### - windows is security problem easily attach virus, we can’t work multiple persons one windows machine.

### - we overcome above problems using Unix

### 1. Uptime Command

In Linux **uptime** command shows since how long your system is running and the number of users are currently logged in and also displays load average for **1,5** and **15** minutes intervals.

**# uptime**

08:16:26 up 22 min, 1 user, load average: 0.00, 0.03, 0.22

#### Check Uptime Version

**Uptime** command don’t have other options other than **uptime** and **version**. It gives information only in **hours:mins** if it less than **1** day.

[tecmint@tecmint ~]$ uptime -V

procps version 3.2.8

### 2. W Command

It will displays users currently logged in and their process along-with shows **load averages**. also shows the **login name**, **tty name**, **remote host**, **login time**, **idle time**, **JCPU**, **PCPU**, command and processes.

**# w**

[root@localhost ~]# w

04:38:51 up 11 min, 3 users, load average: 0.00, 0.01, 0.02

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

vagrant pts/0 10.0.2.2 04:28 3.00s 0.03s 0.24s sshd: vagrant [priv]

srini pts/1 192.168.33.1 04:36 1:55 0.00s 0.00s -bash

venkat pts/2 192.168.33.1 04:36 1:55 0.01s 0.01s -bash

#### Available options

1. **-h** : displays no header entries.
2. **-s** : without JCPU and PCPU.
3. **-f** : Removes from field.
4. **-V** : (upper letter) – Shows versions.

### 3. Users Command

Users command displays currently logged in users. This command don’t have other parameters other than help and version.

**# users**

tecmint

### 4. Who Command

**who** command simply return **user name**, **date**, **time** and **host information**. who command is similar to **w** command. Unlike **w** command **who** doesn’t print what users are doing. Lets illustrate and see the different between **who** and **w** commands.

**# who**

tecmint pts/0 2012-09-18 07:59 (192.168.50.1)

**# w**

08:43:58 up 50 min, 1 user, load average: 0.64, 0.18, 0.06

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

tecmint pts/0 192.168.50.1 07:59 0.00s 0.43s 0.10s w

#### Who command Options

1. **-b** : Displays last system reboot date and time.
2. **-r** : Shows current runlet.
3. **-a, –all** : Displays all information in cumulatively.
4. **USER** – User name.
5. **TTY** – Terminal type such as pts/0 or console.
6. **FROM** – The remote host name or IP address.
7. **LOGIN@** – Login time.
8. **IDLE** – Idel time.
9. **JCPU** – The JCPU time is the time used by all processes attached to the tty.
10. **PCPU** – The PCPU time is the time used by the current process displayed in WHAT field.
11. **WHAT** – The command line of USER’s current process.

### 5. Whoami Command

**whoami** command print the name of current user. You can also use “**who am i**” command to display the current user. If you are logged in as a root using sudo command “**whoami**” command return **root** as current user. Use “**who am i**” command if you want to know the exact user logged in.

**# whoami**

tecmint

### 6. ls Command

ls command display list of files in human readable format.

**# ls -l**

total 114

dr-xr-xr-x. 2 root root 4096 Sep 18 08:46 bin

dr-xr-xr-x. 5 root root 1024 Sep 8 15:49 boot

Sort file as per last modified time.

**# ls -ltr**

total 40

-rw-r--r--. 1 root root 6546 Sep 17 18:42 install.log.syslog

-rw-r--r--. 1 root root 22435 Sep 17 18:45 install.log

-rw-------. 1 root root 1003 Sep 17 18:45 anaconda-ks.cfg

For more examples of ls command, please check out our article on [15 Basic ‘ls’ Command Examples in Linux](https://www.tecmint.com/15-basic-ls-command-examples-in-linux/).

### 8. Less Command

**less** command allows quickly view file. You can page up and down. Press ‘**q**‘ to quit from less window.

**# less install.log**

Installing setup-2.8.14-10.el6.noarch

warning: setup-2.8.14-10.el6.noarch: Header V3 RSA/SHA256 Signature, key ID c105b9de: NOKEY

Installing filesystem-2.4.30-2.1.el6.i686

Installing ca-certificates-2010.63-3.el6.noarch

Installing xml-common-0.6.3-32.el6.noarch

Installing tzdata-2010l-1.el6.noarch

Installing iso-codes-3.16-2.el6.noarch

### 9. More Command

**more** command allows quickly view file and shows details in percentage. You can page up and down. Press ‘**q**‘ to quit out from more window.

**# more install.log**

Installing setup-2.8.14-10.el6.noarch

warning: setup-2.8.14-10.el6.noarch: Header V3 RSA/SHA256 Signature, key ID c105b9de: NOKEY

Installing filesystem-2.4.30-2.1.el6.i686

Installing ca-certificates-2010.63-3.el6.noarch

Installing xml-common-0.6.3-32.el6.noarch

Installing tzdata-2010l-1.el6.noarch

Installing iso-codes-3.16-2.el6.noarch

**--More--(10%)**

### 10. CP Command

Copy file from source to destination preserving same mode.

**# cp -p fileA fileB**

You will be prompted before overwrite to file.

**# cp -i fileA fileB**

### 11. MV Command

Rename **fileA** to **fileB**. **-i** options prompt before overwrite. Ask for confirmation if exist already.

**# mv -i fileA fileB**

### 12. Cat Command

**cat** command used to view multiple file at the same time.

**# cat fileA fileB**

You combine **more** and **less** command with cat command to view file contain if that doesn’t fit in single screen / page.

**# cat install.log | less**

**# cat install.log | more**

For more examples of Linux cat command read our article on [13 Basic Cat Command Examples in Linux](https://www.tecmint.com/13-basic-cat-command-examples-in-linux/).

### 13. Cd command (change directory)

with cd command (change directory) it will goes to **fileA** directory.

**# cd /fileA**

### 14. pwd command (print working directory)

**pwd** command return with present working directory.

**# pwd**

/root

### 15. Sort command

Sorting lines of text files in ascending order. with **-r** options will sort in descending order.

**#sort fileA.txt**

**#sort -r fileA.txt**

### 16. VI Command

**Vi** is a most popular text editor available most of the **UNIX-like OS**. Below examples open file in read only with **-R** option. Press ‘**:q**‘ to quit from vi window.

**# vi -R /etc/shadows**

### 17. SSH Command (Secure Shell)

SSH command is used to login into remote host. For example the below ssh command will connect to remote host (**192.168.50.2**) using user as **narad**.

**# ssh narad@192.168.50.2**

To check the version of ssh use option **-V** (uppercase) shows version of ssh.

**# ssh -V**

OpenSSH\_5.3p1, OpenSSL 1.0.0-fips 29 Mar 2010

### 18. Ftp or sftp Command

**ftp** or **sftp** command is used to connect to remote ftp host. ftp is (**file transfer protocol**) and sftp is (**secure file transfer protocol**). For example, the below commands will connect to ftp host (**192.168.50.2**).

**# ftp 192.168.50.2**

**# sftp 192.168.50.2**

Putting multiple files in remote host with **mput** similarly we can do **mget** to download multiple files from remote host.

**# ftp > mput \*.txt**

**# ftp > mget \*.txt**

### 19. Service Command

**Service** command call script located at **/etc/init.d/** directory and execute the script. There are two ways to start the any service. For example, we start the service called **httpd** with service command.

**# service httpd start**

OR

**# /etc/init.d/httpd start**

### 20. Free command

**Free** command shows **free**, **total** and **swap memory** information in bytes. (**Buffers:** a temporary holding place for data that is being sent to or received from an external device, such as a hard disk drive (HDD), keyboard or printer) ()

**# free**

[vagrant@localhost ~]$ free

total used free shared buff/cache available

Mem: 1016344 94496 767356 7068 154492 757600

Swap: 1048572 0 1048572

Free with **-t** options shows **total memory** used and available to use in bytes.

**# free -t**

[vagrant@localhost ~]$ free -t

total used free shared buff/cache available

Mem: 1016344 94972 766660 7100 154712 757012

Swap: 1048572 0 1048572

Total: 2064916 94972 1815232

[vagrant@localhost ~]$ free -t -m

total used free shared buff/cache available

Mem: 992 90 751 6 150 741

Swap: 1023 0 1023

Total: 2016 90 1775

**# free -m (m is used for show memory in megabytes)**

**# free -g (m is used for show memory in gigabytes)**

### 21. Top Command

**top** command displays processor activity of your system and also displays tasks managed by kernel in real-time. It’ll show **processor** and **memory** are being used. Use top command with **‘u**‘ option this will display specific User process details as shown below. Press ‘**O**‘ (**uppercase letter**) to sort as per desired by you. Press ‘**q**‘ to quit from top screen.

**# top -u tecmint**

[root@localhost ~]# top

top - 05:05:53 up 9 min, 2 users, load average: 0.00, 0.05, 0.05

Tasks: **116** total, **1** running, **115** sleeping, **0** stopped, **0** zombie

%Cpu(s): **0.0** us, **0.0** sy, **0.0** ni,**100.0** id, **0.0** wa, **0.0** hi, **0.0** si, **0.0** st

KiB Mem : **1016344** total, **763764** free, **97024** used, **155556** buff/cache

KiB Swap: **1048572** total, **1048572** free, **0** used. **754476** avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

**3817 root 20 0 157712 2228 1524 R 0.3 0.2 0:00.05 top**

1 root 20 0 128104 6724 3964 S 0.0 0.7 0:01.49 systemd

2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd

3 root 20 0 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/0

6 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kworker/u4:0

7 root rt 0 0 0 0 S 0.0 0.0 0:00.05 migration/0

### Details: https://www.lifewire.com/linux-top-command-2201163

### 22. Tar Command

**tar** command is used to compress files and folders in Linux. For example, the below command will create a archive for **/home** directory with file name as **archive-name.tar**.

**# tar -cvf archive-name.tar /home**

To extract tar archive file, use the option as follows.

**# tar -xvf archive-name.tar**

### 23. Grep Command

grep search for a given string in a file. Only **tecmint** user displays from **/etc/passwd** file. we can use **-i** option for ignoring case sensitive.

**# grep tecmint /etc/passwd**

tecmint:x:500:500::/home/tecmint:/bin/bash

### 24. Find Command

Find command used to search **files**, **strings** and **directories**. The below example of find command search **tecmint** word in ‘**/**‘ partition and return the output.

**# find / -name tecmint**

/var/spool/mail/tecmint

/home/tecmint

/root/home/tecmint

Ex2: find / -name \*.txt | xargs grep -i srini

For complete guide on **Linux find command** examples fount at [35 Practical Examples of Linux Find Command](https://www.tecmint.com/35-practical-examples-of-linux-find-command/).

We can search for all files that end in ".pub" like this:

find / -type f -name "\*.pub"

check the history of commands. Here you are searching for ‘upload’ word.

$ history | grep upload

### 25. lsof Command

**lsof** mean List of all open files. Below lsof command list of all opened files by user **tecmint**.

**# lsof -u tecmint**

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

sshd 1889 tecmint cwd DIR 253,0 40962 /

sshd 1889 tecmint txt REG 253,0 532336 298069 /usr/sbin/sshd

sshd 1889 tecmint DEL REG 253,0 412940 /lib/libcom\_err.so.2.1

For more **lsof command examples** visit [10 lsof Command Examples in Linux](https://www.tecmint.com/10-lsof-command-examples-in-linux/).

### 26. last command

With last command we can watch user’s activity in the system. This command can execute normal user also. It will display complete user’s info like **terminal**, **time**, **date**, **system reboot** or **boot** and **kernel version**. Useful command to troubleshoot.

**# last**

[root@localhost ~]# last

srini pts/1 192.168.33.1 Fri Nov 10 05:04 still logged in

vagrant pts/0 10.0.2.2 Fri Nov 10 04:59 still logged in

reboot system boot 3.10.0-514.16.1. Fri Nov 10 04:56 - 05:24 (00:28)

vagrant pts/0 10.0.2.2 Wed Nov 8 06:04 - crash (1+22:52)

venkat pts/2 192.168.33.10 Wed Nov 8 05:47 - 05:47 (00:00)

venkat pts/1 192.168.33.1 Wed Nov 8 05:16 - crash (1+23:40)

vagrant pts/0 10.0.2.2 Wed Nov 8 05:12 - 06:03 (00:50)

reboot system boot 3.10.0-514.16.1. Wed Nov 8 05:11 - 05:24 (2+00:13)

You can use **last** with **username** to know for specific user’s activity as shown below.

**# last tecmint**

[root@localhost ~]# last srini

srini pts/1 192.168.33.1 Fri Nov 10 05:04 still logged in

[root@localhost ~]# last venkat

venkat pts/2 192.168.33.10 Wed Nov 8 05:47 - 05:47 (00:00)

venkat pts/1 192.168.33.1 Wed Nov 8 05:16 - crash (1+23:40)

### 27. ps command

**ps** command displays about processes running in the system. Below example show **init** process only.

**# ps -ef | grep init**

root 1 0 0 07:53 ? 00:00:04 /sbin/init

root 7508 6825 0 11:48 pts/1 00:00:00 grep init

### 28. kill command

Use **kill** command to terminate process. First find process **id** with **ps** command as shown below and kill process with **kill -9** command.

**# ps -ef | grep init**

root 1 0 0 07:53 ? 00:00:04 /sbin/init

root 7508 6825 0 11:48 pts/1 00:00:00 grep init

**# kill- 9 7508**

### 29. rm command

**rm** command used to remove or delete a file without prompting for confirmation.

**# rm filename**

Using **-i** option to get confirmation before removing it. Using options ‘**-r**‘ and ‘**-f**‘ will remove the file forcefully without confirmation.

**# rm -i test.txt**

rm: remove regular file `test.txt'?

### 30. mkdir command example.

**mkdir** command is used to create directories under Linux.

**# mkdir directoryname**

This is a handy day to day useable basic commands in Linux / Unix-like operating system. Kindly share through our comment box if we missed out.

## locate

Find files on your system by name.

## mount

Mounts a filesystem. There are different options in this command that you can use which enables you to mount even remote file systems

**Lock the password (All commands will work from root user)**  
# passwd -l user1  
Locking password for user user1.  
passwd: Success

**Unlock the password**  
# passwd -u user1  
Unlocking password for user user1.  
passwd: Success

**Status of Lock an account**

# passwd -S user1  
user1 LK 2014-08-17 0 99999 7 -1 (Password locked.)

**Lock an account**  
# usermod -L user1

**Unlock a user account**  
# usermod -U user1

**DF (Disk Free)**

# df -ah

[root@localhost ~]# df

**Filesystem 1K-blocks Used Available Use% Mounted on**

/dev/mapper/cl-root 52403200 1693892 50709308 4% /

devtmpfs 497404 0 497404 0% /dev

tmpfs 508172 0 508172 0% /dev/shm

tmpfs 508172 6712 501460 2% /run

tmpfs 508172 0 508172 0% /sys/fs/cgroup

/dev/sda1 1038336 169288 869048 17% /boot

/dev/mapper/cl-home 29337604 33032 29304572 1% /home

tmpfs 101636 0 101636 0% /run/user/1000

tmpfs 101636 0 101636 0% /run/user/1001

[root@localhost ~]# df -ah

**Filesystem Size Used Avail Use% Mounted on**

rootfs - - - - /

sysfs 0 0 0 - /sys

proc 0 0 0 - /proc

devtmpfs 486M 0 486M 0% /dev

securityfs 0 0 0 - /sys/kernel/security

tmpfs 497M 0 497M 0% /dev/shm

devpts 0 0 0 - /dev/pts

tmpfs 497M 6.6M 490M 2% /run

tmpfs 497M 0 497M 0% /sys/fs/cgroup

cgroup 0 0 0 - /sys/fs/cgroup/systemd

pstore 0 0 0 - /sys/fs/pstore

cgroup 0 0 0 - /sys/fs/cgroup/devices

cgroup 0 0 0 - /sys/fs/cgroup/cpu,cpuacct

srini@SRINIEIS:~$ df

**Filesystem 1K-blocks Used Available Use% Mounted on**

rootfs 422876156 74472176 348403980 18% /

root 422876156 74472176 348403980 18% /root

home 422876156 74472176 348403980 18% /home

data 422876156 74472176 348403980 18% /data

cache 422876156 74472176 348403980 18% /cache

mnt 422876156 74472176 348403980 18% /mnt

none 422876156 74472176 348403980 18% /dev

none 422876156 74472176 348403980 18% /run

none 422876156 74472176 348403980 18% /run/lock

none 422876156 74472176 348403980 18% /run/shm

none 422876156 74472176 348403980 18% /run/user

C: 422876156 74472176 348403980 18% /mnt/c

F: 552958972 178266300 374692672 33% /mnt/f

**DU (Disk Use)**

# du -sh etc/

[root@localhost /]# du -sh /usr/

1.1G /usr/

**netstart (Check Open Ports)**

# netstart -tulpn

**ps aux | grep ngix (Check CPU usage for process)**

[root@localhost /]# ps aux | grep sshd

**USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND**

root 1010 0.0 0.3 105480 4020 ? Ss 04:56 0:00 /usr/sbin/sshd -D

root 3720 0.0 0.5 145696 5084 ? Ss 04:59 0:00 sshd: vagrant [priv]

vagrant 3723 0.0 0.2 145696 2640 ? S 04:59 0:00 sshd: vagrant@pts/0

root 3792 0.0 0.4 145696 5048 ? Ss 05:04 0:00 sshd: srini [priv]

srini 3796 0.0 0.2 145696 2816 ? S 05:04 0:00 sshd: srini@pts/1

root 3919 0.0 0.0 112652 956 pts/0 R+ 05:42 0:00 grep --color=auto ######################

* 1. # touch
  2. Debug shall script (start debug with “set -x”, end debug with “set +x”)
  3. # nl filename (show lines in a file with line numbers)
  4. # cut -f 2,3 filename
  5. # cut -c 2-5 filename
  6. # tr ‘source character’ ‘replace character’ < filename (The **tr** command automatically translates (substitutes, or maps) one set of characters to another.). it will not convert actual content
  7. # tr ‘a-z’ ‘A-Z’ < filename (convert all lower case to upper case letters)
  8. tree (multi re directional command)
  9. # grep Java course.txt (find where “Java” is there in course.txt file, case sensitive. It will show in red color )
  10. # grep -i Java course.txt (find where “Java” is there in course.txt file, ignore the case sensitive)
  11. # grep -c Java course.txt (count where “Java” is there in course.txt file, case sensitive )
  12. # grep -ni Java course.txt (find where “Java” is there in course.txt file with line number, ignore case sensitive )
  13. # grep -l “Java” \* (display all filename if it contains “Java”)
  14. Ancores (start ^, end character $)
  15. # grep -in “c[aeiou]ll” course.txt (display string where start with 1st letter ‘c’ , end letter ‘ll’, in between ant mach ‘aeiou’ letters in course.txt, ignore case sensitive)
  16. # grep -i “^[aeiou]” course.txt (display all Lines if it contains “Java”)
  17. # grep -i “a$” course.txt (display all lines in ‘course.txt’ which ends with ‘a’)
  18. # grep “^$” course.txt (get all empty lines)
  19. # grep -c “^$” course.txt (get all empty lines count)
  20. # grep -n “^$” course.txt (get which lines are empty)
  21. # grep -v “^$” course.txt (get all data lines)
  22. # tty (it is used for check the terminal)
  23. # cd /var/log/ (we can check the all log files in this path)

**Get operating system information**

# uname -r

**Crons jobs in Linux**

* It contains \* \* \* \* \* (5 starts)
* “Minutes” “hours” “day of month” “month of year” “Day of week”
* Ex: 30 April 2.30 = 30 14 30 4 0
* Create crons job on Linux:
* 1) check cronsjob
* # crontab -l
* 2) go to cd /var/spool/cron/
* 3) cat /etc/crontab

SHELL=/bin/bash

PATH=/sbin:/bin:/usr/sbin:/usr/bin

MAILTO=root

# For details see man 4 crontabs

# 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,fri,sat

# | | | | |

# \* \* \* \* \* user-name command to be executed

* go to cd /var/spool/cron/
* # cronetab -e (it is used for create crone jab, it will open VI editor)

47 19 29 4 \* echo “my crone job success” > /dev/pts/0

* cron.allow – by default we can’t get this, we need to create manually in /etc/crone.allow
* when we create empty crone.allow file, no user log in OR allow in Linux, if we want give permission to any user than will give username in crone.allow file.
* # crontab -r (remove cronetab)

################################################################### Part I

* 1. **alias** **c=clea**r (use ‘c’ command for clear the screen instead of clear command)
  2. **LILO** is Linux loader, it is boot loader
  3. Using **free, vmstat** command for see the physical memory & virtual memory.
  4. **cat /proc/meminfo** command used for show Complete information memory (OR **free -g**)
  5. Using **sar** command**,** we can see CPU utilization & other status
  6. If sar command not found than install:

yum install sysstat

systemctl start sysstat

systemctl enable sysstat

[root@localhost ~]# sar 2 5

Linux 3.10.0-123.el7.x86\_64 (localhost.localdomain)     Monday 26 October 2015     \_x86\_64\_    (2 CPU)

01:43:55  EDT     CPU     %user     %nice   %system   %iowait    %steal     %idle

01:43:57  EDT     all      0.00      0.00      0.25      0.00      0.00     99.75

01:43:59  EDT     all      0.00      0.00      0.00      0.00      0.00    100.00

01:44:01  EDT     all      0.00      0.00      0.25      0.00      0.00     99.75

01:44:03  EDT     all      0.00      0.00      0.25      0.25      0.00     99.50

01:44:05  EDT     all      0.00      0.00      0.00      0.00      0.00    100.00

Average:        all      0.00      0.00      0.15      0.05      0.00     99.80

**https://www.linuxtechi.com/generate-cpu-memory-io-report-sar-command/**

* 1. **vmstat -a, vmstat -t (time stamp) , vmstat -t 1 5 (every 1 sec , after stop 5 intervals)**
  2. **sar** (system activity report), **sar -u** (show CPU usage), **sar -r (memory free & used) , sar -u 1 6 (every 1 sec , after stop 6 intervals)**
  3. If use **fdisk** command for partitioning, that size is fixed later we can’t modify or resize the partition
  4. how to reduce or shrink the size of LVM (logical volume manager) partition. This is advanced partitioning tool
  5. What is swap space:
  6. **chmod** : change the permission of files and directories.
  7. **chown:** to change the owner of files and directories.
  8. **chgrp:** to change the group ownership of the files and directories.
  9. **Symbolic links (Syntax: ln -s path shortcut\_path)(Ex: ln -s /etc/sysconfig/network/hots /myhost)** are similar to shortcuts in windows. Also called soft link.
  10. Maximum length of file name: 255 characters. Path length: 4096 characters.
  11. **env** command is used for get all the environment variables
  12. **ps aux:** show all process status.
  13. **du -sh directory\_name** command will show size of the directory.
  14. **su –** (substitute user)
  15. **nohub** command used for process will run in the background.
  16. **inode** : contents of the file will be stored in data blocks. Whereas information (meta data) about that file will be stored in **inode**
  17. **nice** command is used for priority of the jobs can be changed. (Syn : nice ). priority range from -20(highest) to 19(lowest).
  18. **renice** command is used for program is already used.
  19. **Shadow passwords**: every user passwords are stored in **/etc/passwd** file. This file will access will all the users. **/etc/shadow** file will access only super user
  20. #ifconfig | grep inet -> check for ip address (OR) # ip a | grep inet

#######################

**Networking Commands**

**######################**

1. $ping google.com
2. $ftp hostname or ip-address
3. $ftp amrood.com
4. $ traceroute fermion (The traceroute command will attempt to provide a list of all the routers your connections cross when reaching out to a remote system)
5. $ netstat -a | grep LISTEN | grep -v LISTENING (The netstat command can tell you about ongoing connections on the local system and ports (i.e., services) that are listening, indicating that services are waiting for requests to come through. By itself, netstat gives you a \*lot\* information. With a -a option, it gives you even more.)
6. $ whois valero.com (You can often find a lot more information about a domain using the whois comman)
7. The **tcpdump** command can print out the headers of network packets as they reach your server or can be used with various filters to select just the packets you want to see. You can also save packets for later analysis. I'll get into some interesting uses of tcpdump in my next post.
8. checking status of destination host – **telnet**
9. $ **nslookup blogger.com (show the IP address of the host)**

[root@host1 ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

0.0.0.0 192.168.0.254 0.0.0.0 UG 100 0 0 eno1

192.168.0.0 0.0.0.0 255.255.255.0 U 100 0 0 eno1

- This is the first file the network script will read; it sets several environment variables. The first two variables set the computer to run networking programs (even though it is not on a network) but not to forward packets (since it has nowhere to send them). The last two variables are generic entries.

*/etc/sysconfig/network*

NETWORKING=yes  
FORWARD\_IPV4=false  
HOSTNAME=localhost.localdomain  
GATEWAY=

This is the first file the network script will read; again the first variables simply determine that the computer will do networking but that it will not forward packets. The last four variables identify the computer and its link to the rest of the Internet (everything that is not on the LAN).

*/etc/sysconfig/network*

NETWORKING=yes  
FORWARD\_IPV4=false  
HOSTNAME=stealth.cs.u.edu  
DOMAINNAME=cs.u.edu  
GATEWAY=172.16.1.1  
GATEWAYDEV=eth0

After setting these variables, the network script will configure the network devices. This file sets up environment variables for the Ethernet card. It names the device and gives it its IP address, network mask, and broadcast address as well as any other device specific variables. This kind of computer would also have a loopback configuration file exactly like the one for a non-networked computer.

*/etc/sysconfig/network-scripts/ifcfg-eth0*

DEVICE=eth0  
IPADDR=172.16.1.4  
NMASK=255.255.255.0  
NETWORK=172.16.1.0  
BCAST=172.16.1.255  
ONBOOT=yes  
BOOTPROTO=none

After setting these variables, the network script will run the *ifconfig program* to start the device. Finally, the script will run the *route* program to add the default route (GATEWAY) and any other specified routes (found in the */etc/sysconfig/static-routes file*, if any). In this case only the default route is specified, since all traffic either stays on the LAN (where the computer will use ARP to find other hosts) or goes through the router to get to the outside world.

Linux TCP/IP Network Configuration Files:

| **File** | **Description** |
| --- | --- |
| /etc/resolv.conf | List DNS servers for internet domain name resolution. Manual page for: /etc/[resolv.conf](http://man.yolinux.com/cgi-bin/man2html?cgi_command=resolv.conf) |
| /etc/hosts | Lists hosts to be resolved locally (not by DNS). Manual page for: /etc/[hosts](http://man.yolinux.com/cgi-bin/man2html?cgi_command=hosts) |
| /etc/nsswitch.conf | List order of host name search. Typically look at local files, then NIS server, then DNS server. Manual page for: /etc/[nsswitch.conf](http://man.yolinux.com/cgi-bin/man2html?cgi_command=resolv.conf) |
| Red Hat/Fedora/CentOS: /etc/sysconfig/network | Specify network configuration. eg. Static IP, DHCP, NIS, etc. |
| Red Hat/Fedora/CentOS: /etc/sysconfig/network-scripts/ifcfg-***device*** | Specify TCP network information. |
| Ubuntu/Debian: /etc/network/interfaces | Specify network configuration and devices. eg. Static IP and info, DHCP, etc. |

------------------------------------------------------########################