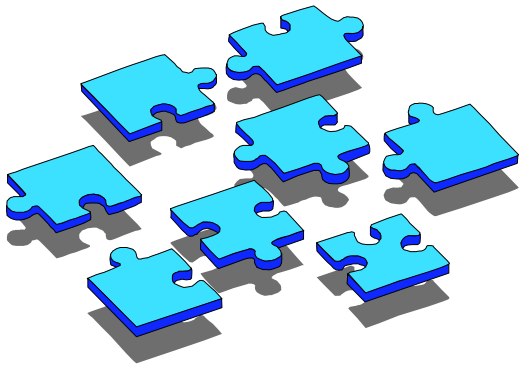
Controlling OS/390 UNIX System Services Servers and Daemons Using RACF

Vanguard Enterprise Security Expo 2001
Session 124

Bruce R. Wells
RACF Development, IBM
845-435-7498
brwells@us.ibm.com



Disclaimer

The information contained in this document is distributed on as "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.

Trademarks

- **The following are trademarks or registered trademarks of the International Business Machines Corporation:**
  - OS/390
  - RACF
  - SecureWay
  - z/OS
- **UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.**

Agenda

- z/OS UNIX System Services
- Daemons
  - Typical UNIX vs z/OS UNIX
  - z/OS UNIX Security setup
- Servers
  - Typical UNIX vs z/OS UNIX
  - z/OS UNIX Security setup
  - Resource limits
- Terms and Considerations
- References

Some new terms we will hear

- process - a program using kernel services running in an address space
- thread - a task doing the same
- daemon - a process that changes identities
- 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 plus exec)
- server - a process that does work for clients
- client - a user

What is OS/390 UNIX System Services?

- Product formerly known as OpenEdition
- Base element of z/OS
- UNIX interface for MVS providing
  - Hierarchical File System (HFS) containing directories and files
  - Application Interfaces (Callable Services)
  - Commands (Shells and Utilities)
- 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 z/OS 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 z/OS
  - 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

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

Controlling Daemons ... Selecting your level of security

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

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

- Activated by defining FACILITY BPX.DAEMON
- Restricts the use of identity changing services
- Only trusted daemons should be given authority

- The daemon address space must be kept clean
  - If a program that is NOT a controlled program is loaded, the address space is marked dirty and cannot perform daemon activities
- Clean environment ensures daemons perform their intended function

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

- 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*
    - Issuer needs authority to BPX.FILEATTR.PROGCTL
    - Turned off automatically if file is changed
    - Ignored if HFS mounted with *nosetuid* or *nosecurity*
- IBM-supplied daemons shipped in /usr/bin with sticky bit on so SYS1.LINKLIB copy will be used, or shipped with +p extended attribute

Daemon Setup ...

UNIX-Level Security

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

AU OMVSKERN OMVS(UID(0)) NOPASSWORD
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 ...

z/OS 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 ...

z/OS UNIX-Level Security

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

```
RDEF  FACILITY  BPX.FILEATTR.PROGCTL
     UACC(NONE)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY)
     ID(uxadmin) 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
```

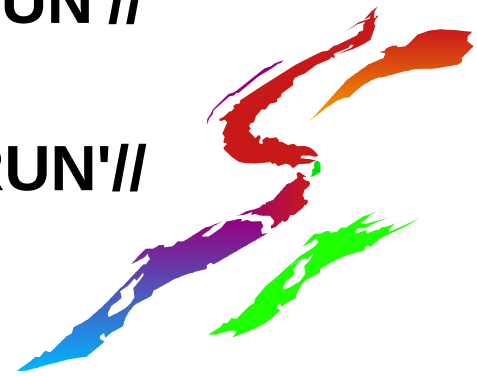
program-controlled attribute

Daemon Setup ...

z/OS 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
```



Agenda

- z/OS UNIX System Services
- Daemons
  - Typical UNIX vs z/OS UNIX
  - z/OS UNIX Security setup
- Servers
  - Typical UNIX vs z/OS UNIX
  - z/OS UNIX Security setup
- Resource limits
- Terms and Considerations
- References

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 (BPX1TLS)
  - Checks the client's authority when accessing z/OS resources
    - auth_check_resource_np (BPX1ACK)
    - _check_resource_auth_np()
- Not all servers are well-behaved

Controlling Servers ... Selecting your level of security

- Typical UNIX-level vs. z/OS UNIX level security applies
  - Activated by defining FACILITY BPX.SERVER
  - Restricts use of *pthread_security_np* and *auth_check_resource_np*
  - 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..

AG OMVSGRP OMVS(GID(n))
AU DATASRVR DFLTGRP(OMVSGRP) OMVS(UID(0))

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 ...

z/OS UNIX-Level Security

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

```
ALTUSER DATASRVR  OMVS(UID(7))
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')
SETR WHEN(PROGRAM) RACLIST(FACILITY)
    REFRESH
```

Server Setup ...

z/OS UNIX-Level Security

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

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

**** UNLESS THE SERVER DOES NOT TAKE USERIDS ****

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

Server Setup ...

Bypassing System Resource Limits

- UNIX System Services provide global resource limits on a per user basis in BPXPRMxx:
  - MAXCPUTIME: 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 OS/390 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 OS/390 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:
  - CPUTIMEMAX(cpu-time)
  - ASSIZEMAX(address-space-size)
  - FILEPROCMAX(files-per-process)
  - PROCUSERMAX(processes-per-UID)
  - THREADSMAX(threads-per-process)
  - MMAPAREAMAX(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
- 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 plus exec)
- server - a process that does work for clients
- client - a user

Considerations

- Is your environment well controlled?
  - BPX.DAEMON and BPX.SERVER
  - BPX.DEFAULT.USER
- Who has Superuser authority (UID=0)?
- Who has authority to issue *extattr +p*?
  - BPX.FILEATTR.PROGCTL
- Who can update the Program Controlled libraries?
- Who can change the PROGRAM profile?

What do we need to remember?

- Read the security chapters of the OS/390 or z/OS 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
  - OW44371 on OS/390 V2R8, and V2R10 and higher, provides additional diagnostics
- 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-1.ibm.com/servers/eserver/zseries/zos/unix/
  UNIX wizard: installation assistance
- UNIX System Services Planning manual SC28-1890 (for your release)
  - Available online at
    http://www-1.ibm.com/servers/s390/os390/bkserv/
- mvs-oe mailing list (see the Forums link at the UNIX web site above for information)