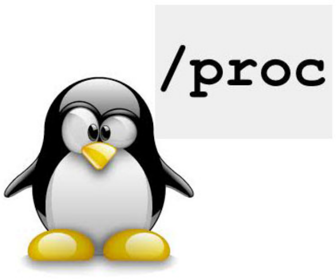
Configure Linux users to see only their own user processes with Hidepid - Stop users to see what others are doing

Author : admin



If you administer a *university shared free shell Linux server, have a small community of *NIX users offering free accounts* for them, or responsible for **Linux software company** with **development servers**, where **programmers login and use daily to program software / websites** its necessery to have tightened security rules with a major goal to **keep the different user accounts processes separate one from other (hide all system and user processes from single logged in user).**

Preventing users to see other users processes is essential for Linux servers which are at **high risk to be hacked.** At earlier times to achieve hiding all processes besides own ones from a logged in user was possible by using A kernel security module Grsecurity.

In latest currenlt **Linux kernel version 3.2**+ (on both *Debian (unstable) / Ubuntu 14.04 / RHEL/CentOS v6.5*+ above) you can hide process from other user so only **root** (useruser) can see all running process with (**ps auxwwf**) with a native kernel option hidepid.

Configuring Hidepid

To **enable hidepid option** you have to remount the **/proc** filesystem with the *Linux kernel* *hardening* **hidepid** option, to make it one time setting on already running server issue:

```
mount -o remount,rw,hidepid=2 /proc
```

To make the hidepid setting permanently active its necessery to modify ***/proc*** **filesystem settings**

in ***/etc/fstab***

```
vim /etc/fstab
```

```
proc    /proc    proc    defaults,hidepid=2    0    0
```

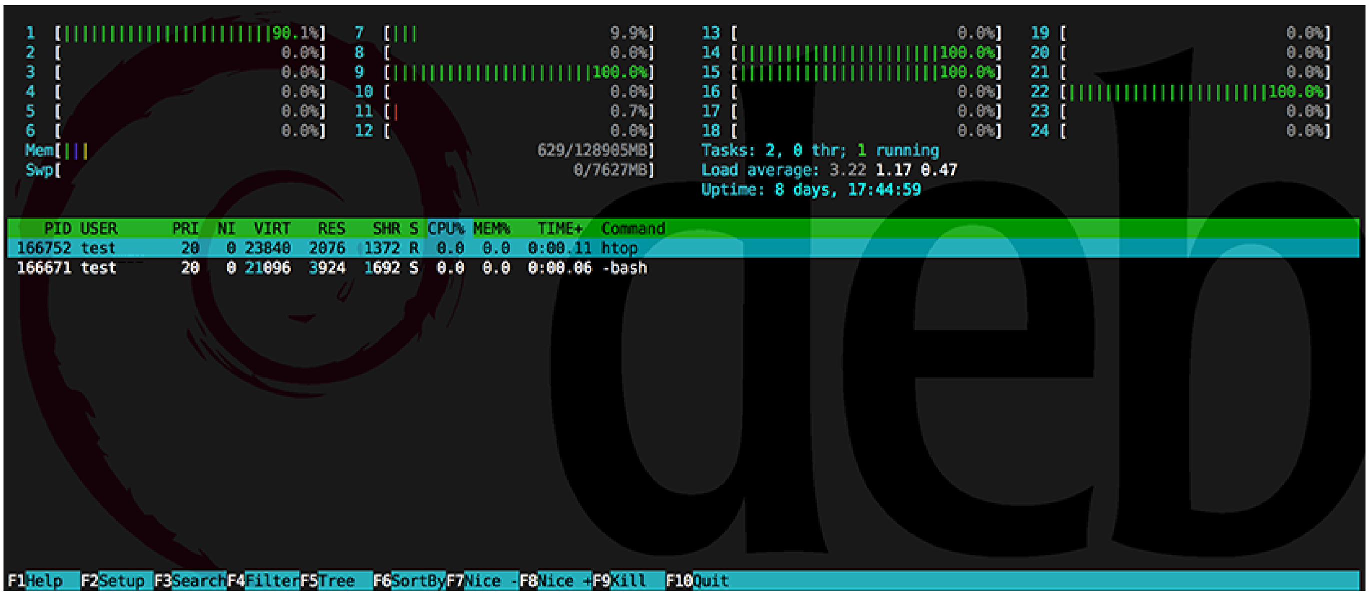
- **hidepid=0** - Anybody may read all world-readable **/proc/PID/*** files (default).
- **hidepid=1** - Means users may not access any **/proc/ /** directories, but only ones owned by them.Important files like cmdline, sched*, status are now protected to read from other other users.
- **hidepid=2** - Means **hidepid=1** plus all **/proc/PID/** will be invisible to other users besides logged in. Using this options stops Cracker's from gathering info about running processes, indication of daemon (services) which runs with elevated privileges, other user running processes (some might contain password) passed as argument or some sensitive data. Revealing such data is *frequently*

  .

  used to get versions of local / remote running services that can be exploited

Below is output of **htop** of a logged in user on **hidepid** activated server:

:



```
  1  [||||||||||||||||||||||90.1%]   7  [|||                  9.9%]    13 [                     0.0%]   19 [                     0.0%]
  2  [                     0.0%]   8  [                     0.0%]    14 [||||||||||||||||||||100.0%]   20 [                     0.0%]
  3  [                     0.0%]   9  [||||||||||||||||||||100.0%]    15 [||||||||||||||||||||100.0%]   21 [                     0.0%]
  4  [                     0.0%]   10 [                     0.0%]    16 [                     0.0%]   22 [||||||||||||||||||||100.0%]
  5  [                     0.0%]   11 [|                    0.7%]    17 [                     0.0%]   23 [                     0.0%]
  6  [                     0.0%]   12 [                     0.0%]    18 [                     0.0%]   24 [                     0.0%]
  Mem[|||                                     629/128905MB]    Tasks: 2, 0 thr; 1 running
  Swp[                                          0/7627MB]    Load average: 3.22 1.17 0.47
                                                             Uptime: 8 days, 17:44:59

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 166752 test       20   0 23840  2076  1372 R  0.0  0.0  0:00.11 htop
 166671 test       20   0 21096  3924  1692 S  0.0  0.0  0:00.06 -bash

F1Help  F2Setup  F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```