GALGOTIAS UNIVERSITY

School of Computing Science and Engineering

Program: BSC (Hons) CS

Course Code: BSCS3560

Course Name: Linux Administration

Lecture : 15

UNIT II

MONITORING AND MANAGING LINUX PROCESS AND LOGS

- Linux process - Controlling Jobs - Background Process and Foreground Process - Monitoring Process Activity - Killing Processes - Reviewing syslog files.

Print Process Tree

A process tree shows how processes on the system are linked to each other; processes whose parents have been killed are adopted by the init (or systemd).

$ ps -e --forest

```
  577 ?        00:00:00 NetworkManager
  900 ?        00:00:00   \_ dhclient
  773 ?        00:00:00 polkitd
 1146 ?        00:00:00 sshd
 2226 ?        00:00:00   \_ sshd
 2232 pts/0    00:00:00   |    \_ bash
 2557 pts/0    00:00:00   |         \_ ps
 2332 ?        00:00:00   \_ sshd
 2336 ?        00:00:00        \_ sshd
 2337 pts/1    00:00:00             \_ bash
 1154 ?        00:00:00 httpd
 2210 ?        00:00:00   \_ httpd
 2211 ?        00:00:00   \_ httpd
 2212 ?        00:00:00   \_ httpd
 2213 ?        00:00:00   \_ httpd
 2214 ?        00:00:00   \_ httpd
 1164 ?        00:00:00 postmaster
 2150 ?        00:00:00   \_ postmaster
 2204 ?        00:00:00   \_ postmaster
 2205 ?        00:00:00   \_ postmaster
 2206 ?        00:00:00   \_ postmaster
 2207 ?        00:00:00   \_ postmaster
 2208 ?        00:00:00   \_ postmaster
 1178 ?        00:00:09 ts3server
 1180 ?        00:00:00 iprinit
 1181 ?        00:00:00 iprupdate
 1380 ?        00:00:00 iprdump
 2199 ?        00:00:00 master
 2200 ?        00:00:00   \_ pickup
 2201 ?        00:00:00   \_ qmgr
[root@tecmint ~]#
```

```
  577 ?          00:00:00 NetworkManager
  900 ?          00:00:00  \_ dhclient
  773 ?          00:00:00 polkitd
 1146 ?          00:00:00 sshd
 2226 ?          00:00:00  \_ sshd
 2232 pts/0      00:00:00  |    \_ bash
 2557 pts/0      00:00:00  |         \_ ps
 2332 ?          00:00:00  \_ sshd
 2336 ?          00:00:00       \_ sshd
 2337 pts/1      00:00:00            \_ bash
 1154 ?          00:00:00 httpd
 2210 ?          00:00:00  \_ httpd
 2211 ?          00:00:00  \_ httpd
 2212 ?          00:00:00  \_ httpd
 2213 ?          00:00:00  \_ httpd
 2214 ?          00:00:00  \_ httpd
 1164 ?          00:00:00 postmaster
 2150 ?          00:00:00  \_ postmaster
 2204 ?          00:00:00  \_ postmaster
 2205 ?          00:00:00  \_ postmaster
 2206 ?          00:00:00  \_ postmaster
 2207 ?          00:00:00  \_ postmaster
 2208 ?          00:00:00  \_ postmaster
 1178 ?          00:00:09 ts3server
 1180 ?          00:00:00 iprinit
 1181 ?          00:00:00 iprupdate
 1380 ?          00:00:00 iprdump
 2199 ?          00:00:00 master
 2200 ?          00:00:00  \_ pickup
 2201 ?          00:00:00  \_ qmgr
[root@tecmint ~]#
```

You can also print a process tree for a given process like this.

$ ps -f --forest -C sshd

OR

$ ps -ef --forest | grep -v grep | grep sshd

```
[root@tecmint ~]# ps -f --forest -C sshd
UID        PID  PPID  C STIME TTY          TIME CMD
root      1029     1  0 08:31 ?        00:00:00 /usr/sbin/sshd -D
root      2093  1029  0 08:31 ?        00:00:01  \_ sshd: root@pts/0
root      4409  1029  0 09:58 ?        00:00:00  \_ sshd: tecmint [priv]
tecmint   4413  4409  0 09:58 ?        00:00:00      \_ sshd: tecmint@pt
[root@tecmint ~]#
[root@tecmint ~]#
[root@tecmint ~]# ps -ef --forest | grep -v grep | grep sshd
root      1029     1  0 08:31 ?        00:00:00 /usr/sbin/sshd -D
root      2093  1029  0 08:31 ?        00:00:01  \_ sshd: root@pts/0
root      4409  1029  0 09:58 ?        00:00:00  \_ sshd: tecmint [priv]
tecmint   4413  4409  0 09:58 ?        00:00:00      \_ sshd: tecmint@pt
s/1
[root@tecmint ~]#
```

Print Process Threads

To print all threads of a process, use the -H flag, this will show the **LWP** (**light weight process**) as well as **NLWP** (**number of light weight process**) columns.

$ ps -fL -C httpd

```
[root@tecmint ~]# ps -fL -C httpd
UID        PID  PPID   LWP  C NLWP STIME TTY          TIME CMD
root      1023     1  1023  0    1 08:31 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2079  1023  2079  0    1 08:31 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2080  1023  2080  0    1 08:31 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2081  1023  2081  0    1 08:31 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2082  1023  2082  0    1 08:31 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    2083  1023  2083  0    1 08:31 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
[root@tecmint ~]#
```

List Process Threads

Specify Custom Output Format

Using the **-o** or **–format** options, ps allows you to build user-defined output formats as shown below.

To list all format specifiers, include the L flag.

$ ps L

The command below allows you to view the **PID**, **PPID**, user name and command of a process.

$ ps -eo pid, ppid, user, cmd

```
[root@tecmint ~]# ps -eo pid,ppid,user,cmd
  PID  PPID USER     CMD
    1     0 root     /usr/lib/systemd/systemd --switched-root --syste
    2     0 root     [kthreadd]
    4     2 root     [kworker/0:0H]
    5     2 root     [kworker/u2:0]
    6     2 root     [mm_percpu_wq]
    7     2 root     [ksoftirqd/0]
    8     2 root     [rcu_sched]
    9     2 root     [rcu_bh]
   10     2 root     [rcuos/0]
   11     2 root     [rcuob/0]
   12     2 root     [migration/0]
   13     2 root     [watchdog/0]
   14     2 root     [cpuhp/0]
   15     2 root     [kdevtmpfs]
   16     2 root     [netns]
   17     2 root     [khungtaskd]
   18     2 root     [oom_reaper]
   19     2 root     [writeback]
```

Cont..

- There is another example of a custom output format showing file system group, nice value, start time and elapsed time of a process.

$ ps -p 1154 -o pid,ppid,fgroup,ni,lstart,etime

```
[root@tecmint ~]# ps -p 1154 -o pid,ppid,fgroup,ni,lstart,etime
  PID  PPID FGROUP    NI                  STARTED     ELAPSED
 1154     1 root       0 Fri Sep  8 13:05:10 2017    01:18:37
[root@tecmint ~]#
```

- To find a process name using its PID.

  $ ps -p 1154 -o comm=

```
[root@tecmint ~]# ps -p 1154  -o comm=
httpd
[root@tecmint ~]#
```

Find Process using PID

Display Parent and Child Processes

- To select a specific process by its name, use the -C flag, this will also display all its child processes.

$ ps -C sshd

```
[root@tecmint ~]# ps -C sshd
  PID TTY          TIME CMD
 1146 ?        00:00:00 sshd
 2226 ?        00:00:02 sshd
 2332 ?        00:00:00 sshd
 2336 ?        00:00:00 sshd
[root@tecmint ~]#
```

Find Parent Child Process

Display Parent and Child Processes

- Find all **PIDs** of all instances of a process, useful when writing scripts that need to read **PIDs** from a std output or file.
- $ ps -C httpd -o pid=

```
[root@tecmint ~]# ps -C  httpd -o pid=
 1154
 2210
 2211
 2212
 2213
 2214
[root@tecmint ~]#
```

Find All PIDs

Display Parent and Child Processes

- Check execution time of a process.

$ ps-eo comm,etime,user | grep httpd

The output below shows the HTTPD service has been running for 1 hours, 48 minutes and 17 seconds.

```
[root@tecmint ~]# ps -eo comm,etime,user | grep  httpd
httpd             01:48:17 root
httpd             01:48:10 apache
httpd             01:48:10 apache
httpd             01:48:10 apache
httpd             01:48:10 apache
httpd             01:48:10 apache
[root@tecmint ~]#
```

Uptime

Cont…

- Find top running processes by highest memory and CPU usage in Linux.

$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head

OR

$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%cpu | head

```
[root@tecmint ~]# ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
  PID  PPID CMD                         %MEM %CPU
  342     1 /usr/bin/python -Es /usr/sb  2.0  0.8
  344     1 /usr/bin/python -Es /usr/sb  1.5  0.4
  776   428 /sbin/dhclient -d -sf /usr/  1.5  0.0
 1039     1 /usr/pgsql-9.6/bin/postmast  1.4  0.0
 1508     1 ./ts3server inifile=ts3serv  1.4  4.6
 1023     1 /usr/sbin/httpd -DFOREGROUN  1.2  0.3
  630     1 /usr/lib/polkit-1/polkitd -  1.1  0.1
  428     1 /usr/sbin/NetworkManager --  0.8  0.2
 2079  1023 /usr/sbin/httpd -DFOREGROUN  0.6  0.0
[root@tecmint ~]#
```

Cont…

- To kill an Linux processes/unresponsive applications or any process that is consuming high CPU time.
- First, find the **PID** of the unresponsive process or application.
- $ ps -A | grep -i stress
- Then use the kill command to terminate it immediately.
- $ kill -9 2583 2584

```
[root@tecmint ~]#  ps -A | grep -i stress
 2583 tty1     00:00:00 stress
 2584 tty1     00:00:06 stress
[root@tecmint ~]#
[root@tecmint ~]# kill -9 2583 2584
[root@tecmint ~]#
[root@tecmint ~]#  ps -A | grep -i stress
[root@tecmint ~]#
```

Print Security Information

- Show security context (specifically for **SELinux**) like this.
- $ ps -eM
- OR
- $ ps --context

Print Security Information

```
[root@tecmint ~]# ps -e --context
  PID CONTEXT                          COMMAND
    1 system_u:system_r:init_t:s0      /usr/lib/systemd/systemd --switched-r
oot --s
    2 system_u:system_r:kernel_t:s0    [kthreadd]
    3 system_u:system_r:kernel_t:s0    [ksoftirqd/0]
    5 system_u:system_r:kernel_t:s0    [kworker/0:0H]
    6 system_u:system_r:kernel_t:s0    [kworker/u2:0]
    7 system_u:system_r:kernel_t:s0    [migration/0]
    8 system_u:system_r:kernel_t:s0    [rcu_bh]
    9 system_u:system_r:kernel_t:s0    [rcuob/0]
   10 system_u:system_r:kernel_t:s0    [rcu_sched]
   11 system_u:system_r:kernel_t:s0    [rcuos/0]
   12 system_u:system_r:kernel_t:s0    [watchdog/0]
```

Find SELinux Context

You can also display security information in user-defined format with this command.

$ ps -eo euser,ruser,suser,fuser,f,comm,label

```
[root@tecmint ~]# ps  -eo  euser,ruser,suser,fuser,f,comm,label
EUSER    RUSER    SUSER    FUSER    F COMMAND         LABEL
root     root     root     root     4 systemd         system_u:system_r:init_t:s0
root     root     root     root     1 kthreadd        system_u:system_r:kernel_t:s0
root     root     root     root     1 ksoftirqd/0     system_u:system_r:kernel_t:s0
root     root     root     root     1 kworker/0:0H    system_u:system_r:kernel_t:s0
root     root     root     root     1 kworker/u2:0    system_u:system_r:kernel_t:s0
root     root     root     root     1 migration/0     system_u:system_r:kernel_t:s0
root     root     root     root     1 rcu_bh          system_u:system_r:kernel_t:s0
root     root     root     root     1 rcuob/0         system_u:system_r:kernel_t:s0
root     root     root     root     1 rcu_sched       system_u:system_r:kernel_t:s0
root     root     root     root     1 rcuos/0         system_u:system_r:kernel_t:s0
root     root     root     root     5 watchdog/0      system_u:system_r:kernel_t:s0
root     root     root     root     1 khelper         system_u:system_r:kernel_t:s0
```

List SELinux Context by Users

Perform Real-time Process Monitoring Using Watch Utility

- Finally, since **ps** displays static information, you can employ the watch utility to perform real-time process monitoring with repetitive output, displayed after every second as in the command below (specify a custom **ps command** to achieve your objective).
- $ watch -n 1 'ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head'

Perform Real-time Process Monitoring Using Watch Utility

```
Every 1.0s: ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%cpu | head  Sat Sep  9 09:21:07 2017

  PID  PPID CMD                         %MEM %CPU
 3206  2097 dd if=/dev/zero of=/dev/nul  0.0 48.5
 3215  2221 dd if=/dev/zero of=/dev/nul  0.0 38.0
 3472  3471 stress --cpu 4               0.0 16.5
 3473  3471 stress --cpu 4               0.0 16.5
 3474  3471 stress --cpu 4               0.0 16.5
 3475  3471 stress --cpu 4               0.0 16.5
 1508     1 ./ts3server inifile=ts3serv  1.8  0.3
    1     0 /usr/lib/systemd/systemd --  0.6  0.0
    2     0 [kthreadd]                   0.0  0.0
```

Real Time Process Monitoring

Fields described by ps are described as:

UID: User ID that this process belongs to (the person running it)
PID: Process ID
PPID: Parent process ID (the ID of the process that started it)
C: CPU utilization of process
STIME: Process start time
TTY: Terminal type associated with the process
TIME: CPU time taken by the process
CMD: The command that started this process

- There are other options which can be used along with ps command :

-a: Shows information about all users
-x: Shows information about processes without terminals
-u: Shows additional information like -f option
-e: Displays extended information


Thank You