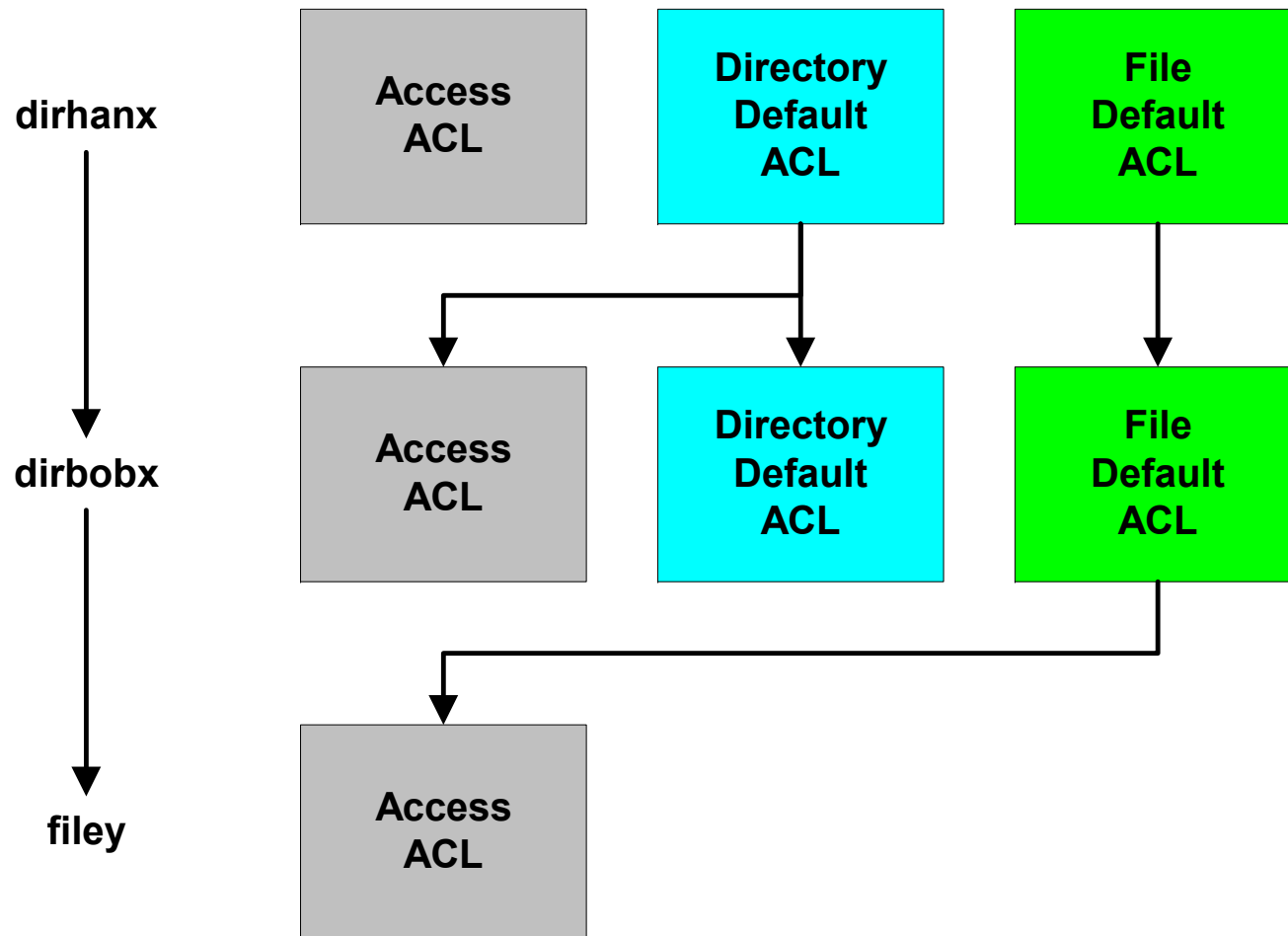
- Grants global access, even if denied access by FSP permission bits or ACL
- READ Read all files and search all directories
- UPDATE Write to any file
- CONTROL Write to any directory

ACCESS AUTHORIZATION - ACL



(c) 2008 RSH Consulting, Inc.

ACL INHERITANCE



ACL INHERITANCE

/dirhanx

user:bob:rwx

group:systems:r-x

fdefault:group:techsup:rwx

default:group:techsup:rwx

/dirhanx/dirbobx

group:techsup:rwx

fdefault:group:techsup:rwx

default:group:techsup:rwx

/dirhanx/dirbobx/file

group:techsup:rwx

FSP & ACLs - cp (copy) & mv (move)

cp - when copying a file or directory into another directory, Unix treats the target as if its a new object being created

- Normal inheritance for owner and group
- Permission bits set based on umask
- ACLs set based on defaults for target directory

mv - when moving a file or directory into a directory, Unix keeps all of the original attributes unchanged

- Owner, Group, Permission bits, & ACLs retained as is
- Useful if transferring large directory structure from one upper level directory to another

If developers are creating files and directories under test or personal directories for subsequent transfer to production directories, advise them to cp as opposed to mv

FINDING SECURITY PERMISSIONS

ACLs of given type

```
find path -acl type                type = a | d | f
```

ACL entries

```
find path -acl_user userid  
find path -acl_group group  
find path -acl_entry text
```

ACLs with n or more entries

```
find path -acl_count n
```

777 permissions

```
find path -perm 777 -a -type type    type = f | d
```

(note: if omit *type*, will list all symbolic links)

IMPLEMENTATION CONSIDERATIONS

(IBM) Rely on lack of x permission to upper level directory to deny access to subdirectories, where permissions can be relaxed

(RSH) Rely more on ACLs and less on permission bits

- **Do not rely on Owner for granting access (creator could be anyone with w to the directory)**
- **Do not use Group permission-bits for granting access - set to a 'place-holder' group with no users or permissions**
- **Set all default umasks to 077**
- **Use Other for UACC-type access only as needed**
- **Grant access to all users and groups who need access via ACLs, even when such users might also be Owners**
- **Set default file and directory ACLs to maintain permissions, generally they will match the ACL permissions**

IMPLEMENTATION CONSIDERATIONS

(RSH) If individuals maintaining Unix file security do not need Root/Superuser authority for other duties, grant them the following access to administer security (consider granting directory r-w access to RACF Admin staff, too)

```
UNIXPRIV SUPERUSER.FILESYS.CHANGEPERMS ACCESS(READ)
```

```
UNIXPRIV SUPERUSER.FILESYS.CHOWN ACCESS(READ)
```

- and either -

```
setfacl -m g:SECGRP:r-x $(find / -type d)
```

```
setfacl -m d:g:SECGRP:r-x $(find / -type d)
```

- or -

```
System-AUDITOR (with APAR OA28648)
```

(RSH) Permit FACILITY BPX.SUPERUSER users READ access to UNIXPRIV SUPERUSER.FILESYS (reduce need to 'su' to root)

Use IBM's IRRHFSU HFS unload utility to list and manage FSPs & ACLs (see IBM's RACF website for a copy)

MONITORING

FSP Audit bits

SETROPTS LOGOPTIONS & AUDIT

FACILITY BPX.SAFFASTPATH

UAUDIT

UNIXPRIV

Events Always Audited

SMF Records & Reporting

FSP AUDIT BITS

Two sets of audit specifications & defaults

- Owner's fff
- Auditor's ---

Interpretation of audit settings as displayed in 'ls -W' output

- f Failure (access not permitted)
- s Success (access permitted)
- a All (both)
- - None

Owner and Auditor settings work in combination to provide the most inclusive auditing

find command options -audit (Owner) and -audit (AUDITOR) can identify FSPs with specific audit settings

CHANGE AUDITING - *chaudit*

chaudit [-Fdai] *attr pathname*

- F Change audit settings on all files in a directory
- d Change audit settings on all subdirectories in a directory
- a Change Auditor audit settings
- i Do not issue error message if cannot change

Authority required:

- Owner's settings - UID 0 -or- File OWNER
- Auditor's settings - AUDITOR authority

CHANGE AUDITING - *chaudit*

attr form: [*operation*] *op* *auditcondition* ...

operation (if not specified, applies to all)

- r Read attempts
- w Write attempts
- x Execute attempts

op (operator)

- + Turn on specified audit condition(s)
- Turn off specified audit condition(s)
- = Replace existing settings with specified audit conditions

auditcondition

- s Audit successful access
- f Audit failed access

CHANGE AUDITING - *chaudit*

| | |
|---|--|
| Audit failures for all types of access | chaudit +f /appl/pay/checkfile |
| Audit successful write access | chaudit w+s /appl/pay/checkfile |
| Remove auditing for successful access | chaudit -s /appl/pay/* |
| Audit all successful access via Auditor bits | chaudit -a +s /* |

Note: There is no +/-a for *auditcondition*. The audit bit is automatically set to 'a' if both 'f' and 's' are specified collectively. Similarly, it is set to '-' if neither 'f' nor 's' are specified.

MONITORING

SETROPTS LOGOPTIONS(*level* (*class*)) - log USS events

- DIRSRCH Directory searches
- DIRACC Directory read/write access
- FSOBJ File & Directory access
- FSSEC File system security changes
- PROCESS Process uid or gid changes and privileged operations
- PROCACT Functions effecting other processes
- IPCOBJ Object access, uid or gid changes

Recommendation:

- SETR LOGOPTIONS(ALWAYS(FSSEC))
- SETR LOGOPTIONS(FAILURES(PROCESS PROCACT IPCOBJ))

LOGOPTIONS(ALWAYS(PROCACT)) required to log 'getpsent'

MONITORING

SETROPTS AUDIT(*class*)

- **F**SOBJ creations and deletions of file system objects
- **I**PCOBJ creations and deletions of objects (e.g., semaphores)
- **P**ROCESS dubbing and undubbing of a process

FACILITY BPX.SAFFASTPATH

- If defined, eliminates RACF calls and related logging if Unix can determine access is allowed (e.g., allowed by FSP permission bits)
- Intended to improve performance
- Can prevent tidal wave of SMF records during rebuilds and upgrades
- If defined after IPL, must issue SETOMVS or SET OMVS operator command to activate
- Do not define if using IRRSXT00 exit to control HFS access
- Does not suppress logging for users with UAUDIT

MONITORING

UAUDIT honored by UNIX

- Can generate massive amounts of SMF records - a DIRSRCH record for each directory access down through the directory tree

UNIXPRIV - can log successes, but not failures

- Except for profile SHARED.IDS, RACROUTE LOG=NOFAIL is used
- To monitor, specify RALT UNIXPRIV *profile* AUDIT(SUCCESS(*level*))

Events always audited

- Attempt to create (i.e., dub) a process by user with missing or incomplete OMVS segment
- Dubbing a process using the BPX.DEFAULT.USER user uid
- Unauthorized attempt to mount or unmount a file system

Auditing produces SMF Type 80 records

Use output from SMF Unload utility (or 3rd party product) to report events (RACFRW provides incomplete results)

REFERENCES

IBM z/OS manuals

- **UNIX System Services Planning**
- **UNIX System Services Command Reference**
- **Security Server RACF Security Administrator's Guide**
- **Security Server RACF Auditor's Guide**

IBM RACF Presentation webpage

- **www.ibm.com/servers/eserver/zseries/zos/racf/presentations.html**

IBM RACF Downloads webpage (IRRHFSU)

- **www.ibm.com/servers/eserver/zseries/zos/racf/goodies.html**

Internet Discussion List - mvs-oe