RSH CONSULTING

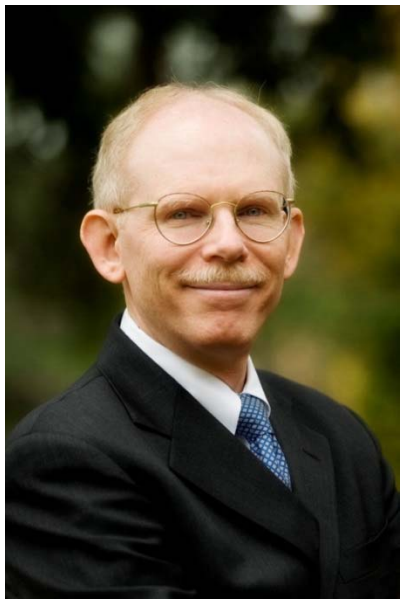
RACF UNIXPRIV Class

GARUG - May 2017

RSH Consulting - Robert S. Hansel

RSH Consulting, Inc. is an IT security professional services firm established in 1992 and dedicated to helping clients strengthen their IBM z/OS mainframe access controls by fully exploiting all the capabilities and latest innovations in RACF. RSH's services include RACF security reviews and audits, initial implementation of new controls, enhancement and remediation of existing controls, and training.

- www.rshconsulting.com
- 617-969-9050



Robert S. Hansel is Lead RACF Specialist and founder of RSH Consulting, Inc. He began working with RACF in 1986 and has been a RACF administrator, manager, auditor, instructor, developer, and consultant. Mr. Hansel is especially skilled at redesigning and refining large-scale implementations of RACF using role-based access control concepts. He is a leading expert in securing z/OS Unix using RACF. Mr. Hansel has created elaborate automated tools to assist clients with RACF administration, database merging, identity management, and quality assurance.

- 617-969-8211
- R.Hansel@rshconsulting.com
- www.linkedin.com/in/roberthansel
- http://twitter.com/RSH_RACF

UNIXPRIV Class

- Allows delegation of specific Unix Superuser privileges as an alternative to assigning full Superuser authority

- Superuser (root - Unix system administrator) privileges
  - Full access to all Unix directories and files (like TRUSTED)
  - Can change directory and file OWNER, GROUP, and permissions (like SPECIAL)
  - Can perform privileged Unix functions (e.g., mount)
  - Can use privileges related to most unprotected FACILITY BPX resources and to some protected ones even without permission (e.g., BPX.CONSOLE)
  - If BPX.DAEMON is not defined, can initiate processes with other user's identities

- Superuser authority assigned by …
  - OMVS(UID(0)) — Unix and Daemon Started Tasks
  - FACILITY - BPX.SUPERUSER — Tech Support staff and Daemons
  - PRIVILEGED / TRUSTED Started Tasks — Still require UID and GID assignment

UNIXPRIV Class - RACF CDT Entry

ID	= 1	DFTRETC	= 4
POSIT	= 555	DFTUACC	= NONE
		OPER	= NO
MAXLNTH	= 246		
FIRST	= ANY	GENLIST	= DISALLOWED
OTHER	= ANY	RACLIST	= ALLOWED
KEYQUAL	= 0	RACLREQ	= YES

- RACLREQ=YES (RACLIST REQUIRED) is specified because access to most UNIXPRIV resources is checked using FASTAUTH processing
  - If the class is not RACLISTed, it is treated as if it is inactive
  - The Global Access Table (GLOBAL) cannot be used to grant access because FASTAUTH processing ignores it

UNIXPRIV Class - Security Administration

- SUPERUSER.FILESYS.CHANGEPERMS - 'chmod' and 'setfacl' any permit
- SUPERUSER.FILESYS.CHOWN - 'chown' and 'chgrp' any file or directory
  - READ access required
  - Also require x (search) authority to upper directories of target directory or file
  - Permit security administration staff access to CHANGEPERMS and CHOWN instead of permitting them access to BPX.SUPERUSER

- SHARED.IDS
  - Prevents duplicate assignment of existing UID or GID to keep them unique
    - Attempts to assign a UID or GID already in use result in an IRR52174I message
    - Does not control use of shared UIDs or GIDs
  - Existence of profile acts as switch to activate restriction - must be Discrete
  - Requires Application Identity Mapping (AIM) Stage 2 or 3
  - Can be overridden using SHARED keyword when creating or changing OMVS segment

```
ADDUSER FTPD OMVS( UID(0) SHARED )
```

  - Requires System-SPECIAL or READ access to use SHARED keyword
  - Required to implement FACILITY BPX.NEXT.USER
  - Considered essential if delegating OMVS UID administration via FIELD class resource USER.OMVS.UID to prevent inappropriate assignment of UID(0)

UNIXPRIV Class - Security Administration

- CHOWN.UNRESTRICTED
  - Existence of profile acts as switch to activate functionality - must be Discrete profile
  - Enables a file or directory OWNER to change the object's OWNER and GROUP
    - READ Change OWNER to any ID with UID(non-0) or change GROUP to any group
    - UPDATE Change OWNER to ID with UID(0)

- FILE.GROUPOWNER.SETGID
  - Existence of profile acts as switch to activate - must be Discrete
  - Changes method of GROUP inheritance for new files and directories
    - Standard Unix behavior - GROUP taken from parent Directory
    - New optional behavior - GROUP taken from 'effective' GID of creating user's User Security Packet (USP)
  - Behavior depends on SETGID bit for the parent directory
    - If bit OFF (default) - GROUP taken from USP
    - If bit ON - GROUP taken from parent Directory as before
    - Must use 'chmod' command to turn on SETGID bit for directory in order for it to revert to original behavior
    - 'ls' display shows 's' ('x' on) or 'S' ('x' off) for the x-Search-bit for GROUP (e.g., rwxr-s--x)
  - Currently running processes do not recognize the change - they will require a restart

UNIXPRIV Class - Maintenance

- SUPERUSER.FILESYS.MOUNT
  - 'mount', 'chmount', and 'unmount' zFS files
  - READ With NOSETUID
  - UPDATE With SETUID
- SUPERUSER.FILESYS.QUIESCE
  - 'quiesce' and 'unquiesce' zFS files
  - READ With NOSETUID
  - UPDATE With SETUID
- Limit access to Tech Support staff responsible for maintaining UNIX
- Permit Started Task DFHSM UPDATE access to SUPERUSER.FILESYS.QUIESCE if it backs up Unix File Systems (this is an alternative to assigning it UID(0))

- SUPERUSER.FILESYS.USERMOUNT
  - Restricted 'mount' to be permitted to non-Unix administrators
  - Mounts file system with SECURITY and NOSETUID
  - READ access required, plus directory rwx access to both the mount point and file system root

UNIXPRIV Class - Commands and Callable Services

SUPERUSER.FILESYS.PFSCTL	Use Physical File System services - pfsctl()
SUPERUSER.FILESYS.VREGISTER	Register as VFS server (e.g. NFS) - vreg()
SUPERUSER.IPC.RMID	Release IPC resources - ipcrm command
SUPERUSER.PROCESS.GETPSENT	Get status of any process - w_getpsent()
SUPERUSER.PROCESS.KILL	Terminate any process - kill()
SUPERUSER.PROCESS.PTRACE	Use trace through dbx debugger - ptrace()
SUPERUSER.SETPRIORITY	Increase own priority - setpriority()
SUPERUSER.SHMMCV.LIMITS	Increase shared memory variables and mutexes

- All require READ permission to use
- Typically grant access to UNIX processes or to users performing debugging

UNIXPRIV Class - Commands and Callable Services

- Require SUPERUSER.PROCESS.GETPSENT
  - WebFocus IADMIN user
  - Tivoli System Automation (Netview) - if not assigned UID(0)
  - Peoplesoft
  - Users of " ps " command (process status) - can list dubbed IDs and find their attributes [useful for RACF administrators]

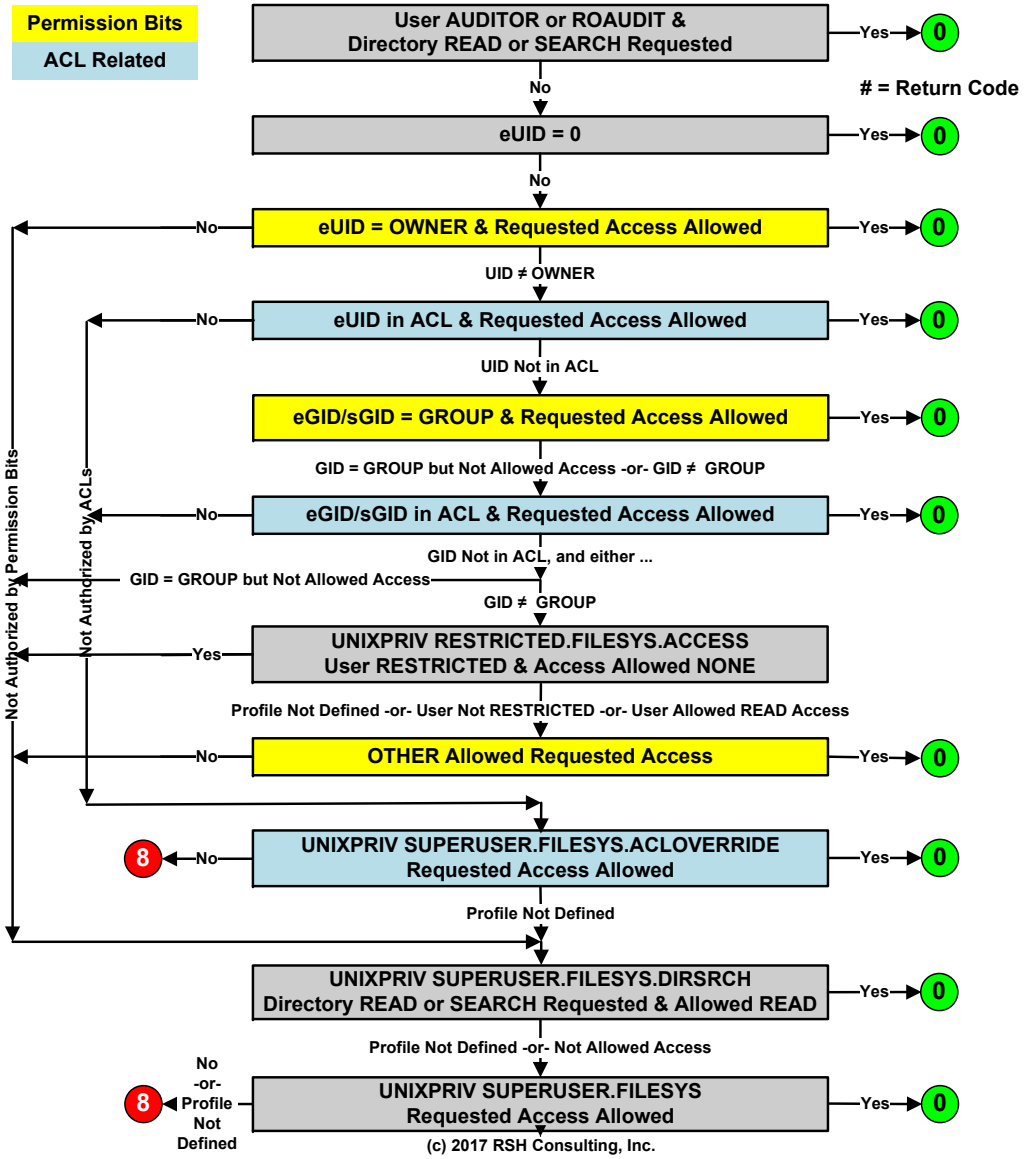
- Require SUPERUSER.FILESYS.PFSCTL
  - SAP's sap-system-identifier-ADM ID
  - CA Mainframe Chorus for DB2
  - Non-Superusers executing zFS utilities and commands (e.g., zfsadm)

- To debug daemons, users need access to ...
  - SUPERUSER.PROCESS.GETPSENT
  - SUPERUSER.PROCESS.KILL
  - SUPERUSER.PROCESS.PTRACE
    - Also requires access to FACILITY profile BPX.DEBUG to trace processes running with either APF-authorization or BPX.SERVER authority

UNIXPRIV Class - Access

- **SUPERUSER.FILESYS**
  - Grants access to all Unix files and directories at specified permit level, even if denied access by permission bits and Access Control Lists (ACLs) [unless ACLOVERRIDE is defined]
    - READ Read all files and search all directories
    - UPDATE Write to any file
    - CONTROL Write to any directory
  - Can replace UID(0) with SUPERUSER.FILESYS access
    - READ Sterling (now CA) Solve NetMaster
    - READ Tivoli Asset Discovery - Inquisitor
    - READ Tivoli System Automation (Netview)
    - CONTROL Tivoli Directory Server (LDAP) [or substitute with specific permits]
- **SUPERUSER.FILESYS.ACLOVERRIDE**
  - Causes ACL permissions to overrule access SUPERUSER.FILESYS would otherwise grant
  - Permissions grant access like SUPERUSER.FILESYS
- **SUPERUSER.FILESYS.DIRSRCH (z/OS 2.2)**
  - Permitting READ access grants READ and SEARCH (r-x) authority to all directories
  - Designed to be paired with SUPERUSER.FILESYS.CHANGEPERMS and .CHOWN in lieu of AUDITOR or ROAUDIT authority
- **RESTRICTED.FILESYS.ACCESS**
  - Prohibits RESTRICTED users from gaining access via OTHER permission bits
  - Permitting READ access bypasses the restriction

UNIXPRIV Class - Access



UNIXPRIV Class

- UNIXPRIV is checked after other authorities (AUDITOR, ROAUDIT, eUID(0), permission bits, ACLs) and grants access not otherwise allowed
  - Generally cannot be used to block or limit other authorities
- Monitoring and logging
  - UNIXPRIV access is checked using LOG=NOFAIL; to monitor use, specify either ...

    RALTER UNIXPRIV *profile* AUDIT(SUCCESS(READ),FAILURES(READ))

    ALTUSER *userid* UAUDIT
  - Cannot monitor UNIXPRIV access with SETROPTS LOGOPTIONS inmost cases
    - Bypassed due to the use of FASTAUTH processing
  - To see w_getpsent events, set SETROPTS LOGOPTIONS(ALWAYS(PROCACT))
- Best Practices:
  - Use UNIXPRIV permissions as an alternative to assigning full Superuser authority whenever possible
  - Permit Technical Support users READ access to SUPERUSER.FILESYS when they also have FACILITY BPX.SUPERUSER access
  - Do not define a catch-all * or ** profile