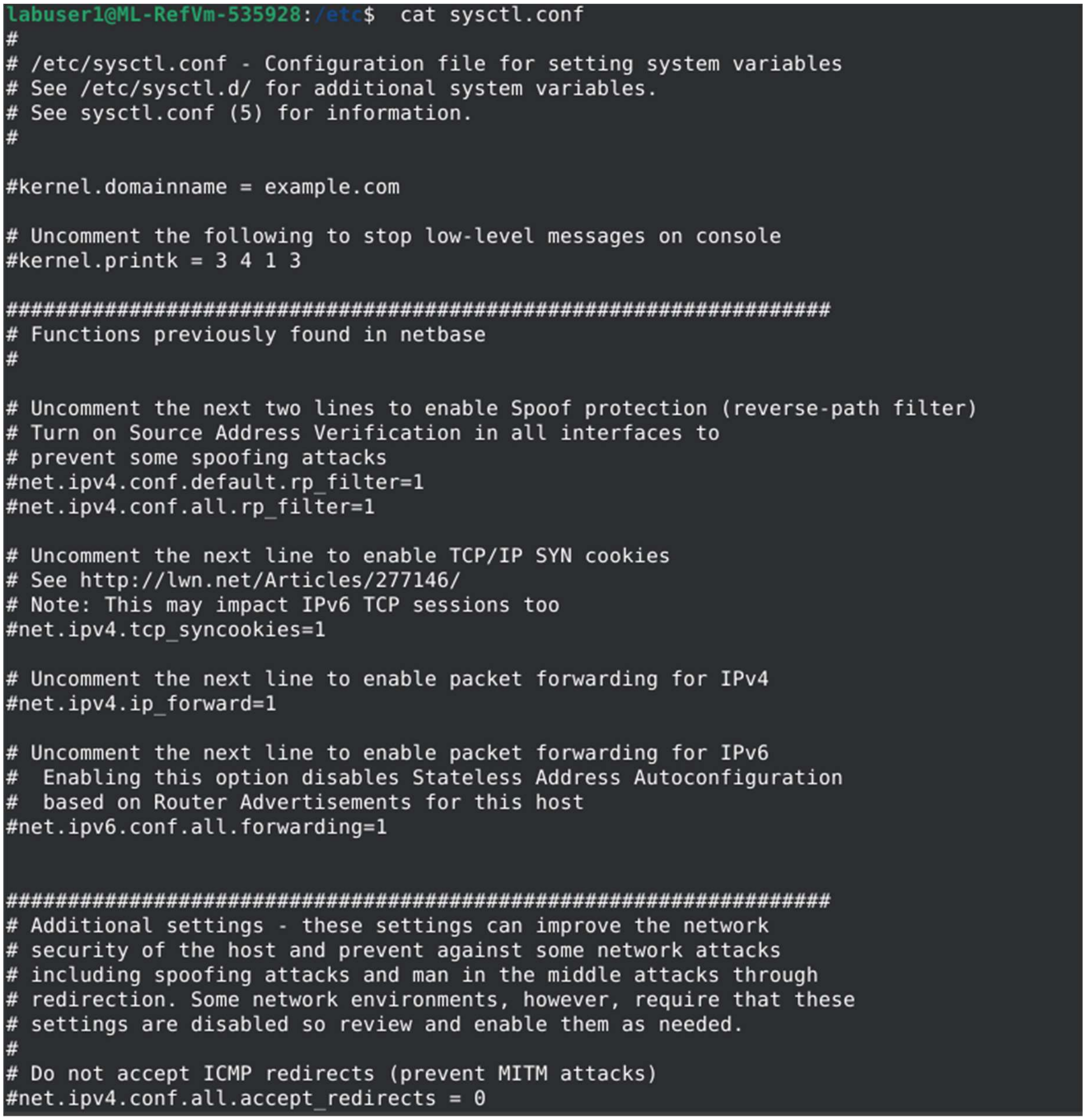
COSC 3364 – Principles of Cybersecurity
Lab 08

Provide screenshots where * is indicated.

Network Security Policy

You can use kernel parameters to modify the behavior of the kernel by adjusting key features. They can be used to change how the kernel manages devices, optimize memory usage, and enhance the security of the system.

- Navigate to the system control configuration file: **/etc/sysctl.conf**
- Determine which kernel parameters are enabled* (only a portion of the file is required for the screenshot)



```
labuser1@ML-RefVm-535928:/etc$  cat sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

###################################################################
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
```

Possible kernel parameters:

a. **Ignoring ping requests** – the ping command is often used to determine if a remote host is accessible through the network. An adversary can use ping to probe for active systems trying to find systems that they can break into. Responding to ping requests can leave a system vulnerable to denial of service attacks. To ignore ping requests, use the following setting in the **/etc/sysctl.conf:**

```
net.ipv4.icmp_echo_ignore_all = 1
```

b. **Ignoring Broadcast Requests** – Broadcast requests can be used for DoS and DDoS attacks. To ignore broadcast requests, use the following setting in the **/etc/sysctl.conf:**

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

c. **Enabling TCP SYN Protection** – A SYN flood attack is another DoS attack where SYN requests are used to make a system unresponsive. To Ignore SYS requests, use the following setting in the **/etc/sysctl.conf:**

```
net.ipv4.tcp_syncookies = 1
```

d. **Disabling IP Source Routing** – a feature that enables the sender of a packet to specify the network route that should be taken. This feature bypasses routing tables and makes your system vulnerable to man-in-the-middle attacks. To disable this feature from a specific network device, use the following setting in the **/etc/sysctl.conf:**

```
net.ipv4.conf.eth0.accept_source_route = 0
```

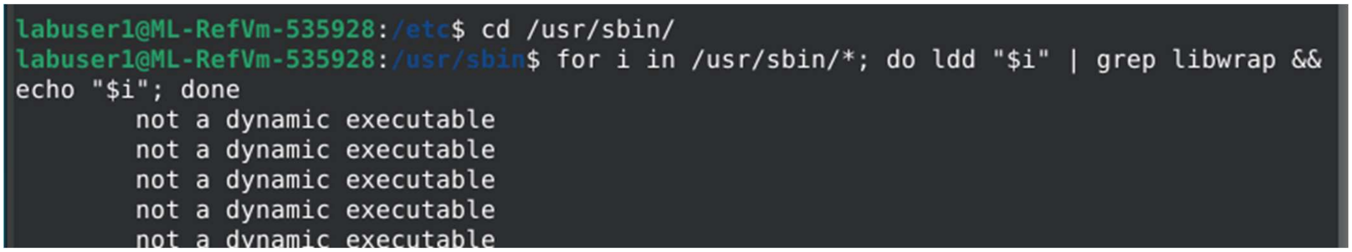
TCP wrappers are used when server programs that have been compiled with the **libwrap** library call that library when a system tries to access the service. An easy way to determine whether a service uses the **libwrap** library is to use the **ldd** command:

```
ldd <program>| grep libwrap
```

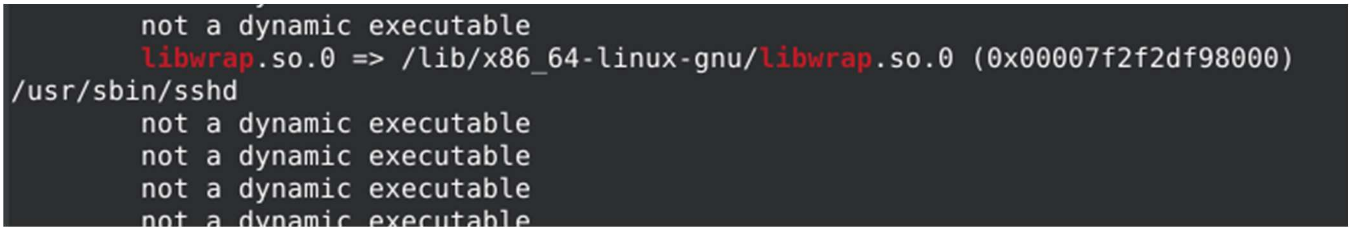
if the command returns **libwrap.so.0**, then the program uses TCP Wrappers.

- Execute to determine which program(s) use TCP Wrappers*:

```
for i in /usr/sbin/*
>do
>ldd $i | grep libwrap && echo $i
>done
```

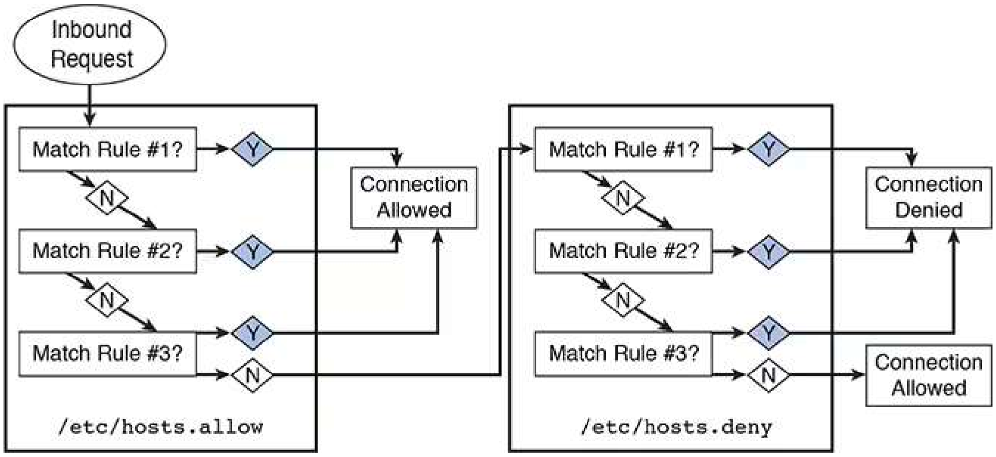


```
labuser1@ML-RefVm-535928:/etc$ cd /usr/sbin/
labuser1@ML-RefVm-535928:/usr/sbin$ for i in /usr/sbin/*; do ldd "$i" | grep libwrap &&
echo "$i"; done
        not a dynamic executable
        not a dynamic executable
        not a dynamic executable
        not a dynamic executable
        not a dynamic executable
```



```
        not a dynamic executable
        libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007f2f2df98000)
/usr/sbin/sshd
        not a dynamic executable
        not a dynamic executable
        not a dynamic executable
        not a dynamic executable
```

The **libwrap** library uses configuration files to determine whether the SSH connection should be allowed, based on what machine is initiating the connection. The files used are the **/etc/hosts.allow** and **/etc/hosts.deny** files.



The syntax of the rules in the **/etc/hosts.allow** and **/etc/hosts.deny** files is

```
service_list: client_list [options]
```

The service is the name of the binary executable service program (for example, sshd or xinetd). The client list is what system(s) this rule should apply to.

The **client_list** is also flexible. The following list details the different values you can provide:

- **IP address**: Example: 192.168.0.100.
- **Network**: Example: 192.168.0.0/255.255.255.0 or 192.168.0.
- **Entire domain**: Example: **.example.com**.
- **ALL**: Matches every possible client.
- **LOCAL**: Matches clients without a dot in their hostname. Example: test1.

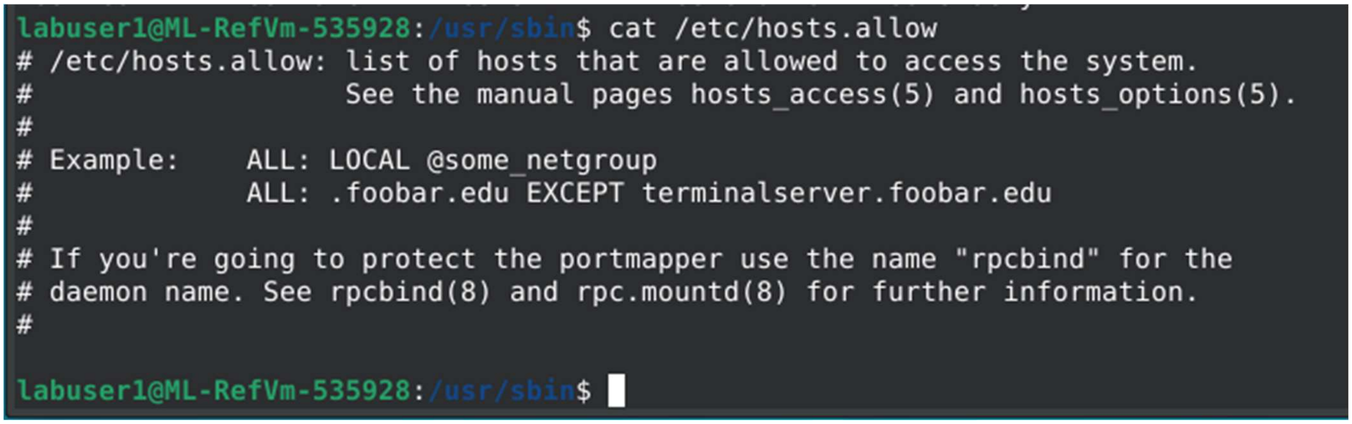
  - **UNKNOWN**: Matches clients that can't be resolved via the hostname resolver (DNS, local hosts file, and so on).
  - **KNOWN**: Matches clients that can be resolved via the hostname resolver (DNS, local hosts file, and so on).

Example: sshd: test.onecoursesource.com

Example: xinetd,sshd: test.onecoursesource.com

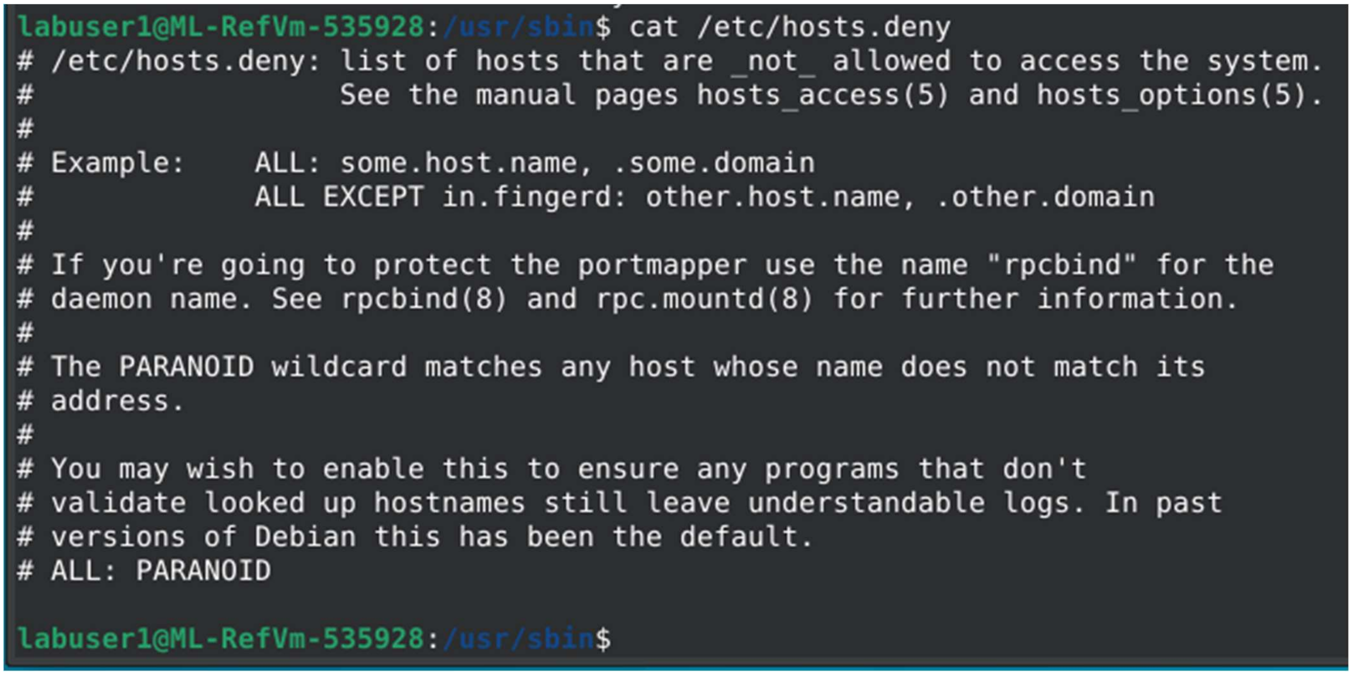
Example: ALL: test.onecoursesource.com

- Navigate to **/etc/hosts.allow** and determine what connections are allowed*



```
labuser1@ML-RefVm-535928:/usr/sbin$ cat /etc/hosts.allow
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#

labuser1@ML-RefVm-535928:/usr/sbin$ 
```

- Navigate to **/etc/hosts.deny** and determine what connections are denied*



```
labuser1@ML-RefVm-535928:/usr/sbin$ cat /etc/hosts.deny
# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
#                  See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: some.host.name, .some.domain
#             ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: PARANOID

labuser1@ML-RefVm-535928:/usr/sbin$
```

Process Control

The **ps** command is used to list processes that are running on the system. With no arguments, the command will list any child process of the current shell as well as the BASH shell itself.

Each line describes one process. By default, the **ps** command displays the following information:

- **PID**: Process ID number; each process has a unique ID that can be used to control the process.
- **TTY**: This is the terminal window that started the process. A terminal window is essentially a place where a user is able to issue commands from the command line.
- **TIME**: CPU time; how much time the process has used to execute code on the CPU. Although this number can grow over time, it is typically a very small number (a handful of seconds or maybe a few minutes, but rarely more), unless something is wrong with the process.
- **CMD**: The command.

To list all processes running on the system, add the **-e** option. The command **wc -l** can be used with the **ps** command to display the number of lines of data produced by the command to determine the number of processes running on a system. Each process is displayed on a separate line, so to be able to display a specific process you will want to use a **grep** command to filter the output.

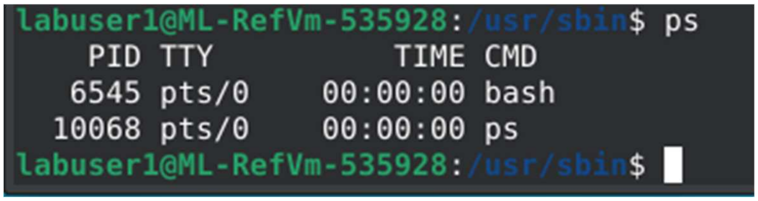
The **top** command displays process information that is updated on a regular basis (by default, every two seconds). The first half of the output of the top command contains overall information, whereas the second half displays a select list of processes (by default, the processes with the most CPU utilization).

The **free** command displays memory statistics.

To execute a process in the background, add an ampersand (**&**) character to the end of the command. Running a process in the background allows you to continue to work in the BASH shell and execute additional commands. Each BASH shell keeps track of the processes that are running from that BASH shell. These processes are referred to as jobs. To list the currently running jobs, execute the **jobs** command from the BASH shell.

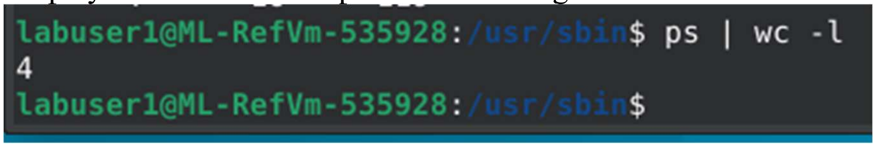
The phrase “kill a process” is used to describe when you completely stop a process. Several methods are available: the **kill** command, the **pkill** command, the **killall** command and the **xkill** command. The **kill** command can be used to change the state of a process, including to stop (kill) a process. To stop a process, first determine its process ID or (%) job number and then provide that number as an argument to the **kill** command. You can kill a process by running the xkill command and then just clicking the process that you want to stop.

- Display the processes running*



```
labuser1@ML-RefVm-535928:/usr/sbin$ ps
    PID TTY          TIME CMD
   6545 pts/0    00:00:00 bash
  10068 pts/0    00:00:00 ps
labuser1@ML-RefVm-535928:/usr/sbin$ 
```

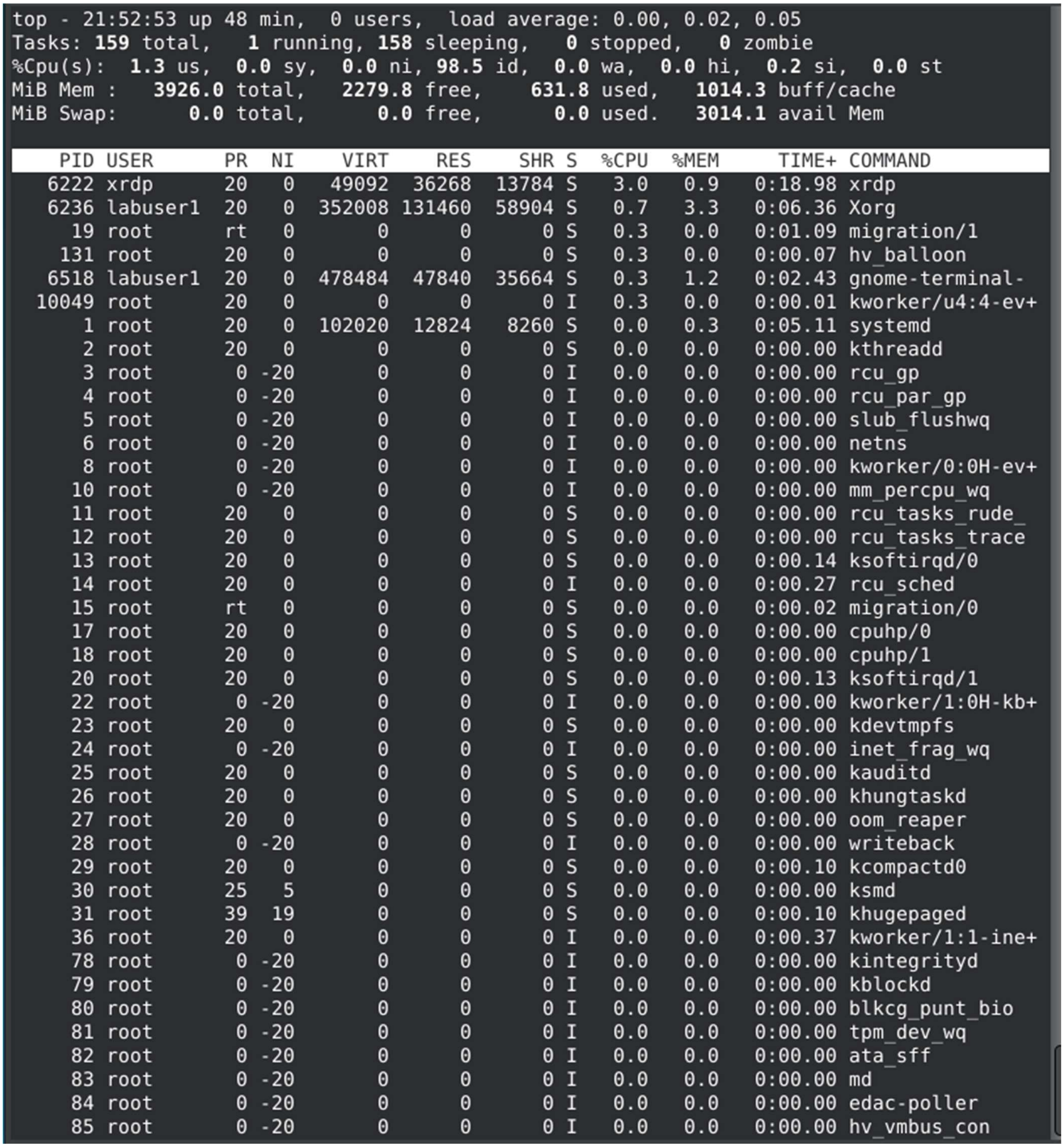
- Display the number of all processes running*



```
labuser1@ML-RefVm-535928:/usr/sbin$ ps | wc -l
4
labuser1@ML-RefVm-535928:/usr/sbin$
```

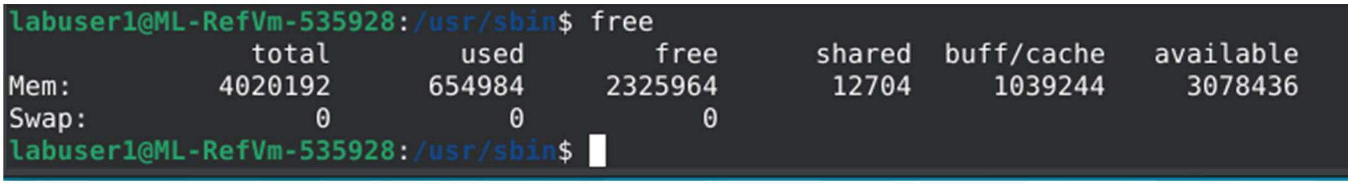
- Display process information*

`



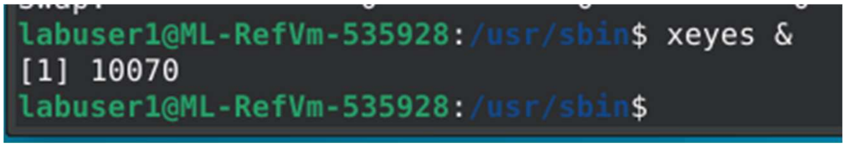
```
top - 21:52:53 up 48 min,  0 users,  load average: 0.00, 0.02, 0.05
Tasks: 159 total,   1 running, 158 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.3 us,  0.0 sy,  0.0 ni, 98.5 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   3926.0 total,   2279.8 free,    631.8 used,   1014.3 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   3014.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   6222 xrdp      20   0   49092  36268  13784 S   3.0   0.9   0:18.98 xrdp
   6236 labuser1  20   0  352008 131460  58904 S   0.7   3.3   0:06.36 Xorg
     19 root      rt   0       0      0      0 S   0.3   0.0   0:01.09 migration/1
    131 root      20   0       0      0      0 S   0.3   0.0   0:00.07 hv_balloon
   6518 labuser1  20   0  478484  47840  35664 S   0.3   1.2   0:02.43 gnome-terminal-
  10049 root      20   0       0      0      0 I   0.3   0.0   0:00.01 kworker/u4:4-ev+
      1 root      20   0  102020  12824   8260 S   0.0   0.3   0:05.11 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-ev+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.14 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.27 rcu_sched
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migration/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     20 root      20   0       0      0      0 S   0.0   0.0   0:00.13 ksoftirqd/1
     22 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-kb+
     23 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
     24 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 inet_frag_wq
     25 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
     26 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
     27 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
     28 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
     29 root      20   0       0      0      0 S   0.0   0.0   0:00.10 kcompactd0
     30 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd
     31 root      39  19       0      0      0 S   0.0   0.0   0:00.10 khugepaged
     36 root      20   0       0      0      0 I   0.0   0.0   0:00.37 kworker/1:1-ine+
     78 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kintegrityd
     79 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd
     80 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 blkcg_punt_bio
     81 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
     82 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 ata_sff
     83 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 md
     84 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 edac-poller
     85 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 hv_vmbus_con
```

- Display memory statistics*



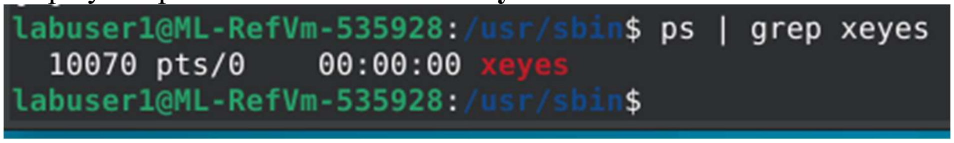
```
labuser1@ML-RefVm-535928:/usr/sbin$ free
              total        used        free      shared  buff/cache   available
Mem:        4020192      654984     2325964       12704     1039244     3078436
Swap:             0           0           0
labuser1@ML-RefVm-535928:/usr/sbin$ 
```

- Begin running **xeyes** in the background*



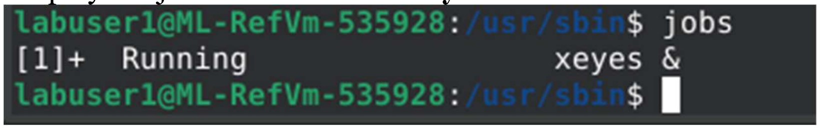
```
labuser1@ML-RefVm-535928:/usr/sbin$ xeyes &
[1] 10070
labuser1@ML-RefVm-535928:/usr/sbin$
```

- Display the process information for **xeyes***



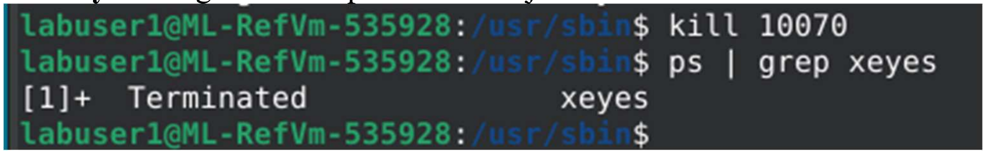
```
labuser1@ML-RefVm-535928:/usr/sbin$ ps | grep xeyes
  10070 pts/0    00:00:00 xeyes
labuser1@ML-RefVm-535928:/usr/sbin$
```

- Display the job information for **xeyes***



```
labuser1@ML-RefVm-535928:/usr/sbin$ jobs
[1]+  Running                 xeyes &
labuser1@ML-RefVm-535928:/usr/sbin$ 
```

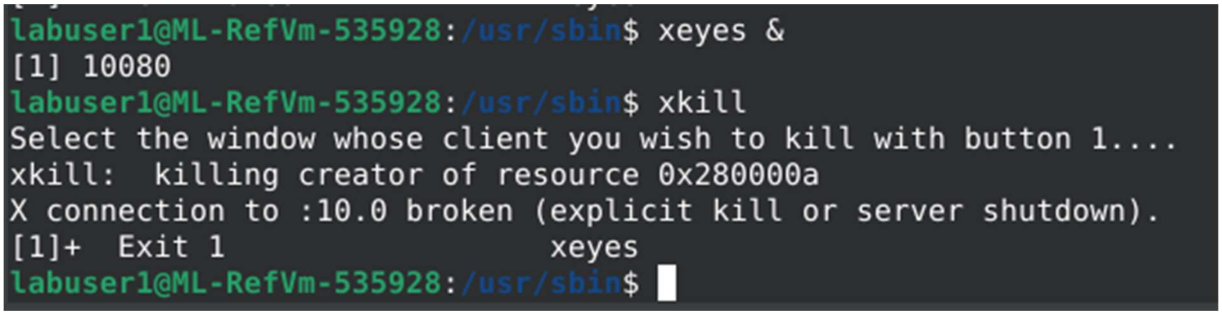
- Kill **xeyes** using either the process ID or job number*



```
labuser1@ML-RefVm-535928:/usr/sbin$ kill 10070
labuser1@ML-RefVm-535928:/usr/sbin$ ps | grep xeyes
[1]+  Terminated              xeyes
labuser1@ML-RefVm-535928:/usr/sbin$
```

- Begin running **xeyes**

- Kill **xeyes** by clicking the process* (the screenshot of the command used)

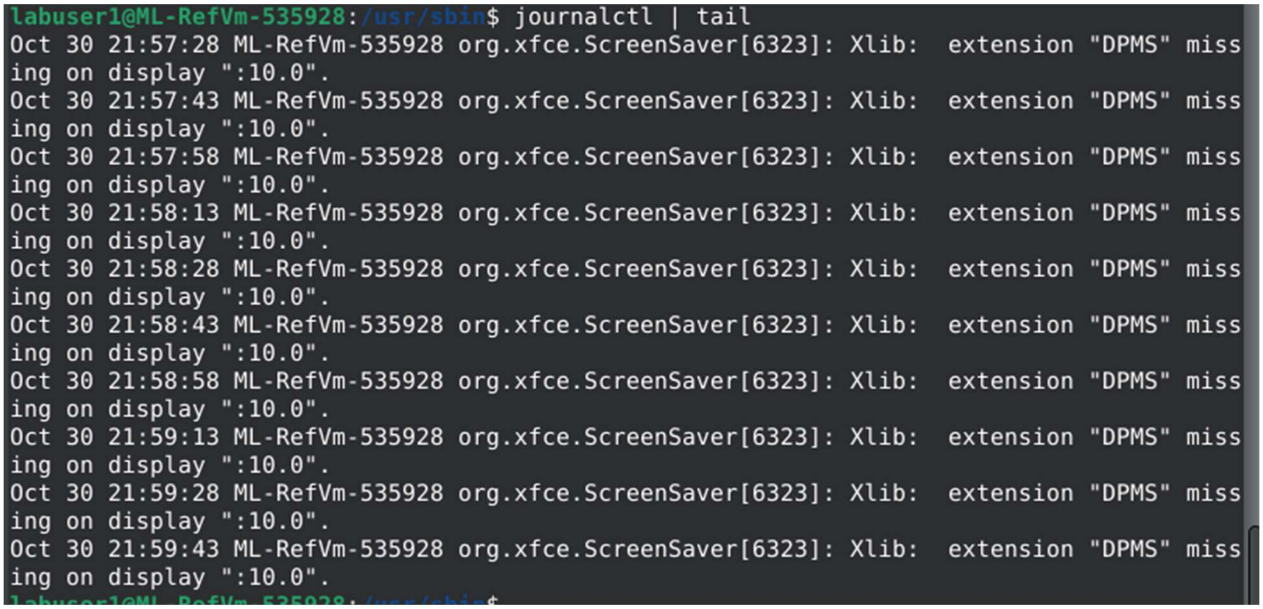


```
labuser1@ML-RefVm-535928:/usr/sbin$ xeyes &
[1] 10080
labuser1@ML-RefVm-535928:/usr/sbin$ xkill
Select the window whose client you wish to kill with button 1....
xkill:  killing creator of resource 0x280000a
X connection to :10.0 broken (explicit kill or server shutdown).
[1]+  Exit 1                  xeyes
labuser1@ML-RefVm-535928:/usr/sbin$ 
```

System Logging

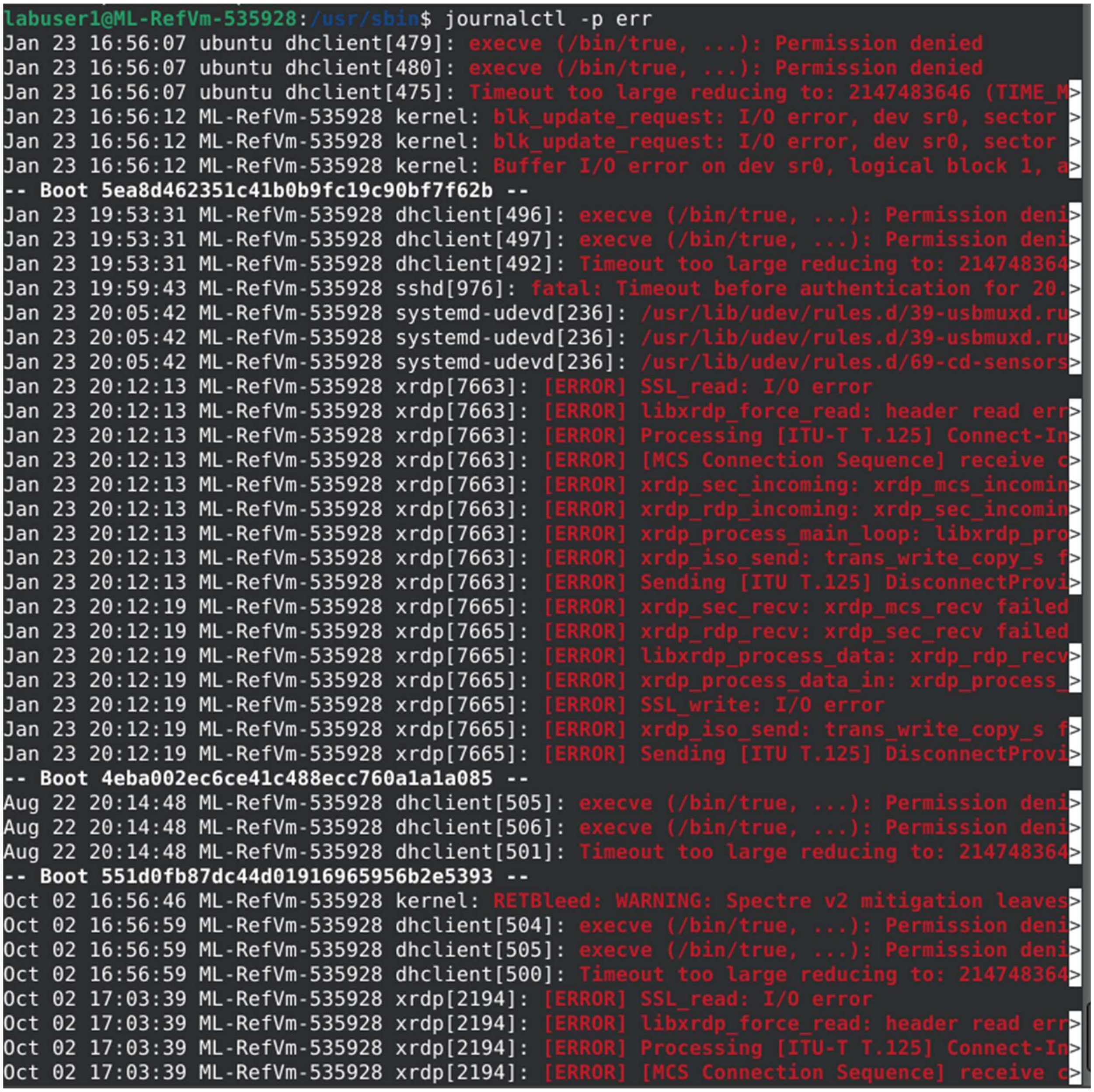
System logs are critical for several reasons: These logs provide administrators with useful information to aid in troubleshooting problems. They are also useful in identifying potential hacking attempts. Additionally, logs can be used to provide general information about services, such as which web pages have been provided by a web server. On modern Linux systems, the logging process is handled by the **systemd-journald** service. To query **systemd** log entries, use the **journalctl** command. The log entries can be filtered in a variety of forms such as by priority (**-p**), unit/process (**-u**), and boot logs(**-b**).

- View the tail of the log entries*



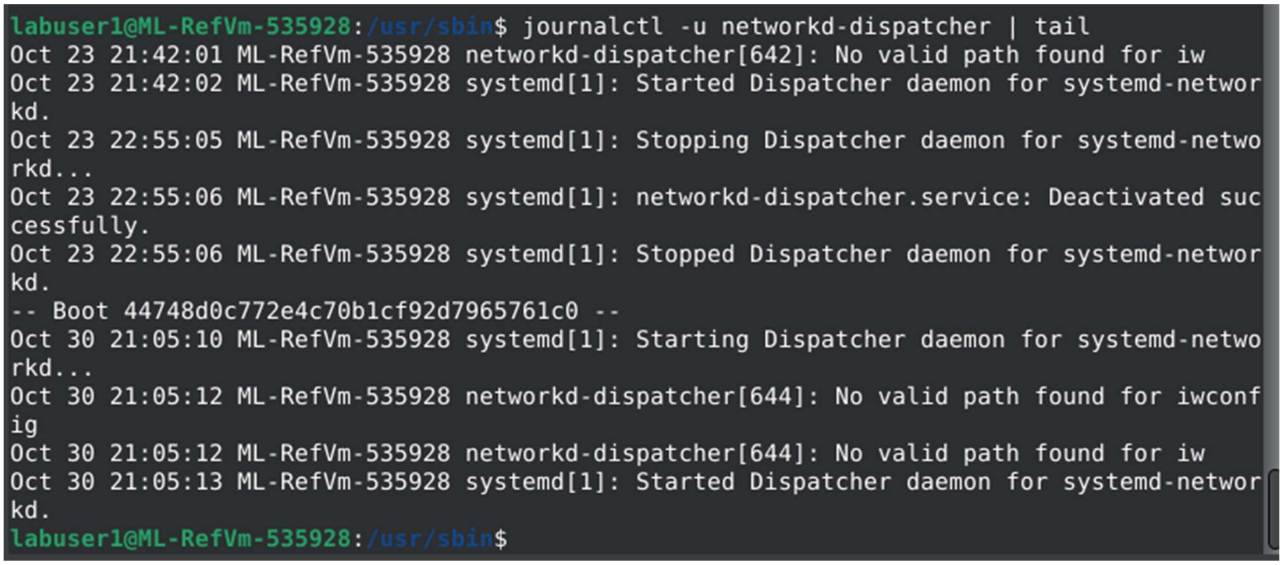
```
labuser1@ML-RefVm-535928:/usr/sbin$ journalctl | tail
Oct 30 21:57:28 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:57:43 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:57:58 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:58:13 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:58:28 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:58:43 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:58:58 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:59:13 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:59:28 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
Oct 30 21:59:43 ML-RefVm-535928 org.xfce.ScreenSaver[6323]: Xlib:  extension "DPMS" miss
ing on display ":10.0".
labuser1@ML-RefVm-535928:/usr/sbin$
```

- View the log entries of the **err** priority*



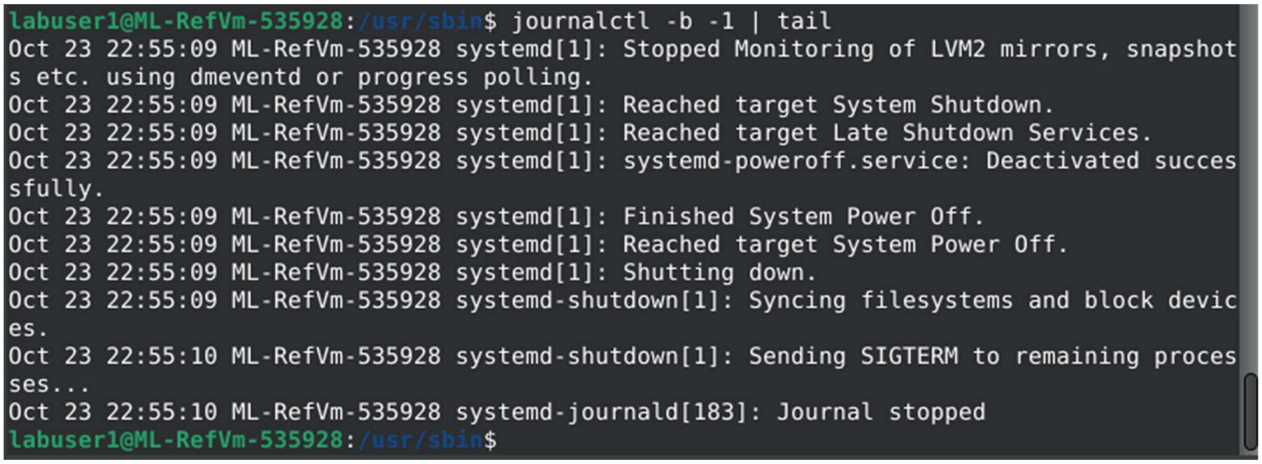
```
labuser1@ML-RefVm-535928:/usr/sbin$ journalctl -p err
Jan 23 16:56:07 ubuntu dhclient[479]: execve (/bin/true, ...): Permission denied
Jan 23 16:56:07 ubuntu dhclient[480]: execve (/bin/true, ...): Permission denied
Jan 23 16:56:07 ubuntu dhclient[475]: Timeout too large reducing to: 2147483646 (TIME_M>
Jan 23 16:56:12 ML-RefVm-535928 kernel: blk_update_request: I/O error, dev sr0, sector >
Jan 23 16:56:12 ML-RefVm-535928 kernel: blk_update_request: I/O error, dev sr0, sector >
Jan 23 16:56:12 ML-RefVm-535928 kernel: Buffer I/O error on dev sr0, logical block 1, a>
-- Boot 5ea8d462351c41b0b9fc19c90bf7f62b --
Jan 23 19:53:31 ML-RefVm-535928 dhclient[496]: execve (/bin/true, ...): Permission deni>
Jan 23 19:53:31 ML-RefVm-535928 dhclient[497]: execve (/bin/true, ...): Permission deni>
Jan 23 19:53:31 ML-RefVm-535928 dhclient[492]: Timeout too large reducing to: 214748364>
Jan 23 19:59:43 ML-RefVm-535928 sshd[976]: fatal: Timeout before authentication for 20.>
Jan 23 20:05:42 ML-RefVm-535928 systemd-udevd[236]: /usr/lib/udev/rules.d/39-usbmuxd.ru>
Jan 23 20:05:42 ML-RefVm-535928 systemd-udevd[236]: /usr/lib/udev/rules.d/39-usbmuxd.ru>
Jan 23 20:05:42 ML-RefVm-535928 systemd-udevd[236]: /usr/lib/udev/rules.d/69-cd-sensors>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] SSL_read: I/O error
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] libxrdp_force_read: header read err>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] Processing [ITU-T T.125] Connect-In>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] [MCS Connection Sequence] receive c>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] xrdp_sec_incoming: xrdp_mcs_incomin>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] xrdp_rdp_incoming: xrdp_sec_incomin>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] xrdp_process_main_loop: libxrdp_pro>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] xrdp_iso_send: trans_write_copy_s f>
Jan 23 20:12:13 ML-RefVm-535928 xrdp[7663]: [ERROR] Sending [ITU T.125] DisconnectProvi>
Jan 23 20:12:19 ML-RefVm-535928 xrdp[7665]: [ERROR] xrdp_sec_recv: xrdp_mcs_recv failed
Jan 23 20:12:19 ML-RefVm-535928 xrdp[7665]: [ERROR] xrdp_rdp_recv: xrdp_sec_recv failed
Jan 23 20:12:19 ML-RefVm-535928 xrdp[7665]: [ERROR] libxrdp_process_data: xrdp_rdp_recv>
Jan 23 20:12:19 ML-RefVm-535928 xrdp[7665]: [ERROR] xrdp_process_data_in: xrdp_process_>
Jan 23 20:12:19 ML-RefVm-535928 xrdp[7665]: [ERROR] SSL_write: I/O error
Jan 23 20:12:19 ML-RefVm-535928 xrdp[7665]: [ERROR] xrdp_iso_send: trans_write_copy_s f>
Jan 23 20:12:19 ML-RefVm-535928 xrdp[7665]: [ERROR] Sending [ITU T.125] DisconnectProvi>
-- Boot 4eba002ec6ce41c488ecc760a1a1a085 --
Aug 22 20:14:48 ML-RefVm-535928 dhclient[505]: execve (/bin/true, ...): Permission deni>
Aug 22 20:14:48 ML-RefVm-535928 dhclient[506]: execve (/bin/true, ...): Permission deni>
Aug 22 20:14:48 ML-RefVm-535928 dhclient[501]: Timeout too large reducing to: 214748364>
-- Boot 551d0fb87dc44d01916965956b2e5393 --
Oct 02 16:56:46 ML-RefVm-535928 kernel: RETBleed: WARNING: Spectre v2 mitigation leaves>
Oct 02 16:56:59 ML-RefVm-535928 dhclient[504]: execve (/bin/true, ...): Permission deni>
Oct 02 16:56:59 ML-RefVm-535928 dhclient[505]: execve (/bin/true, ...): Permission deni>
Oct 02 16:56:59 ML-RefVm-535928 dhclient[500]: Timeout too large reducing to: 214748364>
Oct 02 17:03:39 ML-RefVm-535928 xrdp[2194]: [ERROR] SSL_read: I/O error
Oct 02 17:03:39 ML-RefVm-535928 xrdp[2194]: [ERROR] libxrdp_force_read: header read err>
Oct 02 17:03:39 ML-RefVm-535928 xrdp[2194]: [ERROR] Processing [ITU-T T.125] Connect-In>
Oct 02 17:03:39 ML-RefVm-535928 xrdp[2194]: [ERROR] [MCS Connection Sequence] receive c>
```

- View the tail of the log entries of the **networkd-dispatcher** process*



```
labuser1@ML-RefVm-535928:/usr/sbin$ journalctl -u networkd-dispatcher | tail
Oct 23 21:42:01 ML-RefVm-535928 networkd-dispatcher[642]: No valid path found for iw
Oct 23 21:42:02 ML-RefVm-535928 systemd[1]: Started Dispatcher daemon for systemd-networ
kd.
Oct 23 22:55:05 ML-RefVm-535928 systemd[1]: Stopping Dispatcher daemon for systemd-netwo
rkd...
Oct 23 22:55:06 ML-RefVm-535928 systemd[1]: networkd-dispatcher.service: Deactivated suc
cessfully.
Oct 23 22:55:06 ML-RefVm-535928 systemd[1]: Stopped Dispatcher daemon for systemd-networ
kd.
-- Boot 44748d0c772e4c70b1cf92d7965761c0 --
Oct 30 21:05:10 ML-RefVm-535928 systemd[1]: Starting Dispatcher daemon for systemd-netwo
rkd...
Oct 30 21:05:12 ML-RefVm-535928 networkd-dispatcher[644]: No valid path found for iwconf
ig
Oct 30 21:05:12 ML-RefVm-535928 networkd-dispatcher[644]: No valid path found for iw
Oct 30 21:05:13 ML-RefVm-535928 systemd[1]: Started Dispatcher daemon for systemd-networ
kd.
labuser1@ML-RefVm-535928:/usr/sbin$
```

- View the tail of the log entries for the previous system boot (**-1**)*



```
labuser1@ML-RefVm-535928:/usr/sbin$ journalctl -b -1 | tail
Oct 23 22:55:09 ML-RefVm-535928 systemd[1]: Stopped Monitoring of LVM2 mirrors, snapshot
s etc. using dmeventd or progress polling.
Oct 23 22:55:09 ML-RefVm-535928 systemd[1]: Reached target System Shutdown.
Oct 23 22:55:09 ML-RefVm-535928 systemd[1]: Reached target Late Shutdown Services.
Oct 23 22:55:09 ML-RefVm-535928 systemd[1]: systemd-poweroff.service: Deactivated succes
sfully.
Oct 23 22:55:09 ML-RefVm-535928 systemd[1]: Finished System Power Off.
Oct 23 22:55:09 ML-RefVm-535928 systemd[1]: Reached target System Power Off.
Oct 23 22:55:09 ML-RefVm-535928 systemd[1]: Shutting down.
Oct 23 22:55:09 ML-RefVm-535928 systemd-shutdown[1]: Syncing filesystems and block devic
es.
Oct 23 22:55:10 ML-RefVm-535928 systemd-shutdown[1]: Sending SIGTERM to remaining proces
ses...
Oct 23 22:55:10 ML-RefVm-535928 systemd-journald[183]: Journal stopped
labuser1@ML-RefVm-535928:/usr/sbin$
```