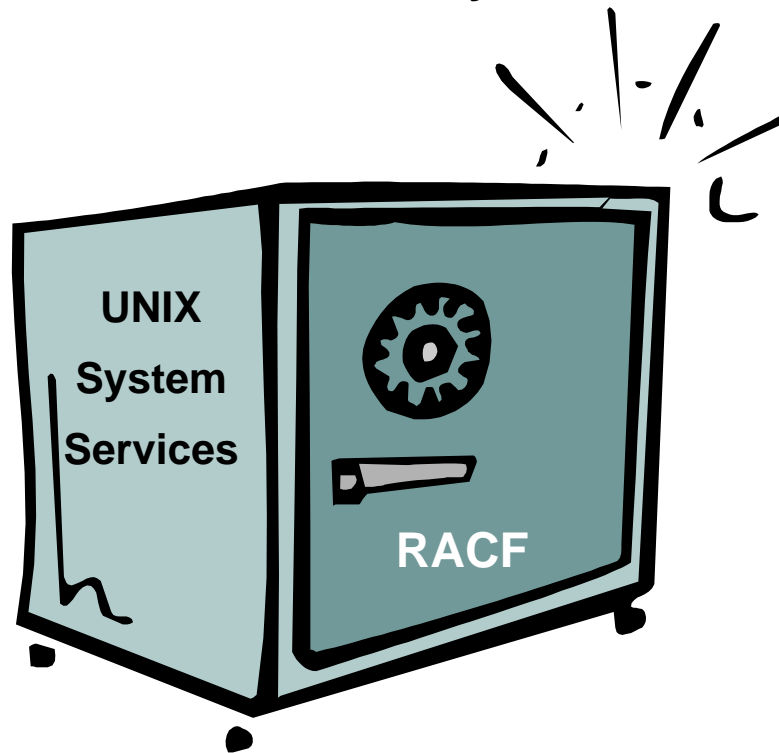


RACF & Unix System Services

KOIRUG - May 2004



Robert S. Hansel

RACF Specialist - RSH Consulting, Inc.

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

TOPICS

Introduction to Unix System Services

User & Group Identifiers

Superuser & Unix Privileges

Controlling Daemons & Servers

Monitoring

RACF, OS/390, and z/OS are Trademarks of the International Business Machines Corporation

UNIX SYSTEM SERVICES (USS)

POSIX-compliant UNIX address space

First introduced in MVS/ESA 5.1

Core z/OS - OS/390 component - part of base product

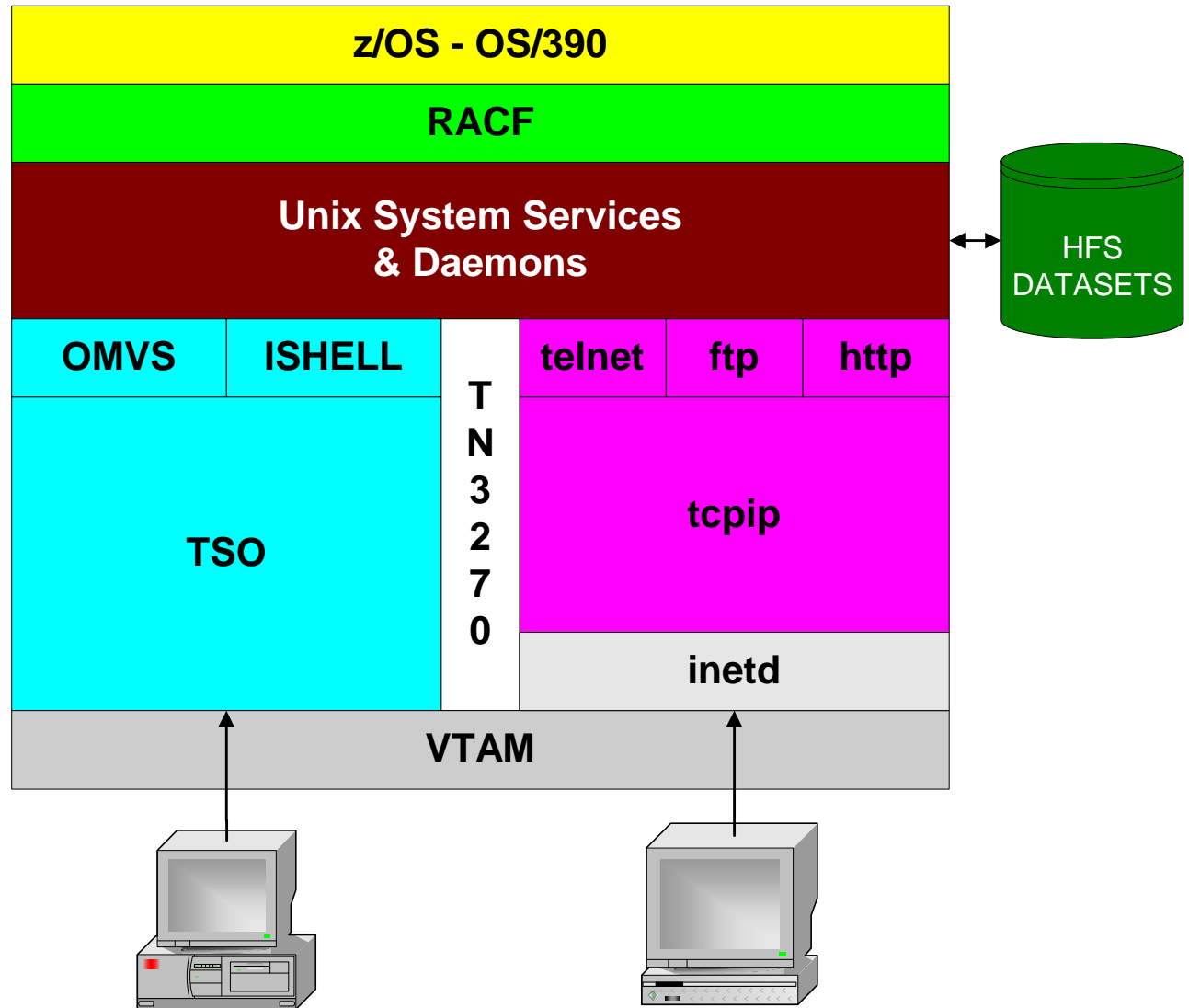
Provides integrated Unix services to MVS

- **Programs in Unix can be invoked from MVS and vice versa**
- **Files / Datasets can be accessed via either environment**

Required to support TCP/IP services (2.6)

Formerly OpenEdition/MVS (OMVS) - now a.k.a. z/OS Unix

UNIX SYSTEM SERVICES



UNIX SYSTEM SERVICES

Runs as a separate address space

Starts at IPL as part of system initialization

- **OMVS PROC**

```
//OMVS PROC
```

```
//OMVS EXEC      PGM=BPXINIT,REGION=0K,TIME=NOLIMIT
```

- **BPXOINIT PROC**

```
//BPXOINIT PROC
```

```
//BPXOINIT EXEC PGM=BPXPINPR,REGION=0K,TIME=NOLIMIT
```

- **Typically assigned ID OMVSKERN (Manual / SAMPLIB examples)**

Initialization PARMs

- **PARMLIB(IEASYSxx) OMVS=xx**
- **PARMLIB(BPXPRMxx)**

BPXPRMxx - PARMs

STARTUP_PROC(<u>OMVS</u> <i>procname</i>)	USS startup Started Task Procedure
MAXUIDS(<u>200</u> 1 - 32767)	Maximum currently active uids allowed
SUPERUSER(<u>BPXROOT</u> <i>userid</i>)	Default Daemon setuid(0) user unknown
TTYGROUP(<u>TTY</u> <i>groupid</i>)	Group given slave pseudo-terminals
AUTHPGMLIST(' <i>path/file</i> ' NONE)	File listing APF auth programs
ROOT FILESYSTEM(' <i>fs-dsname</i> ')	Identifies Root File System DSN
TYPE(HFS)	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(' <i>fs-dsname</i> ')	Identifies File System DSN to mount
TYPE(HFS)	Specifies File System type
MOUNTPOINT(' <i>pathname</i> ')	Directory where to be mounted
MODE(<u>RDWR</u> READ)	Access allowed (RDWR = read/write)
<u>SETUID</u> NOSETUID	Support setuid() & setgid() mode bit
<u>SECURITY</u> NOSECURITY	Perform security checks

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

BPXPRMxx - SAMPLE

MAXUIDS(200)

SUPERUSER(OMVSKERN)

ROOT FILESYSTEM('HFS.PLEXID.ROOT')

 TYPE(HFS) MODE(RDWR)

MOUNT FILESYSTEM('HFS.SYSID.HFS')

 TYPE(HFS) MODE(RDWR) NOAUTOMOVE

 MOUNTPOINT('/SYSID')

MOUNT FILESYSTEM('HFS.SYSID.ETC')

 TYPE(HFS) MODE(RDWR) NOAUTOMOVE

 MOUNTPOINT('/SYSID/etc')

MOUNT FILESYSTEM('HFS.TMP')

 TYPE(HFS) MODE(RDWR) NOAUTOMOVE

 MOUNTPOINT('/tmp')

MOUNT FILESYSTEM('HFS.USERS')

 TYPE(HFS) MODE(RDWR)

 MOUNTPOINT('/u')

HIERARCHICAL FILE SYSTEM (HFS)

Contains Unix directory structure and files

Unique DSORG type

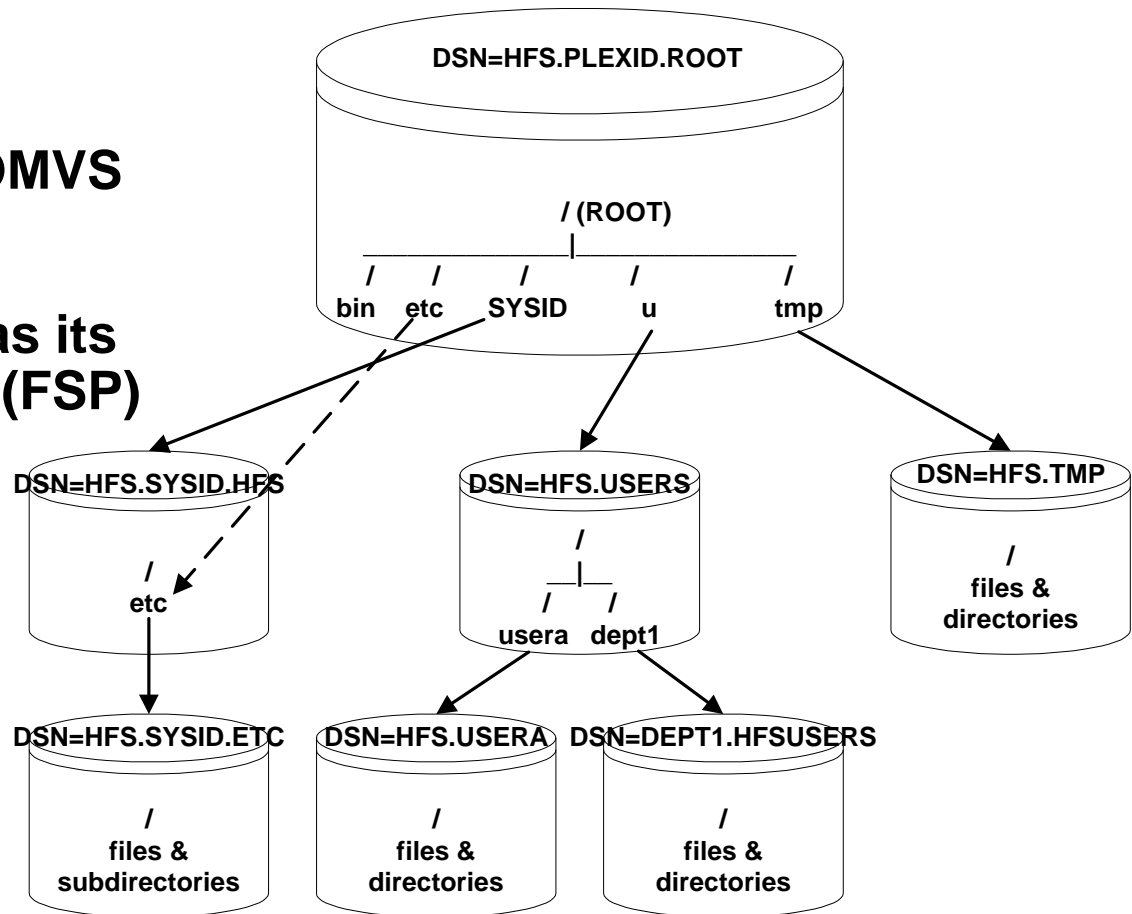
Limit access to just the OMVS Started Task

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

Names are mixed-case

Common directories

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



HIERARCHICAL FILE SYSTEM (HFS)

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
```

HIERARCHICAL FILE SYSTEM (HFS)

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

Owner uid	Group gid	set uid	set gid	sticky bit	Owner			Group			World			Auditing						Extended Attributes		
					Read	Write	Execute	Read	Write	Execute	Read	Write	Execute	Owner			Auditor			APF	Prog Cntrl	Run Sh
chown	chgrp				chmod									chaudit						extattr		

Security

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

a APF authorized
 p program-controlled
 s run shared address space
 l load from shared library region

All

- null

DAEMON

Program providing a service

Usually initiated as a Started Task at IPL

Long-running, unattended process

USS & Communications Server Daemons

- **inetd** **internet daemon**
- **rlogind** **remote login daemon**
- **cron** **batch scheduler**
- **uucpd** **Unix-to-Unix Copy Program (UUCP)**
- **syslogd** **message routing**
- **ftpd** **FTP server**
- **httpd** **HTTP server**

DAEMON

Provides service by ...

- **Identifying the user requesting services**
- **Creating a new process to perform the work (fork / spawn)**
- **Assigning the user's identity to the new process**

Runs authorized and as Superuser

Calls USS services to assign uid and RACF USERID to process

- **BPX1SEU seteuid() Set effective uid**
- **BPX1SUI setuid() Set uid**
- **BPX1SPN spawn() Spawn with USERID**
- **BPX1PWD Change password**
- **BPX1SEC Create security environment**

Can become any user with an OMVS segment or any other user if BPX.DEFAULT.USER is defined

SERVER

Program providing a service

Usually initiated as a Started Task at IPL

Long-running, unattended process

Provides services by ...

- **Identifying the user requesting services**
- **Creating a new thread to perform the work**
- **Associating the user's identity to the new thread**

May (or may not) run authorized or as Superuser

SERVER

Verify user identity

- RACF ID/password
- Application ID/password
- Digital Certificate

Calls USS services to associate client's ID with the thread

- BPX1TLS pthread_security_np()

Calls USS services to check client's access authority to resources

- BPX1ACK auth_check_resource_np()

USS SECURITY LEVELS

Unix Security

- POSIX file and directory access rule bits are used - but are processed by RACF
- Users defined to and validated by RACF
- FACILITY BPX.DAEMON not defined
- ROOT / uid(0) Rules! - equal to SPECIAL, OPERATIONS, AUDITOR

z/OS Security - managed externally by RACF

- Posix file and directory access rule bits are used - but are processed by RACF
- Users defined to and validated by RACF
- FACILITY BPX.DAEMON defined
- Requires program controlled environment
- Control over Superuser and Daemon authorities

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

OMVS SEGMENT

ADDUSER / ALTUSER

```
OMVS( ASSIZEMAX( address-space-size )  
      AUTOUID | UID( user-identifier ) [ SHARED ]  
      CPUTIMEMAX( cpu-time )  
      FILEPROCMAX( files-per-process )  
      HOME( initial-directory-name )  
      MMAPAREAMAX( memory-map-size )  
      PROCUSERMAX( processes-per-UID )  
      PROGRAM( program-name )  
      THREADSMAX( threads-per-process ) )
```

```
ALU USMITH01 OMVS( UID(123) PROGRAM(/bin/sh) HOME(/u/usmith01) )
```

ADDGROUP / ALTGROUP

```
OMVS( AUTOGID | GID( group-identifier ) [ SHARED ] )
```

PREVENT UID & GID SHARING

Intended to keeping uids and gids unique

UNIXPRIV SHARED.IDS (must be discrete)

Profile acts as a switch to activate this feature

Prevents assignment of existing uid or gid

Must be at Application Identity Mapping level 2 or 3 to implement

Can be overridden using the SHARED keyword with commands creating or altering the OMVS segment (e.g., multiple uid 0)

Authority to specify SHARED

- **System-SPECIAL**
- **READ access to SHARED.IDS**

AUTOMATIC UID & GID ASSIGNMENT

Automatically generates uids and gids to ensure uniqueness

**FACILITY BPX.NEXT.USER APPLDATA('starting-uid-value/
starting-gid-value') - e.g., (1000/500) or (2500-6000/1-100)**

Invoked with AUTOUID and AUTOUID keywords in commands effecting OMVS Segment - e.g., ALU *userid* OMVS(AUTOUID)

If an APPLDATA value is null or 'NOAUTO', no assignments for that type of id are done - e.g., APPLDATA(NOAUTO/2200)

RACF automatically increments APPLDATA as it assigns ids

With RRSF, explicit uid/gid value passed - establish different ranges for each system

Requires shared id prevention be implemented first

USS DEFAULT USER

Certain TCP/IP applications require a uid and gid for every user

Default User alternative to assigning each user OMVS segment

FACILITY BPX.DEFAULT.USER APPLDATA('userid/groupid')

Create matching ID and group with OMVS segments

Assign unique uid (not 0) and gid with no shell and home directory of /tmp; make ID PROTECTED, RESTRICTED, & REVOKED

Default uid and gid assigned when no OMVS segment specified

- Overridden by assigning uid and/or gid via OMVS segment
- OMVS segment with no uid or gid specified nullifies USS identity

Some applications still require user & logon group have OMVS segments

USER SECURITY PACKET (USP)

USP is created when Unix service is invoked

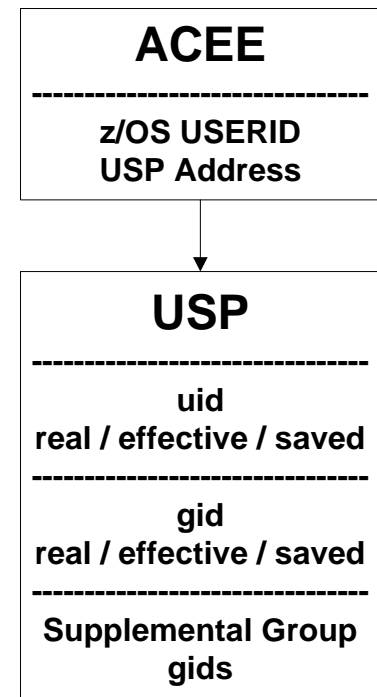
uid/gid obtained from OMVS segment or
BPX.DEFAULT.USER

'Effective' uid/gid used for USS directory and
file access authorization

Maximum of 300 supplemental groups

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

```
id ibmuser
uid=0(OMVSKERN) gid=0(SYS1)
```



OMVS SIGNON CONTROL

Used to restrict who can access OMVS

APPL OMVSAPPL

Requires READ access (UACC or access list) to use USS

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**
- **Can change directory and file security bits**

Alternative - UNIXPRIV profiles - limited / tailored authority

CONTROLLING DAEMONS

Daemons alter address space uid and associated RACF ID

FACILITY BPX.DAEMON controls use of calls

- BPX1SEU seteuid() Set effective uid
- BPX1SUI setuid() Set uid
- BPX1SPN spawn() Spawn with USERID
- BPX1PWD Change password
- BPX1SEC Create security environment

Permit READ access only to OMVS and 'trusted' Daemons that must set identities without prompting for password

Permit UPDATE access to allow use of BPX1SEC

Requires program controlled environment

CONTROLLING SERVERS

Servers set uids on threads

FACILITY BPX.SERVER controls use of calls

- **BXP1TLS** `pthread_security_np()`
- **BPX1ACK** `auth_check_resource_np()`

Authority to access resources is partially governed by level of access server's ID is permitted to BPX.SERVER

- **UPDATE** Resource access granted only on authority of user
- **READ** Resource access granted only if both server and user have authority (server is untrusted)
 - **SURROGAT** authority required to represent user
 - Supports anonymous users

Requires program controlled environment

PROGRAM CONTROLLED ENVIRONMENT

PROGRAM class profile(s) cover programs from MVS Libraries

- **Some IBM Daemon programs have sticky bit on - must use MVS program (STEPLIB, Link Pack Area (LPA), Link List concatenation)**

Controlled-program extended attribute set on Unix programs

- **To identify, execute command 'find / -ext a'**
- **BPXPRMxx - AUTHPGMLIST('path/file') - file listing APF programs**

FACILITY BPX.DAEMON.HFSCTL

- **Allow load of any MVS programs - bypass PROGRAM profile check**
- **Will only result in checks of HFS APF attribute**

FACILITY BPX.MAINCHECK

- **Used with Enhanced Program control**
- **Requires initial program be marked "MAIN" & loaded from MVS**

PROGRAM CONTROLLED ENVIRONMENT

Recommended PROGRAM profiles (support USS and PADS)

PROGRAM ** UACC(READ) ADDMEM(...) [APPLDATA('MAIN')]

'SYS1.LINKLIB'//NOPADCHK

'SYS1.MIGLIB'//NOPADCHK

'SYS1.COMDLIB'//NOPADCHK

'SYS1.LINKLIB'//NOPADCHK

'cee.version.SCEERUN'//NOPADCHK

'tcpip.SEZALINK'//NOPADCHK

'tcpip.SEZATCP'//NOPADCHK

'ftp.userexits'//NOPADCHK

'db2.DSNLOAD'//NOPADCHK

'db2.DSNEXIT'//NOPADCHK

PROGRAM ICHDSM00 UACC(NONE) ADDMEM(...) DATA('DSMON')

'SYS1.LINKLIB'//NOPADCHK

Permit access to Security Administrators, Auditors, RACF Tech Support

PROGRAM IRRDPI00 UACC(NONE) ADDMEM(...) DATA('LOAD RACF PARSING TABLE')

'SYS1.LINKLIB'//NOPADCHK

Permit access to IRRDPTAB and RACF Started Tasks

PROGRAM IEHINITT UACC(NONE) ADDMEM(...) DATA('INITIALIZE TAPES')

'SYS1.LINKLIB'//NOPADCHK

Permit access to Tape Librarians or others who initialize tapes

SETUID & SETGID PROTECTION

Programs run under authority of the owner or group (as identified in the FSP) rather than those of the invoker

- **Uses variants of setuid & setgid function calls**
- **Grants users temporary authority beyond their normal authority**
- **Programs have execute bit set to 's' under owner and/or group**
- **Changes invoking user's 'effective' uid/gid (but not RACF USERID)**
- **Example - 'su' command**

Bit set using 'chmod' command

- **Must be file owner or have SUPERUSER privilege to set**
- **Changing file or its owner resets bits**

Mounting file system with NOSETUID ignores this bit

- **Appropriate for remote or untrusted file systems**

SURROGATE CONTROL

Allow user to switch identity to another user without a password

SURROGAT BPX.SRV.*userid* ACC(READ)

Typically used with untrusted server processes or to set up anonymous type IDs

Permitted user can issue 'su' command

- `su -s userid` -s option bypasses password prompt
- Changes 'effective' uid and RACF USERID

FILE EXTENDED ATTRIBUTE CONTROL

Authority to change program extended attributes

Controls use of 'extattr' command

Needed to apply maintenance without uid 0

FACILITY Class profiles

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

OTHER FACILITY CLASS BPX AUTHORITIES

BPX.JOBNAME **Set job name for new process (_BPX_JOBNAME)**

BPX.SMF **Allowed to write SMF record (BPX1SMF)**

BPX.STOR.SWAP **Make address space non-swappable (BPX1ENV)**

BPX.DEBUG **Run ptrace to debug APF programs (BPX1PTR)**

BPX.WLMSEVER **Access to WLM functions (BPX1SIN, BPX1WLM)**

BPX.MAP **Use storage mapping services (BPX1MMI)**

BPX.SHUTDOWN **Special treatment at shutdown (BPX1ENV)**

BPX.CP **Use Coupling Facility sizer tool (_cpl())**

UNIXPRIV CLASS

Allows limited delegation of Unix Superuser privileges

Checked after all other authorities - uid(0), permissions, TRUSTED, etc.

SUPERUSER.FILESYS

- Grants global access at given permit level, even if denied access per FSP
- READ Read all files and search all directories
- UPDATE Write to any files
- CONTROL Write to any directory

SUPERUSER.FILESYS.ACLOVERRIDE

- If defined, causes User or Group permissions to supercede access granted by SUPERUSER.FILESYS
- Used to explicitly deny access SUPERUSER.FILESYS would otherwise grant
- However, permitting access to this profile at a given level overrides the override

UNIXPRIV CLASS

RESTRICTED.FILESYS.ACCESS

- Existence of profile acts as switch to activate
- RESTRICTED users cannot gain access via OTHER permission bits

SUPERUSER.FILESYS.CHANGEPERMS

'chmod' any permits

SUPERUSER.FILESYS.CHOWN

'chown' any file or directory

SUPERUSER.FILESYS.MOUNT

- 'mount' & 'chmount' HFS files
- READ With NOSETUID only
- UPDATE With SETUID or NOSETUID

SUPERUSER.FILESYS.QUIESCE

- 'quiesce' & 'unquiesce' HFS
- READ With NOSETUID only
- UPDATE With SETUID or NOSETUID

UNIXPRIV CLASS

SUPERUSER.FILESYS.PFCTL	Physical File System services
SUPERUSER.FILESYS.VREGISTER	Register as VFS server
SUPERUSER.IPC.RMID	Release IPC resources ('ipcrm')
SUPERUSER.PROCESS.GETPSENT	Get process status info
SUPERUSER.PROCESS.KILL	Issue kill to processes
SUPERUSER.PROCESS.PTRACE	
<ul style="list-style-type: none">• Use ptrace function through dbx debugger• Also requires access to BPX.DEBUG with APF or BPX.SERVER process	
SUPERUSER.SETPRIORITY	Increase own priority
CHOWN.UNRESTRICTED	
<ul style="list-style-type: none">• Existence of profile acts as switch to activate• Allow any user to 'chown' their files & directories to any other user	

FSP GROUP INHERITANCE

Standard Unix behavior - GROUP for FSP taken from 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 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 eXecute bit for GROUP**

Currently running processes do not recognize the change

IDENTITY MAPPING

Required to look up RACF user and group ID with only USS uid

UNIXMAP Class

- **Profiles - Unn (user uids) and Gnn (group gids) - e.g., U340**
- **RACF automatically maintains, even if class is not active**
- **Access list indicates users and groups having these ids**
- **Must activate UNIXMAP before profiles are interrogated**
- **When displaying corresponding user or group, first one on the access list is the one shown**

Application Identity Mapping (AIM)

- **New index structure available since OS/390 2.10**
- **Replaces UNIXMAP class (and others)**
- **Requires RACF database be converted to use the new structure**
- **Enables use of SEARCH UID() GID() keywords**

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

SETROPTS AUDIT(*class*)

- FSOBJ creations and deletions of file system objects
- IPCOBJ creations and deletions of objects (e.g., semaphores)
- PROCESS dubbing and undubbing of a process

MONITORING

User UAUDIT attribute honored

AUDITOR authority

- Needed to issue 'chaudit' command to change audit 'auditor' settings
- Grants search and read access to all directories and FSPs

FACILITY BPX.SAFFASTPATH

- Bypasses logging of successful access events to improve performance
- 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
- Can prevent tidal wave of SMF records during rebuilds and upgrades

MONITORING

UNIXPRIV - can log successes, but not failures

Events always audited

- **Attempt to create process by user with missing or incomplete OMVS segment**
- **Creation of process using BPX.DEFAULT.USER OMVS segments**

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 - GA22-7800**
- **UNIX System Services Command Reference - SA22-7802**
- **UNIX System Services Programming: Assembler Callable Services Reference - SA22-7803**
- **C/C++ Run-Time Library Reference - SA22-7821**

IBM RACF Presentation website

- **www-1.ibm.com/servers/eserver/zseries/zos/racf/presentations.html**

Internet Discussion List - mvs-oe

IBM UNIX Tools & Toys website

- **www-1.ibm.com/servers/eserver/zseries/zos/unix/bpxa1toy.html**