OS monitoring, debugging and logging

Command Line Tools to Monitor Linux Performance

It’s really very tough job for every System or Network administrator to monitor and debug Linux System Performance problems every day. I’ve compiled the list of frequently used command line monitoring tools that might be useful for every Linux/Unix System Administrator. These commands are available under all flavors of Linux and can be useful to monitor and find the actual causes of performance problem. This list of commands shown here are very enough for you to pick the one that is suitable for your monitoring and debugging scenario.

Top – Linux Process Monitoring

Linux Top command is a performance monitoring program which is used frequently by many system administrators to monitor Linux performance and it is available under many Linux/Unix like operating systems. The top command used to display all the running and active real-time processes in ordered list and updates it regularly. It display *CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size, Process PID, User, Commands* and much more. It also shows high *memory and cpu* utilization of a running processes. The top command is much useful for system administrator to monitor and take correct action when required. Let's see top command in action.

```
# top
```

VmStat – Virtual Memory Statistics

Linux VmStat command used to display statistics of virtual memory, kernel threads, disks, system processes, I/O blocks, interrupts, CPU activity and much more. By default vmstat command is not available under Linux systems you need to install a package called sysstat that includes a vmstat program. The common usage of command format is.

```
# vmstat

procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free  inact active   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 810420  97380  70628    0    0   115     4   89   79  1  6 90  3  0
```

Lsof – List Open Files

Lsof command used in many Linux/Unix like system that is used to display list of all the open files and the processes. The open files included are disk files, network sockets, pipes, devices and processes. One of the main reason for using this command is when a disk cannot be unmounted and displays the error that files are being used or opened. With this command you can easily identify which files are in use. The most common format for this command is.

```
# lsof

COMMAND     PID      USER   FD      TYPE     DEVICE     SIZE       NODE NAME
init          1      root  cwd       DIR      104,2     4096          2 /
init          1      root  rtd       DIR      104,2     4096          2 /
init          1      root  txt       REG      104,2    38652   17710339 /sbin/init
init          1      root  mem       REG      104,2   129900     196453 /lib/ld-2.5.so
init          1      root  mem       REG      104,2  1693812     196454 /lib/libc-2.5.so
init          1      root  mem       REG      104,2    20668     196479 /lib/libdl-2.5.so
init          1      root  mem       REG      104,2   245376     196419 /lib/libsepol.so.1
init          1      root  mem       REG      104,2    93508     196431 /lib/libselinux.so.1
init          1      root   10u     FIFO       0,17                 953 /dev/initctl
```

Tcpdump – Network Packet Analyzer

Tcpdump one of the most widely used command-line network packet analyzer or packets sniffer program that is used capture or filter TCP/IP packets that received or transferred on a specific interface over a network. It also provides a option to save captured packages in a file for later analysis. tcpdump is almost available in all major Linux distributions.

```
# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
22:08:59.617628 IP tecmint.com.ssh > 115.113.134.3.static-mumbai.vsnl.net.in.28472: P 2532133365:2532133481(116) ack 3561562349 win 9648

22:09:07.653466 IP tecmint.com.ssh > 115.113.134.3.static-mumbai.vsnl.net.in.28472: P 116:232(116) ack 1 win 9648

22:08:59.617916 IP 115.113.134.3.static-mumbai.vsnl.net.in.28472 > tecmint.com.ssh: . ack 116 win 64347
```

Netstat – Network Statistics

Netstat is a command line tool for monitoring incoming and outgoing network packets statistics as well as interface statistics. It is very useful tool for every system administrator to monitor network performance and troubleshoot network related problems.

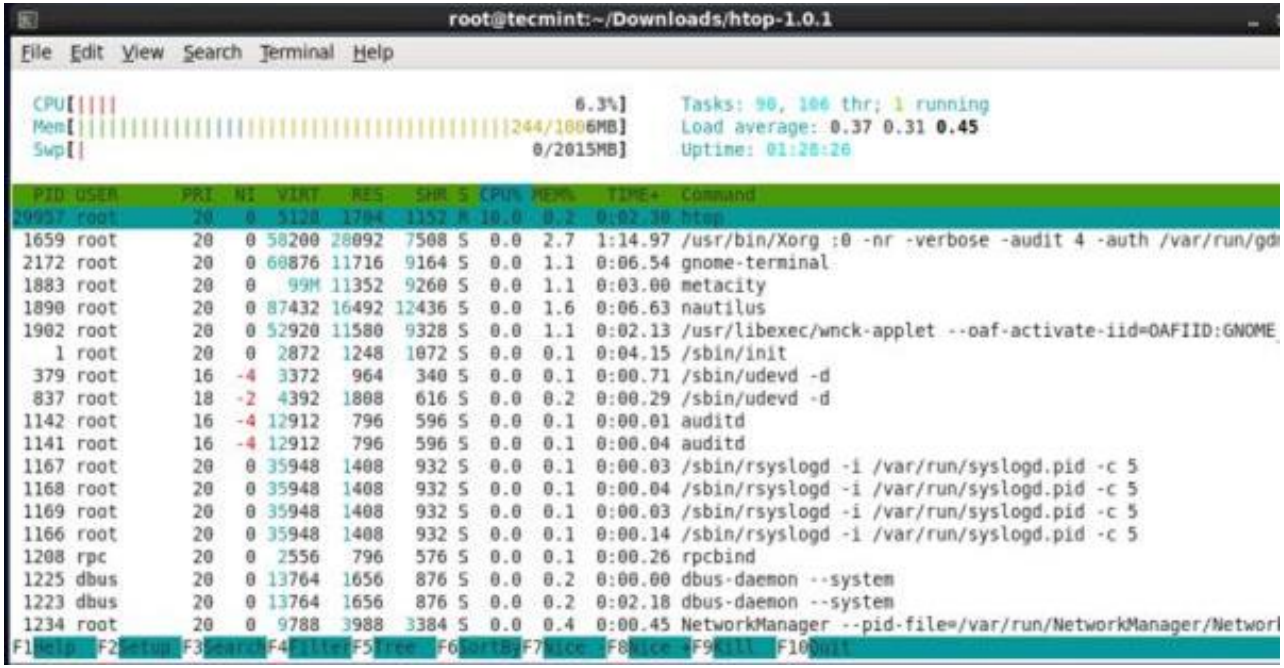
```
# netstat -a | more

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 *:mysql                     *:*                         LISTEN
tcp        0      0 *:sunrpc                    *:*                         LISTEN
tcp        0      0 *:realm-rusd                *:*                         LISTEN
tcp        0      0 *:ftp                       *:*                         LISTEN
tcp        0      0 localhost.localdomain:ipp   *:*                         LISTEN
tcp        0      0 localhost.localdomain:smtp  *:*                         LISTEN
tcp        0      0 localhost.localdomain:smtp  localhost.localdomain:42709 TIME_WAIT
tcp        0      0 localhost.localdomain:smtp  localhost.localdomain:42710 TIME_WAIT
tcp        0      0 *:http                      *:*                         LISTEN
tcp        0      0 *:ssh                       *:*                         LISTEN
tcp        0      0 *:https                     *:*                         LISTEN
```

Htop – Linux Process Monitoring

Htop is a much advanced interactive and real time Linux process monitoring tool. This is much similar to Linux top command but it has some rich features like user friendly interface to manage process, shortcut keys, vertical and horizontal view of the processes and much more. Htop is a third party tool and doesn't included in Linux systems, you need to install it using YUM package manager tool. For more information on installation read our article below.

```
# htop
```

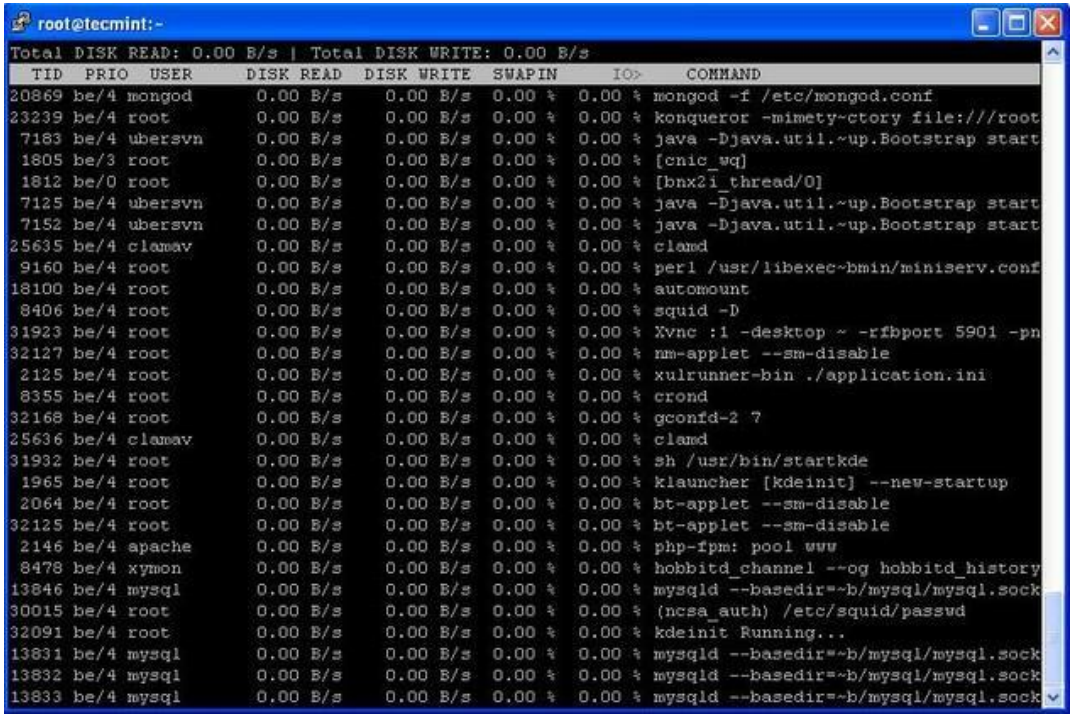
Htop – Linux Process Monitoring



Iotop – Monitor Linux Disk I/O

Iotop is also much similar to top command and Htop program, but it has accounting function to monitor and display real time Disk I/O and processes. This tool is much useful for finding the exact process and high used disk read/writes of the processes.

```
# iotop
```



Iostat – Input/Output Statistics

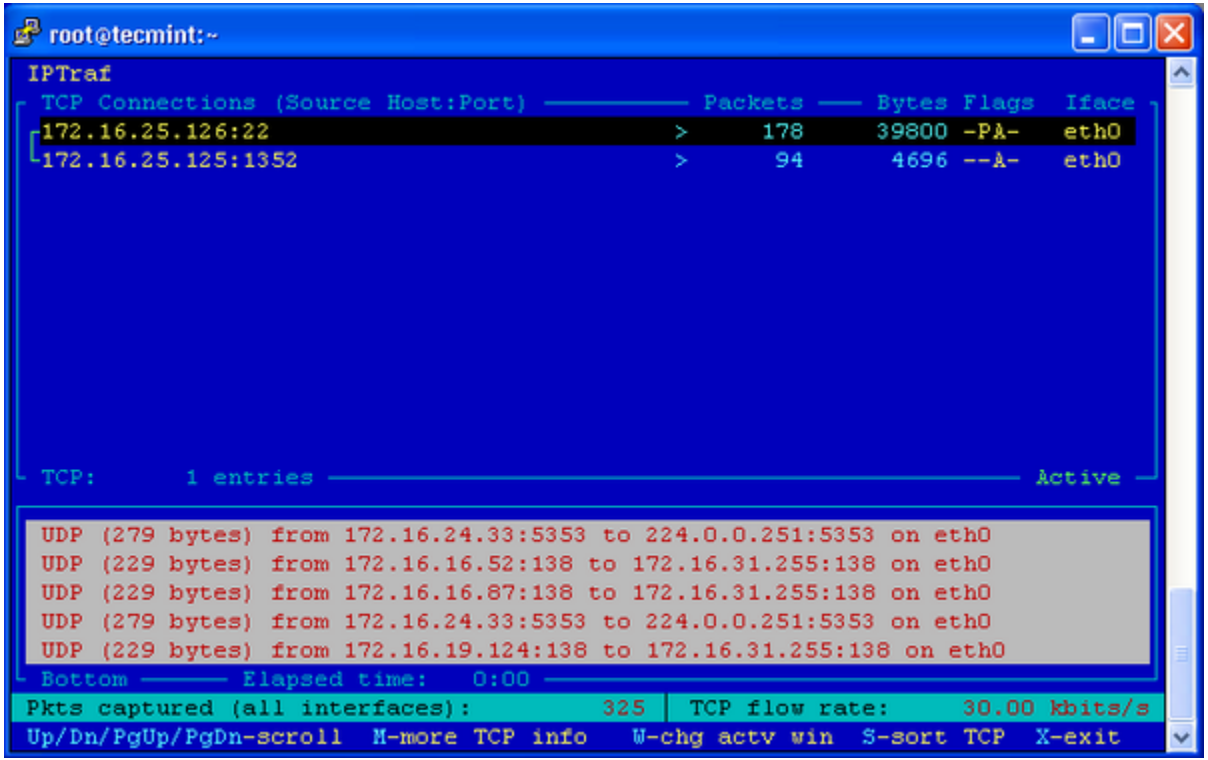
IoStat is simple tool that will collect and show system input and output storage device statistics. This tool is often used to trace storage device performance issues including devices, local disks, remote disks such as NFS.

```
root@shuttle:~# iostat
Linux 6.2.0-35-generic (shuttle)        20.02.24        _x86_64_        (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5,70    0,02    3,48    0,43    0,00   90,37

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
loop0             0,00         0,00         0,00         0,00         64          0          0
loop1             0,01         0,20         0,00         0,00    1973490          0          0
loop10            0,00         0,29         0,00         0,00    2873086          0          0
loop11            0,00         0,00         0,00         0,00       8100          0          0
loop12            0,00         0,00         0,00         0,00       9512          0          0
loop13            0,00         0,04         0,00         0,00     448637          0          0
loop14            0,00         0,00         0,00         0,00      38458          0          0
loop15            0,00         0,00         0,00         0,00       1285          0          0
loop16            0,00         0,57         0,00         0,00    5771310          0          0
loop17            0,00         0,11         0,00         0,00    1127506          0          0
loop18            0,00         0,00         0,00         0,00      13600          0          0
loop19            0,00         0,00         0,00         0,00       1364          0          0
loop2             0,00         0,15         0,00         0,00    1494186          0          0
loop3             0,00         0,02         0,00         0,00     212728          0          0
loop4             0,00         0,00         0,00         0,00       6215          0          0
loop5             0,01         1,12         0,00         0,00   11273767          0          0
loop6             0,00         0,16         0,00         0,00    1566835          0          0
loop7             0,00         0,03         0,00         0,00     307092          0          0
loop8             0,00         0,01         0,00         0,00     147844          0          0
loop9             0,01         1,59         0,00         0,00   16020937          0          0
sda              18,22        89,10       273,39        54,84  896238335 2749977885  551615048
sdb               3,62       215,38        25,49         0,00 2166505207  256355920          0
```

IPTraf – Real Time IP LAN Monitoring

IPTraf is an open source console-based real time network (IP LAN) monitoring utility for Linux. It collects a variety of information such as IP traffic monitor that passes over the network, including TCP flag information, ICMP details, TCP/UDP traffic breakdowns, TCP connection packet and by one counts. It also gathers information of general and detailed interface statistics of TCP, UDP, IP, ICMP, non-IP, IP checksum errors, interface activity etc.

IPTraf – Real Time IP LAN Monitoring



Psacct or Acct – Monitor User Activity

psacct or acct tools are very useful for monitoring each users activity on the system. Both daemons runs in the background and keeps a close watch on the overall activity of each user on the system and also what resources are being consumed by them.

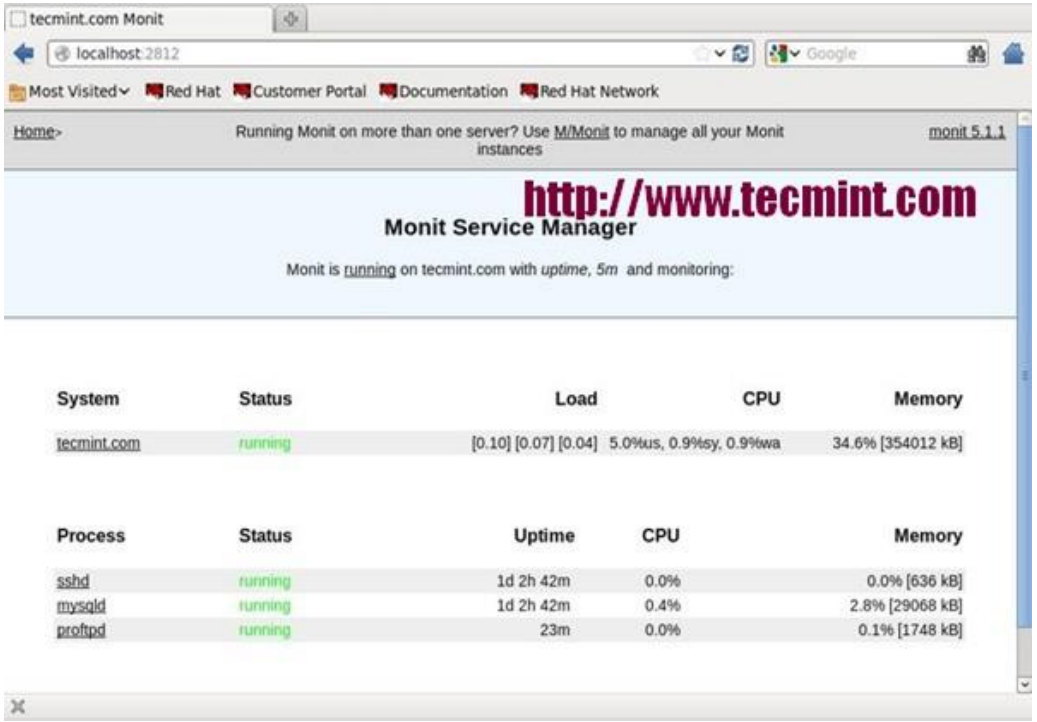
These tools are very useful for system administrators to track each users activity like what they are doing, what commands they issued, how much resources are used by them, how long they are active on the system etc.

Monit – Linux Process and Services Monitoring

Monit is a free open source and web based process supervision utility that automatically monitors and managers system processes, programs, files, directories, permissions, checksums and filesystems.

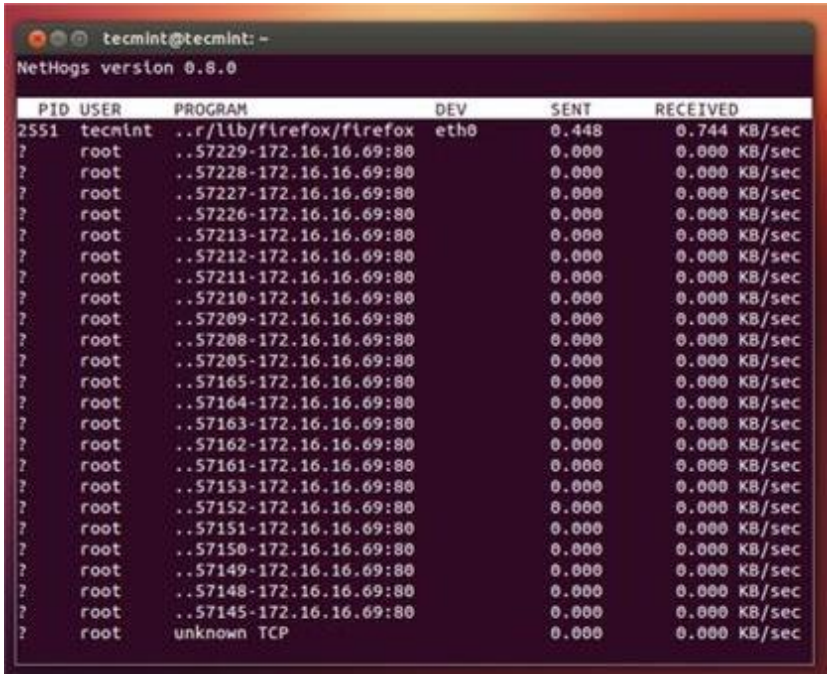
It monitors services like Apache, MySQL, Mail, FTP, ProFTP, Nginx, SSH and so on. The system status can be viewed from the command line or using it own web interface.

Monit – Linux Process and Services Monitoring



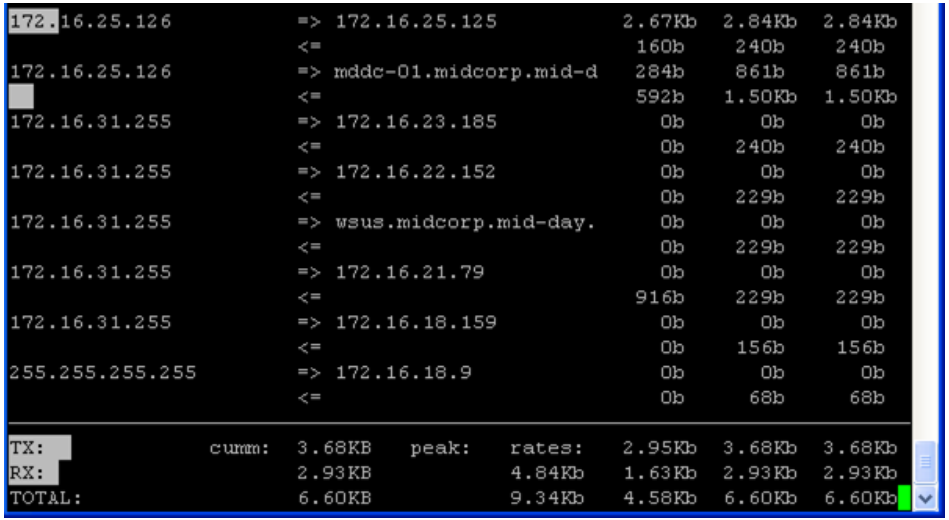
NetHogs – Monitor Per Process Network Bandwidth

NetHogs is an open source nice small program (similar to Linux top command) that keeps a tab on each process network activity on your system. It also keeps a track of real time network traffic bandwidth used by each program or application.



iftop – Network Bandwidth Monitoring

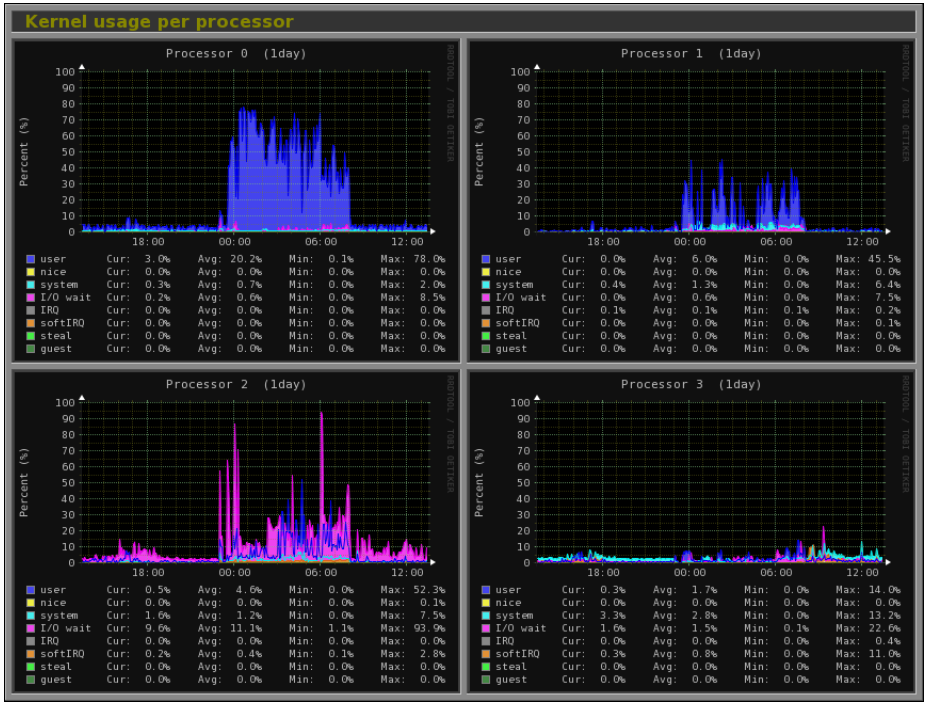
iftop is another terminal-based free open source system monitoring utility that displays a frequently updated list of network bandwidth utilization (source and destination hosts) that passing through the network interface on your system. iftop is considered for network usage, what ‘top‘ does for CPU usage. iftop is a ‘top‘ family tool that monitor a selected interface and displays a current bandwidth usage between two hosts.



Monitorix – System and Network Monitoring

Monitorix is a free lightweight utility that is designed to run and monitor system and network resources as many as possible in Linux/Unix servers. It has a built in HTTP web server that regularly collects system and network information and display them in graphs. It Monitors system load average and usage, memory allocation, disk driver health, system services, network ports, mail statistics (Sendmail, Postfix, Dovecot, etc), MySQL statistics and many more. It designed to monitor overall system performance and helps in detecting failures, bottlenecks, abnormal activities etc.

Monitorix – System and Network Monitoring



Arpwatch – Ethernet Activity Monitor

Arpwatch is a kind of program that is designed to monitor Address Resolution (MAC and IP address changes) of Ethernet network traffic on a Linux network. It continuously keeps watch on Ethernet traffic and produces a log of IP and MAC address pair changes along with a timestamps on a network. It also has a feature to send an email alerts to administrator, when a pairing added or changes. It is very useful in detecting ARP spoofing on a network.

Suricata – Network Security Monitoring

Suricata is an high performance open source Network Security and Intrusion Detection and Prevention Monitoring System for Linux, FreeBSD and Windows.It was designed and owned by a non-profit foundation OISF (Open Information Security Foundation).

VnStat PHP – Monitoring Network Bandwidth

VnStat PHP a web based frontend application for most popular networking tool called “vnstat“. VnStat PHP monitors a network traffic usage in nicely graphical mode. It displays a total IN and OUT network traffic usage in hourly, daily, monthly and full summary report.

The strace Command

The strace command can be used to intercept and record the system calls made, and the signals received by a process. This allows examination of the boundary layer between the user and kernel space which can be very useful for identifying why a process is failing.

```
# strace ls file1
execve("/bin/ls", ["ls", "file1"], [/* 21 vars */]) = 0
brk(0)                                  = 0xadb000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f516bb79000
.....
close(1)                                = 0
munmap(0x7f516bb78000, 4096)            = 0
close(2)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Nagios – Network/Server Monitoring

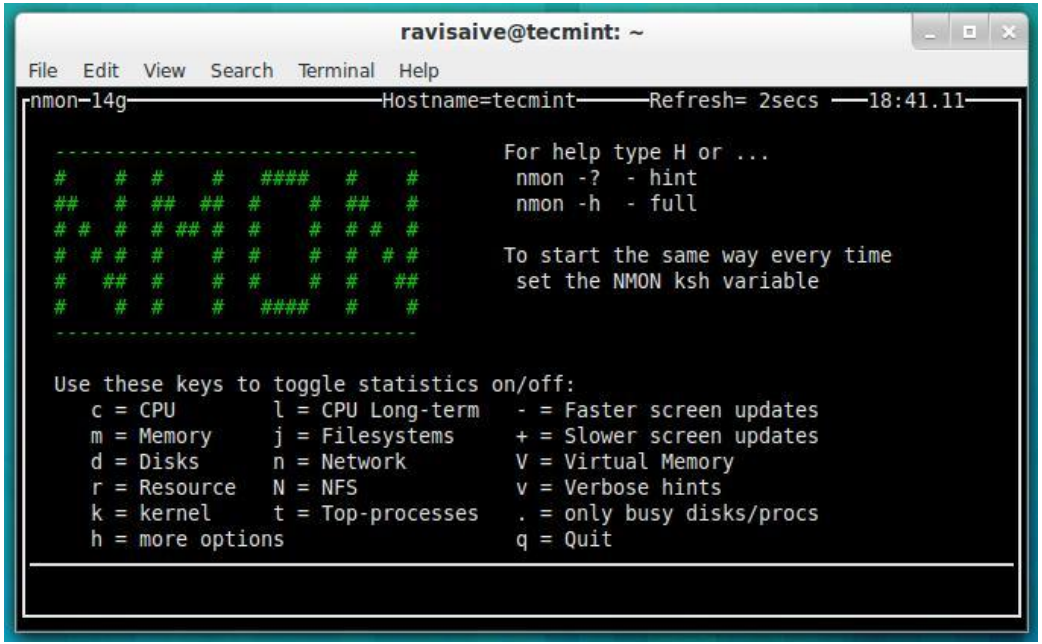
Nagios is an leading open source powerful monitoring system that enables network/system administrators to identify and resolve server related problems before they affect major business processes. With the Nagios system, administrators can able to monitor remote Linux, Windows, Switches, Routers and Printers on a single window. It shows critical warnings and indicates if something went wrong in your network/server which indirectly helps you to begin remediation processes before they occur.

Nmon: Monitor Linux Performance

Nmon (stands for Nigel's performance Monitor) tool, which is used to monitor all Linux resources such as CPU, Memory, Disk Usage, Network, Top processes, NFS, Kernel and much more. This tool comes in two modes: Online Mode and Capture Mode.

The Online Mode, is used for real-time monitoring and Capture Mode, is used to store the output in CSV format for later processing.

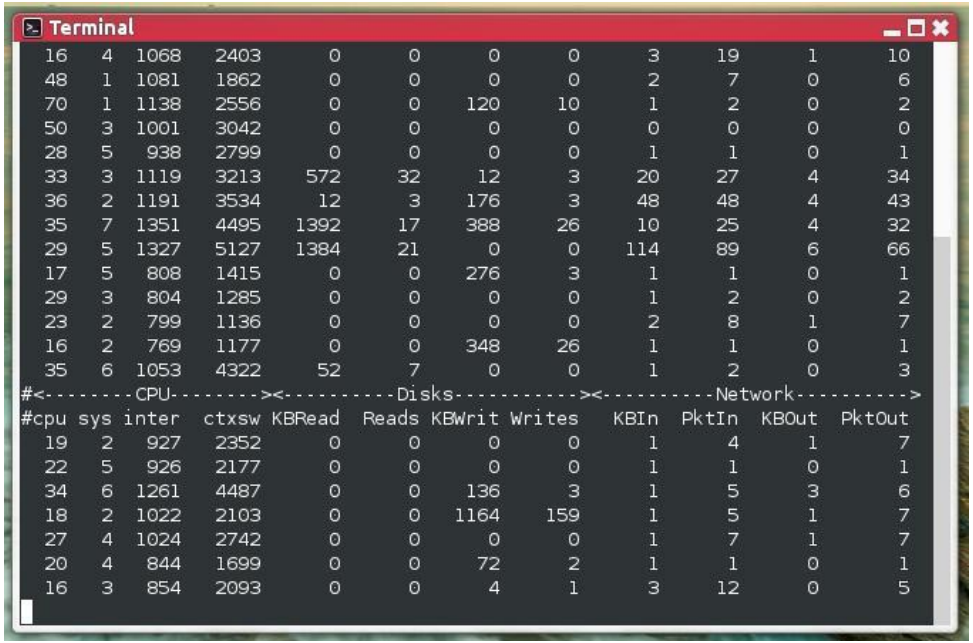
Nmon: Monitor Linux Performance



Collectl: All-in-One Performance Monitoring Tool

Collectl is a yet another powerful and feature rich command line based utility, that can be used to gather information about Linux system resources such as CPU usage, memory, network, inodes, processes, nfs, tcp, sockets and much more.

Collectl: All-in-One Performance Monitoring Tool



/proc

/proc is a virtual file system. For example, if you do ls -l /proc/stat, you'll notice that it has a size of 0 bytes, but if you do "cat /proc/stat", you'll see some content inside the file.

Do a ls -l /proc, and you'll see lot of directories with just numbers. These numbers represents the process ids, the files inside this numbered directory corresponds to the process with that particular PID.

The following are the important files located under each numbered directory (for each process):

- cmdline – command line of the command.
- environ – environment variables.
- fd – Contains the file descriptors which is linked to the appropriate files.
- limits – Contains the information about the specific limits to the process.
- mounts – mount related information

The following are the important links under each numbered directory (for each process):

- cwd – Link to current working directory of the process.
- exe – Link to executable of the process.
- root – Link to the root directory of the process.

User management

Adding User Accounts

To add a new user account, you can run either of the following two commands as root.

```
# adduser [new_account]
# useradd [new_account]
```

When a new user account is added to the system, the following operations are performed.

1. His/her home directory is created (/home/username by default).
2. The following hidden files are copied into the user's home directory, and will be used to provide environment variables for his/her user session.

```
.bash_logout
.bash_profile
.bashrc
```

3. A mail spool is created for the user at /var/spool/mail/username.
4. A group is created and given the same name as the new user account.

Adding User Accounts: **Understanding /etc/passwd**

The full account information is stored in the /etc/passwd file. This file contains a record per system user account and has the following format (fields are delimited by a colon).

```
[username]:[x]:[UID]:[GID]:[Comment]:[Home directory]:[Default shell]
```

Fields [username] and [Comment] are self explanatory.

- The x in the second field indicates that the account is protected by a shadowed password (in /etc/shadow), which is needed to logon as [username].
- The [UID] and [GID] fields are integers that represent the User IDentification and the primary Group IDentification to which [username] belongs, respectively.
- The [Home directory] indicates the absolute path to [username]'s home directory, and
- The [Default shell] is the shell that will be made available to this user when he or she logins the system.

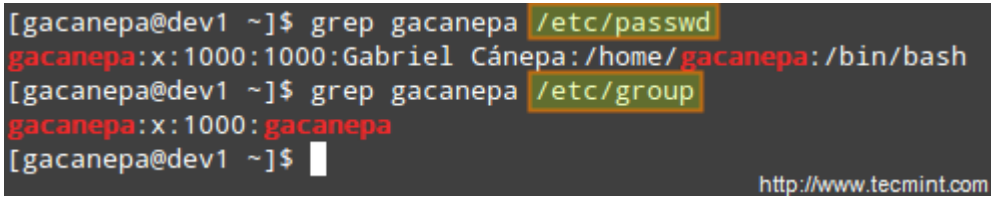
Adding User Accounts: **Understanding /etc/group**

Group information is stored in the /etc/group file. Each record has the following format.

```
[Group name]:[Group password]:[GID]:[Group members]
```

- [Group name] is the name of group.
- An x in [Group password] indicates group passwords are not being used.
- [GID]: same as in /etc/passwd.
- [Group members]: a comma separated list of users who are members of [Group name].

Adding User Accounts: **Understanding /etc/group**



After adding an account, you can edit the following information (to name a few fields) using the usermod command, whose basic syntax of usermod is as follows.

```
# usermod [options] [username]
```

Adding User Accounts: **Understanding /etc/group**

Setting the expiry date for an account

Use the **–expiredate** flag followed by a date in YYYY-MM-DD format.

```
# usermod --expiredate 2014-10-30 tecmint
```

Adding the user to supplementary groups

Use the combined **-aG**, or **–append –groups** options, followed by a comma separated list of groups.

```
# usermod --append --groups root,users tecmint
```

Adding User Accounts: **Understanding /etc/group**

Changing the default location of the user's home directory

Use the **-d**, or **–home** options, followed by the absolute path to the new home directory.

```
# usermod --home /tmp tecmint
```

Changing the shell the user will use by default

Use –shell, followed by the path to the new shell.

```
# usermod --shell /bin/sh tecmint
```

Adding User Accounts: **Understanding /etc/group**

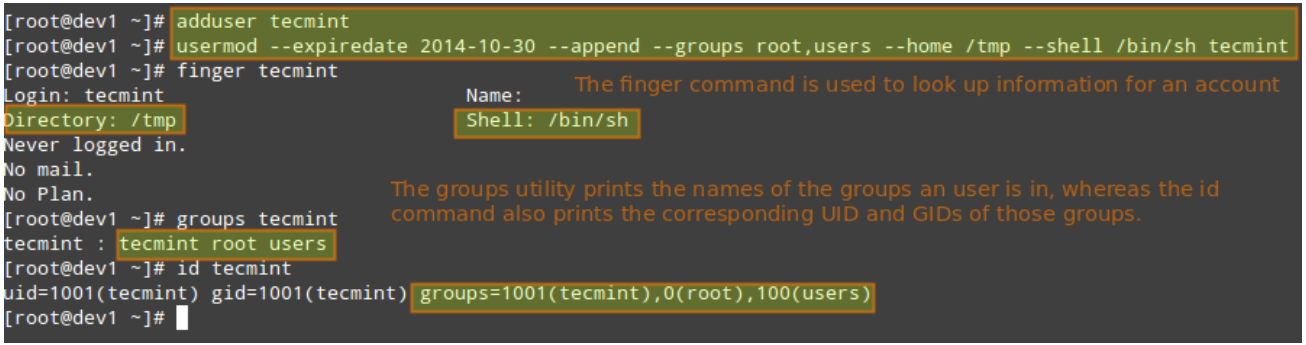
Displaying the groups an user is a member of

```
# groups tecmint
# id tecmint
```

Now let’s execute all the above commands in one go.

```
# usermod --expiredate 2014-10-30 --append --groups root,users --home /tmp --shell /bin/sh tecmint
```

Adding User Accounts: **Understanding /etc/group**



In the example above, we will set the expiry date of the tecmint user account to October 30th, 2014. We will also add the account to the root and users group. Finally, we will set sh as its default shell and change the location of the home directory to /tmp:

Adding User Accounts: **Understanding /etc/group**

For existing accounts, we can also do the following.

Disabling account by locking password

Use the -L (uppercase L) or the –lock option to lock a user's password.

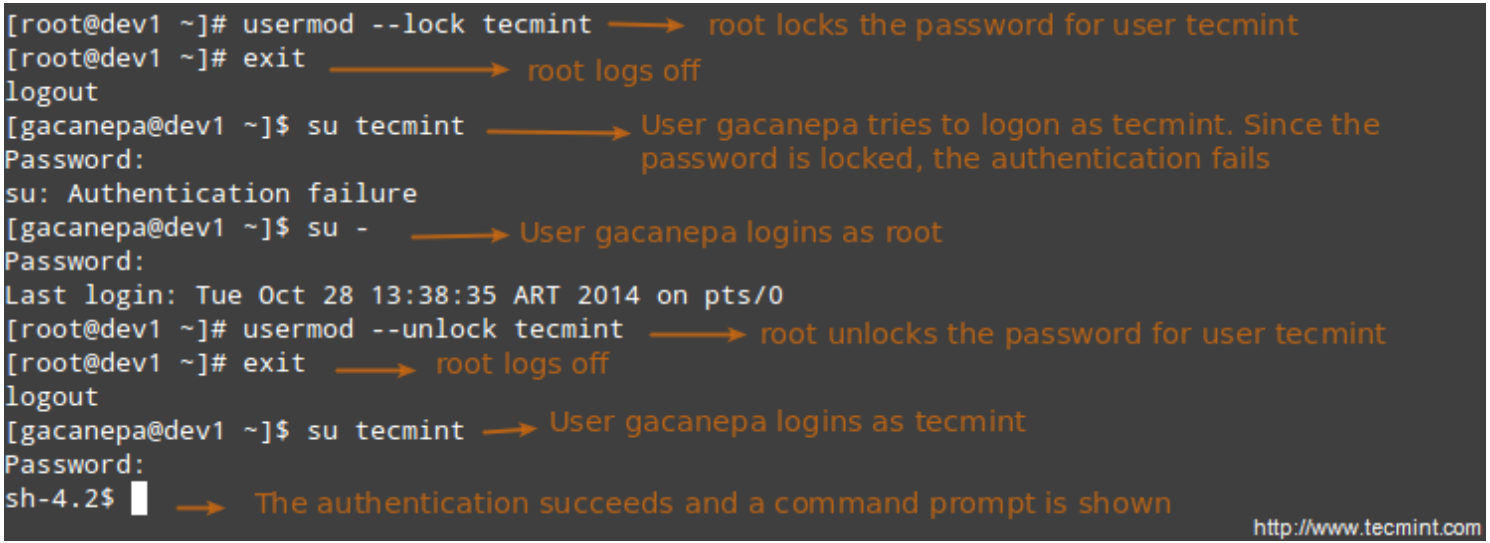
```
# usermod --lock tecmint
```

Unlocking user password

Use the –u or the –unlock option to unlock a user's password that was previously blocked.

```
# usermod --unlock tecmint
```

Adding User Accounts: **Understanding /etc/group**



Adding User Accounts: **Understanding /etc/group**

Creating a new group for read and write access to files that need to be accessed by several users

Run the following series of commands to achieve the goal.

```
# groupadd common_group # Add a new group
# chown :common_group common.txt # Change the group owner of common.txt to common_group
# usermod -aG common_group user1 # Add user1 to common_group
# usermod -aG common_group user2 # Add user2 to common_group
# usermod -aG common_group user3 # Add user3 to common_group
```

Adding User Accounts: **Understanding /etc/group**

Deleting a group

You can delete a group with the following command.

```
# groupdel [group_name]
```

If there are files owned by **group_name**, they will not be deleted, but the group owner will be set to the *GID* of the group that was deleted.

Linux File Permissions

Besides the basic read, write, and execute permissions that we discussed in Archiving Tools and Setting File Attributes – Part 3 of this series, there are other less used (but not less important) permission settings, sometimes referred to as “special permissions”.

Like the basic permissions discussed earlier, they are set using an octal file or through a letter (symbolic notation) that indicates the type of permission.

Deleting user accounts

You can delete an account (along with its home directory, if it’s owned by the user, and all the files residing therein, and also the mail spool) using the userdel command with the –remove option.

```
# userdel --remove [username]
```

Group Management

Every time a new user account is added to the system, a group with the same name is created with the username as its only member. Other users can be added to the group later. One of the purposes of groups is to implement a simple access control to files and other system resources by setting the right permissions on those resources.

For example, suppose you have the following users.

- user1 (primary group: user1)
- user2 (primary group: user2)
- user3 (primary group: user3)

All of them need read and write access to a file called common.txt located somewhere on your local system, or maybe on a network share that user1 has created. You may be tempted to do something like

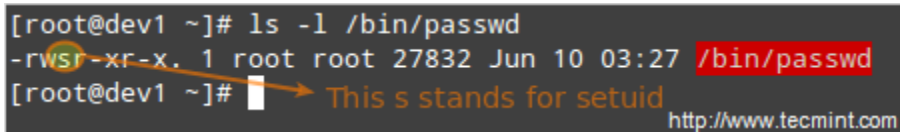
```
# chmod 660 common.txt
OR
# chmod u=rw,g=rw,o= common.txt [notice the space between the last equal sign and the file name]
```

However, this will only provide read and write access to the owner of the file and to those users who are members of the group owner of the file (user1 in this case). Again, you may be tempted to add user2 and user3 to group user1, but that will also give them access to the rest of the files owned by user user1 and group user1.

Understanding Setuid

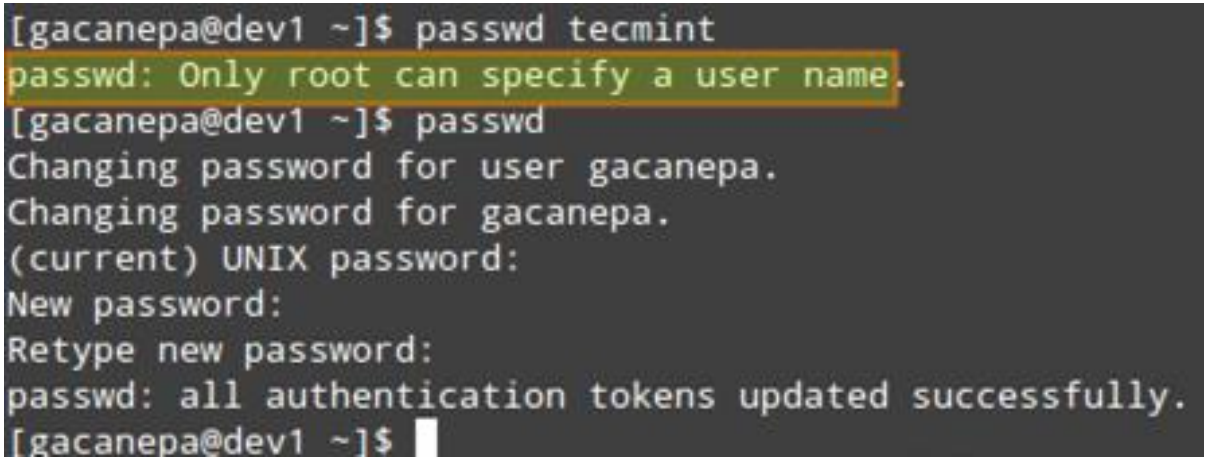
When the setuid permission is applied to an executable file, an user running the program inherits the effective privileges of the program's owner. Since this approach can reasonably raise security concerns, the number of files with setuid permission must be kept to a minimum. You will likely find programs with this permission set when a system user needs to access a file owned by root.

Summing up, it isn't just that the user can execute the binary file, but also that he can do so with root's privileges. For example, let's check the permissions of /bin/passwd. This binary is used to change the password of an account, and modifies the /etc/shadow file. The superuser can change anyone's password, but all other users should only be able to change their own.



Understanding Setuid

Thus, any user should have permission to run /bin/passwd, but only root will be able to specify an account. Other users can only change their corresponding passwords.



Understanding Setgid

When the setgid bit is set, the effective GID of the real user becomes that of the group owner. Thus, any user can access a file under the privileges granted to the group owner of such file. In addition, when the setgid bit is set on a directory, newly created files inherit the same group as the directory, and newly created subdirectories will also inherit the setgid bit of the parent directory. You will most likely use this approach whenever members of a certain group need access to all the files in a directory, regardless of the file owner's primary group.

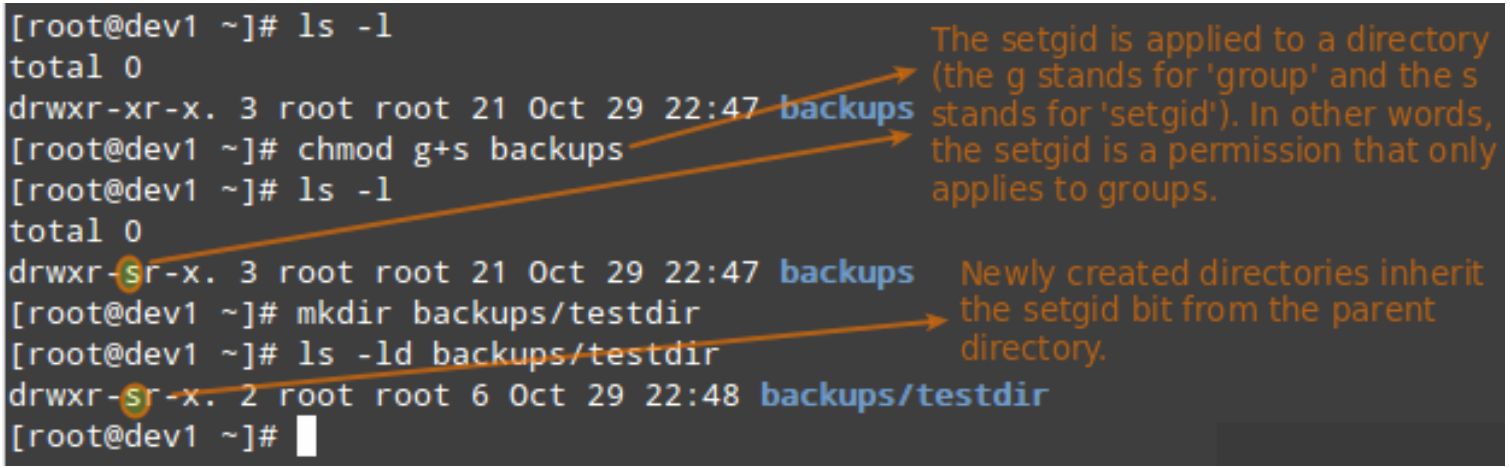
```
# chmod g+s [filename]
```

To set the setgid in octal form, prepend the number 2 to the current (or desired) basic permissions.

```
# chmod 2755 [directory]
```

Understanding Setgid

Setting the SETGID in a directory



Understanding Sticky Bit

When the *"sticky bit"* is set on files, Linux just ignores it, whereas for directories it has the effect of preventing users from deleting or even renaming the files it contains unless the user owns the directory, the file, or is root.

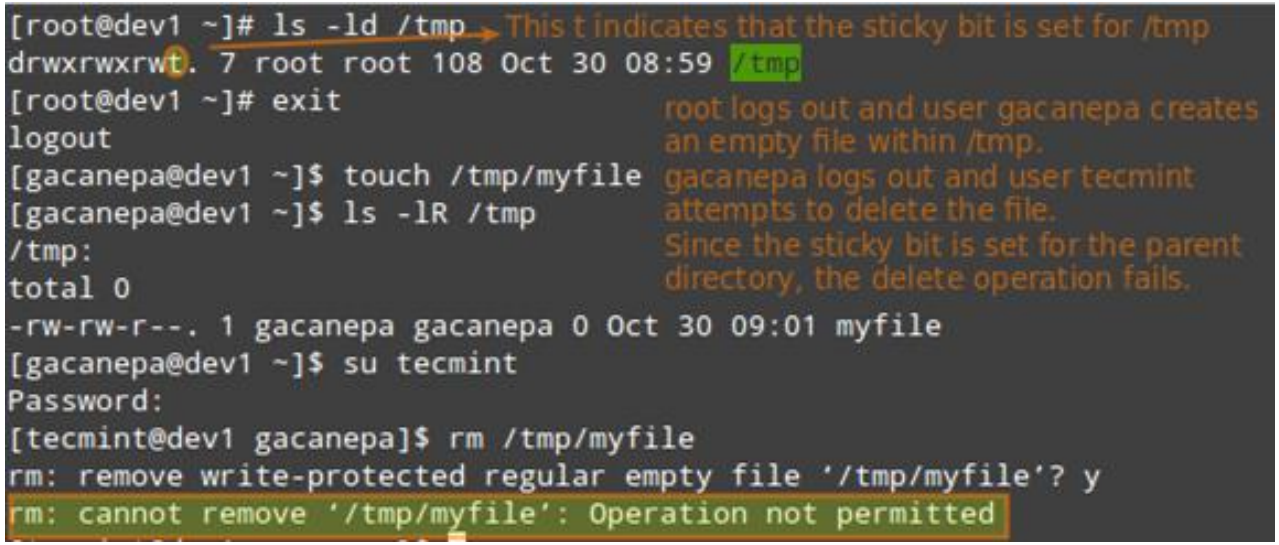
```
# chmod o+t [directory]
```

To set the sticky bit in octal form, prepend the number 1 to the current (or desired) basic permissions.

```
# chmod 1755 [directory]
```

Understanding Sticky Bit

Without the sticky bit, anyone able to write to the directory can delete or rename files. For that reason, the sticky bit is commonly found on directories, such as /tmp, that are world-writable.



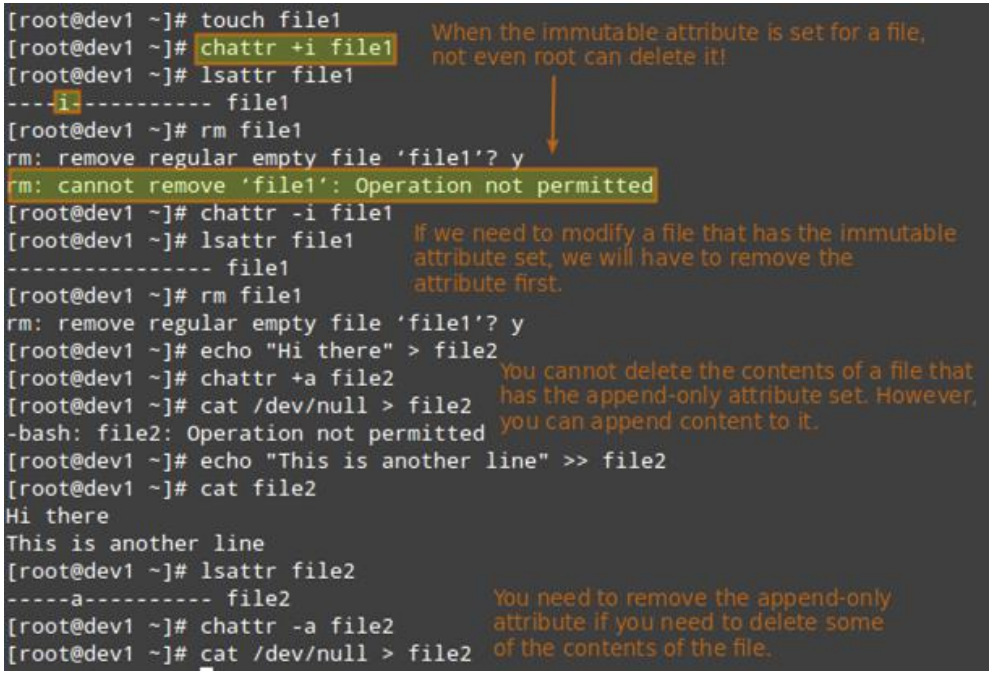
Special Linux File Attributes

There are other attributes that enable further limits on the operations that are allowed on files. For example, prevent the file from being renamed, moved, deleted, or even modified. They are set with the chattr command and can be viewed using the lsattr tool, as follows.

```
# chattr +i file1
# chattr +a file2
```

Special Linux File Attributes

After executing those two commands, file1 will be immutable (which means it cannot be moved, renamed, modified or deleted) whereas file2 will enter append-only mode (can only be open in append mode for writing).



Accessing the root Account and Using sudo

One of the ways users can gain access to the root account is by typing.

```
$ su
```

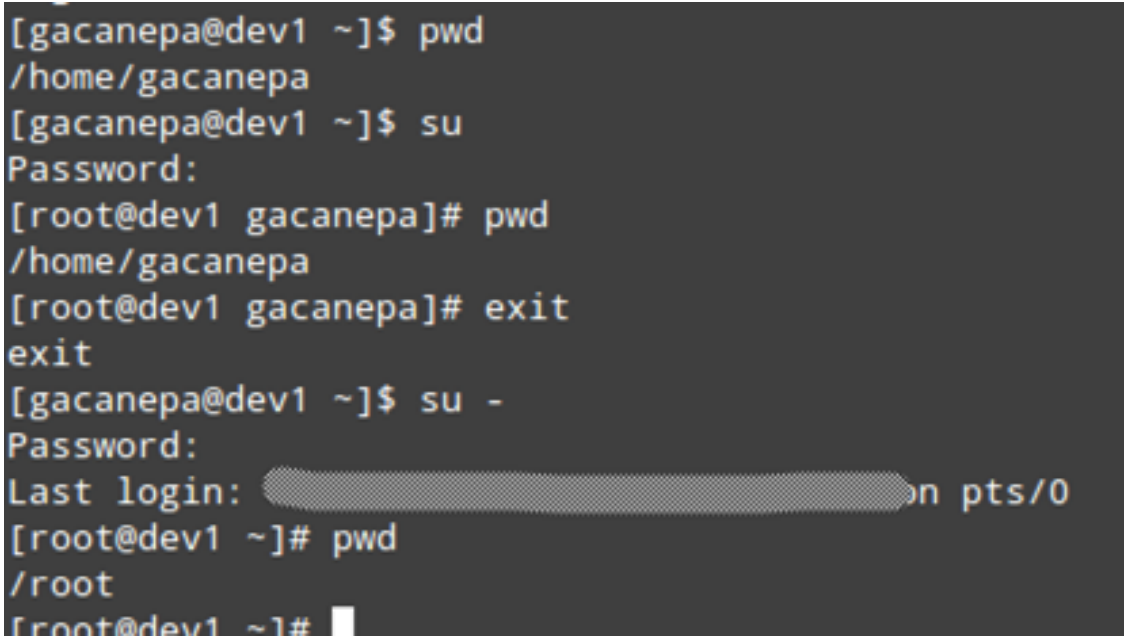
and then entering root's password.

If authentication succeeds, you will be logged on as root with the current working directory as the same as you were before. If you want to be placed in root's home directory instead, run.

```
$ su -
```

and then entering root's password.

Accessing the root Account and Using sudo



Accessing the root Account and Using sudo

The above procedure requires that a normal user knows root's password, which poses a serious security risk. For that reason, the sysadmin can configure the sudo command to allow an ordinary user to execute commands as a different user (usually the superuser) in a very controlled and limited way. Thus, restrictions can be set on a user so as to enable him to run one or more specific privileged commands and no others.

To authenticate using sudo, the user uses his/her own password. After entering the command, we will be prompted for our password (not the superuser) and if the authentication succeeds (and if the user has been granted privileges to run the command), the specified command is carried out.

To grant access to sudo, the system administrator must edit the /etc/sudoers file. It is recommended that this file is edited using the visudo command instead of opening it directly with a text editor.

```
# visudo
```

Accessing the root Account and Using sudo

This opens the /etc/sudoers file using **vim**

These are the most relevant lines.

```
Defaults    secure_path="/usr/sbin:/usr/bin:/sbin"
root        ALL=(ALL) ALL
tecmint     ALL=/bin/yum update
gacanepa    ALL=NOPASSWD:/bin/updatedb
%admin      ALL=(ALL) ALL
```

Let’s take a closer look at them.

```
Defaults    secure_path="/usr/sbin:/usr/bin:/sbin:/usr/local/bin"
```

This line lets you specify the directories that will be used for sudo, and is used to prevent using user-specific directories, which can harm the system.

Accessing the root Account and Using sudo

The next lines are used to specify permissions.

```
root        ALL=(ALL) ALL
```

- The first ALL keyword indicates that this rule applies to all hosts.
- The second ALL indicates that the user in the first column can run commands with the privileges of any user.
- The third ALL means any command can be run.

```
tecmint     ALL=/bin/yum update
```

Accessing the root Account and Using sudo

If no user is specified after the = sign, sudo assumes the root user. In this case, user tecmint will be able to run yum update as root.

```
gacanepa    ALL=NOPASSWD:/bin/updatedb
```

The **NOPASSWD** directive allows user gacanepa to run */bin/updatedb* without needing to enter his

```
%admin      ALL=(ALL) ALL
```

Accessing the root Account and Using sudo

The % sign indicates that this line applies to a group called "*admin*". The meaning of the rest of the line is identical to that of an regular user. This means that members of the group "*admin*" can run all commands as any user on all hosts.

To see what privileges are granted to you by sudo, use the "**-l**" option to list them.

Before

```
[gacanepa@dev1 root]$ sudo -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for gacanepa:
Sorry, user gacanepa may not run sudo on dev1.
[gacanepa@dev1 root]$
```

Before and after adding user gacanepa to the sudoers file

After

```
User gacanepa may run the following commands on this host:
    (root) NOPASSWD: /bin/updatedb
```

Q&A