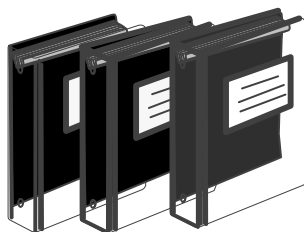




RACF & OS/390 UNIX SYSTEM SERVICES OVERVIEW



Debbie Mapes
RACF Development



What is OS/390 UNIX System Services?

- Product formerly known as OpenEdition
- Base element of OS/390
- UNIX interface for MVS providing
 - Hierarchical File System (HFS) containing directories and files
 - Application Interfaces
 - Commands
- MVS externals with a UNIX feel
 - Installation and Service
 - Administration
 - Operation



What is it for?

- Makes application development easier
 - Standard (open) programming interface
 - Interoperability in networks
 - Portable programs
 - Portable data
- Required by some products



How is it related to RACF?

- External security product is required
- User identification and authentication
- Protection of services
- Protection of files
- Auditing of security events



What needs to be done?

User definition

- Users must be defined to RACF
- User profiles need OMVS segments
 - UID - 0 to 2147483647
 - HOME - current working directory
 - PROGRAM - initial program to execute
- UID required, others will default
- Current connect group needs GID
- UID and GID should be unique



User Definition ...

TSO LOGON
AG UNIXGR..
ALU ADMIN..

```
ADDGROUP UNIXGRP OMVS(GID(100))
ALTUSER ADMIN OMVS(UID(1) HOME(/u/admin)
PROGRAM(/bin/sh)) !!!! Note the mixed case !!!!
CONNECT ADMIN GROUP(UNIXGRP)
ADDUSER JOHN PASSW(xxxx) DFLTGRP(UNIXGRP)
OMVS(UID(2) HOME(/u/john))
TSO(ACCTNUM(12345) PROC(PROC01))
LISTUSER JOHN OMVS NORACF
USER=JOHN
OMVS INFORMATION
-----
UID = 0000000002
HOME = /u/john
```



User Definition ...

- Panels can be used
- List-of-groups applies to resource access only
- Ensure uniqueness of UIDs and GIDs
 - Use a value that's already unique (Serial Number)
 - Use the ISHELL
 - Use sample DBunload reports
- Delegate with the FIELD class
 - Consider an OS/390 UNIX administrator to assign UIDs and GIDs
 - Allow users to list their own info and change some of it



User Definition ...

TSO LOGON
AU UXADM...
SETR CLASS...

```
ADDUSER UXADM PASSW(yyyyyy) DFLTGRP(UNIXGRP)  
OMVS(UID(1000)) TSO(...)  
SETR CLASSACT(FIELD) GENERIC(FIELD)  
RDEF FIELD USER.OMVS.* UACC(NONE)  
RDEF FIELD USER.OMVS.PROGRAM UACC(NONE)  
PE USER.OMVS.* CL(FIELD) ID(UXADM) ACC(UPDATE)  
PE USER.OMVS.PROGRAM CL(FIELD) ID(UXADM)  
ACC(UPDATE)  
PE USER.OMVS.* CL(FIELD) ID(&RACUID) ACC(READ)  
PE USER.OMVS.PROGRAM CL(FIELD) ID(&RACUID)  
ACC(UPDATE)  
SETR RACLIST(FIELD)
```


User Definition ... SUPERUSER!

- Defined in a number of ways
 - BPX.SUPERUSER, any UID, any GID
 - UID 0, any GID
 - Trusted or privileged, any UID, any GID
- A superuser can:
 - Pass all OS/390 UNIX security checks
 - Change his identity to another UID
 - Use setrlimit to increase system limits
- Not used when accessing MVS resources
- No special meaning for GID 0



What needs to be done?

System Setup


- Define the kernel and initialization proc
 - Create user and group with UID and GID
 - Create STARTED class profiles (or use ICHRIN03 table)
- Protect data sets containing the HFS
- Define other started procedures such as RMFGAT and TCP/IP
- Define BPX.DEFAULT.USER
- Define BPX.SUPERUSER



System Setup ...

TSO LOGON
AG OMVSG...
AU OMVSK...

```
ADDGROUP OMVSGRP OMVS(GID(1))  
AU OMVSKERN DFLTGRP(OMVSGRP) PASSWORD(xyz)  
OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))  
NAME('OMVS KERNEL')  
SETR GENERIC(STARTED)  
RDEF STARTED OMVS.* STDATA(USER(OMVSKERN)  
GROUP(OMVSGRP) TRUSTED(YES))  
RDEF STARTED BPXOINIT.* STDATA(USER(OMVSKERN)  
GROUP(OMVSGRP) TRUSTED(NO))  
SETR CLASSACT(STARTED) RACLIST(STARTED)
```



System Setup ...

```
TSO LOGON
AG UXDFL...
AU UXDFL...
```

```
AG UXDFLTG OMVS(GID(999))
AU UXDFLTU DFLTGRP(UXDFLTG) PASSWORD(abc)
  OMVS(UID(999)) NAME('DEFAULT UNIX USER')
RDEF FACILITY BPX.DEFAULT.USER
  APPLDATA('UXDFLTU/UXDFLTG')
RDEF FACILITY BPX.SUPERUSER UACC(NONE)
AG SUPERUSE OMVS(GID(3))
PERMIT BPX.SUPERUSER CLASS(FACILITY) ID(SUPERUSE)
  ACCESS(READ)
SETR CLASSACT(FACILITY) RACLIST(FACILITY)
  ...OR...
SETR RACLIST(FACILITY) REFRESH
```

What needs to be done?

Other BPX profiles

- BPX.DAEMON - restricts the use of sensitive services
- BPX.DEBUG - allows debugging of authorized programs
- BPX.FILEATTR.APF - controls marking files authorized
- BPX.FILEATTR.PROGCTL - controls marking files program controlled
- BPX.SERVER - restricts the use of sensitive services
- BPX.SMF - allows the writing of SMF records
- BPX.STOR.SWAP - controls making address spaces non-swappable
- BPX.WLMSEVER - controls access to WLM interface
- BPX.SAFFASTPATH - improves performance but prevents auditing of successful events



What needs to be done?

Protecting Files

- Occurs automatically
- No profiles in RACF database
- File Security Packets stored with file
- Access control by permission bits
- 3 types of permissions
 - Owner - owning UID (creating user)
 - Group - owning GID (group owning directory)
 - Other - similar to UACC
- 3 levels of authority - read, write, execute (search)



Protecting Files ...

- TSO or Shell commands, or ISHELL
 - OSHELL mkdir project2
 - TSO MKDIR '/u/myid/project2'
 - ISHELL to bring up ISPF shell menu
- umask sets permission bit mask to override defaults during file creation
- chown changes file owner
 - superuser authority required
- chgrp changes file group
 - superuser or owner required
- chmod changes permission bits



Protecting Files ...


TSO LOGON
OMVS
ls -l

```
.s -l
lrwxr-x--- 2 USER1 UNIXGRP 0 Jun 1 08:45 user1
.rwxr--r-- 1 USER1 UNIXGRP 88 Jun 2 10:20 mydata
.rwx----- 1 USER1 UNIXGRP 50 Jun 2 10:22 myfile.c
```

 **PERMISSION BITS !!!!**

```
!kdir -m 700 user1sub1
!hmod a=rx myfile.c
!hown user1:statgrp /u/user1/mydata
```

```
.s -l
lrwxr-x--- 2 USER1 UNIXGRP 0 Jun 1 08:45 user1
.rwxr--r-- 1 USER1 STATGRP 88 Jun 2 10:20 mydata
.r-xr-xr-x 1 USER1 UNIXGRP 50 Jun 2 10:22 myfile.c
lrwx----- 1 USER1 UNIXGRP 0 Jun 3 09:10 user1sub1
```



Protecting Files ...

```
ICH408I USER(USER3) GROUP(UXGRP4) NAME(OOPS)  
/U/PRIV/PRG.C CL(FSOBJ) FID(01C7D5D9D3F0F100010400007000)  
INSUFFICIENT AUTHORITY TO OPEN  
ACCESS INTENT(RW-) ACCESS ALLOWED(OTHER --X)
```

USER3 tried to open this file for READ and WRITE access

The owner of this file wasn't USER3

UXGRP4 wasn't the owning group for this file

USER3 didn't belong to the group that owns this file

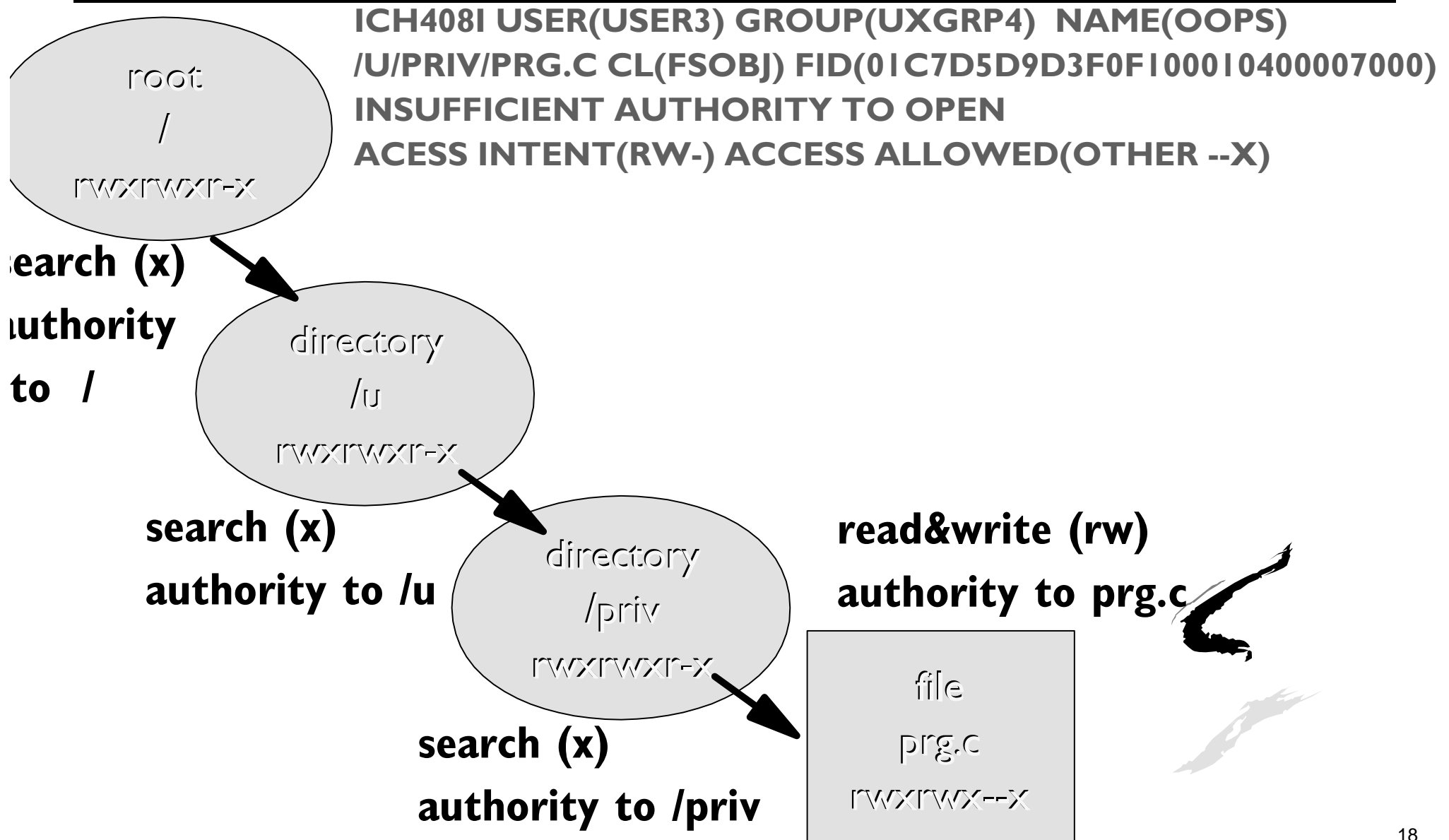
The other permissions only allow execute access

Auditing of failures was set for this file,

or with SETR for class FSOBJ



Protecting Files ...



What needs to be done?

Audit Setup

- Controlled by audit classes
 - SETR LOGOPTIONS, SETR AUDIT
 - DIRSRCH, DIRACC, FSOBJ, FSSEC
 - PROCESS, PROCACT
 - CLASSACT/NOCLASSACT has no effect
- Missing OMVS segment always audited
- Failing mounts/unmounts always audited
- Files have user and auditor options
 - Similar to AUDIT() and GLOBALAUDIT()
 - Set with chaudit, not ALTDSD or RALT



Audit Setup ...

The results

- Type 80 SMF records and ICH408ls
- ICH408ls for resources and services

```
ICH408I USER(SYS) GROUP(TST) NAME(OOPS)  
CLASS(PROCESS)  
OMVS SEGMENT NOT DEFINED
```

- Settings can cause excessive records
SETR LOGOPTIONS(ALWAYS(...))
- No write-to-programmers are issued
- RACFRW information is incomplete
- Use SMF data unload



The ISPF Shell ... A Panel Interface

- User-friendly way to do administration
 - Create and set up the file system
 - Display file system attributes
 - Mount/Unmount the file system
 - Change attributes for OS/390 UNIX users
 - Display a list of users, sorted by UID, GID
 - Print a list of users
 - Set up OS/390 UNIX users and groups
- Normal RACF authority checking applies



What new terms have we heard?

- kernel - address space containing the OS/390 UNIX services
- shell - a program that interprets input text as commands
- HFS - hierarchical file system
- root - the first directory in the system
- directory - a container for files and subdirectories
- file - a container for data and programs
- superuser - passes all UNIX security checks



What do we need to remember?

- Read the security chapters of the OS/390 UNIX System Services Planning manual for YOUR release level (SC28-1890)
- To set up security for an OS/390 UNIX application, read its documentation for recommendations
- Use the ISPF Shell (ISHELL) if you don't like that UNIX feel
- It's still RACF and MVS under the surface

