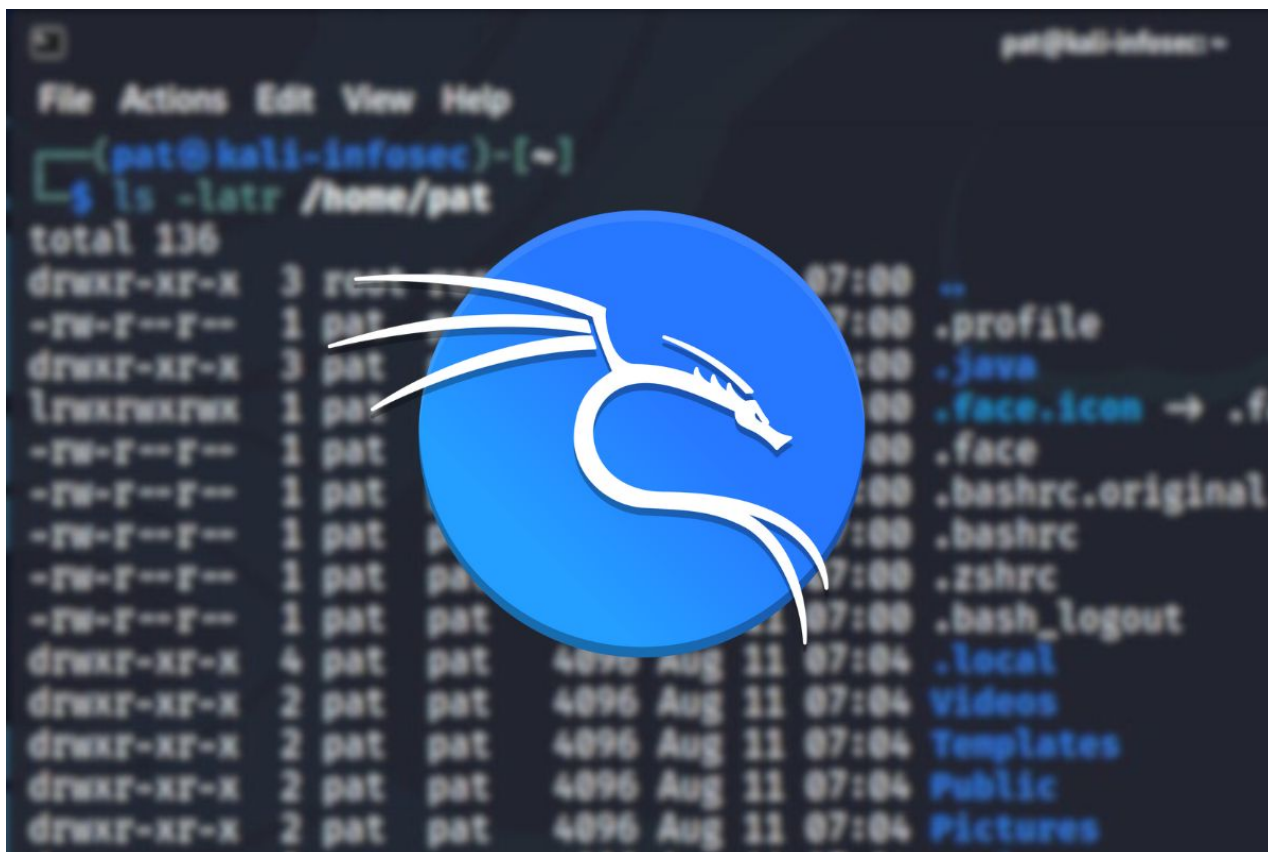


# 50 Basic Linux Commands you Need to Know on Kali Linux

 infosecscout.com/basic-kali-linux-commands

Patrick Fromaget



Even if Kali Linux includes powerful tools, that can often be used via the GUI (graphical user interface), knowing some basic Linux commands is highly recommended to do a bit more with your system and targets. That's exactly the goal of this article, where I'll give you 50 of the most useful Linux commands you need to know.

I worked about 20 years as a Linux system administrator, and these were the commands I used every day. So, if there are only 50 Linux commands you can remember, start with these. It will save you time regularly.

I sorted them by category, so you don't necessarily have to read from start to end, you can use the table of content below to quickly get to what you are looking for.

Master Linux Commands

Your essential Linux handbook

Want to level up your Linux skills? Here is the perfect solution to become efficient on Linux. **20% off today!**

[Download now](#)

# Files management

## Hide your IP address and location with a free VPN:

[Try it for free now](#), with advanced security features.

2900+ servers in 65 countries. It's free. Forever.

These are the basic commands are that every Linux beginner should learn.

You might already know these commands, so this might be a reminder.

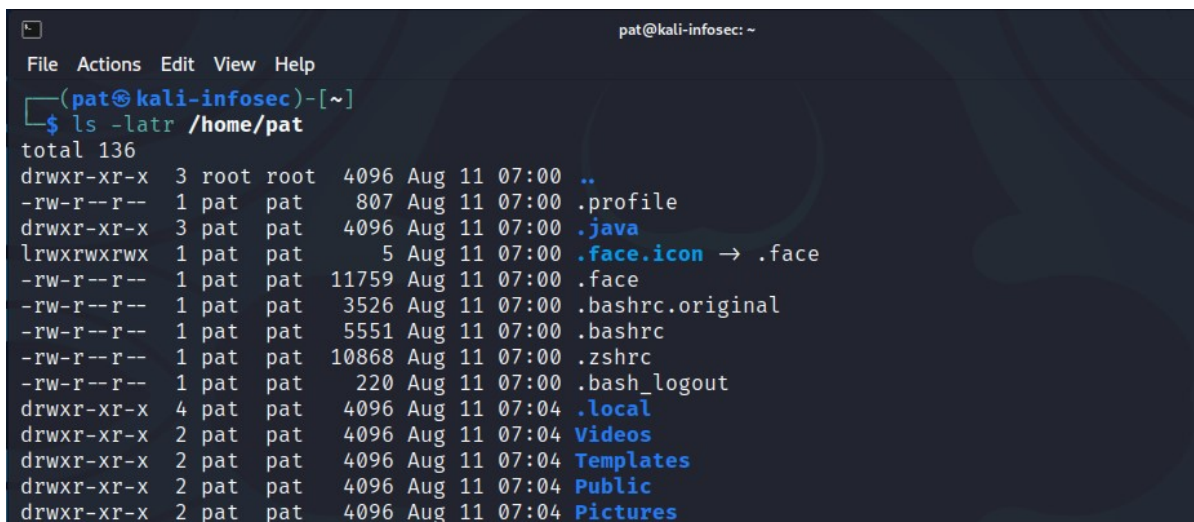
The Linux files organization is a tree, starting at the root: /.

Each subfolder created under it is accessible starting with /.

For example: /home/pat ⇒ pat is a subfolder of /home, home is a subfolder in /.

Don't forget to use sudo if you are not allowed to access the file or directory (sudo gives you administrator privileges for the command that follows).

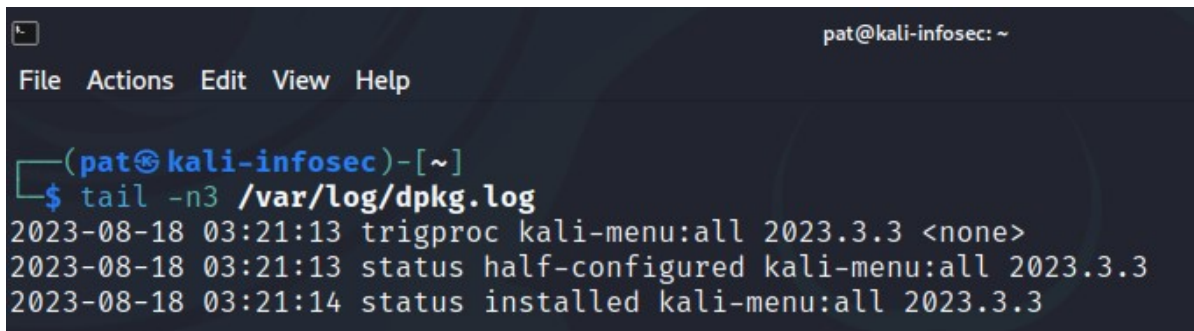
- **cd <path>**: Changes directory, go to the specified folder in the files tree. **cd** /home/pat.
- **ls**: Lists the files and directories in the current or specified folder.  
**ls ls /home/pat**  
**ls -latr /home/pat**



```
pat@kali-infosec: ~
File Actions Edit View Help
(pat@kali-infosec)-[~]
$ ls -latr /home/pat
total 136
drwxr-xr-x  3 root root  4096 Aug 11 07:00 ..
-rw-r--r--  1 pat  pat   807 Aug 11 07:00 .profile
drwxr-xr-x  3 pat  pat  4096 Aug 11 07:00 .java
lrwxrwxrwx  1 pat  pat    5 Aug 11 07:00 .face.icon → .face
-rw-r--r--  1 pat  pat 11759 Aug 11 07:00 .face
-rw-r--r--  1 pat  pat  3526 Aug 11 07:00 .bashrc.original
-rw-r--r--  1 pat  pat  5551 Aug 11 07:00 .bashrc
-rw-r--r--  1 pat  pat 10868 Aug 11 07:00 .zshrc
-rw-r--r--  1 pat  pat   220 Aug 11 07:00 .bash_logout
drwxr-xr-x  4 pat  pat  4096 Aug 11 07:04 .local
drwxr-xr-x  2 pat  pat  4096 Aug 11 07:04 Videos
drwxr-xr-x  2 pat  pat  4096 Aug 11 07:04 Templates
drwxr-xr-x  2 pat  pat  4096 Aug 11 07:04 Public
drwxr-xr-x  2 pat  pat  4096 Aug 11 07:04 Pictures
```

- **mkdir <folder>**: Creates a new subfolder in the current or specified folder.  
**mkdir myfolder**  
**mkdir /home/pat/myfolder**
- **cp <file> <destination>**: Copies a file or a directory to another location (to copy a complete directory, you need to add the -r parameter for “recursive”).  
**cp test.txt /home/pat/Documents/**  
**cp /home/pat/test.txt /home/pat/Documents/**  
**cp -r /home/pat/test/ /home/pat/Documents/**
- **mv <source> <destination>**: Moves a file or a directory to another location.  
**mv /home/pat/test.txt /home/Documents/**  
**mv /home/pat/test/ /home/Documents/**

- **cat <file>**: Displays all the content of the specified file.  
`cat /home/pat/README.txt`
- **more <file>**: Displays the content of the specified file, page per page (enter or space to continue, q to quit).  
`more /var/log/syslog`
- **tail <file>**: Tail allows you to display the end of a file, it's useful to check new entries in a log file.  
`tail /var/log/syslog`  
 You can specify the number of lines to display with -n.  
`tail -n20 /var/log/syslog`  
 And finally, my favorite is the option -f to display new lines in real-time.  
`tail -f /var/log/syslog`



The screenshot shows a terminal window with a dark background. The title bar at the top reads 'pat@kali-infosec: ~'. Below the title bar is a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal prompt is '(pat@kali-infosec)-[~]'. The user has entered the command '\$ tail -n3 /var/log/dpkg.log'. The output shows three lines of log data: '2023-08-18 03:21:13 trigproc kali-menu:all 2023.3.3 <none>', '2023-08-18 03:21:13 status half-configured kali-menu:all 2023.3.3', and '2023-08-18 03:21:14 status installed kali-menu:all 2023.3.3'.

```
(pat@kali-infosec)-[~]
$ tail -n3 /var/log/dpkg.log
2023-08-18 03:21:13 trigproc kali-menu:all 2023.3.3 <none>
2023-08-18 03:21:13 status half-configured kali-menu:all 2023.3.3
2023-08-18 03:21:14 status installed kali-menu:all 2023.3.3
```

- **head <file>**: It's the same as tail but displays the beginning of a file.  
`head /home/pat/file.txt`  
`head -n20 /home/pat/file.txt`
- **grep <string>**: Grep is a powerful tool to search string in a text. You can use it to search for something in a file or to filter the output of another command or script.  
 Basic usage:  
`grep dhcp /var/log/syslog`

As I said, you can use it with a file or a script output:

```
cat /var/log/syslog | grep dhcp
/home/pat/myscript.sh | grep error
ls -latr | grep php
```

And finally, there are many options to use with grep, like regular expressions or options to display lines before (-B), after (-A) or around (-C) the search string. You can also use -v to display everything except the input string.

```
grep 'dhcp|dns' /var/log/syslog
grep -A2 -B4 'Fatal error' /var/log/apache/error.log
grep -v 'Notice' /var/log/apache/error.log
```

If you like this tool, I recommend reading the man page to know exactly what you can do with it.

```
man grep
```

- **nano <file>**: Nano is text editor.

It allows you to edit a file, and save your changes with (CTRL + O, Enter, CTRL + X).

```
nano /home/pat/myscript.sh
```

You'll find all actions available at the bottom of the screen.

- **rm <file>**: Deletes a file. For a folder, add option -rf (recursive and force)

```
rm monscript.sh
```

```
rm -rf /home/pat/scripts/
```

Be careful if you are using sudo with this command. Using it on system folders will delete everything without any warning message, and can break your system in a few seconds.

- **tar -c**: You can use tar to store files into an archive. It's often used with gzip to compress files.

```
tar -cvfz archive.tar.gz /home/pat/Documents/mydirectory
```

-c: create an archive

-v: verbose

-f: filename of the archive follow

-z: compress files with gzip

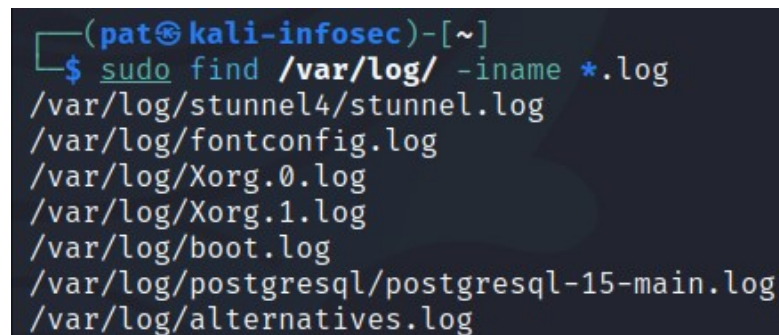
- **tar -x**: It's the same command, but to extract files.

```
tar -xvfz archive.tar.gz
```

-x: extract an archive

- **find**: As the name suggests, find is useful to locate files on your computer.

```
find /home/pat -iname *.tar.gz
```



```
(pat@kali-infosec)-[~]
$ sudo find /var/log/ -iname *.log
/var/log/stunnel4/stunnel.log
/var/log/fontconfig.log
/var/log/Xorg.0.log
/var/log/Xorg.1.log
/var/log/boot.log
/var/log/postgresql/postgresql-15-main.log
/var/log/alternatives.log
```

There are many options to help you find the good file (size, last modification date, ...).

- **pwd**: Pwd lets you see in which directory you are.

```
pwd
```

- **tree**: Another great tool to analyze your current location in the file tree. It will show you the entire lower tree (see the example below).

tree

```
(pat@kali-infosec)-[~]
$ cd /var/log

(pat@kali-infosec)-[/var/log]
$ tree
.
├── alternatives.log
├── apache2
│   ├── access.log
│   ├── error.log
│   └── other_vhosts_access.log
├── apt
│   ├── eipp.log.xz
│   ├── history.log
│   └── term.log
├── boot.log
├── boot.log.1
├── btmp
└── dpkg.log
```

Note: you may need to install this one depending on your distribution:

sudo apt install tree

## Network commands

---

You may need these commands just after the Kali Linux installation, if you only have access to the system via a terminal (SSH, for example). It can also be useful for pentesting, when connected to a network you don't know anything about.

## Configuration

---

Most computers come with 2 interfaces (Ethernet and Wi-Fi). The Ethernet is generally called eth0 and the Wi-Fi is wlan0. You have to use these names with some commands below. On Kali, "lo" is the local interface, don't worry about it.



- **ip**: Displays your current network configuration, mainly your IP address if connected.

`ip a`

```

pat@kali-infosec: /var/log
File Actions Edit View Help

(pat@kali-infosec)-[/var/log]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether 00:15:5d:de:19:1a brd ff:ff:ff:ff:ff:ff
   inet 192.168.227.171/20 brd 192.168.239.255 scope global dynamic noprefixroute eth0
       valid_lft 86109sec preferred_lft 86109sec
   inet6 fe80::d732:1e78:21b0:bebf/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

```

- **ping <ip>**: Sends a ping packet to another IP on the network to check if the host is alive.

`ping 192.168.1.1`

- **ifup <interface>**: Enables the specified interface.

`sudo ifup eth0`

- **ifdown <interface>**: Disables the specified interface. It can be useful to disable Wi-Fi if you are already connected by cable, for example.

`sudo ifdown wlan0`

## File transfer and remote connection

- **wget <url>**: This command allows you to download a file from the Internet.

`wget https://wordpress.org/latest.zip`

```

pat@kali-infosec: ~/Downloads
File Actions Edit View Help

(pat@kali-infosec)-[~/Downloads]
$ wget https://wordpress.org/latest.zip
--2023-08-25 10:33:02-- https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org) ... 198.143.164.253
Connecting to wordpress.org (wordpress.org)[198.143.164.253]:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 24937774 (24M) [application/zip]
Saving to: 'latest.zip'

latest.zip          13%[====>]          ] 3.17M  999KB/s  eta 22s

```

- **ssh <user>@<ip>**: SSH is a network protocol that provides you a way to connect securely to a remote computer.

`ssh root@192.168.1.201`

- **scp <file> <user>@<ip>:<path>**: scp can transfer a file to a remote computer over SSH.

`scp test.txt root@192.168.1.201:/root/`

- **rsync <file> <user>@<ip>:<path>**: rsync does almost the same thing, but with a delta comparison algorithm and some optimizations to transfer files faster.

```
rsync test.txt root@192.168.1.201:/root/
```

```
rsync -auzr /home/pat/Documents/* /home/pat/backups/Documents/
```

As you can see, you can also use rsync for local file synchronization.

## System updates

---

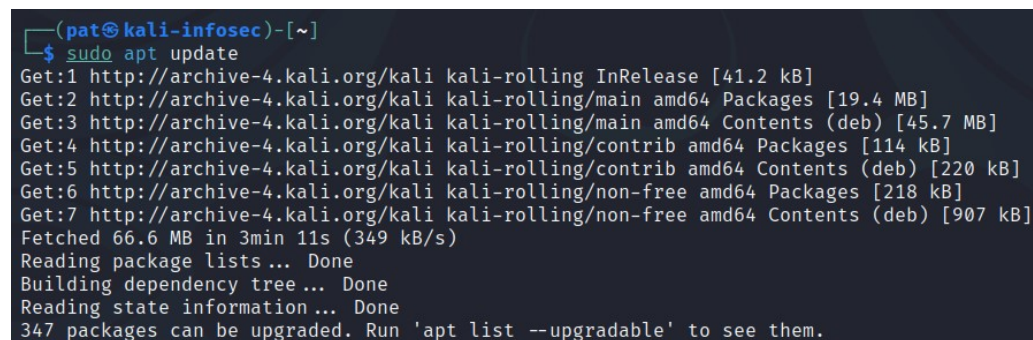
Just after the network configuration, you'll have to update your system to get the latest version of each default package installed.

On Kali Linux, as with most Linux distributions, you have at least one package installed for each app. A repository is the server where all the packages are available for download. Once the system installed, you need to sync your system with this repository, so all your packages can be updated to the latest version available.

These commands explain how to do this on Kali Linux, but the idea is the same with other distributions. **You need sudo for all these commands if you are not connected as root.**

- **apt update**: Downloads the last repository version for each one you have in your configuration (/etc/apt/sources.list).

```
sudo apt update
```



```
(pat@kali-infosec)-[~]
$ sudo apt update
Get:1 http://archive-4.kali.org/kali kali-rolling InRelease [41.2 kB]
Get:2 http://archive-4.kali.org/kali kali-rolling/main amd64 Packages [19.4 MB]
Get:3 http://archive-4.kali.org/kali kali-rolling/main amd64 Contents (deb) [45.7 MB]
Get:4 http://archive-4.kali.org/kali kali-rolling/contrib amd64 Packages [114 kB]
Get:5 http://archive-4.kali.org/kali kali-rolling/contrib amd64 Contents (deb) [220 kB]
Get:6 http://archive-4.kali.org/kali kali-rolling/non-free amd64 Packages [218 kB]
Get:7 http://archive-4.kali.org/kali kali-rolling/non-free amd64 Contents (deb) [907 kB]
Fetched 66.6 MB in 3min 11s (349 kB/s)
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
347 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

- **apt upgrade**: Updates all installed packages if needed.

```
sudo apt upgrade
```

## Packages management

---

After that, you may want to install new packages on your Linux system.

Here are the commands you need to know to do this:

- **apt install <package>**: Installs the specified package(s).

```
sudo apt install phpmyadmin
```

```
sudo apt install vim htop
```

- **apt remove <package>**: Removes a previously selected package.

```
sudo apt remove vim
```

- **apt search <search>:** Searches for a package name in the packages list (repository).

```
sudo apt search myadmin  
sudo apt search php
```

```
(pat@kali-infosec)-[~]  
$ apt search myadmin  
Sorting ... Done  
Full Text Search ... Done  
adminer/kali-rolling 4.8.1-1 all  
  Web-based database administration tool  
  
php-phpmyadmin-motranslator/kali-rolling 5.3.0-1 all  
  translation API for PHP using Gettext MO files  
  
php-phpmyadmin-shapefile/kali-rolling 3.0.1-2 all  
  ESRI ShapeFile library for PHP  
  
php-phpmyadmin-sql-parser/kali-rolling 5.7.0-2 all  
  validating SQL lexer and parser
```

- **dpkg -l:** Lists all installed packages on your system. You can use grep to find a specific package.

```
dpkg -l  
dpkg -l | grep myadmin
```

## System management

---

Here are the commands you'll often use to manage your Linux system:

- **reboot:** As the name says, this command will restart the computer immediately.  

```
sudo reboot
```
- **shutdown -h now:** This is to stop the computer immediately.  

```
sudo shutdown -h now
```

You can replace “now” by a specific time (shutdown -h 12:05).



- **service <servicename> <action>**: This command allows you to start or stop services.

```
service apache2 start
service apache2 stop
```

Sometimes there are other options, depending on the service, for example:

```
service apache2 reload
service apache2 restart
```

```
(pat@kali-infosec)-[~]
$ service cron status
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Fri 2023-08-25 10:27:16 CEST; 9min ago
     Docs: man:cron(8)
    Main PID: 598 (cron)
      Tasks: 1 (limit: 2187)
    Memory: 596.0K
       CPU: 6ms
    CGroup: /system.slice/cron.service
            └─598 /usr/sbin/cron -f

Aug 25 10:27:16 kali-infosec systemd[1]: Started cron.service - Regular background program processing dae>
Aug 25 10:27:16 kali-infosec (cron)[598]: cron.service: Referenced but unset environment variable evaluat>
Aug 25 10:27:16 kali-infosec cron[598]: (CRON) INFO (pidfile fd = 3)
```

Don't type any action to see all those available:

```
service apache2
```

For information, you can also use systemctl to do the same thing, on Debian-based distributions, you still have the choice.

- **update-rc.d <service> <action>**: On Kali Linux, this command allows you to manage the service start or stop on the system boot.

To start a service on boot:

```
sudo update-rc.d ssh enable
```

To disable start of the service:

```
sudo update-rc.d -f ssh remove
```

The -f option is here to force the symbolic link deletion. This command is only for service. To start other scripts or commands on boot, you have to edit the /etc/rc.local file.

```
sudo nano /etc/rc.local
```

- **ps:** This command displays all running process on your computer. The basic command is this one to display everything:

`ps aux`

You can also display the processes started by a specific user:

`ps -u pat`

This will give you a list like this:

```
(pat@kali-infosec)-[~]
$ ps -u pat
  PID TTY          TIME CMD
  938 ?            00:00:00 systemd
  939 ?            00:00:00 (sd-pam)
  955 ?            00:00:00 pipewire
  956 ?            00:00:00 pipewire
  957 ?            00:00:00 wireplumber
  958 ?            00:00:00 pipewire-pulse
```

The process ID (PID) can be useful for other commands, to stop it, for example (next command).

- **kill <pid>:** The kill command allows you to terminate a process. You'll need the process ID to do this (see the previous command).

`kill 12345`

Sometimes you may need to use the -9 option to force all related commands to stop. For example, if you run 20 commands in a script and kill it, it'll continue to the next line, not exit the program, except if you use the -9 option.

`kill -9 12345`

You can also use killall to stop all occurrences of a program.

`killall php`

This command will stop all PHP scripts.

Be aware that this command will immediately stop the process asked, no matter what was going on. It isn't a clean stop. You don't know what the script is doing so it can damage data or corrupt files.

This should be used as a last step, and if possible on the non-critical process.

- **htop:** This tool is an alternative to top. It's more user-friendly than top, with colors and dynamic load bars.

`htop`

- **df:** Displays the partition list, with the disk space used and available for each one.

`df`

`df -h`

-h option is for the human-readable format.

## Misc

Here are some other useful commands that I haven't managed to place in the other categories :

## Master Ethical Hacking Skills!

Join the [Complete Ethical Hacking Course Bundle](#) and step into the world of cybersecurity.

Learn to think like a hacker and protect systems with this comprehensive course.

- **history:** Linux store any command you type in an archive file. History is the command to use to display this list.

`history`

```
(pat@kali-infosec)-[~]
$ history
 1 sudo apt update
 2 sudo apt upgrade -y
 3 clear
 4 sudo shutdown -h now
 5 sudo dpkg -l | grep linux
```

You can also clear all the history.

`history -c`

Or clear one specific entry.

`history -d 12`

- **crontab:** Cron is a tool to schedule tasks on a Linux computer. Crontab is the file where you enter lines for each task to run.

`crontab -l`

`crontab -e`

-l option to display lines.

-e option to edit lines.

You can use sudo before to schedule tasks to run with root privileges.

I recommend using [a crontab tool like this one](#) to quickly find the correct syntax.

- **screen:** This tool allows you to let something run in the background even if you close your session. `screen -s <name>` `screen -r <name>`

-s option to start a new screen with the following name.

-r option to resume a running screen with this name.

You can forget the name if you want, an ID will be generated, use `screen -r` to find it and `screen -r <ID>` to resume it.

With only one screen running, `screen -r` will resume it directly.

## Warrior commands

---

In this last part, I'll introduce some powerful commands to master the Linux command line.

If you are just starting using Linux, you may not need to know this one, but if you want to save time or go further with it, you should probably take the time to learn these commands.

- **awk:** awk is almost a programming language in itself. It allows you to search string and transform them to display it differently.

So it'll be difficult to summarize all the possibilities in a few lines, but I'll try to give you some examples to understand it.

The basic syntax of awk is this one:

```
awk -F":" '{print $1}' /etc/passwd
```

/etc/passwd is the file to parse. The field separator is ":" so we use it in the -F option.

Then in the program string, we ask to display only the first column.

So this command will display only a list of usernames.

This is the simple way to use it if you want to know more, I recommend reading a dedicated tutorial like [this one](#).

- **sed:** sed allows you to do similar things to awk. This command will transform text to what you want.

As for awk, it's a complex command to master, and I'll only introduce it here.

The basic syntax looks like this:

```
sed <option> <script> <file>
```

So it's very close to awk on this.

Let's see an example:

```
sed '/^#/d' /etc/apache2/apache2.conf
```

```
(pat@kali-infosec)-[~]
$ cat /etc/ssh/sshd_config | tail -n10

# override default of no subsystems
Subsystem      sftp      /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server

(pat@kali-infosec)-[~]
$ sed '/^#/d' /etc/ssh/sshd_config | tail -n10
UsePAM yes

X11Forwarding yes
PrintMotd no

AcceptEnv LANG LC_*

Subsystem      sftp      /usr/lib/openssh/sftp-server
```

In each configuration file, you'll find a lot of comments to explain what each line is.

This command will display the apache configuration file without comments.

We use a regular expression to delete lines starting with #.

You have to redirect the output to another file to save it.

```
sed '/^#/d' /etc/apache2/apache2.conf > /etc/apache2/apache2-
nocomment.conf
```

Like for awk, this is just a glimpse of what sed can do.

If you want to know more, there is also a [good sed tutorial on the same website](#).



- **cut:** cut is the last way to transform text that I'll introduce. It's less powerful but it's simpler to use, so if cut can do it, you'll probably prefer to use it rather than awk or sed.

As the name suggests, cut allows you to extract part of a text or file.

The basic syntax is:

```
cut <options> <file>
echo <string> | cut <options>
```

The first one is for a file, and the second one to cut a string directly.

A basic example now:

```
echo "abcdefghi" | cut -c 2-4
```

This will display only "bcd".

-c option is for the character, so basically, it'll extract character 2 to 4.

Here are other options with a file:

```
cut -d : -f 1 /etc/passwd
```

This will do the same thing as the first example of the awk command.

/etc/passwd is a file with ":" use as a delimiter.

-d option is to give the delimiter character (":").

-f option is to indicate the column to extract (f stands for the field).

So, this will display only the first column and you'll get a list of usernames.

- **wc:** wc stands for Word Count, it allows you to count everything in a file or stream. There are three main options: -l for lines, -w for words and -m for characters. There is also the -c option to get the file size.

Wc without option will give you all of this.

```
wc .bash_history
668 1977 17979 .bash_history
```

The first column is line count, the second is word count, and the last is the file size in bytes.

Here are some examples of options:

```
wc -l .bash_history
ls -latr | wc -l
wc -w myfile.txt
```

- **lsof:** lsof stands for "List open files".

This command displays all files open on your Linux system.

This can be useful to know why you can't edit a file, or which file lock the unmount process.

```
lsof
```

- **watch:** If you are waiting for something, in a file or directory, the watch command can help you to monitor what happens. This will execute the same command every two seconds.

```
watch date
```

```
watch ls -latr
```

```
watch cat output.txt
```

You can also change the refresh rate with the -n option.

```
watch -n10 date
```

This will display the current date every ten seconds.

- **netstat:** Netstat is a powerful tool to monitor what your Linux server is doing with the network. For example, you can see every port open and every traffic flow. But netstat is a complex tool that i can't explain in detail in a few lines. I will only introduce some basic usages to display all listening connections you can use:

```
netstat -l
```

-p option will add the process id (PID).

```
netstat -lp
```

-c option allows you to refresh data continuously.

```
netstat -lpc
```

You can find all options in the [man page](#) of netstat.

- **dmesg:** This command is useful to understand your Linux system boot problems. It will show you every event that happened in the start sequence. Here you could see errors with drivers or services and understand why something doesn't work the way you want.

```
dmesg
```

You will get a column with the time elapsed since the beginning of the boot and a text explaining what happened.

There are also normal messages when everything is fine.

```
(pat@kali-infosec)-[~]
$ sudo dmesg
[ 0.000000] Linux version 6.1.0-kali9-amd64 (devel@kali.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld
(GNU Binutils for Debian) 2.40) #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-6.1.0-kali9-amd64 root=UUID=a5661f59-4507-4b27-9e24-
9d3224419c0b ro quiet splash
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x020: 'AVX-512 opmask'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x040: 'AVX-512 Hi256'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x080: 'AVX-512 ZMM_Hi256'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: xstate_offset[5]: 832, xstate_sizes[5]: 64
[ 0.000000] x86/fpu: xstate_offset[6]: 896, xstate_sizes[6]: 512
[ 0.000000] x86/fpu: xstate_offset[7]: 1408, xstate_sizes[7]: 1024
[ 0.000000] x86/fpu: Enabled xstate features 0xe7, context size is 2432 bytes, using 'compact' format
```

## Conclusion

Here we are.

Now you should have a better idea of commands to learn to get the best of your Linux computer or server.

And you, what are your more used commands?

You can see it by analyzing your history file with this kind of command:

```
history | awk '{CMD[$2]++;count++;}END { for (a in CMD)print CMD[a] " "
CMD[a]/count*100 "% " a;}' | grep -v "./" | column -c3 -s " " -t | sort -nr | nl |
head -n10
```

Want to experiment a few concretes examples with the command line? Try these tutorials I wrote on this website:

- [How To Format USB Drives The Right Way On Kali Linux](#)
- [How To Change DNS Servers On Kali Linux \(GUI & Commands\)](#)
- [Mastering Netcat on Kali Linux: A beginner's guide](#)

**Whenever you're ready for more security, here are things you should think about:**

- **Break free from Gmail**: You should be able to choose what happens to your data. With Proton, only you can read your emails. [Get private email](#).
- **Protect yourself online**: Use a high-speed Swiss VPN that safeguards your privacy. Open-source, no activity logs. [Get Proton VPN risk-free](#).
- **Master Linux commands**: A sure method to learn (and remember) Linux commands. Useful ones only, one at a time, with clear explanations. [Download the e-book](#).