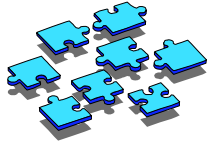




Controlling OS/390 UNIX System Services Daemons and Servers with RACF

Vanguard Enterprise Security Expo 2000
Session 86



Walt Farrell
RACF Design, IBM
914-435-7750
wfarrell@us.ibm.com




Disclaimer

The information contained in this document is distributed on an "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 or registered trademarks of the International Business Machines Corporation:
 - OS/390
 - RACF
 - SecureWay
- UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.



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 (Callable Services)
 - Commands (Shells and Utilities)
 - MVS externals with a UNIX feel
 - Makes application development easier
 - Portable programs and data
 - Interoperability in networks
- 

What is a Daemon?

- A program that starts at initialization time (a started task or cataloged procedure)
- A long-lived process, running unattended
- A service provider
- An authorized, superuser process
- Daemons perform work for users by:
 - Verifying the user requesting the service
 - Creating a new process to do the work
 - Giving the new process the user's identity



What is a Daemon?

- Daemons supplied with OS/390 UNIX:
 - inetd - the internet daemon
 - rlogind - the remote login daemon
 - cron - the batch scheduler
 - lm - the Communications Server login monitor
 - uucpd - the UUCP (UNIX-to-UNIX Copy Program) daemon
- Daemons supplied with IBM Communications Server for OS/390
 - syslog - message routing daemon



How Does a Daemon Change Identity?

- Daemon programs call identity changing services to alter the UID and RACF user ID of an address space
 - seteuid()
 - setuid()
 - spawn() with a user ID
- Daemons can become any user that has an OMVS segment defined (any user if you have BPX.DEFAULT.USER setup)



Controlling Daemons ... Selecting your level of security

- UNIX-level security
 - A superuser is equivalent to a daemon
 - Superusers can change UNIX & MVS identities
 - Superusers can access all resources
 - A superuser is equivalent to a system programmer



Controlling Daemons ... Selecting your level of security

- OS/390 UNIX-level security
 - A superuser is not equivalent to a daemon
 - Superusers cannot change MVS identity
 - Superusers cannot access MVS resources
 - A superuser maintains the file systems



Controlling Daemons ... OS/390 UNIX-Level Security

- Activated by defining FACILITY BPX.DAEMON
- Restricts the use of identity changing services
- Only trusted daemons should be given authority
- Daemon programs must be protected
 - Sticky bit on in file system copy
 - SYS1.LINKLIB protected with PROGRAM profile and DATASET profile
 - "extattr +p" for HFS-resident programs
- IBM-supplied daemons shipped in /usr/bin with sticky bit on so SYS1.LINKLIB copy will be used or shipped with +p external attribute



Controlling Daemons ... OS/390 UNIX-Level Security

- The daemon address space must be kept clean
- All programs loaded must be controlled
 - PROGRAM profiles covering all programs from MVS libraries (UACC READ is OK)
 - Controlled attribute for programs from the HFS
 - ▶ Set with extattr +p
 - ▶ Requires authority to BPX.FILEATTR.PROGCTL
 - ▶ Turned off automatically if file is changed
 - ▶ Ignored if HFS mounted with nosetuid or nosecurity
- Clean environment ensures daemons perform their intended function



Daemon Setup ... UNIX-Level Security

```
TSO LOGON  
RDEF START...  
SETR RACL....
```

```
RDEF STARTED INETD.* STDATA(USER(OMVSKERN)  
GROUP(OMVSGRP) TRUSTED(NO))
```

```
SETR RACLIST(STARTED) REFRESH
```

```
WARNING: INETD has SUPERUSER authority  
and can become any RACF defined user that  
has an OMVS segment. Any user that can  
change the function of the INETD program has  
full authority over the system.
```



Daemon Setup ... OS/390 UNIX-Level Security

```
TSO LOGON
RDEF FACIL.....
PE BPX.DAE...
```

```
RDEF FACILITY BPX.DAEMON UACC(NONE)
PERMIT BPX.DAEMON CLASS(FACILITY) ID(OMVSKERN)
ACCESS(READ)
SETR RACLIST(FACILITY) REFRESH

RDEF PROGRAM * ADDMEM('CEE.SCEERUN'//NOPADCHK
'SYS1.SEZALINK'//NOPADCHK
'SYS1.LINKLIB'//NOPADCHK) UACC(READ)
SETR WHEN(PROGRAM) ...OR ...
SETR WHEN(PROGRAM) REFRESH
```



Daemon Setup ... OS/390 UNIX-Level Security

```
TSO LOGON
RDEF FACIL.....
PE BPX.FILE...
```

```
RDEF FACILITY BPX.FILEATTR.PROGCTL
UACC(NONE)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY)
ID(UXADM) ACCESS(READ)
SETR RACLIST(FACILITY) REFRESH
OMVS
extattr +p /user/sbin/daemonx
ls -E daemonx
-rwxr-xr-x -p- 1 ROOT SYS1 101 Mar 12 19:32 daemonx
```

← program-controlled
attribute



Daemon Setup ... OS/390 UNIX-Level Security

```
TSO LOGON
RDEF PROG....
RDEF PROG....
```

```
RDEF PROGRAM INETD ADDMEM('SYS1.LINKLIB'//
NOPADCHK) UACC(READ)
RDEF PROGRAM RLOGIND ADDMEM('SYS1.LINKLIB'//
NOPADCHK) UACC(READ)
RDEF PROGRAM CRON ADDMEM('SYS1.LINKLIB'//
NOPADCHK) UACC(READ)
RDEF PROGRAM SU ADDMEM('SYS1.LINKLIB'//
NOPADCHK) UACC(READ)
RDEF PROGRAM CEE* ADDMEM('CEE.SCEERUN'//
NOPADCHK) UACC(READ)
RDEF PROGRAM EDC* ADDMEM('CEE.SCEERUN'//
NOPADCHK) UACC(READ)
SETR WHEN(PROGRAM) REFRESH
```



What is a Server?

- A program that starts at initialization time (a started task or cataloged procedure)
- A long-lived process, running unattended
- A service provider
- **Not necessarily authorized or superuser**
- Servers perform work for users by:
 - Verifying or **trusting** the user requesting the service
 - Creating a new **thread** to do the work
 - Giving the new **thread** the user's identity



How Does a Server Do Work for Clients?

- A well-behaved server does the following:
 - Verifies the client's identity
 - RACF or application password, digital certificate
 - Creates a thread for the client's work
 - Associates the client's user ID with the thread
 - pthread_security_np
 - Checks the client's authority when accessing OS/390 resources
 - BPX1ACK
- Not all servers are well-behaved



Controlling Servers ... Selecting your level of security

- UNIX-level vs. OS/390 UNIX level security applies
 - Activated by defining BPX.SERVER
 - Restricts use of pthread_security_np and BPX1ACK
 - UPDATE for trustworthy servers
 - Only the client's authority is checked
 - READ for servers that need added control
 - Client's and server's authority checked
 - SURROGAT allows server to represent client
 - Support for anonymous users
 - Clean address space is required



Server Setup ... UNIX-Level Security

```
TSO LOGON
AU DATASR...
RDEF START...
```

```
AU DATASRVR DFLTGRP(OMVSGRP) OMVS(UID(7))
```

```
RDEF STARTED MYSERVER.* STDATA(USER(DATASRVR)
GROUP(OMVSGRP) TRUSTED(NO))
```

```
SETR RACLIST(STARTED) REFRESH
```

```
RDEFINE APPL OMVSAPPL UACC(READ)
```

Continue to next page for better security...



Server Setup ... OS/390 UNIX-Level Security

```
TSO LOGON
RDEF FACIL...
PE BPX.SER...
```

```
RDEF FACILITY BPX.SERVER UACC(NONE)
PERMIT BPX.SERVER CLASS(FACILITY) ID(DATASRVR)
ACCESS(READ)
```

```
** WITH READ ACCESS, CLIENT AND SERVER AUTHORITY IS
CHECKED UNLESS THE CLIENT'S RACF PASSWORD
IS SUPPLIED **
```

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(DATASRVR)
ACCESS(UPDATE)
```

```
** WITH UPDATE ACCESS, ONLY THE CLIENT'S AUTHORITY
IS CHECKED **
```

```
RALT PROGRAM * ADDMEM('MYLIB.SRVRS/VOL003)
SETR WHEN(PROGRAM) RACL(FACILITY) REFR
```



Server Setup ... OS/390 UNIX-Level Security

```
TSO LOGON
PE BPX.SRV.....
RDEF SURR...
```

```
PE BPX.SERVER CLASS(FACILITY) ID(DATASVR)
ACCESS(READ)
RDEF SURROGAT BPX.SRV.USERA UACC(NONE)
PE BPX.SRV.USERA CLASS(SURROGAT)
ID(DATASVR) ACCESS(READ)
SETR RACLIST(FACILITY SURROGAT) REFRESH
```

**** UNLESS THE SERVER DOESN'T TAKE USERIDS ****

```
ADDUSER ANONYMOS NOPASSWORD
RDEF SURROGAT BPX.SRV.ANONYMOS UACC(NONE)
PE BPX.SRV.ANONYMOS CLASS(SURROGAT)
ID(DATASVR) ACCESS(READ)
```



Server Setup ... Bypassing System Resource Limits

- UNIX System Services imposes resource limits on a per user basis in BPXPRMxx:
 - MAXCPU TIME: cpu time
 - MAXASSIZE: address space region size
 - MAXFILEPROC: open files per process
 - MAXPROCUSER: processes per UID
 - MAXTHREADS: threads per process
 - MAXMMAPAREA: amount of storage mapped by mmap()
- Can be reset by SETOMVS or SET OMVS
- SUPERUSER can choose to exceed these limits



Server Setup ... Bypassing System Resource Limits

- Often servers need to use more resources than normal users
- Choices before V2R8:
 - increase system limit for all users
 - bad for system reliability, performance
 - give server UID(0) or BPX.SUPERUSER authority and modify server code to request a higher limit
 - requires modification to server code
- With V2R8:
 - assign higher limits specifically to server user IDs (or others) that need them




Server Setup ... Bypassing System Resource Limits


- New OMVS segment keywords on ADDUSER allow specific limits for individual users:
 - CPU TIME MAX(cpu-time)
 - ADDRESS SPACE MAX(address-space-size)
 - FILES PER PROCESS MAX(files-per-process)
 - PROCESSES PER UID MAX(processes-per-UID)
 - THREADS PER PROCESS MAX(threads-per-process)
 - MEMORY MAP SIZE MAX(memory-map-size)
- Also added, altered, deleted via ALTUSER
- Listed via LISTUSER
- Limits not specified in segment taken from BPXPRMxx.



What new terms have we heard?

- process - a program using kernel services running in an address space
 - thread - a task doing the same
 - daemon - a process that changes identities
 - sticky bit - file permission bit allowing multiple users to share a single copy of a program
 - fork - creation of a child process in a new address space
 - spawn - creation and starting of a child process that runs a named program (fork and exec)
 - server - a process that does work for clients
 - client - a user
- 

What do we need to remember?

- Read the security chapters of the OpenEdition or OS/390 UNIX System Services Planning manual for YOUR release level (SC28-1890)
 - Check the documentation for the daemon or server for security setup
 - Brush up on RACF Program Control
 - Check informational APARs ii08176, ii10548 and ii11345
 - Have a copy of the RACF Diagnosis Guide handy for your system programmer, just in case ...
- 

Good Sources of Information

- UNIX System Services web site, at <http://www.ibm.com/s390/unix/>
 - UNIX System Services Planning manual SC28-1890 (for your release)
 - Available online at <http://www.ibm.com/s390/os390/bkserv/>
 - mvs-oe mailing list (see the Forums link at the UNIX web site above for information)
- 