# Linux

## Incident Response

security incident occurs. As an Incident Responder, you should always be aware of what should be and should not be present in your systems.

The security incidents that could be overcome by:

- By examining the running processes
- By having insights into the contents of physical memory.
- By gathering details on the hostname, IP address, operating systems etc
- Gathering information on system services.
- By identifying all the known and unknown users logged onto the system.
- By inspecting network connections, open ports and any network activity.
- By determining the various files present

## User Accounts

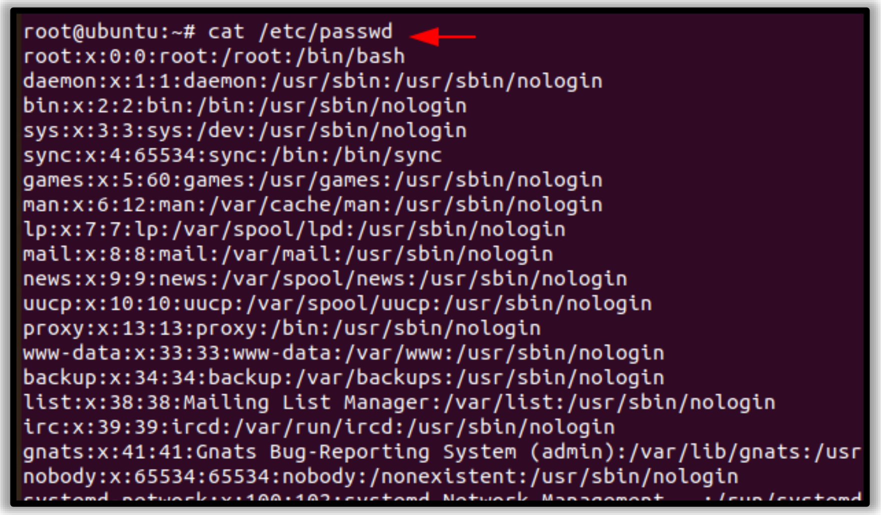
As an Incident Responder, it is very important to investigate the user account's activity. It helps you understand the logged-in users, the existing users, usual or unusual logins, failed login attempts, permissions, access by sudo etc.

The various commands to check the user account activity:

### /etc/passwd

To identify whether there is an account entry in your system that may seem suspicious. This command usually fetches all the information about the user account. To do so, type

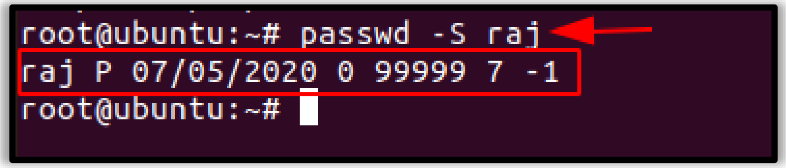
```
cat  /etc/passwd
```



```
root@ubuntu:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd
```

The '**Setuid**' option in Linux is unique file permission. So, on a Linux system when a user wants to make the change of password, they can run the '**passwd**' command. As the root account is marked as setuid, you can get temporary permission.

```
passwd -S raj
```

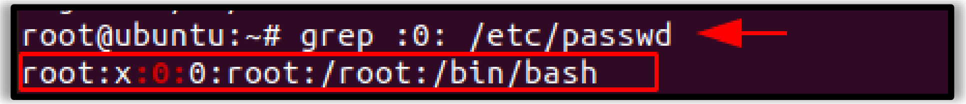


```
root@ubuntu:~# passwd -S raj
raj P 07/05/2020 0 99999 7 -1
root@ubuntu:~#
```

## grep

Grep is used for searching plain- text for lines that match a regular expression. :0: is used to display '**UID 0**' files in /etc/passwd file.

```
grep :0: /etc/passwd
```

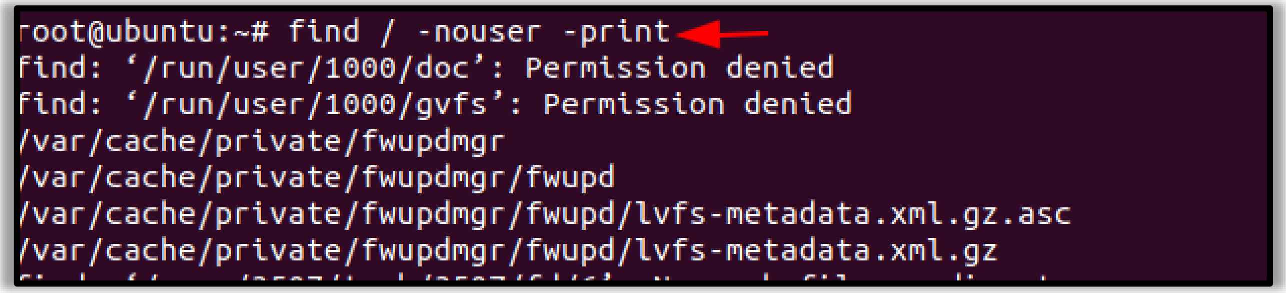


```
root@ubuntu:~# grep :0: /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

## find /-nouser

To Identify and display whether an attacker created any temporary user to perform an attack, type

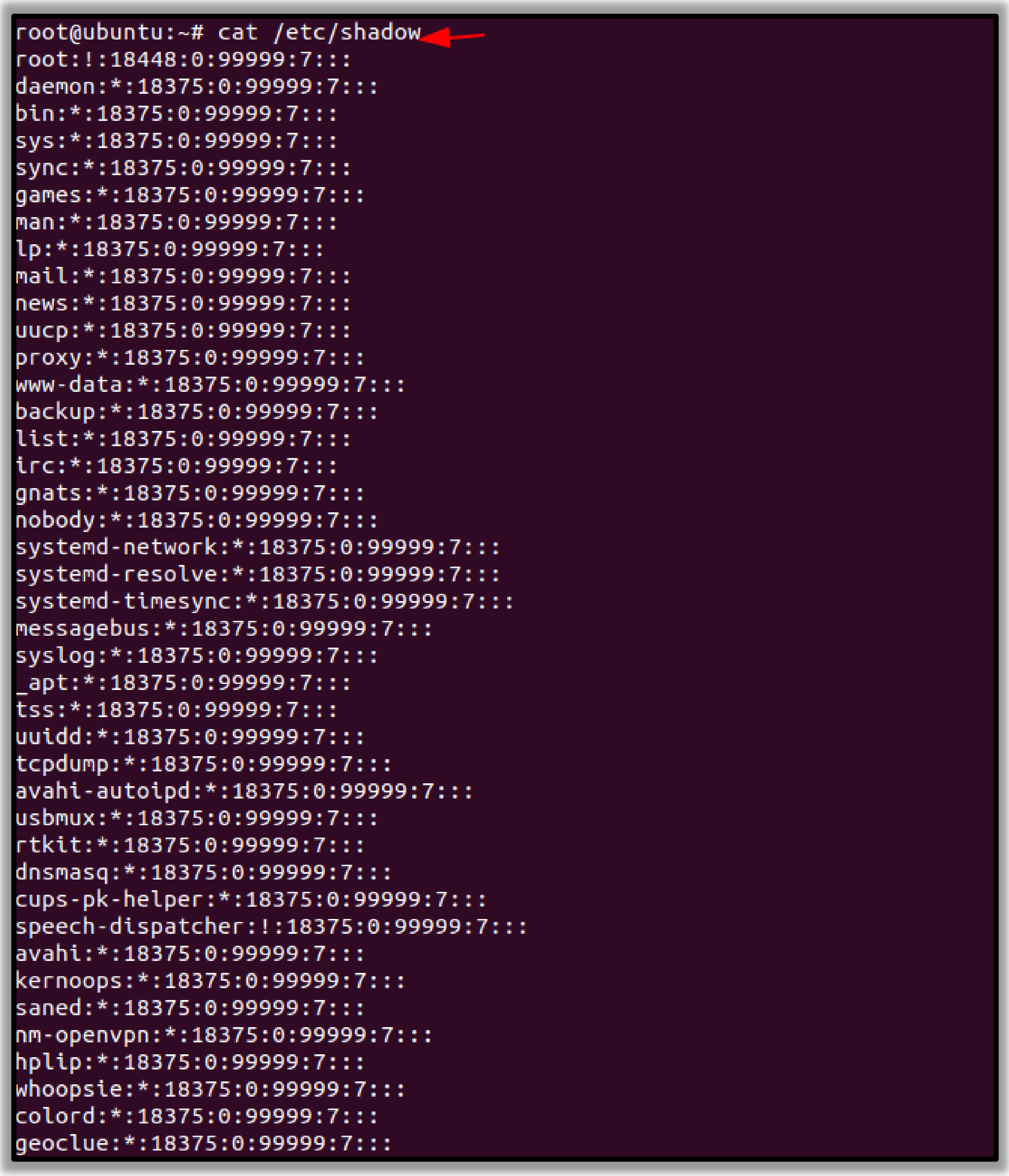
```
find / -nouser -print
```



```
root@ubuntu:~# find / -nouser -print
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
/var/cache/private/fwupdmgr
/var/cache/private/fwupdmgr/fwupd
/var/cache/private/fwupdmgr/fwupd/lvfs-metadata.xml.gz.asc
/var/cache/private/fwupdmgr/fwupd/lvfs-metadata.xml.gz
```

The /etc/shadow contains the encrypted password, details about the passwords and is only accessible by the root users.

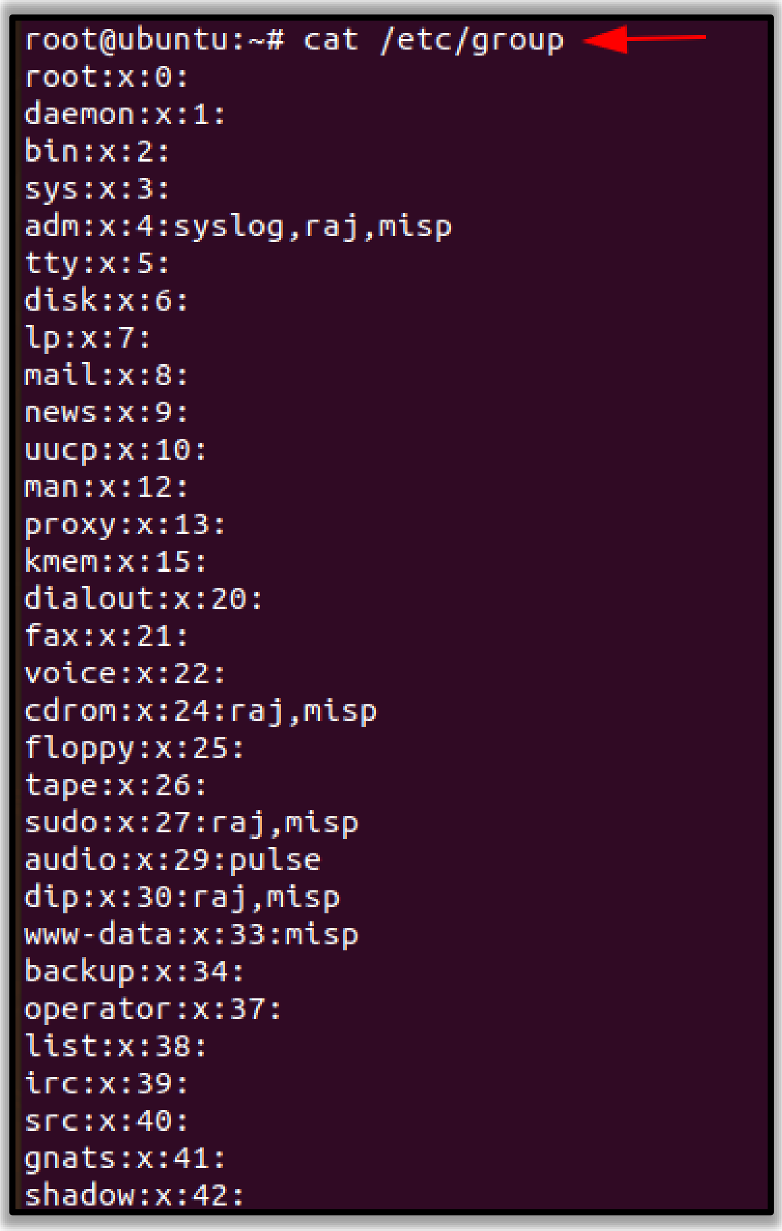
```
cat /etc/shadow
```



```
root@ubuntu:~# cat /etc/shadow
root:!:18448:0:99999:7:::
daemon:*:18375:0:99999:7:::
bin:*:18375:0:99999:7:::
sys:*:18375:0:99999:7:::
sync:*:18375:0:99999:7:::
games:*:18375:0:99999:7:::
man:*:18375:0:99999:7:::
lp:*:18375:0:99999:7:::
mail:*:18375:0:99999:7:::
news:*:18375:0:99999:7:::
uucp:*:18375:0:99999:7:::
proxy:*:18375:0:99999:7:::
www-data:*:18375:0:99999:7:::
backup:*:18375:0:99999:7:::
list:*:18375:0:99999:7:::
irc:*:18375:0:99999:7:::
gnats:*:18375:0:99999:7:::
nobody:*:18375:0:99999:7:::
systemd-network:*:18375:0:99999:7:::
systemd-resolve:*:18375:0:99999:7:::
systemd-timesync:*:18375:0:99999:7:::
messagebus:*:18375:0:99999:7:::
syslog:*:18375:0:99999:7:::
_apt:*:18375:0:99999:7:::
tss:*:18375:0:99999:7:::
uuidd:*:18375:0:99999:7:::
tcpdump:*:18375:0:99999:7:::
avahi-autoipd:*:18375:0:99999:7:::
usbmux:*:18375:0:99999:7:::
rtkit:*:18375:0:99999:7:::
dnsmasq:*:18375:0:99999:7:::
cups-pk-helper:*:18375:0:99999:7:::
speech-dispatcher:!:18375:0:99999:7:::
avahi:*:18375:0:99999:7:::
kernoops:*:18375:0:99999:7:::
saned:*:18375:0:99999:7:::
nm-openvpn:*:18375:0:99999:7:::
hplip:*:18375:0:99999:7:::
whoopsie:*:18375:0:99999:7:::
colord:*:18375:0:99999:7:::
geoclue:*:18375:0:99999:7:::
```

The group file displays the information of the groups used by the user. To view the details, type

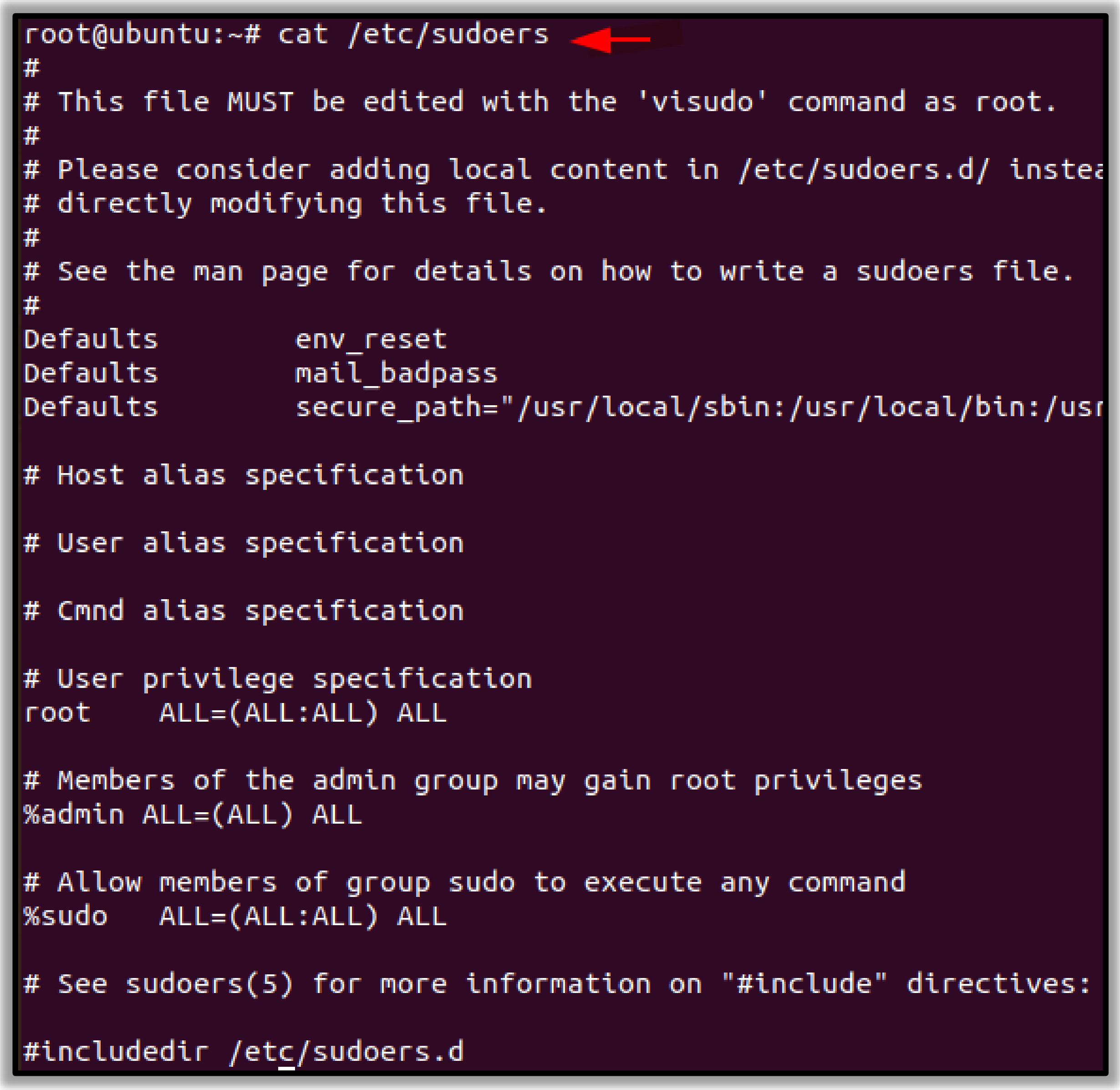
```
cat /etc/group
```



```
root@ubuntu:~# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,raj,misp
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:raj,misp
floppy:x:25:
tape:x:26:
sudo:x:27:raj,misp
audio:x:29:pulse
dip:x:30:raj,misp
www-data:x:33:misp
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
```

If you want to view information about user and group privileges to be displayed, the/ etc/sudoers file can be viewed

```
cat /etc/sudoers
```

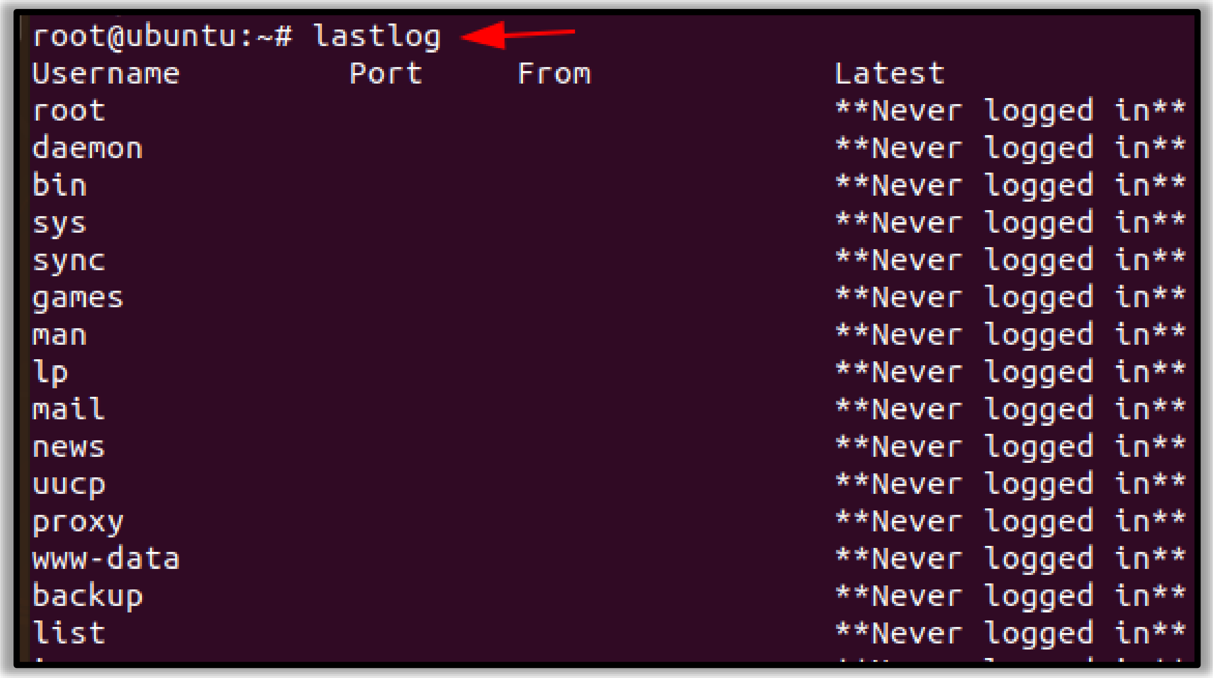


```
root@ubuntu:~# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instea
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

# Lastlog

To view the reports of the most recent login of a particular user or all the users in the Linux system, you can type,

```
lastlog
```

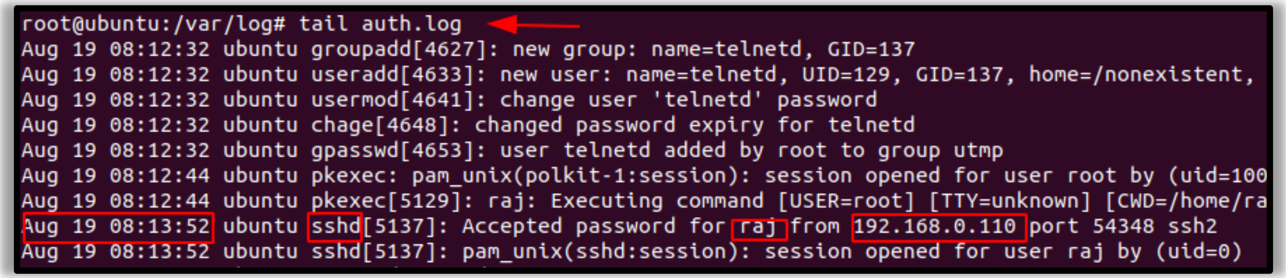


```
root@ubuntu:~# lastlog
Username         Port     From             Latest
root                                       **Never logged in**
daemon                                     **Never logged in**
bin                                        **Never logged in**
sys                                        **Never logged in**
sync                                       **Never logged in**
games                                      **Never logged in**
man                                        **Never logged in**
lp                                         **Never logged in**
mail                                       **Never logged in**
news                                       **Never logged in**
uucp                                       **Never logged in**
proxy                                      **Never logged in**
www-data                                   **Never logged in**
backup                                     **Never logged in**
list                                       **Never logged in**
```

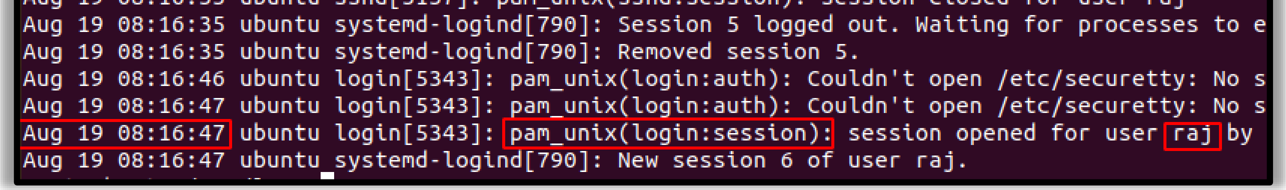
# Auth.log

To identify any curious SSH & telnet logins or authentication in the system, you can go to /var/log/ directory and then type

```
tail auth.log
```



```
root@ubuntu:/var/log# tail auth.log
Aug 19 08:12:32 ubuntu groupadd[4627]: new group: name=telnetd, GID=137
Aug 19 08:12:32 ubuntu useradd[4633]: new user: name=telnetd, UID=129, GID=137, home=/nonexistent,
Aug 19 08:12:32 ubuntu usermod[4641]: change user 'telnetd' password
Aug 19 08:12:32 ubuntu chage[4648]: changed password expiry for telnetd
Aug 19 08:12:32 ubuntu gpasswd[4653]: user telnetd added by root to group utmp
Aug 19 08:12:44 ubuntu pkexec: pam_unix(polkit-1:session): session opened for user root by (uid=100
Aug 19 08:12:44 ubuntu pkexec[5129]: raj: Executing command [USER=root] [TTY=unknown] [CWD=/home/ra
Aug 19 08:13:52 ubuntu sshd[5137]: Accepted password for raj from 192.168.0.110 port 54348 ssh2
Aug 19 08:13:52 ubuntu sshd[5137]: pam_unix(sshd:session): session opened for user raj by (uid=0)
```

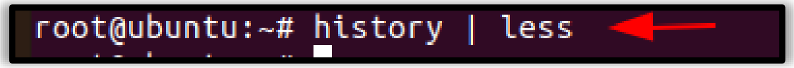


```
Aug 19 08:16:35 ubuntu sshd[5137]: pam_unix(sshd:session): session closed for user raj
Aug 19 08:16:35 ubuntu systemd-logind[790]: Session 5 logged out. Waiting for processes to e
Aug 19 08:16:35 ubuntu systemd-logind[790]: Removed session 5.
Aug 19 08:16:46 ubuntu login[5343]: pam_unix(login:auth): Couldn't open /etc/securetty: No s
Aug 19 08:16:47 ubuntu login[5343]: pam_unix(login:auth): Couldn't open /etc/securetty: No s
Aug 19 08:16:47 ubuntu login[5343]: pam_unix(login:session): session opened for user raj by
Aug 19 08:16:47 ubuntu systemd-logind[790]: New session 6 of user raj.
```

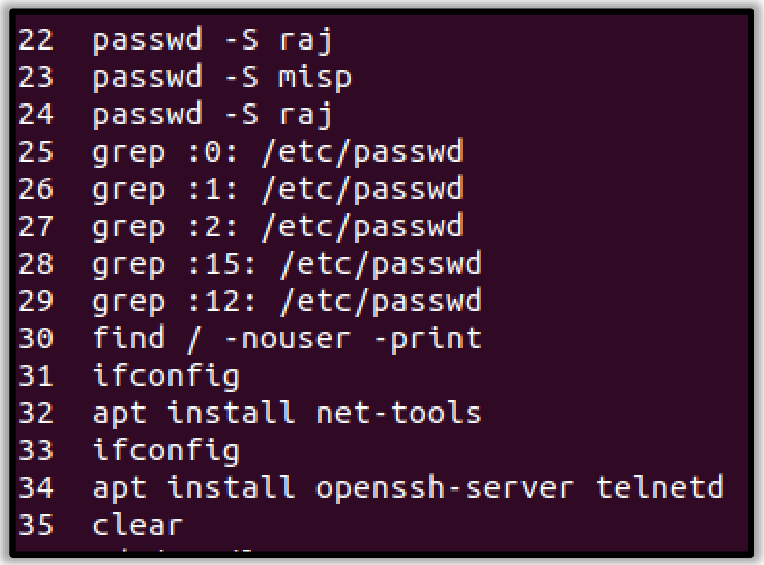
## History

To view the history of commands that the user has typed, you can type history with less or can even mention up to the number of commands you typed last. To view history, you can type

```
history| less
```



```
root@ubuntu:~# history | less
```



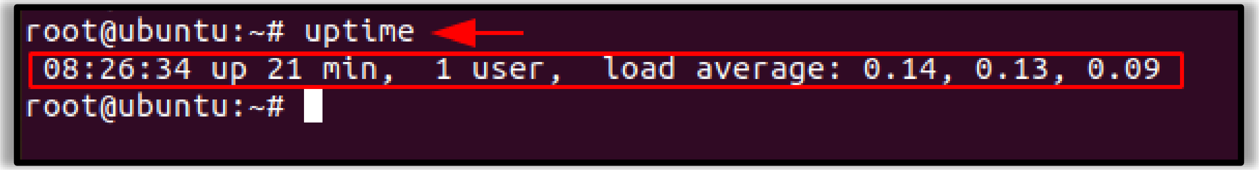
```
22  passwd -S raj
23  passwd -S misp
24  passwd -S raj
25  grep :0: /etc/passwd
26  grep :1: /etc/passwd
27  grep :2: /etc/passwd
28  grep :15: /etc/passwd
29  grep :12: /etc/passwd
30  find / -nouser -print
31  ifconfig
32  apt install net-tools
33  ifconfig
34  apt install openssh-server telnetd
35  clear
```

memory space and utilisation of the system etc.

## Uptime

To know whether your Linux system has been running overtime or to see how long the server has been running for, the current time in the system, how many users have currently logged on, and the load averages of the system, then you can type:

```
uptime
```

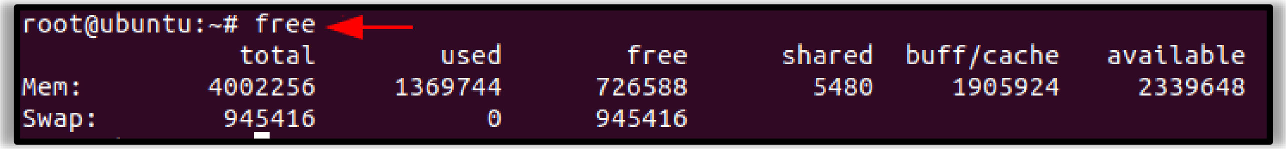


```
root@ubuntu:~# uptime
 08:26:34 up 21 min,  1 user,  load average: 0.14, 0.13, 0.09
root@ubuntu:~#
```

## Free

To view the memory utilisation by the system in Linux, the used physical and swap memory in the system, as well as the buffers used by the kernel, you can type,

```
free
```

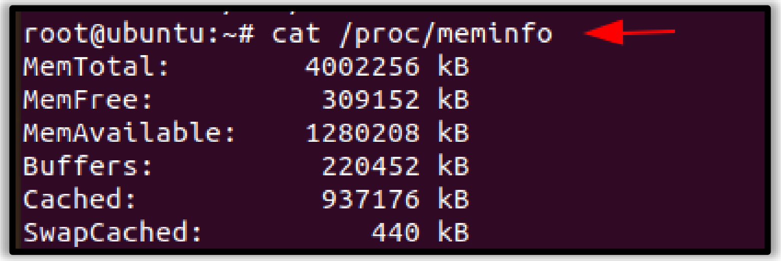


```
root@ubuntu:~# free
              total        used        free      shared  buff/cache   available
Mem:        4002256     1369744      726588        5480     1905924     2339648
Swap:        945416           0      945416
```

## /proc/memory

As an incident responder to check the detail information of the ram, memory space available, buffers and swap on the system, you can type

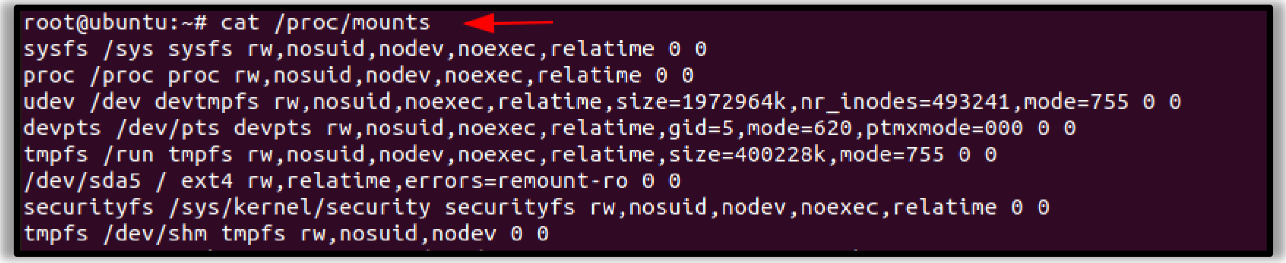
```
cat /proc/meminfo
```



```
root@ubuntu:~# cat /proc/meminfo
MemTotal:        4002256 kB
MemFree:          309152 kB
MemAvailable:    1280208 kB
Buffers:          220452 kB
Cached:           937176 kB
SwapCached:          440 kB
```

As an incident responder, it's your responsibility to check if there is an unknown mount on your system, to check the mount present on your system, you can type

```
cat /proc/mounts
```



```
root@ubuntu:~# cat /proc/mounts
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,nosuid,noexec,relatime,size=1972964k,nr_inodes=493241,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,nodev,noexec,relatime,size=400228k,mode=755 0 0
/dev/sda5 / ext4 rw,relatime,errors=remount-ro 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
```

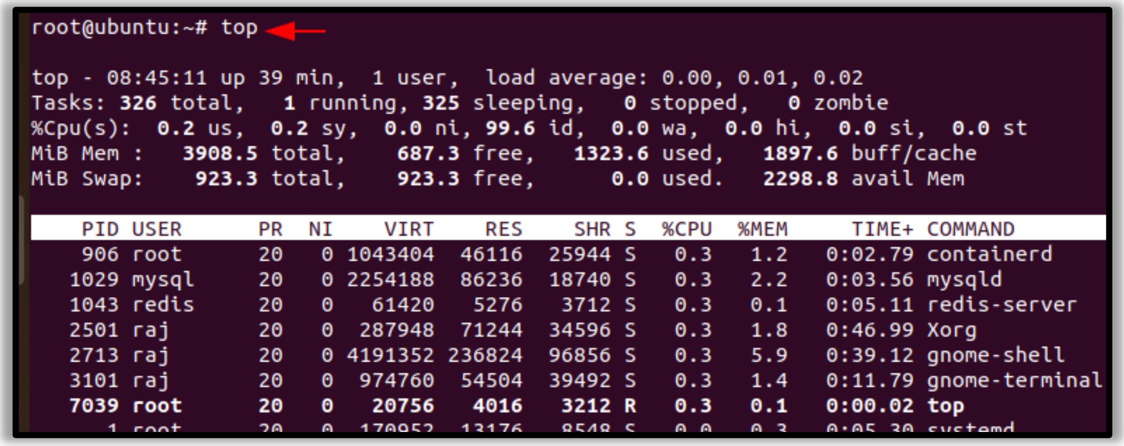
# Processes

As an incident responder, you should be always curious when you are looking through the output generated by your system. Your curiosity should compel you to view the programs that are currently running in the system, if they necessary to run and if they should be running, and usage of the CPU usage by these processes etc.

## top

To get a dynamic and a real-time visual of all the processes running in the Linux system, a summary of the information of the system and the list of processes and their ID numbers or threads managed by Linux Kernel, you can make use of

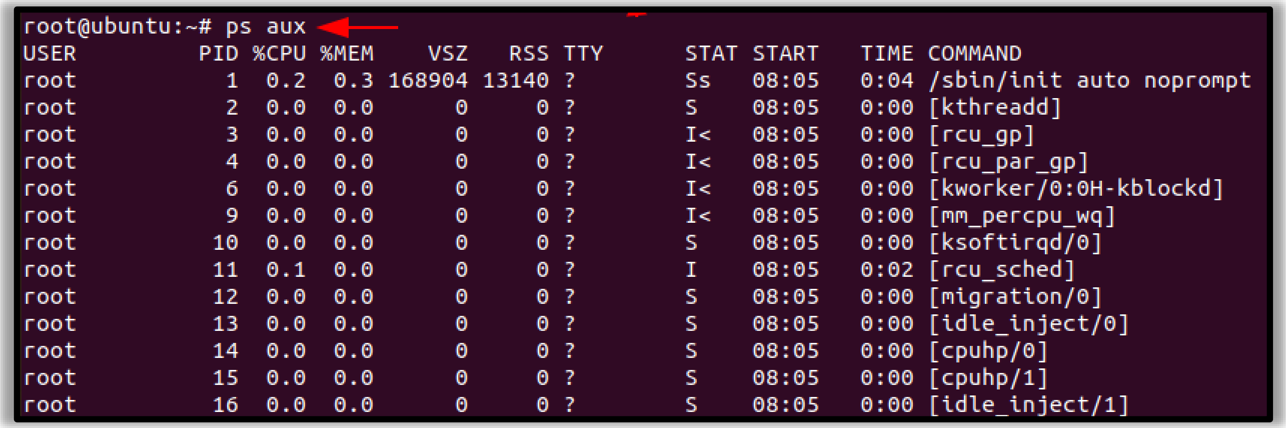
```
top
```



```
root@ubuntu:~# top

top - 08:45:11 up 39 min,  1 user,  load average: 0.00, 0.01, 0.02
Tasks: 326 total,   1 running, 325 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.2 sy,  0.0 ni, 99.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  3908.5 total,   687.3 free,  1323.6 used,  1897.6 buff/cache
MiB Swap:   923.3 total,   923.3 free,     0.0 used.  2298.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    906 root      20   0 1043404  46116  25944 S   0.3   1.2   0:02.79 containerd
   1029 mysql     20   0 2254188  86236  18740 S   0.3   2.2   0:03.56 mysqld
   1043 redis     20   0   61420   5276   3712 S   0.3   0.1   0:05.11 redis-server
   2501 raj       20   0  287948  71244  34596 S   0.3   1.8   0:46.99 Xorg
   2713 raj       20   0 4191352 236824  96856 S   0.3   5.9   0:39.12 gnome-shell
   3101 raj       20   0  974760  54504  39492 S   0.3   1.4   0:11.79 gnome-terminal
   7039 root      20   0   20756   4016   3212 R   0.3   0.1   0:00.02 top
      1 root      20   0  170952  13176   8548 S   0.0   0.3   0:05.30 systemd
```

To see the process status of your Linux and the currently running processes system and the PID. To identify abnormal processes that could indicate any malicious activity in the Linux system, you can use

```
ps aux
```

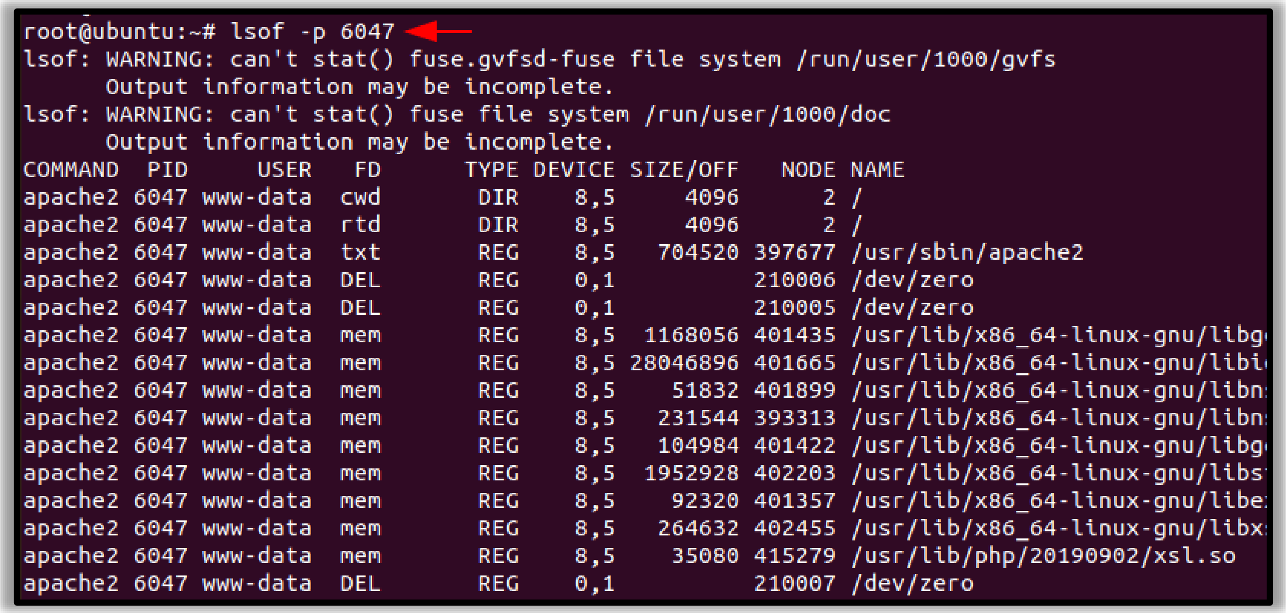


```
root@ubuntu:~# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.2  0.3 168904 13140 ?        Ss   08:05   0:04 /sbin/init auto noprompt
root           2  0.0  0.0      0     0 ?        S    08:05   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   08:05   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   08:05   0:00 [rcu_par_gp]
root           6  0.0  0.0      0     0 ?        I<   08:05   0:00 [kworker/0:0H-kblockd]
root           9  0.0  0.0      0     0 ?        I<   08:05   0:00 [mm_percpu_wq]
root          10  0.0  0.0      0     0 ?        S    08:05   0:00 [ksoftirqd/0]
root          11  0.1  0.0      0     0 ?        I    08:05   0:02 [rcu_sched]
root          12  0.0  0.0      0     0 ?        S    08:05   0:00 [migration/0]
root          13  0.0  0.0      0     0 ?        S    08:05   0:00 [idle_inject/0]
root          14  0.0  0.0      0     0 ?        S    08:05   0:00 [cpuhp/0]
root          15  0.0  0.0      0     0 ?        S    08:05   0:00 [cpuhp/1]
root          16  0.0  0.0      0     0 ?        S    08:05   0:00 [idle_inject/1]
```

## PID

To display more details on a particular process, you can use,

```
lsof -p [pid]
```



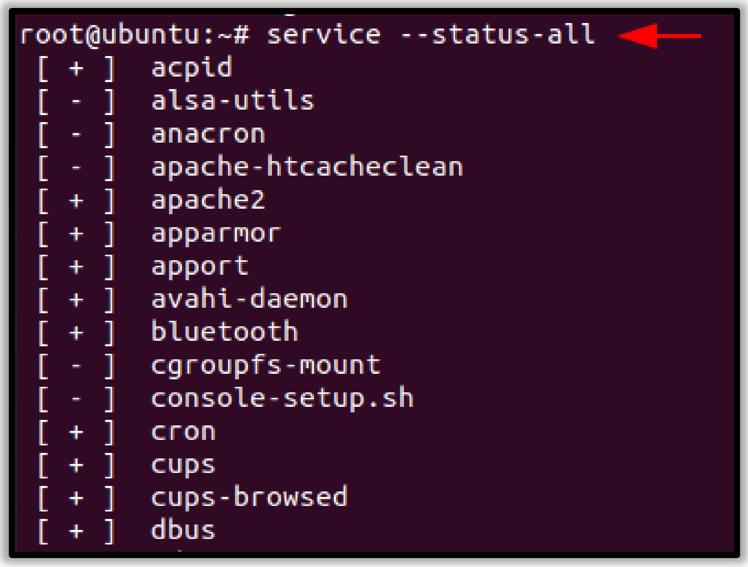
```
root@ubuntu:~# lsof -p 6047
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1000/gvfs
      Output information may be incomplete.
lsof: WARNING: can't stat() fuse file system /run/user/1000/doc
      Output information may be incomplete.
COMMAND   PID     USER   FD      TYPE DEVICE SIZE/OFF   NODE NAME
apache2 6047 www-data  cwd       DIR    8,5     4096      2 /
apache2 6047 www-data  rtd       DIR    8,5     4096      2 /
apache2 6047 www-data  txt       REG    8,5   704520 397677 /usr/sbin/apache2
apache2 6047 www-data  DEL       REG    0,1          210006 /dev/zero
apache2 6047 www-data  DEL       REG    0,1          210005 /dev/zero
apache2 6047 www-data  mem       REG    8,5  1168056 401435 /usr/lib/x86_64-linux-gnu/libg
apache2 6047 www-data  mem       REG    8,5 28046896 401665 /usr/lib/x86_64-linux-gnu/libi
apache2 6047 www-data  mem       REG    8,5    51832 401899 /usr/lib/x86_64-linux-gnu/libn
apache2 6047 www-data  mem       REG    8,5   231544 393313 /usr/lib/x86_64-linux-gnu/libn
apache2 6047 www-data  mem       REG    8,5   104984 401422 /usr/lib/x86_64-linux-gnu/libg
apache2 6047 www-data  mem       REG    8,5  1952928 402203 /usr/lib/x86_64-linux-gnu/libs
apache2 6047 www-data  mem       REG    8,5    92320 401357 /usr/lib/x86_64-linux-gnu/libe
apache2 6047 www-data  mem       REG    8,5   264632 402455 /usr/lib/x86_64-linux-gnu/libx
apache2 6047 www-data  mem       REG    8,5    35080 415279 /usr/lib/php/20190902/xsl.so
apache2 6047 www-data  DEL       REG    0,1          210007 /dev/zero
```

include the status of services, cron, etc and network services include file transfer, domain name resolution, firewalls, etc. As an incident responder, you identify if there is an anomaly in the services.

## Service

To find any abnormally running services, you can use

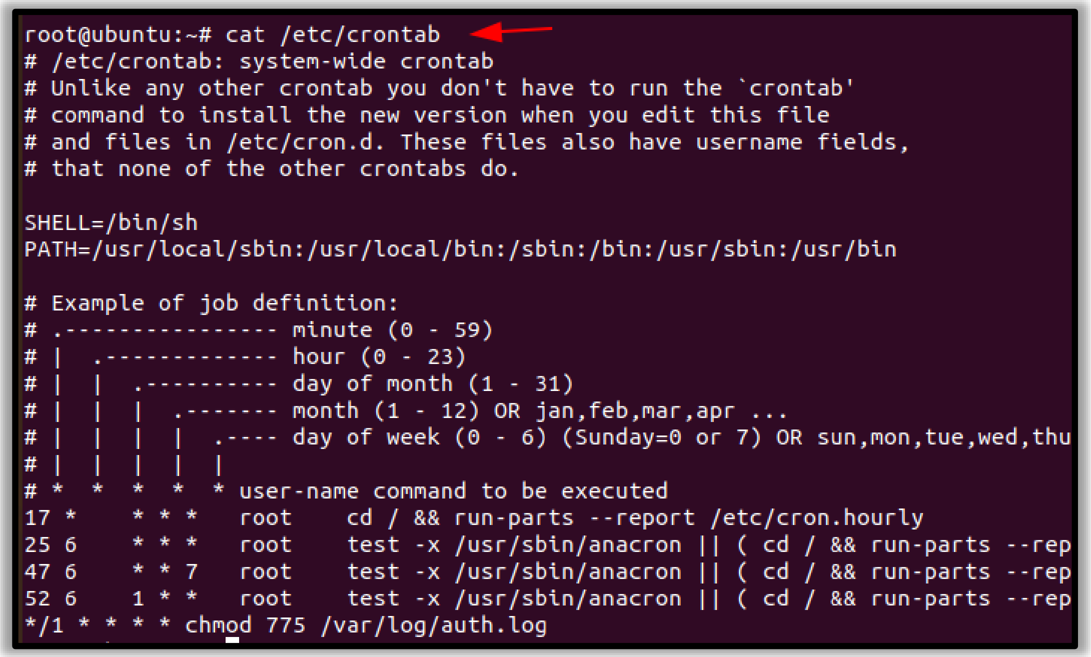
```
service --status-all
```



```
root@ubuntu:~# service --status-all
 [ + ]  acpid
 [ - ]  alsa-utils
 [ - ]  anacron
 [ - ]  apache-htcacheclean
 [ + ]  apache2
 [ + ]  apparmor
 [ + ]  apport
 [ + ]  avahi-daemon
 [ + ]  bluetooth
 [ - ]  cgroupfs-mount
 [ - ]  console-setup.sh
 [ + ]  cron
 [ + ]  cups
 [ + ]  cups-browsed
 [ + ]  dbus
```

The incident responder should look for any suspicious scheduled tasks and jobs. To find the scheduled tasks, you can use,

```
cat /etc/crontab
```

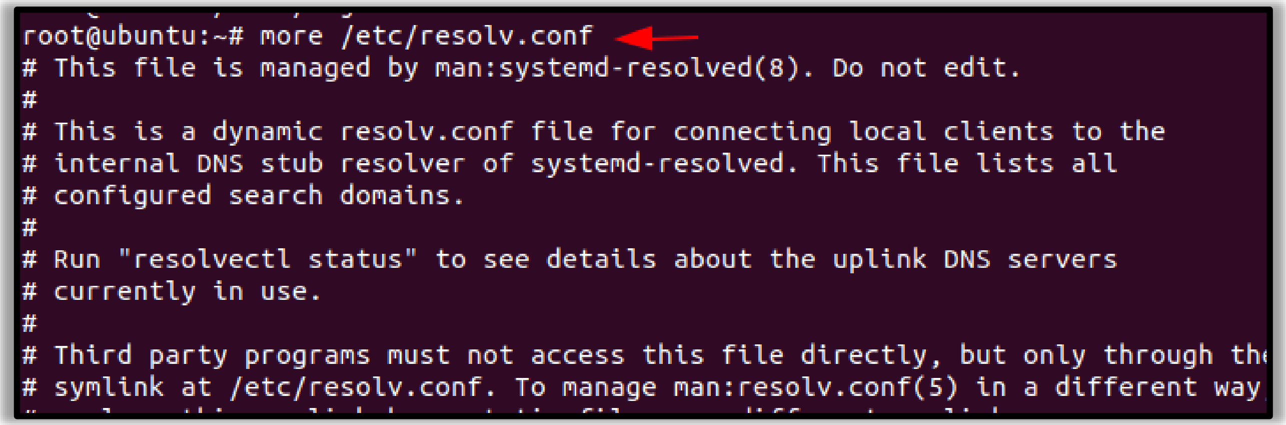


```
root@ubuntu:~# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --rep
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --rep
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --rep
*/1 * * * * chmod 775 /var/log/auth.log
```

## /etc/resolv.conf

To resolve DNS configuration issues and to avail a list of keywords with values that provide the various types of resolver information, you can use

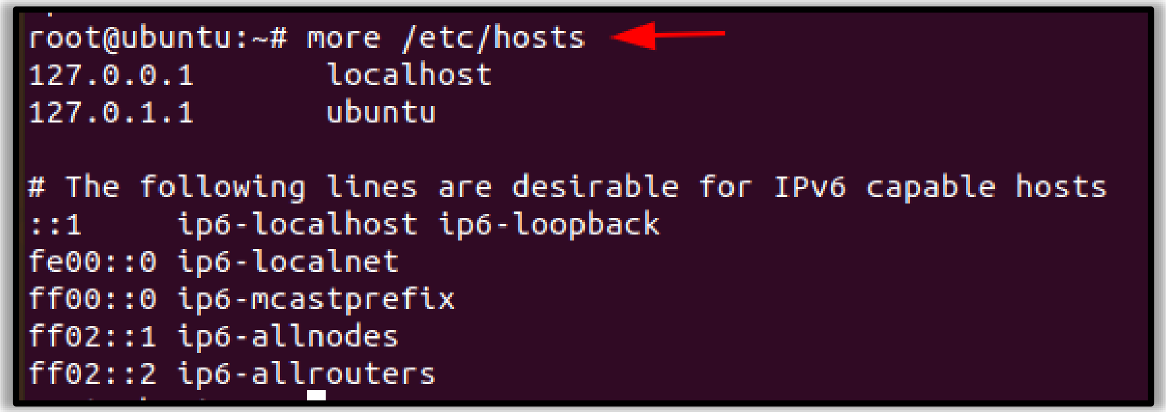
```
more /etc/resolv.conf
```



```
root@ubuntu:~# more /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way
```

To check file that translates hostnames or domain names to IP addresses, which is useful for testing changes to the website or the SSL setup, you can use

```
more /etc/hosts
```

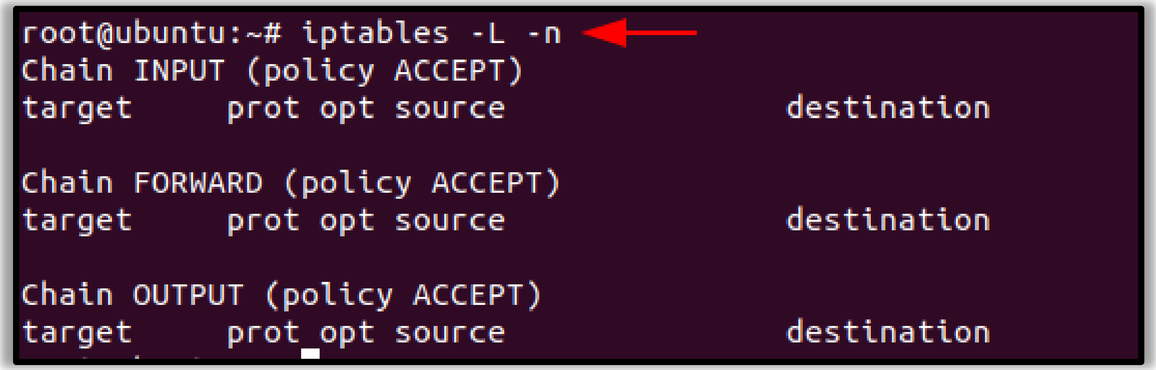


```
root@ubuntu:~# more /etc/hosts
127.0.0.1       localhost
127.0.1.1       ubuntu

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

## iptables

To check and manage the IPv4 packet filtering and NAT in Linux systems, you can use iptables and can make use of a variety of commands like:

```
iptables -L -n
```

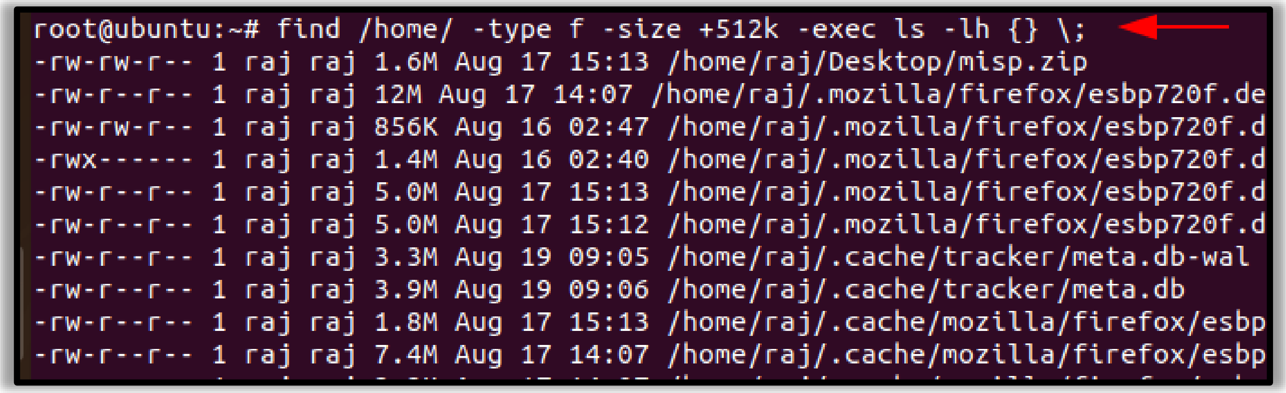


```
root@ubuntu:~# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

## Large Files

To identify any overly large files in your system and their permissions with their destination, you can use

```
find /home/ -type f -size +512k -exec ls -lh {} \;
```

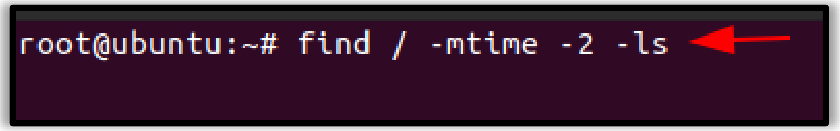


```
root@ubuntu:~# find /home/ -type f -size +512k -exec ls -lh {} \;
-rw-rw-r-- 1 raj raj 1.6M Aug 17 15:13 /home/raj/Desktop/misp.zip
-rw-r--r-- 1 raj raj 12M Aug 17 14:07 /home/raj/.mozilla/firefox/esbp720f.de
-rw-rw-r-- 1 raj raj 856K Aug 16 02:47 /home/raj/.mozilla/firefox/esbp720f.d
-rwx------ 1 raj raj 1.4M Aug 16 02:40 /home/raj/.mozilla/firefox/esbp720f.d
-rw-r--r-- 1 raj raj 5.0M Aug 17 15:13 /home/raj/.mozilla/firefox/esbp720f.d
-rw-r--r-- 1 raj raj 5.0M Aug 17 15:12 /home/raj/.mozilla/firefox/esbp720f.d
-rw-r--r-- 1 raj raj 3.3M Aug 19 09:05 /home/raj/.cache/tracker/meta.db-wal
-rw-r--r-- 1 raj raj 3.9M Aug 19 09:06 /home/raj/.cache/tracker/meta.db
-rw-r--r-- 1 raj raj 1.8M Aug 17 15:13 /home/raj/.cache/mozilla/firefox/esbp
-rw-r--r-- 1 raj raj 7.4M Aug 17 14:07 /home/raj/.cache/mozilla/firefox/esbp
```

## mtime

As an incident responder, if you want to see an anomalous file that has been present in the system for 2 days, you can use the command,

```
find / -mtime -2 -ls
```



```
root@ubuntu:~# find / -mtime -2 -ls
```

extremely vital to identify the overall picture of a system network and its health.

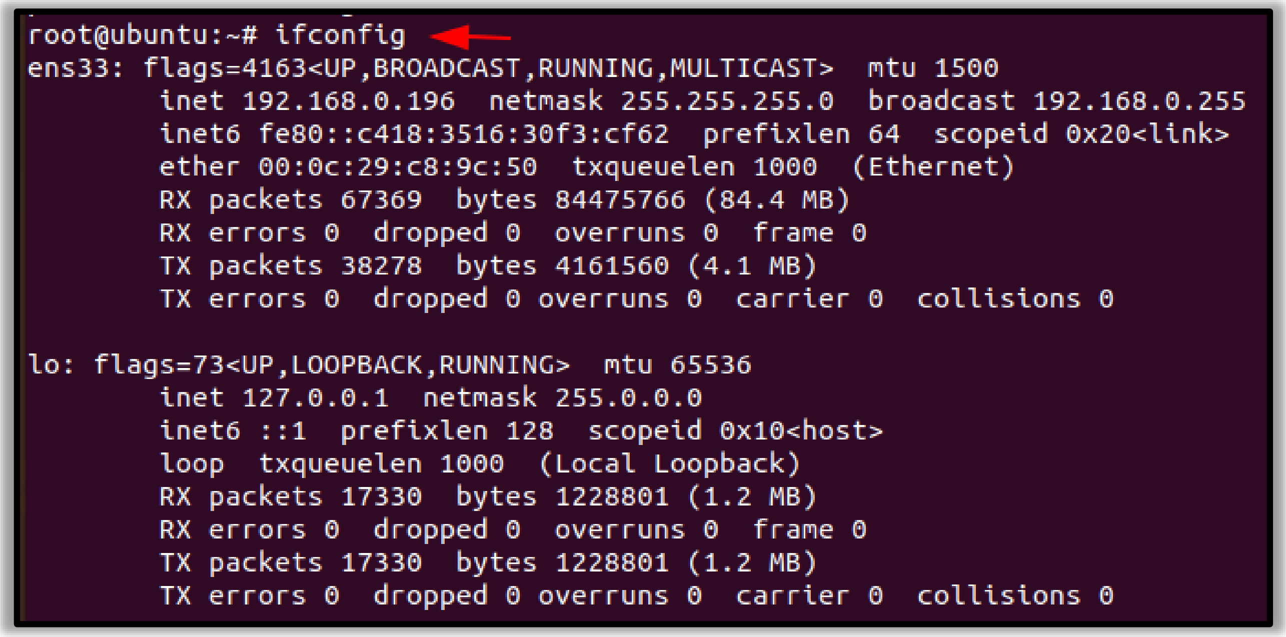
## ifconfig

To obtain the network activity information, you can use various commands.

```
ifconfig
```

To see all the network interfaces, you can use

```
ifconfig -a
```

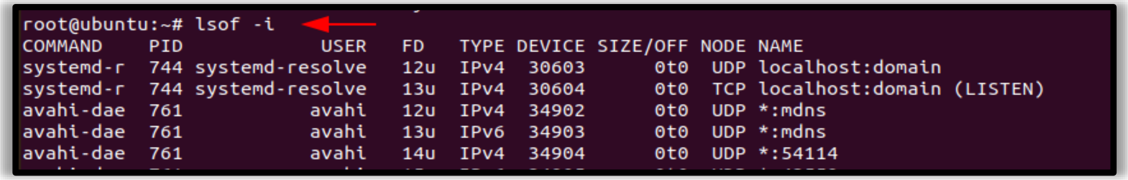


```
root@ubuntu:~# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.196  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::c418:3516:30f3:cf62  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:c8:9c:50  txqueuelen 1000  (Ethernet)
        RX packets 67369  bytes 84475766 (84.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 38278  bytes 4161560 (4.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 17330  bytes 1228801 (1.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17330  bytes 1228801 (1.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

## Open files

To list all the processes that are listening to ports with their PID, you can use

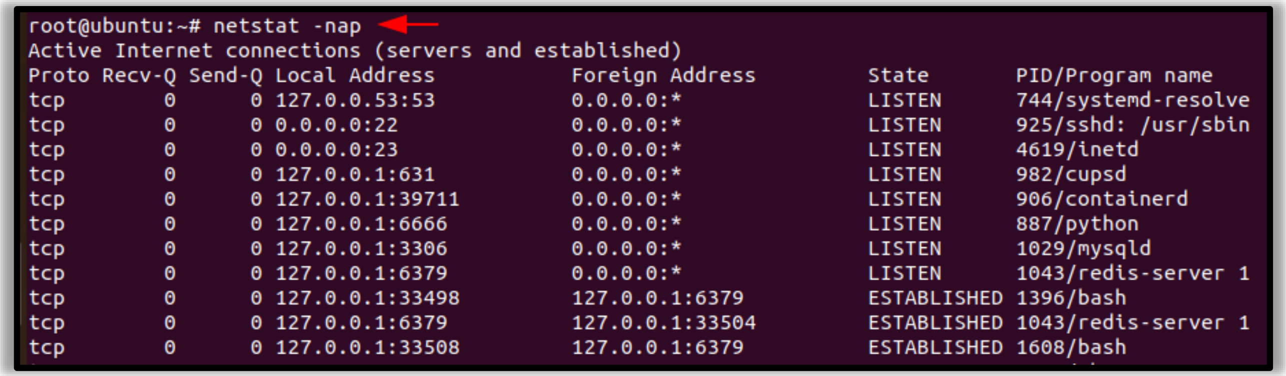
```
lsof -i
```



```
root@ubuntu:~# lsof -i
COMMAND     PID            USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r   744 systemd-resolve   12u  IPv4  30603      0t0  UDP localhost:domain
systemd-r   744 systemd-resolve   13u  IPv4  30604      0t0  TCP localhost:domain (LISTEN)
avahi-dae   761           avahi   12u  IPv4  34902      0t0  UDP *:mdns
avahi-dae   761           avahi   13u  IPv6  34903      0t0  UDP *:mdns
avahi-dae   761           avahi   14u  IPv4  34904      0t0  UDP *:54114
```

To display all the listening ports in the network use

```
netstat -nap
```

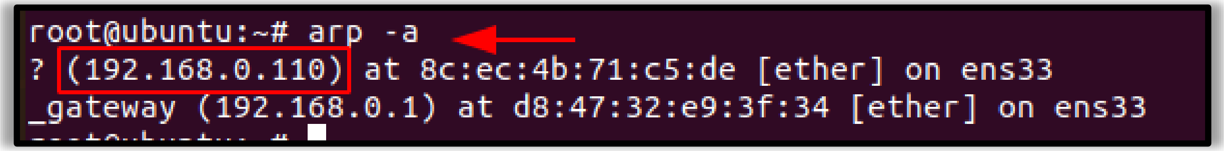


```
root@ubuntu:~# netstat -nap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      744/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      925/sshd: /usr/sbin
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN      4619/inetd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      982/cupsd
tcp        0      0 127.0.0.1:39711         0.0.0.0:*               LISTEN      906/containerd
tcp        0      0 127.0.0.1:6666          0.0.0.0:*               LISTEN      887/python
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      1029/mysqld
tcp        0      0 127.0.0.1:6379          0.0.0.0:*               LISTEN      1043/redis-server 1
tcp        0      0 127.0.0.1:33498         127.0.0.1:6379          ESTABLISHED 1396/bash
tcp        0      0 127.0.0.1:6379          127.0.0.1:33504         ESTABLISHED 1043/redis-server 1
tcp        0      0 127.0.0.1:33508         127.0.0.1:6379          ESTABLISHED 1608/bash
```

## arp

To display the system ARP cache, you can type

```
arp -a
```

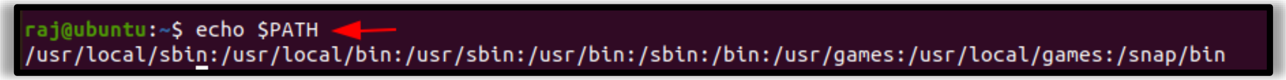


```
root@ubuntu:~# arp -a
? (192.168.0.110) at 8c:ec:4b:71:c5:de [ether] on ens33
_gateway (192.168.0.1) at d8:47:32:e9:3f:34 [ether] on ens33
root@ubuntu:~#
```

## path

The $PATH displays a list of directories that tells the shell which directories to search for executable files, to check for directories that are in your path you can use.

```
echo $PATH
```



```
raj@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```