

Get dmesg command kernel log report with human date / time timestamp on older Linux distributions

Author : admin



If you're a sysadmin you surely love to take a look at **dmesg** kernel log output. Usually on many Linux distributions there is a setup that **dmesg** keeps logging to log files **/var/log/dmesg** or **/var/log/kern.log**. But if you get some inherited old Linux servers it is quite possible that the previous machine maintainer did not enable the output of syslog to get logged in **/var/log/{dmesg,kern.log,kernel.log}** or even have disabled the kernel log for some reason. Even though that in **dmesg** output you might find some interesting events reporting issues with Hard drives on its way to get broken / a bad / reads system processes crashing or whatever of other interesting information that could help you prevent severe servers downtimes or problems earlier but due to an old version of Linux distribution lets say **Redhat 5 / Debian 6 or old CentOS / Fedora**, the version of **dmesg** command shipped does not support the '**-T**' option that is present in **util-linux** package shipped with newer versions of **Redhat 7.X / 8.X / SuSEs** etc.

-T, --ctime

Print human readable timestamps. The timestamp could be inaccurate!

To illustrate better what I mean, here is an example from the non-human readable timestamp provided by older **dmesg** command

```
root@web-server~:# dmesg |tail -n 5
[4505913.361095] hid-generic 0003:1C4F:0002.000E: input,hidraw1: USB HID v1.10 Device
[SIGMACHIP USB Keyboard] on usb-0000:00:1d.0-1.3/input1
[4558251.034024] Process accounting resumed
[4615396.191090] r8169 0000:03:00.0 eth1: Link is Down
[4615397.856950] r8169 0000:03:00.0 eth1: Link is Up - 100Mbps/Full - flow control rx/tx
```

[4644650.095723] Process accounting resumed

Thankfully using below few lines of shell or perl scripts the **dmesg -T** functionality could be added to the system, so you can easily get the proper timestamp out of the obscure default generated timestamp in the same manner as on newer distros.

Here is how to do with it with bash script:

```
#!/bin/sh paste in .bashrc and use dmesgt to get human readable timestamp
dmesg_with_human_timestamps () {
    FORMAT="%a %b %d %H:%M:%S %Y"

    now=$(date +%s)
    cputime_line=$(grep -m1 "\.clock" /proc/sched_debug)

    if [[ $cputime_line =~ [^0-9]*([0-9]*).* ]]; then
        cputime=$((BASH_REMATCH[1] / 1000))
    fi

    dmesg | while IFS= read -r line; do
        if [[ $line =~ ^\[ *([0-9]+)\.[0-9]+\] (.*) ]]; then
            stamp=$((now-cputime+BASH_REMATCH[1]))
            echo "[$(date +"${FORMAT}" --date=@${stamp})] ${BASH_REMATCH[2]}"
        else
            echo "$line"
        fi
    done
}
```

Copy the script somewhere under lets say **/usr/local/bin** or wherever you like on the server and add into your HOME **~/.bashrc** some alias like:

```
alias dmesgt=dmesg_with_timestamp.sh
```

You can get a [copy dmesg_with_timestamp.sh of the script from here](#)

Or you can use below few lines perl script to get the proper **dmesg kernel date / time**

```
#!/bin/perl
# on old Linux distros CentOS 6.0 etc. with dmesg (part of util-linux-
ng-2.17.2-12.28.el6_9.2.x86_64) etc. dmesg -T not available
# workaround is little pl script below
dmesg_with_human_timestamps () {
    $(type -P dmesg) "$@" | perl -w -e 'use strict;
    my ($uptime) = do { local @ARGV="/proc/uptime";}; ($uptime) = ($uptime =~
/^\(d+\)\.)/);
    foreach my $line () {
        printf( ($line=~/\[s*(\d+)\.\d+\](.+)\)/ ? ( "[%s] %s\n", scalar localtime(time -
$uptime + $1), $2 ) : $line )
    }
}
```

Again to make use of the script put it under [/usr/local/bin/check_dmesg_timestamp.pl](#)

```
alias dmesgt=dmesg_with_human_timestamps
```

```
root@web-server:~# dmesgt | tail -n 20
```

```
[Sun Jun 13 15:51:49 2021] usb 2-1.3: USB disconnect, device number 9
[Sun Jun 13 15:51:50 2021] usb 2-1.3: new low-speed USB device number 10 using ehci-pci
[Sun Jun 13 15:51:50 2021] usb 2-1.3: New USB device found, idVendor=1c4f,
idProduct=0002, bcdDevice= 1.10
[Sun Jun 13 15:51:50 2021] usb 2-1.3: New USB device strings: Mfr=1, Product=2,
SerialNumber=0
[Sun Jun 13 15:51:50 2021] usb 2-1.3: Product: USB Keyboard
[Sun Jun 13 15:51:50 2021] usb 2-1.3: Manufacturer: SIGMACHIP
```

[Sun Jun 13 15:51:50 2021] input: SIGMACHIP USB Keyboard as /devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.3/2-1.3:1.0/0003:1C4F:0002.000D/input/input25

[Sun Jun 13 15:51:50 2021] hid-generic 0003:1C4F:0002.000D: input,hidraw0: USB HID v1.10 Keyboard [SIGMACHIP USB Keyboard] on usb-0000:00:1d.0-1.3/input0

[Sun Jun 13 15:51:50 2021] input: SIGMACHIP USB Keyboard Consumer Control as /devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.3/2-1.3:1.1/0003:1C4F:0002.000E/input/input26

[Sun Jun 13 15:51:50 2021] input: SIGMACHIP USB Keyboard System Control as /devices/pci0000:00/0000:00:1d.0/usb2/2-1/2-1.3/2-1.3:1.1/0003:1C4F:0002.000E/input/input27

[Sun Jun 13 15:51:50 2021] hid-generic 0003:1C4F:0002.000E: input,hidraw1: USB HID v1.10 Device [SIGMACHIP USB Keyboard] on usb-0000:00:1d.0-1.3/input1

[Mon Jun 14 06:24:08 2021] Process accounting resumed

[Mon Jun 14 22:16:33 2021] r8169 0000:03:00.0 eth1: Link is Down

[Mon Jun 14 22:16:34 2021] r8169 0000:03:00.0 eth1: Link is Up - 100Mbps/Full - flow control rx/tx