

UNIT-I

Linux introduction and file system–Basic Features, Advantage, Installing requirement , Basic Architecture of Unix/Linux System,Kernel, Shell. Linux File System-Boot block, super block, Inode Table, data blocks, how Linux Access Files, storage files,Linux Standard directories.Commands for files and directories cd , ls , cp ,md, rm , mkdir , rmdir , pwd ,file ,more, less, creating and viewing files using cat , file comparison -cmp & comm., view files, disk related commands, checking disk free spaces

Index:

| S.no. | Topic | Page |
|-------|--------------------------------|-------|
| A | Introduction to linux | 1 -15 |
| B | file realted commands in linux | 16-62 |
| B.1 | cat command | 16-20 |
| B.2 | ls command | 21-31 |
| B.3 | cd command | 32-34 |
| B.4 | pwd command | 35-35 |
| B.5 | cp command | 36-38 |
| B.6 | mv command | 39-41 |
| B.7 | rm command | 42-45 |
| B.8 | mkdir command | 46-47 |
| B.9 | rmdir command | 48-48 |
| B.10 | more command | 49-52 |
| B.11 | less command | 53-57 |
| B.14 | file command | 58-62 |
| C | File comparison commands | 63-69 |
| D | Disk related commands | 70-86 |
| D.1 | fdisk command | 70-76 |
| D.2 | df command | 77-80 |
| D.# | du command | 81-86 |

Linux Introduction

Linux is a UNIX-base operating system. Its original creator was a Finnish student name Linus Torvalds, although being ‘open source’ it has change a great deal since its original conception. It belongs to nobody, and is free to download and use. We will start with an overview of how Linux became the operating system it is today, but first we will know what is an operating system?

Operating System:

An operating system is the software that manages the computer's hardware and provides a convenient and safe environment for running programs. It acts as an interface between programs and the hardware resources that these programs access (like memory, hard disk , printers etc). It is loaded into memory when a computer is booted and remains active as long as the machine is up.

The operations performed by the Operating system also depend on the operating system we are using, but the following actions are common to most systems:

1. The operating system allocates memory for the program and loads the program to the allocated memory.
2. It also loads the CPU registers with control information related to the program. The registers maintain the memory locations where each segment of a program is stored.
3. The instructions provided in the program are executed by the CPU. The operating system keeps track of the instruction that was last executed. this enables it to resume a program if it had to be taken out of the CPU before it completed execution.
4. If the program needs to access the hardware, it makes a call to the operating system rather than attempt to do the job itself. For instance, if the program needs to read a file on disk, the operating system directs the disk controller to open the file and make the data available to the program.
5. After the program has completed execution, the OS cleans up the memory and registers and make them available for the next program.

Linux Operating System:

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its functionality list is quite similar to that of UNIX.

Components of Linux System:

Linux Operating System has primarily three components

Kernel - "The kernel" is in charge for maintaining all the vital abstractions of the operating system, together with such things as virtual memory and processes. The Linux kernel forms the central part of Linux operating system. It provides all the functionality compulsory to run processes, and it also provides "system services" to give arbitrated and sheltered or protected access to hardware resources. The kernel implements every feature that is required to be eligible as an operating system.

System Library - System libraries are special functions or programs using which application programs or system utilities access Kernel's features. These libraries implement most of the functionalities of the operating system and do not require kernel module's code access rights.

System Utility - "the system utilities" are the programs that execute individual, particular and specialized managing tasks. Some of the system utilities may be invoked just once to initialize and configure some features of the system; others (known as daemons in UNIX language) may run enduringly, conducting such tasks as responding to inward or incoming network connections, accepting logon requests from terminals or updating log records and files.

Basic Features

Following are some of the important features of Linux Operating System.

Portable - Portability means softwares can work on different types of hardwares in same way. Linux kernel and application programs support their installation on any kind of hardware platform.

Open Source - Linux source code is freely available and it is community based development project. Multiple teams work in collaboration to enhance the capability of Linux operating system and it is continuously evolving.

Multi-User - Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.

Multiprogramming - Linux is a multiprogramming system means multiple applications can run at same time.

Hierarchical File System - Linux provides a standard file structure in which system files/ user files are arranged.

Shell - Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs etc.

Security - Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

Advantages of linux

A lot of the advantages of Linux are a consequence of Linux' origins, deeply rooted in UNIX, except for the first advantage, of course:

Linux is free:

Linux can be downloaded in its entirety from the Internet completely for free. No registration fees, no costs per user, free updates, and freely available source code in case you want to change the behavior of your system.

Most of all, Linux is free as in free speech:

The license commonly used is the GNU Public License (GPL). The license says that anybody who may want to do so, has the right to change Linux and eventually to

redistribute a changed version, on the one condition that the code is still available after redistribution. In practice, you are free to grab a kernel image, for instance to add support for teletransportation machines or time travel and sell your new code, as long as your customers can still have a copy of that code.

Linux is portable to any hardware platform:

A vendor who wants to sell a new type of computer and who doesn't know what kind of OS his new machine will run (say the CPU in your car or washing machine), can take a Linux kernel and make it work on his hardware, because documentation related to this activity is freely available.

Linux was made to keep on running:

As with UNIX, a Linux system expects to run without rebooting all the time. That is why a lot of tasks are being executed at night or scheduled automatically for other calm moments, resulting in higher availability during busier periods and a more balanced use of the hardware. This property allows for Linux to be applicable also in environments where people don't have the time or the possibility to control their systems night and day.

Linux is secure and versatile:

The security model used in Linux is based on the UNIX idea of security, which is known to be robust and of proven quality. But Linux is not only fit for use as a fort against enemy attacks from the Internet: it will adapt equally to other situations, utilizing the same high standards for security. Your development machine or control station will be as secure as your firewall.

Linux is scalable:

From a Palmtop with 2 MB of memory to a petabyte storage cluster with hundreds of nodes: add or remove the appropriate packages and Linux fits all. You don't need a supercomputer anymore, because you can use Linux to do big things using the building blocks provided with the system. If you want to do little things, such as

making an operating system for an embedded processor or just recycling your old 486, Linux will do that as well.

The Linux OS and most Linux applications have very short debug-times:

Because Linux has been developed and tested by thousands of people, both errors and people to fix them are usually found rather quickly. It sometimes happens that there are only a couple of hours between discovery and fixing of a bug.

Linux Vs Windows

The main benefits and advantages of Linux over other operating systems, particularly Microsoft Windows, are:

- It is free to use and distribute.
- Support is free through online help sites, blogs and forums.
- It is very reliable – more so than most other operating systems with very few crashes.
- A huge amount of free open source software has been developed for it.
- It is very resistant to malware such as spyware, adware and viruses.
- It runs in a wide variety of machines than cannot be updated to use newer Windows versions.
- Since the source code is visible, ‘backdoors’ are easily spotted, so Linux offers greater security for sensitive applications.
- Linux offers a high degree of flexibility of configuration, and significant customization is possible without modifying the source code.

Linux System Requirements

The following table lists the minimum and the recommended requirements for using Helix Producer on Linux.

| Requirement | Minimum | Recommended |
|-------------|---|-------------|
| Version | Linux 2.2 and 2.4 with glibc 2.1 or greater | |

| | | |
|----------------------------|--|----------|
| CPU | 400 MHz | 800+ MHz |
| RAM | 32 MB (file to file encoding) 96 MB (live broadcasting) | 256 MB |
| Hard Disk space (software) | 20 MB | |
| Hard Disk space (data) | 500 MB | 1 GB |

Linux Operating System Architecture

Linux System Architecture consists of following layers

Hardware layer - Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).

Kernel - Core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.

Shell - An interface to kernel, hiding complexity of kernel's functions from users. Takes commands from user and executes kernel's functions.

Utilities - Utility programs giving user most of the functionalities of an operating systems.

Linux kernel and Shell

Kernel Definition

The kernel is a program that constitutes the central core of a computer operating system. It has complete control over everything that occurs in the system.

A kernel can be contrasted with a shell (such as bash, csh or ksh in Unix-like operating systems), which is the outermost part of an operating system and a program that interacts with user commands. The kernel itself does not interact directly with the user, but rather interacts with the shell and other programs as well as with the hardware devices on the system, including the processor (also called the central processing unit or CPU), memory and disk drives.

The kernel is the first part of the operating system to load into memory during booting (i.e., system startup), and it remains there for the entire duration of the computer session because its services are required continuously. Thus it is important for it to be as small as possible while still providing all the essential services needed by the other parts of the operating system and by the various application programs.

Because of its critical nature, the kernel code is usually loaded into a protected area of memory, which prevents it from being overwritten by other, less frequently used parts of the operating system or by application programs. The kernel performs its tasks, such as executing processes and handling interrupts, in kernel space, whereas everything a user normally does, such as writing text in a text editor or running programs in a GUI (graphical user interface), is done in user space. This separation is made in order to prevent user data and kernel data from interfering with each other and thereby diminishing performance or causing the system to become unstable (and possibly crashing).

When a computer crashes, it actually means the kernel has crashed. If only a single program has crashed but the rest of the system remains in operation, then the kernel itself has not crashed. A crash is the situation in which a program, either a user application or a part of the operating system, stops performing its expected function(s) and responding to other parts of the system. The program might appear to the user to freeze. If such program is a critical to the operation of the kernel, the entire computer could stall or shut down.

The kernel provides basic services for all other parts of the operating system, typically including memory management, process management, file management and I/O (input/output) management (i.e., accessing the peripheral devices). These services are requested by other parts of the operating system or by application programs through a specified set of program interfaces referred to as system calls.

Shell:

In early days of computing, instruction are provided using binary language, which is difficult for all of us, to read and write. So in Os there is special program called Shell. Shell accepts your instruction or commands in English (mostly) and if its a valid command, it is pass to kernel.

Shell is a user program or it's environment provided for user interaction. Shell is an command language interpreter that executes commands read from the standard input device (keyboard) or from a file.

Shell is not part of system kernel, but uses the system kernel to execute programs, create files etc.

Previously, it was the only user interface available on a Unix computer. Nowadays, we have graphical user interfaces (GUIs) in addition to command line interfaces (CLIs) such as the shell.

Types Of Shells:

There are several different shells available for Unix; the most popular are described here.

Sh:

The Bourne shell, called "sh," is one of the original shells, developed for Unix computers by Stephen Bourne at AT&T's Bell Labs in 1977. Its long history of use means many software developers are familiar with it. It offers features such as input and output redirection, shell scripting with string and integer variables, and condition testing and looping.

Bash:

The popularity of sh motivated programmers to develop a shell that was compatible with it, but with several enhancements. Linux systems still offer the sh shell, but "bash" -- the "Bourne-again Shell," based on sh -- has become the new default standard. One attractive feature of bash is its ability to run sh shell scripts unchanged. Shell scripts are complex sets of commands that automate programming and maintenance chores; being able to reuse these scripts saves

programmers time. Conveniences not present with the original Bourne shell include command completion and a command history.

csh and tcsh:

Developers have written large parts of the Linux operating system in the C and C++ languages. Using C syntax as a model, Bill Joy at Berkeley University developed the "C-shell," csh, in 1978. Ken Greer, working at Carnegie-Mellon University, took csh concepts a step forward with a new shell, tcsh, which Linux systems now offer. Tcsh fixed problems in csh and added command completion, in which the shell makes educated "guesses" as you type, based on your system's directory structure and files. Tcsh does not run bash scripts, as the two have substantial differences.

Ksh:

David Korn developed the Korn shell, or ksh, about the time tcsh was introduced. Ksh is compatible with sh and bash. Ksh improves on the Bourne shell by adding floating-point arithmetic, job control, command aliasing and command completion. AT&T held proprietary rights to ksh until 2000, when it became open source.

File system:

"On a UNIX system, everything is a file; if something is not a file, it is a process."

This statement is true because there are special files that are more than just files (named pipes and sockets, for instance), but to keep things simple, saying that everything is a file is an acceptable generalization. A Linux system, just like UNIX, makes no difference between a file and a directory, since a directory is just a file containing names of other files. Programs, services, texts, images, and so forth, are all files. Input and output devices, and generally all devices, are considered to be files, according to the system.

Most files are just files, called regular files; they contain normal data, for example text files, executable files or programs, input for or output from a program and so on.

While it is reasonably safe to suppose that everything you encounter on a Linux system is a file, there are some exceptions.

Directories: files that are lists of other files.

Special files: the mechanism used for input and output. Most special files are in /dev, we will discuss them later.

Links: a system to make a file or directory visible in multiple parts of the system's file tree. We will talk about links in detail.

(Domain) sockets: a special file type, similar to TCP/IP sockets, providing inter-process networking protected by the file system's access control.

Named pipes: act more or less like sockets and form a way for processes to communicate with each other, without using network socket semantics.

A file system is a logical collection of files on a partition or disk. A partition is a container for information and can span an entire hard drive if desired.

Your hard drive can have various partitions which usually contains only one file system, such as one file system housing the / file system or another containing the /home file system.

One file system per partition allows for the logical maintenance and management of differing file systems.

Everything in Unix is considered to be a file, including physical devices such as DVD-ROMs, USB devices, floppy drives, and so forth.

Directory Structure:

Unix uses a hierarchical file system structure, much like an upside-down tree, with root (/) at the base of the file system and all other directories spreading from there.

A UNIX filesystem is a collection of files and directories that has the following properties:

- It has a root directory (/) that contains other files and directories.
- Each file or directory is uniquely identified by its name, the directory in which it resides, and a unique identifier, typically called an inode.
- By convention, the root directory has an inode number of 2 and the lost+found directory has an inode number of 3. Inode numbers 0 and 1 are

not used. File inode numbers can be seen by specifying the -i option to ls command.

- It is self contained. There are no dependencies between one filesystem and any other.
-

The directories have specific purposes and generally hold the same types of information for easily locating files. Following are the directories that exist on the major versions of Unix:

Directory Description

/ This is the root directory which should contain only the directories needed at the top level of the file structure.

/bin This is where the executable files are located. They are available to all user.

/dev These are device drivers.

/etc Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages.

/lib Contains shared library files and sometimes other kernel-related files.

/boot Contains files for booting the system.

/home Contains the home directory for users and other accounts.

/mnt Used to mount other temporary file systems, such as cdrom and floppy for the CD-ROM drive and floppy diskette drive, respectively

/proc Contains all processes marked as a file by process number or other information that is dynamic to the system.

/tmp Holds temporary files used between system boots

/usr Used for miscellaneous purposes, or can be used by many users. Includes administrative commands, shared files, library files, and others

/var Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data

/sbin Contains binary (executable) files, usually for system administration. For example fdisk and ifconfig utilities.

/kernel Contains kernel files

Basic Unix File System Structure

Overview

The structure of a simple Unix file system can generally be separated into four parts, **the boot block, the super block, the inode list and the data blocks.**

The **boot block** is located at the beginning of the file system and can be accessed with the minimal code incorporated in the computer's rom bios. The boot block of the bootable partition contains the code needed to further initialize the operating system.

The **super block** describes the state of a file system - how large it is, how many files it can store, what parts of the storage area are already in use and what parts are available, etc.

The **inode list (table)** is a list of inodes that are used to track and maintain information about each file created on the filesystem. All access to a file is based on the data in the inode list. The inode list is found immediately

after the super block, and the first usable inode is reserved for accessing the root directory of the filesystem.

The **data blocks** are where the data of a file is stored. These blocks follow the inode table and occupy most of the storage device's space.

Super Block

The super block contains the following information:

- the size of the file system.

This is the storage size of the device or current partition on the device.

- the list of storage blocks

The storage space is divided up into a series of standard size blocks.

When data is moved to or from the filesystem, it is moved in block units.

- the number of free blocks.

- the location of all free blocks.

- the index of the next free block in the free block list.

- the size of the inode list.

The inode list is initialized to track the maximum number of files

which cannot be more than the maximum number of storage blocks.

- the number of free inodes in the file system.

- the index of the next free inode in the free inode list.

iNode: Every file have one inode and vice versa. In linux every file is recognized with integer number known as inode number. This structure consists of info of file about

1. file ownership indication
2. file type (e.g., regular, directory, special device, pipes, etc.)
3. file access permissions. May have setuid (sticky) bit set.
4. time of last access, and modification
5. number of links (aliases) to the file
6. pointers to the data blocks for the file

7. size of the file in bytes (for regular files), major and minor device numbers for special devices.

Inodes include pointers to the data blocks. Each inode contains 15 pointers:

- the first 12 pointers point directly to data blocks
- the 13th pointer points to an indirect block, a block containing pointers to data blocks
- the 14th pointer points to a doubly-indirect block, a block containing 128 addresses of singly indirect blocks
- the 15th pointer points to a triply indirect block (which contains pointers to doubly indirect blocks, etc.)

Commands for files and directories:

Commands for files and directories :

1) cat :

The cat (short for "concatenate") command allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files.

The sample file used for cat command is "test".

i) Display the contents of a file.

cat command can be used to view files content.using filename with cat command can help us to view the file content.Its syntax is as follows:

Syantx: cat [filename]

```
root@kali:~/Desktop# cat test
Why Do a Penetration Test?
Companies invest millions of dollars in security programs to protect critical
infrastructures, identify chinks in the armor, and prevent serious data breaches.
A penetration test is one of the most effective ways to identify systemic weak-
nesses and deficiencies in these programs. By attempting to circumvent secu-
rity controls and bypass security mechanisms, a penetration tester is able to
identify ways in which a hacker might be able to compromise an organization's
security and damage the organization as a whole.
As you read through this book, remember that you're not necessarily
targeting one system or multiple systems. Your goal is to show, in a safe and
controlled manner, how an attacker might be able to cause serious harm to
an organization and impact its ability to, among other things, generate reve-
nue, maintain its reputation, and protect its customers.
Why Metasploit?
Metasploit isn't just a tool; it's an entire framework that provides the infra-
structure needed to automate mundane, routine, and complex tasks. This
allows you to concentrate on the unique or specialized aspects of penetration
testing and on identifying flaws within your information security program.
As you progress through the chapters in this book and establish a well-
rounded methodology, you will begin to see the many ways in which Meta-
sploit can be used in your penetration tests. Metasploit allows you to easily
build attack vectors to augment its exploits, payloads, encoders, and more
in order to create and execute more advanced attacks. At various points in
this book we explain several third-party tools—including some written by the
authors of this book—that build on the Metasploit Framework. Our goal is to
get you comfortable with the Framework, show you some advanced attacks,
and ensure that you can apply these techniques responsibly.root@kali:~/Desktop#
```

Note : here content of file "test' is viewed.we can also view contents of more than one files also. for this we have to specify the name of the files which we wish to view by separating the names of file with spaces.[make sure file exists]

Synatx : cat [file1] [file2]

```

root@kali:~/Desktop# cat test printout
Why Do a Penetration Test?
Companies invest millions of dollars in security programs to protect critical
infrastructures, identify chinks in the armor, and prevent serious data breaches.
A penetration test is one of the most effective ways to identify systemic weak-
nesses and deficiencies in these programs. By attempting to circumvent secu-
rity controls and bypass security mechanisms, a penetration tester is able to
identify ways in which a hacker might be able to compromise an organization's
security and damage the organization as a whole.
As you read through this book, remember that you're not necessarily
targeting one system or multiple systems. Your goal is to show, in a safe and
controlled manner, how an attacker might be able to cause serious harm to
an organization and impact its ability to, among other things, generate reve-
nue, maintain its reputation, and protect its customers.
Why Metasploit?
Metasploit isn't just a tool; it's an entire framework that provides the infra-
structure needed to automate mundane, routine, and complex tasks. This
allows you to concentrate on the unique or specialized aspects of penetration
testing and on identifying flaws within your information security program.
As you progress through the chapters in this book and establish a well-
rounded methodology, you will begin to see the many ways in which Meta-
sploit can be used in your penetration tests. Metasploit allows you to easily
build attack vectors to augment its exploits, payloads, encoders, and more
in order to create and execute more advanced attacks. At various points in
this book we explain several third-party tools—including some written by the
authors of this book—that build on the Metasploit Framework. Our goal is to
get you comfortable with the Framework, show you some advanced attacks,
and ensure that you can apply these techniques responsibly.root@kali:~/Desktop# awk '{print $0}' test
1      Pen      5      20.00
2      Rubber   10     2.00
3      Pencil    3      3.50
4      Cock     2      45.50
root@kali:~/Desktop# cat > pri

```

Note: here file "test" and "printout" are viewed together.

ii) create a new file : > operator

We can create new files using the cat command and operator '>' .

To create the file we have to specify the filename after following syntax.

Syntax: cat > [filename]

```

root@kali:~/Desktop# cat > test2
hello , we are working on linx
root@kali:~/Desktop# cat test2
hello , we are working on linx
root@kali:~/Desktop# █

```

here file "test2" is created.once the file is created we can write the content of the file from next line of command.

Note: use ctrl + d to go back to shell.

iii) Create one file from several files

we can create a new file from combining contents of pre-existing file.for this we have to specify the name of the files between the command "cat" and '>' operator. we can specify any number of files.we wish.the contents of the file will be concatenated together and will be added in the new file.

Syntax ; cat 'old_file1' 'old_file2' > new_file

```

root@kali:~/Desktop# cat test prinout > test3
root@kali:~/Desktop# cat test3
Why Do A Penetration Test?
Companies invest millions of dollars in security programs to protect critical
infrastructures, identify chinks in the armor, and prevent serious data breaches.
A penetration test is one of the most effective ways to identify systemic weak-
nesses and deficiencies in these programs. By attempting to circumvent secu-
rity controls and bypass security mechanisms, a penetration tester is able to
identify ways in which a hacker might be able to compromise an organization's
security and damage the organization as a whole.
As you read through this book, remember that you're not necessarily
targeting one system or multiple systems. Your goal is to show, in a safe and
controlled manner, how an attacker might be able to cause serious harm to
an organization and impact its ability to, among other things, generate reve-
nue, maintain its reputation, and protect its customers.
Why Metasploit?
Metasploit isn't just a tool; it's an entire framework that provides the infra-
structure needed to automate mundane, routine, and complex tasks. This
allows you to concentrate on the unique or specialized aspects of penetration
testing and on identifying flaws within your information security program.
As you progress through the chapters in this book and establish a well-
rounded methodology, you will begin to see the many ways in which Meta-
sploit can be used in your penetration tests. Metasploit allows you to easily
build attack vectors to augment its exploits, payloads, encoders, and more
in order to create and execute more advanced attacks. At various points in
this book we explain several third-party tools—including some written by the
authors of this book—that build on the Metasploit Framework. Our goal is to
get you comfortable with the Framework, show you some advanced attacks,
and ensure that you can apply these techniques responsibly.root@kali:~/Desktop# awk '{print $0}' test
1      Pen      5      20.00
2      Rubber   10     2.00
3      Pencil    3      3.50
4      Cock     2      45.50

```

here file 'test3' is created from file 'test' and 'prinout'.
the contents of file "prinout" are concatenated after the content of file
'test' in new file "test3".

we can view the content of file "test3".

iv) Display Line numbers : -n option

if we wish to display the line number after each line of the file while we
are viewing it.we can do it by using -n option of cat command

Syntax: cat -n filename

```

root@kali:~/Desktop# cat -n test
1 Why Do A Penetration Test?
2 Companies invest millions of dollars in security programs to protect critical
3 infrastructures, identify chinks in the armor, and prevent serious data breaches.
4 A penetration test is one of the most effective ways to identify systemic weak-
5 ness and deficiencies in these programs. By attempting to circumvent secu-
6 rity controls and bypass security mechanisms, a penetration tester is able to
7 identify ways in which a hacker might be able to compromise an organization's
8 security and damage the organization as a whole.
9 As you read through this book, remember that you're not necessarily
10 targeting one system or multiple systems. Your goal is to show, in a safe and
11 controlled manner, how an attacker might be able to cause serious harm to
12 an organization and impact its ability to, among other things, generate reve-
13 nue, maintain its reputation, and protect its customers.
14 Why Metasploit?
15 Metasploit isn't just a tool; it's an entire framework that provides the infra-
16 structure needed to automate mundane, routine, and complex tasks. This
17 allows you to concentrate on the unique or specialized aspects of penetration
18 testing and on identifying flaws within your information security program.
19 As you progress through the chapters in this book and establish a well-
20 rounded methodology, you will begin to see the many ways in which Meta-
21 sploit can be used in your penetration tests. Metasploit allows you to easily
22 build attack vectors to augment its exploits, payloads, encoders, and more
23 in order to create and execute more advanced attacks. At various points in
24 this book we explain several third-party tools—including some written by the
25 authors of this book—that build on the Metasploit Framework. Our goal is to
26 get you comfortable with the Framework, show you some advanced attacks,
27 and ensure that you can apply these techniques responsibly.root@kali:~/Desktop# 
```

v) To Append content to a existing file. : >> operator

If we wish to append the content to the pre-existing content of a file rather than creating a new file we can use the '>>' in place of '>'.

Syntax : cat >> [filename]

[content]

```
root@kali:~/Desktop# cat >> test
its over
root@kali:~/Desktop# cat test
Why Do a Penetration Test?
Companies invest millions of dollars in security programs to protect critical
infrastructures, identify chinks in the armor, and prevent serious data breaches.
A penetration test is one of the most effective ways to identify systemic weak-
nesses and deficiencies in these programs. By attempting to circumvent secu-
rity controls and bypass security mechanisms, a penetration tester is able to
identify ways in which a hacker might be able to compromise an organization's
security and damage the organization as a whole.
As you read through this book, remember that you're not necessarily
targeting one system or multiple systems. Your goal is to show, in a safe and
controlled manner, how an attacker might be able to cause serious harm to
an organization and impact its ability to, among other things, generate reve-
nue, maintain its reputation, and protect its customers.
Why Metasploit?
Metasploit isn't just a tool; it's an entire framework that provides the infra-
structure needed to automate mundane, routine, and complex tasks. This
allows you to concentrate on the unique or specialized aspects of penetration
testing and on identifying flaws within your information security program.
As you progress through the chapters in this book and establish a well-u become, the mi
rounded methodology, you will begin to see the many ways in which Meta-
sploit can be used in your penetration tests. Metasploit allows you to easily
build attack vectors to augment its exploits, payloads, encoders, and more
in order to create and execute more advanced attacks. At various points in
this book we explain several third-party tools—including some written by the
authors of this book—that build on the Metasploit Framework. Our goal is to
get you comfortable with the Framework, show you some advanced attacks,
and ensure that you can apply these techniques responsibly.its over
```

Note : using '>' operator in place of '>>' operator will overwrite the current content of file. If the File does not exist

```
root@kali:~/Desktop# cat >> test5
hello
root@kali:~/Desktop# cat test5
hello
root@kali:~/Desktop#
```

```
root@kali:~/Desktop# cat > test5
old
root@kali:~/Desktop# cat test5
old
root@kali:~/Desktop#
```

vi) view tab space in file : -T

if we want to see all the tab spaces in our file then we can use -T option with the cat command file name.

Syntax : cat -T [filename]

```
root@kali:~/Desktop# cat -T prinout
root@kali:~/Desktop# awk '{print $0}' test
1^IPen^I5^I20.00
2^IRubber^I10^I2.00
3^IPencil^I3^I3.50
4^ICock^I2^I45.50

root@kali:~/Desktop# cat > pri
/pen/ print $3
root@kali:~/Desktop# PATH=/bin:/usr/bin:/etc:.
root@kali:~/Desktop# chmod a+x pri
root@kali:~/Desktop# awk -f pri test
awk: pri: line 1: syntax error at or near print
root@kali:~/Desktop# awk -f pri test
root@kali:~/Desktop# chmod a+x pri
root@kali:~/Desktop# awk -f pri test
root@kali:~/Desktop# awk -f pri test
5
3

root@kali:~# cd Desktop
root@kali:~/Desktop# cat > bill
BEGIN
{
print "----->"
print "Vibhakar Singh"
print "----->"
}
{
total=$3 + $4
}
END
```

In the above picture we can see the '^I' symbols. These symbols represent the tab spaces in the file "prinout".

2) ls :It is usually used to view the contents of current directory. So, when you run this command, the files and sub-directories included under the current directory will be listed before you.

i) List Files with no option



```
root@kali:~/Desktop# ls
10989419_940756325974653_3425107860827932086_n.jpg
598718_3334799490347_667645853_n.jpg
785964.jpg
a
abc
abc1
abc2
abc3
abc4
a.zip
b
backup.tar
bald-rambo.srt
bill
C_201505221211085567.pdf
Declaration_1430992942546.pdf
dfgfgvbbhbb.png
DT20142615202_1431401891359.pdf
er1-2003.doc
file1
file2
Final-Syllabus-of-Signature-Verification-Training.docx
IBPS PO 2014: Online Exam Questionlist without using RTI!.pdf
IBPS PO 2014: Online Exam Questionlist without using RTI!.html
Ignite JL Annexure.pdf
images.jpeg
install_flash_player_11_linux.x86_64.tar.gz
Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
math
Metasploit - The Penetration Testers Guide - 2011.pdf
noti_files
noti.html
pattern_files
```

The most basic use of any command is when it is used without any option or parameter. So, when ls command is executed with no option or parameter, then it will enlist files and sub-directories of current directory.

ii) Open Last Edited File : -t option

```
root@kali:~/Desktop# ls -t
10989419_940756325974653_3425107860827932086_n.jpg
pic
abc
hello
b
a
abc
abc1
abc2
abc3
math
usr
Welcome to Ignite.pdf
IBPS PO 2014: Online Exam Questionlist without using RTI!.html
IBPS PO 2014: Online Exam Questionlist without using RTI!_files
noti_files
noti.html
pattern_files
pattern.html
Pattern_files
Pattern.html
TestBag_files
TestBag.html
Declaration_1430992942546.pdf
images.jpeg
Sample_Service_agreement_1432787607562.pdf
C_201505221211085567.pdf
DT20142615202_1431401891359.pdf
file2
file1
test
test3
Recording_peak_time_traffic_in_your_town_1389261322890_(1)_1432268319187.pdf
dfgfvbhb.png
```

-t sorts the file by modification time, showing the last edited file first.

```
root@kali:~/Desktop# ls -t | head -1
pic
root@kali:~/Desktop#
```

Note : head -1 picks up this first file.

iii)A Long List : -l option

It will display the contents of the current directory in a long list format.

```

root@kali:~/Desktop# ls -l
total 746748
-rw-r--r-- 1 root root 33319 Jun 5 22:01 10989419_940756325974653_3425107860827932086_n.jpg
-rw-r--r-- 1 root root 25060 Jan 17 10:37 598718_3334799490347_667645853_n.jpg
-rw-r--r-- 1 root root 116132 Jan 17 10:37 785964.jpg
drwxr-xr-x 2 root root 4096 Jun 4 09:20 a
drwxrwxrwx 3 root root 4096 Jun 4 10:10 ab1
drwxr-xr-x 2 root root 4096 Jun 4 09:14 abc1
drwxr-xr-x 2 root root 4096 Jun 4 09:14 abc2
drwxr-xr-x 2 root root 4096 Jun 4 09:14 abc3
drwxrwxrwx 2 root root 4096 Jun 4 09:17 ab2
-rw-r--r-- 1 root root 4403948 May 21 07:20 a.zip
drwxr-xr-x 2 root root 4096 Jun 4 09:21 b
-rw-r--r-- 1 root root 10240 May 21 20:26 backup.tar
-rw-r--r-- 1 root root 29442 Jan 24 16:36 bald-rambo.srt
-rw-r--r-x 1 test2 1014 178 Apr 12 21:43 bill
-rw-r--r-- 1 root root 1905622 May 28 17:59 C_201505221211085567.pdf
-rw-r--r-- 1 root root 223035 May 30 22:44 Declaration_1430992942546.pdf
-rw-r--r-- 1 root root 83860 May 22 18:39 dfgfqvhhb.png
-rw-r--r-- 1 root root 247802 May 28 07:18 DT20142615202_1431401891359.pdf
-rw-r--r-- 1 root root 38912 Apr 12 19:18 erl-2003.doc
-rw-r--r-- 1 root root 12 May 24 22:04 file1
-rw-r--r-- 1 root root 12 May 24 22:04 file2
-rwrxrwxrwx 1 root root 19792 Jan 14 16:50 Final-Syllabus-of-Signature-Verification-Training.docx
drwxrwxrwx 2 root root 4096 Jun 4 10:09 ab3
drwxr-xr-x 2 root root 4096 Jun 2 08:45 IBPS PO 2014: Online Exam Questionlist without using RTI!.files
-rw-r--r-- 1 root root 143614 Jun 2 08:45 IBPS PO 2014: Online Exam Questionlist without using RTI!.html
-rw-r--r-- 1 root root 335262 May 21 14:46 Ignite JL Annexure.pdf
-rw-r--r-- 1 root root 10361 May 29 18:54 images.jpeg
-rw-r--r-- 1 root root 7241885 Feb 6 08:30 install_flash_player_11_linux.x86_64.tar.gz
-rw-r--r-- 1 root root 725001235 Feb 6 13:31 Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
-rwrxr-xr-x 1 root root 40 Jun 4 06:39 math
-rwrxr-xr-x 1 root root 7237192 Oct 17 2012 Metasploit - The Penetration Testers Guide - 2011.pdf
drwxr-xr-x 3 root root 4096 Jun 2 08:32 noti_files
-rw-r--r-- 1 root root 10067 Jun 2 08:32 noti.html
drwxr-xr-x 3 root root 4096 Jun 2 08:31 pattern_files

```

In the output:

1st column will give detailed information regarding file permission,
 2nd column will tell you about the number of links to the file,
 3rd and 4th columns are associated with owner and group of the file,
 5th column will be displaying the size of the file in bytes,
 6th column will display the recent time and date at which the file was modified,
 and the last and 7th column is the actual file/directory name.

iv) Displaying Size in Human Readable Format : -h

option

Instead of printing the file size in terms of bytes, we can display it in human readable form, like MB, GB, TB and so on. This can be achieved when ls command is executed with the option -h.

```

root@kali:~/Desktop# ls -lh
total 730M
-rw-r--r-- 1 root root 33K Jun 5 22:01 10989419_940756325974653_3425107860827932086_n.jpg
-rw-r--r-- 1 root root 25K Jan 17 10:37 598718_3334799490347_667645853_n.jpg
-rw-r--r-- 1 root root 114K Jan 17 10:37 785964.jpg
drwxr-xr-x 2 root root 4.0K Jun 4 09:20 a
drwxrwxrwx 3 root root 4.0K Jun 4 10:10 ab1
drwxr-xr-x 2 root root 4.0K Jun 4 09:14 abc1
drwxr-xr-x 2 root root 4.0K Jun 4 09:14 abc2
drwxr-xr-x 2 root root 4.0K Jun 4 09:14 abc3
drwxrwxrwx 2 root root 4.0K Jun 4 09:17 ab2
-rw-r--r-- 1 root root 4.2M May 21 07:20 a.zip
drwxr-xr-x 2 root root 4.0K Jun 4 09:21 b
-rw-r--r-- 1 root root 10K May 21 20:26 backup.tar
-rw-r--r-- 1 root root 29K Jan 24 16:36 bald-rambo.srt
-rw-r--r-x 1 test2 1014 178 Apr 12 21:43 bill
-rw-r--r-- 1 root root 1.9M May 28 17:59 C_201505221211085567.pdf
-rw-r--r-- 1 root root 218K May 30 22:44 Declaration_1430992942546.pdf
-rw-r--r-- 1 root root 82K May 28 18:39 dfgfqvhhb.png
-rw-r--r-- 1 root root 242K May 28 07:18 DT20142615202_1431401891359.pdf
-rw-r--r-- 1 root root 38K Apr 12 19:18 erl-2003.doc
-rw-r--r-- 1 root root 12 May 24 22:04 file1
-rw-r--r-- 1 root root 12 May 24 22:04 file2
-rwrxrwxrwx 1 root root 20K Jan 14 16:50 Final-Syllabus-of-Signature-Verification-Training.docx
drwxrwxrwx 2 root root 4.0K Jun 4 10:09 ab3
drwxr-xr-x 2 root root 4.0K Jun 2 08:45 IBPS PO 2014: Online Exam Questionlist without using RTI!.files
-rw-r--r-- 1 root root 141K Jun 2 08:45 IBPS PO 2014: Online Exam Questionlist without using RTI!.html
-rw-r--r-- 1 root root 328K May 21 14:46 Ignite JL Annexure.pdf
-rw-r--r-- 1 root root 11K May 29 18:54 images.jpeg
-rw-r--r-- 1 root root 7.0M Feb 6 08:30 install_flash_player_11_linux.x86_64.tar.gz
-rw----- 1 root root 692M Feb 6 13:31 Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
-rwrxr-xr-x 1 root root 40 Jun 4 06:39 math
-rwrxr-xr-x 1 root root 7.0M Oct 17 2012 Metasploit - The Penetration Testers Guide - 2011.pdf
drwxr-xr-x 3 root root 4.0K Jun 2 08:32 noti_files
-rw-r--r-- 1 root root 9.9K Jun 2 08:32 noti.html
drwxr-xr-x 3 root root 4.0K Jun 2 08:31 pattern_files

```

Here switch -l and -h are used together.

v) Sort According to File Size : -S option (Capital S)

```
root@kali:~/Desktop# ls -lhS
total 730M
-rw----- 1 root  root 692M Feb  6 13:31 Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
-rwxr-xr-x 1 root  root 14M Dec 11  2011 The Web Application Hacker's Handbook - Finding and Exploiting Security Flaws.pdf
-rw-r--r-- 1 root  root 7.0M Feb  6 08:30 install_flash_player_11_linux.x86_64.tar.gz
-rwxr-xr-x 1 root  root 7.0M Oct 17  2012 Metasploit - The Penetration Testers Guide - 2011.pdf
-rw-r--r-- 1 root  root 4.2M May 21 07:20 a.zip
-rw-r--r-- 1 root  root 2.5M May 28 22:05 Sample_Service_agreement_1432787607562.pdf
-rw-r--r-- 1 root  root 1.9M May 28 17:59 C_201505221211085567.pdf
-rw-r--r-- 1 root  root 407K Jun  3 06:31 Welcome to Ignite.pdf
-rw-r--r-- 1 root  root 328K May 21 14:46 Ignite JL Annexure.pdf
-rw-r--r-- 1 root  root 242K May 28 07:18 DT20142615202_1431401891359.pdf
-rw-r--r-- 1 root  root 218K May 30 22:44 Declaration_1430992942546.pdf
-rw-r--r-- 1 root  root 172K May 19 05:17 Website.rar
-rw-r--r-- 1 root  root 141K Jun  2 08:45 IBPS PO 2014: Online Exam Questionlist without using RTI!.html
-rw-r--r-- 1 root  root 114K Jan 17 10:37 785964.jpg
-rw-r--r-- 1 root  root  82K May 22 18:39 dfgfvvhbb.png
-rw-r--r-- 1 root  root  38K Apr 12 19:18 er1-2003.doc
-rw-r--r-- 1 root  root  33K Jun  5 22:01 10989419_940756325974653_3425107860827932086_n.jpg
-rw-r--r-- 1 root  root  32K Jan 26 10:49 u.zip
-rw-r--r-- 1 root  root  29K Jan 24 16:36 bald-rambo.srt
-rw-r--r-- 1 root  root  29K May 23 03:14 Recording_peak_time_traffic_in_your_town_1389261322890_(1)_1432268319187.pdf
-rw-r--r-- 1 root  root  25K Jan 17 10:37 598718_3334799490347_667645853_n.jpg
-rwxrwxrwx 1 root  root  20K Jan 14 16:58 Final-Syllabus-of-Signature-Verification-Training.docx
-rw-r--r-- 1 root  root  11K May 29 18:54 images.jpeg
-rw-r--r-- 1 root  root  10K May 21 20:26 backup.tar
-rw-r--r-- 1 root  root  9.9K Jun  2 08:31 pattern.html
-rw-r--r-- 1 root  root  9.9K Jun  2 08:30 Pattern.html
-rw-r--r-- 1 root  root  9.9K Jun  2 08:29 TestBag.html
-rw-r--r-- 1 root  root  9.9K Jun  2 08:32 noti.html
drwxr-xr-x 2 root  root  4.0K Jun  4 09:20 a
drwxrwxrwx 3 root  root  4.0K Jun  4 10:10 b
drwxr-xr-x 2 root  root  4.0K Jun  4 09:14 abc1
drwxr-xr-x 2 root  root  4.0K Jun  4 09:14 abc2
drwxr-xr-x 2 root  root  4.0K Jun  4 09:14 abc3
drwxrwxrwx 2 root  root  4.0K Jun  4 09:17 bcc
```

-l -h and -S option are used together.

-S option sorts the file according to their size with file with maximumm size on top and minimum size on bottom.

vi) List the Hidden Files too : -a option

```
root@kali:~/Desktop# ls -a
.

10989419_940756325974653_3425107860827932086_n.jpg
598718_3334799490347_667645853_n.jpg
785964.jpg
a
abc1
abc2
abc3
abc4
a.zip
b
backup.tar
bald-rambo.srt
bill
C_201505221211085567.pdf
Declaration_1430992942546.pdf
dfgfvvhbb.png
DT20142615202_1431401891359.pdf
er1-2003.doc
file1
file2
Final-Syllabus-of-Signature-Verification-Training.docx
ibc
IBPS PO 2014: Online Exam Questionlist without using RTI!.files
IBPS PO 2014: Online Exam Questionlist without using RTI!.html
Ignite JL Annexure.pdf
images.jpeg
install_flash_player_11_linux.x86_64.tar.gz
Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
math
.mediaartlocal
```

The files can be hidden by starting a file name with a "." (dot) symbol.

Here file ".mediaartlocal' is hidden file.

vii) Sort by Extensions : -X option

```
root@kali:~/Desktop# ls -lx
total 746748
drwxr-xr-x 2 root root 4096 Jun 4 09:20 a
drwxrwxrwx 3 root root 4096 Jun 4 10:10 abc
drwxr-xr-x 2 root root 4096 Jun 4 09:14 abc1
drwxr-xr-x 2 root root 4096 Jun 4 09:14 abc2
drwxr-xr-x 2 root root 4096 Jun 4 09:14 abc3
drwxrwxrwx 2 root root 4096 Jun 4 09:17 abc
drwxr-xr-x 2 root root 4096 Jun 4 09:21 b
-rw-rxr-x 1 test2 1014 178 Apr 12 21:43 bill
-rw-r--r-- 1 root root 12 May 24 22:04 file1
-rw-r--r-- 1 root root 12 May 24 22:04 file2
drwxrwxrwx 2 root root 4096 Jun 4 10:09 file
drwxr-xr-x 2 root root 4096 Jun 2 08:45 IBPS_P0_2014:_Online_Exam_Questions_without_using_RTI!.files
-rwrxr-x 1 root root 4096 Jun 4 06:39 math
drwxr-xr-x 3 root root 4096 Jun 2 08:32 noti_files
drwxr-xr-x 3 root root 4096 Jun 2 08:31 pattern_files
drwxr-xr-x 3 root root 4096 Jun 2 08:30 Pattern_files
drwxr-xr-x 2 root root 4096 Jun 5 22:43 pic
-rwrxr-x 1 root root 116 Apr 13 03:35 shiki2
-rwrxr-x 1 test6 1014 310 Apr 13 03:00 shiki3
-rw-r--r-- 1 root root 1901 May 24 00:11 test
-rw-r--r-- 1 root root 3405 May 23 23:44 test3
drwxr-xr-x 3 root root 4096 Jun 2 08:29 TestBag_files
-rwrxr-x 1 root root 87 Apr 13 13:16 testing
-rw-r--r-- 1 root root 8 Apr 13 15:06 tst
drwxrwxr-x 7 501 501 4096 Jun 4 06:31 usr
-rw-r--r-- 1 root root 38912 Apr 12 19:18 erl-2003.doc
-rwrxrwx 1 root root 19792 Jan 14 16:50 Final-Syllabus-of-Signature-Verification-Training.docx
-rw-r--r-- 1 root root 7241885 Feb 6 08:30 install_flash_player_11_linux.x86_64.tar.gz
-rw-r--r-- 1 root root 143614 Jun 2 08:45 IBPS_P0_2014:_Online_Exam_Questions_without_using_RTI!.html
-rw-r--r-- 1 root root 10067 Jun 2 08:32 noti.html
-rw-r--r-- 1 root root 10088 Jun 2 08:31 pattern.html
-rw-r--r-- 1 root root 10088 Jun 2 08:30 Pattern.html
-rw-r--r-- 1 root root 10088 Jun 2 08:29 TestBag.html
-rw-r--r-- 1 root root 10361 May 29 18:54 images.jpeg
```

By using -X option with 'ls' command we can sort (group) the files of same extension together.

Above all html file are together and so do the doc files.

viii) Show the Contents of Sub-Directories : -R option

It will recursively display the contents of the sub-directories too.

Syntax : ls -R

```
root@kali:~/Desktop# ls -R
.:
10989419_940756325974653_3425107860827932086_n.jpg
598718_3334799490347_667645853_n.jpg
785964.jpg
a
abc
abc1
abc2
abc3
abcd
a.zip
b
backup.tar
```

```

TestBag_files
TestBag.html
testing
The Web Application Hacker's Handbook - Finding and Exploiting Security Flaws.pdf
tst
usr
u.zip
WebSite.rar
Welcome to Ignite.pdf

./a:
./abc:
3 4 9
./abc/4:
./abc1:
./abc2:
./abc3:
./abcd:
./b:
./hello:
5

./IBPS PO 2014: Online Exam Questionlist without using RTI!.files:
a.js           cloudflare.js      flipkart_india.png      IBPS-PO-2015-Pie-Chart.gif
autoptimize_07197a4c169d697a26e42e6e167facdf.js  css.css        ga.js          s.gif
autoptimize_2018e22f059d7042fe7a4413cf47bfd2.css  devicepx-jetpack.js IBPS-PO-2015-Line-Graph.gif wp-emoji-release.js

./noti_files:

```

ix) Sort in Reverse Order : -r option

```

root@kali:~/Desktop# ls -r
Welcome to Ignite.pdf
WebSite.rar
u.zip
usr
tst
The Web Application Hacker's Handbook - Finding and Exploiting Security Flaws.pdf
testing
TestBag.html
TestBag_files
test3
test
shiki3
shiki2
Sample_Service_agreement_1432787607562.pdf
Recording_peak_time_traffic_in_your_town_1389261322890_(1)_1432268319187.pdf
pic
Pattern.html
pattern.html
Pattern_files
pattern_files
noti.html
noti_files
Metasploit - The Penetration Testers Guide - 2011.pdf
math
Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv to hear
install_flash_player_11_linux.x86_64.tar.gz
images.jpeg
Ignite JL Annexure.pdf
IBPS PO 2014: Online Exam Questionlist without using RTI!.html
IBPS PO 2014: Online Exam Questionlist without using RTI!.files
noti
Final-Syllabus-of-Signature-Verification-Training.docx
file2
file1

```

It sorts the file in reverse order. Sorting is done according to the name of the file.

x) Marking the Directories with a Slash (/) Sign : -p option

This is to make identifications of the directories easier. All the directories will have '/' sign after them for easier identification.

```
root@kali:~/Desktop# ls -p
10989419_940756325974653_3425107860827932086_n.jpg
598718_3334799490347_667645853_n.jpg
785964.jpg
a/
abc1/
abc2/
abc3/
abc4/
a.zip
b/
backup.tar
bald-rambo.srt
bill
C_201505221211085567.pdf
Declaration_1430992942546.pdf
dfgfvbhbb.png
DT20142615202_1431401891359.pdf
erl-2003.doc
file1
file2
Final-Syllabus-of-Signature-Verification-Training.docx
hello/
IBPS PO 2014: Online Exam Questionlist without using RTI!.files/
IBPS PO 2014: Online Exam Questionlist without using RTI!.html
Ignite JL Annexure.pdf
images.jpeg
install_flash_player_11_linux.x86_64.tar.gz
Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
math
Metasploit - The Penetration Testers Guide - 2011.pdf
noti_files/
noti.html
pattern_files/
```

xii) Show Only Sub-Directories : -d option

If we want to see only the sub directories of the current directory and not the contents of the sub directories we use -d option.

```
root@kali:~/Desktop# ls -d */
a/ abc1/ abc3/ b/ IBPS PO 2014: Online Exam Questionlist without using RTI!.files/ pattern_files/ pic/ usr/
abc1/ abc2/ abc3/ b/ noti_files/ Pattern_files/ TestBag_files/
```

Above only the sub directories names are show not the content.

xiii) Display File UID and GID Using : -n option

UID-A user ID (**UID**) is a unique positive integer assigned by a Unix-like operating system to each user. Each user is identified to the system by its **UID**, and user names are generally used only as an interface for humans.

It is defined in /etc/passwd.

GID-The group also has a group system number (gid) associated with it. Each group is identified to the system by its GID. It is defined in /etc/group
We can see the UID and GID with the option -n .

Syntax: ls -n

```
root@kali:~/Desktop# ls -n
total 746748
-rw-r--r-- 1 0 0 33319 Jun 5 22:01 10989419_940756325974653_3425107860827932086_n.jpg
-rw-r--r-- 1 0 0 25056 Jan 17 10:37 598718_3334799490347_667645853_n.jpg
-rw-r--r-- 1 0 0 116132 Jan 17 10:37 785964.jpg
drwxr-xr-x 2 0 0 4096 Jun 4 09:20 a
drwxrwxrwx 3 0 0 4096 Jun 4 10:10 abc
drwxr-xr-x 2 0 0 4096 Jun 4 09:14 abc1
drwxr-xr-x 2 0 0 4096 Jun 4 09:14 abc2
drwxr-xr-x 2 0 0 4096 Jun 4 09:14 abc3
drwxrwxrwx 2 0 0 4096 Jun 4 09:17 abc4
-rw-r--r-- 1 0 0 4403948 May 21 07:20 a.zip
drwxr-xr-x 2 0 0 4096 Jun 4 09:21 b
-rw-r--r-- 1 0 0 10240 May 21 20:26 backup.tar
-rw-r--r-- 1 0 0 29442 Jan 24 16:36 bald-rambo.srt
-rwxr-xr-x 1 1002 1014 178 Apr 12 21:43 bill
-rw-r--r-- 1 0 0 1905622 May 28 17:59 C_201505221211085567.pdf
-rw-r--r-- 1 0 0 223035 May 30 22:44 Declaration_1430992942546.pdf
-rw-r--r-- 1 0 0 83866 May 22 18:39 dfgfgvbbhb.png
-rw-r--r-- 1 0 0 247802 May 28 07:18 DT20142615202_1431401891359.pdf
-rw-r--r-- 1 0 0 38912 April 12 19:18 erl-2003.doc
-rw-r--r-- 1 0 0 12 May 24 22:04 file1
-rw-r--r-- 1 0 0 12 May 24 22:04 file2
-rwxrwxrwx 1 0 0 19792 Jan 14 16:50 Final-Syllabus-of-Signature-Verification-Training.docx
drwxrwxrwx 2 0 0 4096 Jun 4 10:09 file3
drwxr-xr-x 2 0 0 4096 Jun 2 08:45 IBPS PO 2014: Online Exam Questionlist without using RTI!.html
-rw-r--r-- 1 0 0 143614 Jun 2 08:45 IBPS PO 2014: Online Exam Questionlist without using RTI!.html
-rw-r--r-- 1 0 0 335262 May 21 14:46 Ignite JL Annexure.pdf
-rw-r--r-- 1 0 0 10361 May 29 18:54 images.jpeg
-rw-r--r-- 1 0 0 7241885 Feb 6 08:30 install_flash_player_11_linux.x86_64.tar.gz
-rw-r----- 1 0 0 725001235 Feb 6 13:31 Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
-rwxr-xr-x 1 0 0 46 Jun 4 06:39 math
-rwxr-xr-x 1 0 0 7237192 Oct 17 2012 Metasploit - The Penetration Testers Guide - 2011.pdf
drwxr-xr-x 3 0 0 4096 Jun 2 08:32 noti_files
-rw-r--r-- 1 0 0 10067 Jun 2 08:32 noti.html
drwxr-xr-x 3 0 0 4096 Jun 2 08:31 pattern_files
```

shows the uid and gid in numeric format instead of names.

xiii) Display Inode number of File or Directory : -i option

Inode-Data structures that contain information about files in Unix file systems that are created when a file system is created. Each file has an **inode** and is identified by an **inode number (i-number)** in the file system where it resides.

To see the inode number of a file or directory use -i option.

Syntax: ls -i

```
root@kali:~/Desktop# ls -i
2237416 10989419_940756325974653_3425107860827932086_n.jpg
2236171 598718_3334799490347_667645853_n.jpg
2236197 785964.jpg
3160899 a
3160898 abc
2759649 abc1
3160891 abc2
3160892 abc3
3160896 abc4
2365907 a.zip
3160900 b
2235736 backup.tar
2236296 bald-rambo.srt
2235764 bill
2365975 C_201505221211085567.pdf
2366072 Declaration_1430992942546.pdf
2235733 dfgfgvbbhb.png
2366107 DT20142615202_1431401891359.pdf
2366035 erl-2003.doc
2235953 file1
2236021 file2
2236150 Final-Syllabus-of-Signature-Verification-Training.docx
3160897 file3
2366184 IBPS PO 2014: Online Exam Questionlist without using RTI!.html
2237241 IBPS PO 2014: Online Exam Questionlist without using RTI!.html
2365900 Ignite JL Annexure.pdf
2236309 images.jpeg
2365883 install_flash_player_11_linux.x86_64.tar.gz
2236374 Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
2235783 math
2235627 Metasploit - The Penetration Testers Guide - 2011.pdf
2366173 noti_files
2237186 noti.html
2366163 pattern_files
```

Above the inode number of all files and directories is displayed before them.

xiv) List particular Directory Information :

We can see the content of a particular directory also rather than seeing all directories. for this we have to specify the directory name after 'ls' command.

Syntax : ls dir_name

```
root@kali:~/Desktop# ls pic/
100.png 107.png 113.png 11.png 15.png 21.png 28.png 34.png 40.png 47.png 53.png 61.png 69.png 75.png 81.png 88.png 94.png math-
101.png 108.png 114.png 120.png 16.png 22.png 29.png 35.png 41.png 48.png 54.png 62.png 6.png 76.png 82.png 89.png 95.png sci
102.png 109.png 115.png 121.png 17.png 23.png 2.png 36.png 42.png 49.png 55.png 63.png 70.png 77.png 83.png 8.png 96.png unit1.txt
103.png 10.png 116.png 122.png 18.png 24.png 30.png 37.png 43.png 4.png 56.png 64.png 71.png 78.png 84.png 90.png 97.png
104.png 110.png 117.png 12.png 19.png 25.png 31.png 38.png 44.png 50.png 58.png 66.png 72.png 79.png 85.png 91.png 98.png
105.png 111.png 118.png 13.png 1.png 26.png 32.png 39.png 45.png 51.png 5.png 67.png 73.png 7.png 86.png 92.png 99.png
106.png 112.png 119.png 14.png 20.png 27.png 33.png 3.png 46.png 52.png 60.png 68.png 74.png 80.png 87.png 93.png 9.png
root@kali:~/Desktop#
```

xv) List the contents of your home directory : ~

option

We can view the content of the home directory from the current directory. For this we do not have to navigate or move to the home directory.

Syntax: ls ~

```
root@kali:~/Desktop# ls ~
Desktop gerix-wifi-cracker-master hello lscomm pass1 sleepy usr
fcfat gerix-wifi-cracker-master.rar install_flash_player_11_linux.x86_64.tar.gz parenthello readme.txt test
root@kali:~/Desktop#
```

'~' is for home directory.

xvi) List the contents of your root directory : /

To view the content of root directory from the current directory only following command is used.

Syntax : ls /

```

root@kali:~/Desktop# ls /
0
bin
boot
dev
etc
example.conf.json
home
initrd.img
lib
lib32
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
Screenshot from 2015-04-12 08:06:31.png
Screenshot from 2015-04-12 08:07:46.png
Screenshot from 2015-04-12 08:07:47 - 1.png
Screenshot from 2015-04-12 08:07:47.png
Screenshot from 2015-04-12 08:07:48.png
Screenshot from 2015-04-12 08:07:49.png
Screenshot from 2015-04-12 09:32:59.png
Screenshot from 2015-04-12 09:36:45.png
Screenshot from 2015-04-12 09:36:46.png
Screenshot from 2015-04-12 09:36:47.png
Screenshot from 2015-04-12 09:36:49.png
Screenshot from 2015-04-12 11:25:38.png
Screenshot from 2015-04-12 11:25:40.png
Screenshot from 2015-04-12 11:29:24.png
Screenshot from 2015-05-21 11:36:06 - 1.png
Screenshot from 2015-05-21 11:36:06.png
Screenshot from 2015-05-21 11:36:23 - 1.png
Screenshot from 2015-05-21 11:37:04.png
Screenshot from 2015-05-21 11:37:11.png
Screenshot from 2015-05-21 12:00:03.png
Screenshot from 2015-05-21 12:00:48 - 1.png
Screenshot from 2015-05-21 12:00:48.png
Screenshot from 2015-05-21 12:01:38.png
Screenshot from 2015-05-21 12:05:29 - 1.png
Screenshot from 2015-05-21 12:05:29.png
Screenshot from 2015-05-21 12:05:30.png
Screenshot from 2015-05-21 12:05:38 - 1.png
Screenshot from 2015-05-21 12:05:38.png
Screenshot from 2015-05-21 12:05:39.png
Screenshot from 2015-05-21 12:05:40.png
Screenshot from 2015-05-21 12:05:41.png
Screenshot from 2015-05-21 12:05:51 - 1.png
Screenshot from 2015-05-21 12:05:51.png
Screenshot from 2015-05-21 12:05:58 - 1.png
Screenshot from 2015-05-21 12:05:58.png
Screenshot from 2015-05-21 12:05:59 - 1.png
Screenshot from 2015-05-21 12:05:59.png
Screenshot from 2015-05-21 12:06:00 - 1.png
Screenshot from 2015-05-21 12:06:00.png
Screenshot from 2015-05-21 12:06:06.png
Screenshot from 2015-05-21 12:06:07.png
Screenshot from 2015-05-21 12:49:55.png
Screenshot from 2015-05-21 12:50:24 - 1.png
Screenshot from 2015-05-21 12:50:24.png
Screenshot from 2015-05-21 12:52:28.png
Screenshot from 2015-05-21 12:52:32.png
Screenshot from 2015-05-21 12:52:33.png
Screenshot from 2015-05-21 12:52:36.png
Screenshot from 2015-05-21 12:54:33 - 1.png
Screenshot from 2015-05-21 12:54:33.png
Screenshot from 2015-05-23 08:01:44.png
Screenshot from 2015-05-23 08:01:47.png
Screenshot from 2015-05-24 11:57:10.png
Screenshot from 2015-05-24 11:57:16.png
Screenshot from 2015-05-24 11:57:29.png
Screenshot from 2015-05-24 11:57:30.png
Screenshot from 2015-05-24 11:57:32 - 1.png
Screenshot from 2015-05-24 11:57:32.png
Screenshot from 2015-05-24 11:57:33 - 1.png
Screenshot from 2015-05-24 11:57:33 - 2.png
Screenshot from 2015-05-24 11:57:33.png
Screenshot from 2015-05-24 11:59:32.png
Screenshot from 2015-05-24 12:00:07.png
Screenshot from 2015-05-24 12:00:12.png
Screenshot from 2015-05-25 20:37:06.png
Screenshot from 2015-05-25 20:37:45 - 1.png
Screenshot from 2015-05-25 20:37:45.png
Screenshot from 2015-05-25 20:54:50 - 1.png
Screenshot from 2015-05-25 20:54:50.png
Screenshot from 2015-05-25 20:54:51 - 1.png
Screenshot from 2015-05-25 20:54:51 - 1.png
Screenshot from 2015-06-04 06:45:49.png
Screenshot from 2015-06-04 06:46:08.png
Screenshot from 2015-06-04 06:46:09.png
Screenshot from 2015-06-04 06:46:11.png
Screenshot from 2015-06-04 07:51:00.png
Screenshot from 2015-06-04 08:18:27.png

```

here we are viewing the root directory from our current directory (Desktop).

xvii) List the contents of the parent directory : ../

To see the content of the parent directory from the current directory we can use following command :

Syntax: ls .. /

```

root@kali:~/Desktop# ls ../
Desktop gerix-wifi-cracker-master    hello
fcat   gerix-wifi-cracker-master.rar install_flash_player_11_linux.x86_64.tar.gz lscomm  pass1  sleepy  usr
root@kali:~/Desktop# 

```

here we have not changed our location to parent directory to see its content.

xviii) List the contents of all subdirectories : */

To see all the content of sub directories of present directory. We use following command.

Syntax: ls * /

```
root@kali:~/Desktop# ls */
a/:
abc/:
3 4 9
abc1/:
abc2/:
abc3/:
abcd/:
b/:
hello/:
5
IBPS PO 2014: Online Exam Questionlist without using RTI!_files/:
a.js                               cloudflare.js   flipkart.india.png
aut optimize_07197a4c169d697a26e42e6e167facdf.js  css.css      ga.js
aut optimize_2018e22f059d7042fe7a4413cf47bfd2.css devicepx-jetpack.js IBPS-PO-2015-Line-Graph.gif
                                         The quieter you become, the more you are able to hear.
noti_files/:
AssessmentServer.gif  homepage_data  homepage.html  jquery-1.js  jquery-ui-1.css  jquery-ui-1.js  mycss.css  stylesheet.html
pattern_files/:
AssessmentServer.gif  homepage_data  homepage.html  jquery-1.js  jquery-ui-1.css  jquery-ui-1.js  mycss.css  stylesheet.html
Pattern_files/:
AssessmentServer.gif  homepage_data  homepage.html  jquery-1.js  jquery-ui-1.css  jquery-ui-1.js  mycss.css  stylesheet.html
```

3) cd : means Change Directory command is one of the most important and most widely used command. 'cd' is the only way to navigate to a directory to check log, execute a program/application/script and for every other task.

Example :

i) Change from current directory to /usr/local.

TO move navigate to other directory or location from current.

Syntax: cd [new location]

```
root@kali:~/Desktop# cd /usr/local  
root@kali:/usr/local#
```

here the current i.e. Desktop has been changed to /usr/local. This way we can navigate to any other directory from our current directory.we have to specify the location after the command "cd".

ii) Change from current directory to /usr/local/lib using absolute path.

Absolute Path:

Alternatively referred to as the **file path** and **full path**, the **absolute path** contains the root directory and all other subdirectories that contain a file or folder.

Syntax : cd [absolute Path]

```
root@kali:/usr/local# cd /usr/local/lib  
root@kali:/usr/local/lib#
```

here our current location is 'local' and we want to move to 'lib' directory. For this we have specified the full path of 'lib' directory i.e '/usr/local/lib' . This is called absolute path.

iii) Change from current working directory to /usr/local/lib using relative path.

Relative Path:

Alternatively referred to as a **partial path** or **non absolute path**, a **relative path** is only a portion of the full path.

Syntax: cd [Relative path]

```
root@kali:/usr/local# cd lib  
root@kali:/usr/local/lib#
```

here to move to the 'lib' directory we have just specified the directory name 'lib' after the 'cd' command rather than giving full path like '/usr/local/lib'.

iv)a) Move one directory back from where you are now.

To move one directory back from current directory following command is uses:
syntax : cd -

```
root@kali:/usr/local# cd lib  
root@kali:/usr/local/lib# cd -  
/usr/local  
root@kali:/usr/local#
```

Here current directory is 'lib' and we have moved one step back i.e.'local'

b) Change Current directory to parent directory.

TO move to the parent directory of the current directory.

Syntax: cd ..

```
root@kali:/usr/local/lib# cd ..  
root@kali:/usr/local#
```

here the parent directory of current directory(lib) is 'local'.

v)Move to last working directory from where we moved

: - switch

To move to the previous directory from where you moved to current directory by following way.

Syntax: cd --

```
root@kali:/usr/local# cd --  
root@kali:~#
```

The current directory is 'local' and we have moved here from root directory.

vi) Move two directory up from where you are now.

To move two directory up from the current directory following command is used.

Syntax: cd ../../

```
root@kali:~/Desktop/abc/l# cd ../../
root@kali:~/Desktop#
```

here current directory is 'l' and directory above 'l' is 'abc' and above up it is 'Desktop'.

Note: The directories are in the hierarchical order i.e.

'root' is at top then is 'Desktop' after it 'abc' and at last is 'l'.

vii) Move to users home directory from anywhere. : ~

To move to the home directory from current directory (anywhere) '~' option is used with 'cd' command.

Syntax: cd ~

```
root@kali:~/Desktop/abc# cd ~
root@kali:~#
```

here from Desktop we have moved to the home directory.

4) `pwd` : '`pwd`' stands for 'Print Working Directory'. As the name states, command '`pwd`' prints the current working directory or simply the directory user is, at present. It prints the current directory name with the complete path starting from root (/). This command is built in shell command and is available on most of the shell - bash, Bourne shell, ksh, zsh, etc.

It gives following information :

- *) Find the full path to the current directory.
- *) Store the full path to the current directory in the shell variable.
- *) Verify the absolute path.
- *) Verify the physical path i.e exclude symbolic links.

It has two options :

-L -logical

use PWD from environment, even if it contains symlinks.

-P -physical

avoid all symlinks.

i) Simple usage of `pwd` :

```
root@kali:~/Desktop# pwd
/root/Desktop
root@kali:~/Desktop# █
```

5) `cp` : "cp" command is used to copy files and directories.

while suing cp command we have to specify the source (file which is to be copied) and destination (where it is to be copied).

Syntax : cp [OPTION] SOURCE DESTINATION

i) Copy a file or directory from source to destination

Syntax : cp source_path destination_path

here initially the 'pic' folder does not have 'math' file. Error is found on running the "ls" command. Then we copy 'math' file into 'pic' folder. now when we run 'ls' command details of 'math' file in 'pic' folder is shown.

```
root@kali:~/Desktop# ls -l pic/math
ls: cannot access pic/math: No such file or directory
root@kali:~/Desktop# cp math pic
root@kali:~/Desktop# ls -l pic/math
-rwxr-xr-x 1 root root 30 May 25 14:10 pic/math
root@kali:~/Desktop#
```

Note: by this way we can copy only one file at time.

ii) A directory (and all its content) can be copied

from source to destination with the recursive option

: -r option

if we wish to copy a folder with more file and sub files in it we can use -r option

Syntax : cp -r source destination

```
root@kali:~/Desktop# ls abc
Acceptance of Joining.pdf      BGC-Proof of Documents.pdf  Medical Certificate.pdf  On Rs 50 stamp paper.pdf
Affidavit_Notorized Undertaking.pdf  CID Forms          On Rs 100 Stamp Paper.pdf  Service Agreement Guidelines.pdf
root@kali:~/Desktop# ls usr
bin lib lib64 share
root@kali:~/Desktop# cp -r abc/ usr/
root@kali:~/Desktop# ls usr
abc bin lib lib64 share
root@kali:~/Desktop# ls usr/abc
Acceptance of Joining.pdf      BGC-Proof of Documents.pdf  Medical Certificate.pdf  On Rs 50 stamp paper.pdf
Affidavit_Notorized Undertaking.pdf  CID Forms          On Rs 100 Stamp Paper.pdf  Service Agreement Guidelines.pdf
root@kali:~/Desktop#
```

Here initially 'usr' folder does not have the 'abc' folder.

Then the 'abc' folder with all its content is copied to the 'usr' folder.

iii) To avoid overwriting the existing file : -n option

if while copying a file from source to destination both location have file with same name but diff. content we want to avoid the over ride condition we can use -n option.

Syntax : cp -n source destination

```
root@kali:~/Desktop# cat > math
hello i am math.i am outside pic folder.root@kali:~/Desktop#
root@kali:~/Desktop# cp -n math pic/math
root@kali:~/Desktop# cd pic
root@kali:~/Desktop/pic# cat math
Hello i am file Math.I am in folder pic.
root@kali:~/Desktop/pic#
```

here the file with name 'math' is copied into folder pic which also has file named 'math' with diff. content in both 'math' file.

On copying the math file(outside of pic folder) into pic folder ,the math file in pic folder is not overwrite.

iv) Confirm before overwriting in a interactive mode :

-i option

if we want to make our command execution more user interactive we can use -i option. it will ask for user choice before executing a action.

Syntax : cp -i source destination

```
root@kali:~/Desktop# cp -i math pic/math
cp: overwrite `pic/math'? y
root@kali:~/Desktop# cat pic/math
hello i am math.i am outside pic folder.root@kali:~/Desktop#
```

Here input from user is taken in form of Y or N .

v) To make a backup of files if copying file has the same name : -p option

we can also make backup of file to be copied.the back will be created at the source.

Syntax: cp --backup source destination

```
root@kali:~/Desktop# ls pic
10.png 14.png 18.png 21.png 25.png 29.png 32.png 36.png 3.png 43.png 47.png 50.png 54.png 5.png 63.png 68.png 7.png      math
11.png 15.png 19.png 22.png 26.png 2.png 33.png 37.png 40.png 44.png 48.png 51.png 55.png 60.png 64.png 69.png 8.png      unit1.txt
12.png 16.png 1.png 23.png 27.png 30.png 34.png 38.png 41.png 45.png 49.png 52.png 56.png 61.png 66.png 6.png 9.png
13.png 17.png 20.png 24.png 28.png 31.png 35.png 39.png 42.png 46.png 4.png 53.png 58.png 62.png 67.png 70.png change.png
root@kali:~/Desktop# cp --backup math pic/math
root@kali:~/Desktop# ls pic
10.png 14.png 18.png 21.png 25.png 29.png 32.png 36.png 3.png 43.png 47.png 50.png 54.png 5.png 63.png 68.png 7.png      math
11.png 15.png 19.png 22.png 26.png 2.png 33.png 37.png 40.png 44.png 48.png 51.png 55.png 60.png 64.png 69.png 8.png      math~
12.png 16.png 1.png 23.png 27.png 30.png 34.png 38.png 41.png 45.png 49.png 52.png 56.png 61.png 66.png 6.png 9.png
13.png 17.png 20.png 24.png 28.png 31.png 35.png 39.png 42.png 46.png 4.png 53.png 58.png 62.png 67.png 70.png change.png
root@kali:~/Desktop#
```

As you can see here that when 'math' is copied from source to destination i.e. in 'pic' folder, a backup of the original file math in the pic folder is made as 'math~' and new file is copied in the 'pic' as usual.

vi) To show copying details in a verbose manner : -v option

By using the -v option we can see what exactly is happening when our command executes.

Syntax : cp -v source destination

```
root@kali:~/Desktop# cp -v math pic/math
`math' -> `pic/math'
root@kali:~/Desktop#
```

6) mv : command is used to move files and directories from one location to another. Apart from moving the files, it can also rename a file or directory.

Example :

i) Rename a File :

Syntax :mv original_name new_name

we use "mv" command to rename a file.we have give the old name and new name after the 'mv' command.

```

root@kali:~/Desktop/pic# ls
10.png 16.png 21.png 27.png 32.png 38.png 43.png 49.png 54.png 61.png 68.png 73.png 79.png 84.png 8.png 95.png math
11.png 17.png 22.png 28.png 33.png 39.png 44.png 4.png 55.png 62.png 69.png 74.png 7.png 85.png 90.png 96.png math-
12.png 18.png 23.png 29.png 34.png 3.png 45.png 50.png 56.png 63.png 6.png 75.png 80.png 86.png 91.png 97.png unit1.txt
13.png 19.png 24.png 2.png 35.png 40.png 46.png 51.png 58.png 64.png 70.png 76.png 81.png 87.png 92.png 98.png
14.png 1.png 25.png 30.png 36.png 41.png 47.png 52.png 5.png 66.png 71.png 77.png 82.png 88.png 93.png 99.png
15.png 20.png 26.png 31.png 37.png 42.png 48.png 53.png 60.png 67.png 72.png 78.png 83.png 89.png 94.png 9.png
root@kali:~/Desktop/pic# mv math sci
root@kali:~/Desktop/pic# ls
10.png 16.png 21.png 27.png 32.png 38.png 43.png 49.png 54.png 61.png 68.png 73.png 79.png 84.png 8.png 95.png math-
11.png 17.png 22.png 28.png 33.png 39.png 44.png 4.png 55.png 62.png 69.png 74.png 7.png 85.png 90.png 96.png sci
12.png 18.png 23.png 29.png 34.png 3.png 45.png 50.png 56.png 63.png 6.png 75.png 80.png 86.png 91.png 97.png unit1.txt
13.png 19.png 24.png 2.png 35.png 40.png 46.png 51.png 58.png 64.png 70.png 76.png 81.png 87.png 92.png 98.png
14.png 1.png 25.png 30.png 36.png 41.png 47.png 52.png 5.png 66.png 71.png 77.png 82.png 88.png 93.png 99.png
15.png 20.png 26.png 31.png 37.png 42.png 48.png 53.png 60.png 67.png 72.png 78.png 83.png 89.png 94.png 9.png
root@kali:~/Desktop/pic# 
```

here file "math" in folder pic is renamed as "sci".

ii) Rename a Directory :

Syntax :mv original_name new_name

we can also rename a directory as same we did a file by giving the old name and then new name after 'mv' command.

```

root@kali:~/Desktop/pic# ls
100.png 15.png 20.png 26.png 31.png 37.png 42.png 48.png 53.png 60.png 67.png 72.png 78.png 83.png 89.png 94.png 9.png
10.png 16.png 21.png 27.png 32.png 38.png 43.png 49.png 54.png 61.png 68.png 73.png 79.png 84.png 8.png 95.png 90.png a
11.png 17.png 22.png 28.png 33.png 39.png 44.png 4.png 55.png 62.png 69.png 74.png 7.png 85.png 90.png 96.png math-
12.png 18.png 23.png 29.png 34.png 3.png 45.png 50.png 56.png 63.png 6.png 75.png 80.png 86.png 91.png 97.png sci
13.png 19.png 24.png 2.png 35.png 40.png 46.png 51.png 58.png 64.png 70.png 76.png 81.png 87.png 92.png 98.png unit1.txt
14.png 1.png 25.png 30.png 36.png 41.png 47.png 52.png 5.png 66.png 71.png 77.png 82.png 88.png 93.png 99.png
root@kali:~/Desktop/pic# mv a b
root@kali:~/Desktop/pic# ls
100.png 15.png 20.png 26.png 31.png 37.png 42.png 48.png 53.png 60.png 67.png 72.png 78.png 83.png 89.png 94.png 9.png
10.png 16.png 21.png 27.png 32.png 38.png 43.png 49.png 54.png 61.png 68.png 73.png 79.png 84.png 8.png 95.png b
11.png 17.png 22.png 28.png 33.png 39.png 44.png 4.png 55.png 62.png 69.png 74.png 7.png 85.png 90.png 96.png math-
12.png 18.png 23.png 29.png 34.png 3.png 45.png 50.png 56.png 63.png 6.png 75.png 80.png 86.png 91.png 97.png sci
13.png 19.png 24.png 2.png 35.png 40.png 46.png 51.png 58.png 64.png 70.png 76.png 81.png 87.png 92.png 98.png unit1.txt
14.png 1.png 25.png 30.png 36.png 41.png 47.png 52.png 5.png 66.png 71.png 77.png 82.png 88.png 93.png 99.png
root@kali:~/Desktop/pic# 
```

iii) Move all the files from a folder

Here the all content of folder 'abc' is moved in folder 'hello'.

To do this we use character "*" (which means all files) in place of file name and then give the path to the folder where we want to move.

```

root@kali:~/Desktop/abc# ls -l
total 4
drwxr-xr-x 2 root root 4096 Jun  4 09:32 1
-rw-r--r-- 1 root root    0 Jun  4 09:49 2
-rw-r--r-- 1 root root    0 Jun  4 09:49 3

root@kali:~/Desktop/abc# mv * ../hello
root@kali:~/Desktop/abc# cd ..
root@kali:~/Desktop# ls -l hello
total 4
drwxr-xr-x 2 root root 4096 Jun  4 09:32 1
-rw-r--r-- 1 root root    0 Jun  4 09:49 2
-rw-r--r-- 1 root root    0 Jun  4 09:49 3 
```

iv) Move only the files that don't exist in the destination directory : -u option

while moving a large number of files it is possible that few files may already exist at destination.so in order to avoid the moving the pre existing file or moving only those file which are not at destination we can use -u option.

```
Syntax : mv -u source destination
```

```
root@kali:~/Desktop/hello# ls -l
total 4
-rw-r--r-- 1 root root 0 Jun 4 09:49 3
drwxr-xr-x 2 root root 4096 Jun 4 09:32 4
-rw-r--r-- 1 root root 0 Jun 4 09:49 5
root@kali:~/Desktop/hello# ls -l ../abc
total 4
-rw-r--r-- 1 root root 0 Jun 4 10:04 2
drwxr-xr-x 2 root root 4096 Jun 4 10:03 4
-rw-r--r-- 1 root root 0 Jun 4 10:04 5
```

```
root@kali:~/Desktop/hello# mv -u * ../abc/
root@kali:~/Desktop/hello# ls -l ../abc
total 4
-rw-r--r-- 1 root root 0 Jun 4 10:04 2
-rw-r--r-- 1 root root 0 Jun 4 09:49 3
drwxr-xr-x 2 root root 4096 Jun 4 09:32 4
-rw-r--r-- 1 root root 0 Jun 4 10:04 5
root@kali:~/Desktop/hello# █
```

here we used '*' wild card character which means moving all files from source but since we were using -u option and most of file were already in destination only those files were moved which does not exist in desitnation. here file "2" was not in destination folder "abc" so only file "2" was moved.

v) To view current action : -v option or verbose.

By using the -v option we can see what exactly is happening when our command executes.

```
root@kali:~/Desktop/abc# mv -v 8 9
`8' -> `9'
root@kali:~/Desktop/abc# █
```

here the output shows that file '8' is renamed as '9'.

7) rm : This command is used to delete files. rm removes each specified file. By default, it does not remove directories.

Example :

i) To remove the file "add" in the current directory :

Syntax : rm filename

To remove a file we just have specify the filename after 'rm' command.

```
root@kali:~/Desktop# ls
598718_3334799490347_667645853_n.jpg
785964.jpg
aaaaaaaaaaaaaaaaaaaaaaa.txt
abc
add
asd.txt
a.zip
backup.tar
bald-rambo.srt
bill
C_201505221211085567.pdf
Declaration_1430992942546.pdf
dfgfvvhbb.png
DT20142615202_1431401891359.pdf
erl-2003.doc
file1
file2
Final-Syllabus-of-Signature-Verification-Training.docx
IBPS PO 2014: Online Exam Questionlist without using RTI!.files
IBPS PO 2014: Online Exam Questionlist without using RTI!.html
Ignite JL Annexure.pdf
images.jpeg
install_flash_player_11_linux.x86_64.tar.gz
Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv
mat1.txt
mat2.txt
mat3.txt
math
Metasploit - The Penetration Testers Guide - 2011.pdf
noti_files
noti.html
pattern_files
Pattern_files
pattern.html
Pattern.html
```

now file 'add' is removed.

i) To delete a directory named "abc" with all its contents : -r option

```
root@kali:~/Desktop# ls abc
Acceptance of Joining.pdf      BGC-Proof of Documents.pdf  Medical Certificate.pdf  On Rs 50 stamp paper.pdf
Affidavit_Notorized Undertaking.pdf  CID Forms          On Rs 100 Stamp Paper.pdf  Service Agreement Guidelines.pdf
root@kali:~/Desktop# rm -r abc
root@kali:~/Desktop# ls abc
ls: cannot access abc: No such file or directory
root@kali:~/Desktop#
```

By default rm command does not remove a directory. To remove a directory with its content we have to specify the -r option. Here directory named 'abc' is removed.

iii) In order to delete a file that is not in the current directory you can specify the full path.

Syntax: rm [full path]/filename.

here we are going to delete the file "change.png" in 'pic' directory. for this we have specified the full path. i.e. "pic/change.png" with rm command rather than just giving file name.

```
root@kali:~/Desktop# ls pic
10.png 15.png 1.png 24.png 29.png 33.png 38.png 42.png 47.png 51.png 56.png 62.png 68.png 72.png 7.png      math~
11.png 16.png 20.png 25.png 2.png 34.png 39.png 43.png 48.png 52.png 58.png 63.png 69.png 73.png 8.png      unit1.txt
12.png 17.png 21.png 26.png 30.png 35.png 3.png 44.png 49.png 53.png 5.png 64.png 6.png 74.png 9.png
13.png 18.png 22.png 27.png 31.png 36.png 40.png 45.png 4.png 54.png 60.png 66.png 70.png 75.png change.png
14.png 19.png 23.png 28.png 32.png 37.png 41.png 46.png 50.png 55.png 61.png 67.png 71.png 76.png math
root@kali:~/Desktop# rm pic/change.png
root@kali:~/Desktop# ls pic
10.png 15.png 1.png 24.png 29.png 33.png 38.png 42.png 47.png 51.png 56.png 62.png 68.png 72.png 7.png      unit1.txt
11.png 16.png 20.png 25.png 2.png 34.png 39.png 43.png 48.png 52.png 58.png 63.png 69.png 73.png 8.png
12.png 17.png 21.png 26.png 30.png 35.png 3.png 44.png 49.png 53.png 5.png 64.png 6.png 74.png 9.png
13.png 18.png 22.png 27.png 31.png 36.png 40.png 45.png 4.png 54.png 60.png 66.png 70.png 75.png math
14.png 19.png 23.png 28.png 32.png 37.png 41.png 46.png 50.png 55.png 61.png 67.png 71.png 76.png math-
root@kali:~/Desktop#
```

iv) You can selectively delete a subset of files using the wildcard- character “*”.

By using wilcard character we can delete all files for a particular extension.

ex. *.txt for all text file

*.jpg for all pics with jpg format.

```
root@kali:~/Desktop# ls *.txt
aaaaaaaaaaaaaaaaaaaaaaa.txt asd.txt mat1.txt mat2.txt mat3.txt po.txt
root@kali:~/Desktop# rm *.txt
root@kali:~/Desktop# ls *.txt
ls: cannot access *.txt: No such file or directory
root@kali:~/Desktop#
```

Removes all files that end with “.txt”.

v) Delete Directory Which has Content (i.e Directory with Files and Sub-directories) : -r option or -R option

Syntax : rm -r dir_name

To delete directory which as content in it. we can use -r or -R option.

```
root@kali:~/Desktop# ls abc
test1
root@kali:~/Desktop# rm abc
rm: cannot remove `abc': Is a directory
root@kali:~/Desktop# rm -r abc
root@kali:~/Desktop# ls abc
ls: cannot access abc: No such file or directory
root@kali:~/Desktop#
```

vi) Deleting a directory interactively : -i option

To make the command execution more user interactive we can use -i option. Here before executing the command, user choice will be asked. Y or N.

Syntax : rm -i filename.

```
root@kali:~/Desktop# ls abc
1 2
root@kali:~/Desktop# rm -i abc/1
rm: remove regular empty file `abc/1'? y
root@kali:~/Desktop# ls abc
2
root@kali:~/Desktop#
```

Depending upon user's choice action is done.

Note: To Delete a directory recursively & interactively (i.e. before deleting each and every sub directory or file ask use choice) use -i and -r option together.

```
root@kali:~/Desktop# rm -ir abc
rm: descend into directory `abc'? y
rm: remove regular empty file `abc/2'? y
rm: remove directory `abc'? y
root@kali:~/Desktop#
```

vii) Avoid error during deleting if file does not exist : -f option

to avoid display of errors we use -f option

i) delete a folder "abc" which does not exits gives us error

```
root@kali:~/Desktop# rm abc
rm: cannot remove `abc': No such file or directory
root@kali:~/Desktop#
```

ii) using -f option the error is not displayed even if file or folder does not exist.

```
root@kali:~/Desktop# rm abc
rm: cannot remove `abc': No such file or directory
root@kali:~/Desktop# rm -f abc
root@kali:~/Desktop#
```

8) mkdir :command is used To make directories.

Example :

i) command is used To make directories.

To create new directory we use 'mkdir' command with the name of the directory

Syntax : mkdir dir_name

```
root@kali:~/Desktop# mkdir abc  
root@kali:~/Desktop#
```

Directory abc is created.

ii) To create multiple directories

we can also create multiple directory at one time.we have specify the directory names after the 'mkdir' command with spaces between the file names.

Syntax : mkdir dir1,dir2,dir3

```
root@kali:~/Desktop# mkdir abc1 abc2 abc3
root@kali:~/Desktop#
```

here directories abc1,abc2 and abc3 are created together.

iii) To create several subdirectories at one time : -p option

we can also create directories inside other directories i.e. we can create the directories in hierarchy way.we just have to specify the directories and their sub directories with 'mkdir' command.

```
root@kali:~/Desktop# ls abc/
root@kali:~/Desktop# mkdir -p abc/dir1/dir2/dir3
root@kali:~/Desktop# ls abc/
dir1
root@kali:~/Desktop# ls abc/dir1/
dir2
root@kali:~/Desktop# ls abc/dir1/dir2/
dir3
root@kali:~/Desktop#
```

here dir1 ,dir2,dir3 are created in directory abc.

iv) To control the permissions of new directories : -m option

When ever a directory is created it will have the default permission.If we want to specify the permission for newly created directory we can use the -m option.

Syntax : mkdir -m permissions directory_name

```
root@kali:~/Desktop# mkdir -m 777 abc
root@kali:~/Desktop# ls -l abc
total 0
root@kali:~/Desktop#
```

Here the new directory abc has the permission for read ,write and execute for all.

v) To show verbose message : -v option

By using the -v option we can see what exactly is happening when our command executes.

```
root@kali:~/Desktop# mkdir -v b
mkdir: created directory `b'
root@kali:~/Desktop#
```

here information is displayed about creation of directory 'b' in second line of output.

9) rmdir : remove empty directories.

Example :

i) To Delete Empty Directories :

Syntax :`rmdir dir_name`

```
root@kali:~/Desktop# ls abc/
root@kali:~/Desktop# rmdir abc
root@kali:~/Desktop# ls abc/
ls: cannot access abc/: No such file or directory
root@kali:~/Desktop#
```

here empty directory "abc" is deleted.

Note: rmdir command will delete the empty directories. i.e directory without any sub-directories or files.

To ensure that you are deleting an empty directory you should use rmdir command. If there is any

files / directories in that directory it will display the following error.

```
root@kali:~/Desktop# ls usr
abc bin lib lib64 share
root@kali:~/Desktop# rmdir usr
rmdir: failed to remove `usr': Directory not empty
root@kali:~/Desktop#
```

ii) To Delete Nested Empty Directories in Linux : -p option

```
root@kali:~/Desktop# ls dir1
dir2
root@kali:~/Desktop# ls dir1/dir2/
DIR3
root@kali:~/Desktop# rmdir -p dir1/dir2/DIR3
root@kali:~/Desktop# ls dir1
ls: cannot access dir1: No such file or directory
root@kali:~/Desktop#
```

Sometimes it is possible that the directory we are trying to delete can have sub directories and those sub directories can have more sub directories. Then we can delete these directories using -p option.

It deletes the inner most directory first, and makes the next level directory empty then it deletes that directory. And it continue doing so.

-----1

O) more :

The Linux "more" command lets you view text files or other output in a scrollable manner. It displays the text one screen full at a time, and lets you scroll backwards and forwards through the text, and even lets you search the text.

```
root@kali:~# more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
mysql:x:101:103:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
colord:x:103:107:colord colour management daemon,,,:/var/lib/colord:/bin/false
usbmux:x:104:46:usbmux daemon,,,:/home/usbmux:/bin/false
miredo:x:105:65534::/var/run/miredo:/bin/false
ntp:x:106:113::/home/ntp:/bin/false
Debian-exim:x:107:114::/var/spool/exim4:/bin/false
arpwatch:x:108:117:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
avahi:x:109:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
beef-xss:x:110:119::/var/lib/beef-xss:/bin/false
dradis:x:111:121::/var/lib/dradis:/bin/false
pulse:x:112:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
speech-dispatcher:x:113:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
haldaemon:x:114:124:Hardware abstraction layer,,,:/var/run/hald:/bin/false
sshd:x:115:65534::/var/run/sshd:/usr/sbin/nologin
snmp:x:116:126::/var/lib/snmp:/bin/false
```

more command with pipe :

For more better and formatted output on screen,we can use 'more' command with combination with other commands too using pipe(|). Pipe helps us to execute two or more commands together.

Syntax: [command to be excuted] | more

```
root@kali:~# ps aux |more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0 10656   816 ?        Ss  12:49  0:00 init [2]
root         2  0.0  0.0     0   0 ?        S   12:49  0:00 [kthreadd]
root         3  0.0  0.0     0   0 ?        S   12:49  0:02 [ksoftirqd/0]
root         5  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/0:0H]
root         7  0.0  0.0     0   0 ?        S   12:49  0:04 [rcu_sched]
root         8  0.0  0.0     0   0 ?        S   12:49  0:00 [rcu_bh]
root         9  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/0]
root        10  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/0]
root        11  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/1]
root        12  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/1]
root        13  0.0  0.0     0   0 ?        S   12:49  0:00 [ksoftirqd/1]
root        15  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/1:0H]
root        16  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/2]
root        17  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/2]
root        18  0.0  0.0     0   0 ?        S   12:49  0:00 [ksoftirqd/2]
root        20  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/2:0H]
root        21  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/3]
root        22  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/3]
root        23  0.0  0.0     0   0 ?        S   12:49  0:00 [ksoftirqd/3]
root        25  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/3:0H]
root        26  0.0  0.0     0   0 ?        S<  12:49  0:00 [khelper]
root        27  0.0  0.0     0   0 ?        S   12:49  0:00 [kdevtmpfs] , the more you are able to hear.
root        28  0.0  0.0     0   0 ?        S<  12:49  0:00 [netns]
root        29  0.0  0.0     0   0 ?        S<  12:49  0:00 [writeback]
root        30  0.0  0.0     0   0 ?        SN  12:49  0:00 [ksmd]
root        31  0.0  0.0     0   0 ?        SN  12:49  0:00 [khugepaged]
root        32  0.0  0.0     0   0 ?        S<  12:49  0:00 [kintegrityd]
root        33  0.0  0.0     0   0 ?        S<  12:49  0:00 [bioset]
root        34  0.0  0.0     0   0 ?        S<  12:49  0:00 [kblockd]
root        36  0.0  0.0     0   0 ?        S   12:49  0:00 [khungtaskd]
root        37  0.0  0.0     0   0 ?        S   12:49  0:00 [kswapd0]
```

```
JUSER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0 10656   816 ?        Ss  12:49  0:00 init [2]
root         2  0.0  0.0     0   0 ?        S   12:49  0:00 [kthreadd]
root         3  0.0  0.0     0   0 ?        S   12:49  0:02 [ksoftirqd/0]
root         5  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/0:0H]
root         7  0.0  0.0     0   0 ?        S   12:49  0:04 [rcu_sched]
root         8  0.0  0.0     0   0 ?        S   12:49  0:00 [rcu_bh]
root         9  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/0]
root        10  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/0]
root        11  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/1]
root        12  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/1]
root        13  0.0  0.0     0   0 ?        S   12:49  0:00 [ksoftirqd/1]
root        15  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/1:0H]
root        16  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/2]
root        17  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/2]
root        18  0.0  0.0     0   0 ?        S   12:49  0:00 [ksoftirqd/2]
root        20  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/2:0H]
root        21  0.0  0.0     0   0 ?        S   12:49  0:00 [watchdog/3]
root        22  0.0  0.0     0   0 ?        S   12:49  0:00 [migration/3]
root        23  0.0  0.0     0   0 ?        S   12:49  0:00 [ksoftirqd/3]
root        25  0.0  0.0     0   0 ?        S<  12:49  0:00 [kworker/3:0H]
root        26  0.0  0.0     0   0 ?        S<  12:49  0:00 [khelper]
root        27  0.0  0.0     0   0 ?        S   12:49  0:00 [kdevtmpfs]
root        28  0.0  0.0     0   0 ?        S<  12:49  0:00 [netns]
root        29  0.0  0.0     0   0 ?        S<  12:49  0:00 [writeback]
root        30  0.0  0.0     0   0 ?        SN  12:49  0:00 [ksmd]
root        31  0.0  0.0     0   0 ?        SN  12:49  0:00 [khugepaged], the more you are able to hear.
root        32  0.0  0.0     0   0 ?        S<  12:49  0:00 [kintegrityd]
root        33  0.0  0.0     0   0 ?        S<  12:49  0:00 [bioset]
root        34  0.0  0.0     0   0 ?        S<  12:49  0:00 [kblockd]
root        36  0.0  0.0     0   0 ?        S   12:49  0:00 [khungtaskd]
root        37  0.0  0.0     0   0 ?        S   12:49  0:00 [kswapd0]
root        38  0.0  0.0     0   0 ?        S   12:49  0:00 [fsnotify_mark]
root        39  0.0  0.0     0   0 ?        S<  12:49  0:00 [crypto]
root        44  0.0  0.0     0   0 ?        S<  12:49  0:00 [kthrotld]
```

--More--

here 'more' command is used with 'ps' command which displays all the running processes in the system and 'more' command helps in displaying the output on screen in scrollable manner.

Switches:

i)-c : to clear screen before displaying text.

If we want to clear the screen before displaying the output.we use -c option. It will clear the screen first and then display the output.

Syntax: more -c filename

```
root@kali:~/Desktop# more -c test
```

here the command is also removed from the screen before displaying the output on the screen. Without using -c option the command execution line would appear on the screen and the output would be after it.

Why Do a Penetration Test?

Companies invest millions of dollars in security programs to protect critical infrastructures, identify chinks in the armor, and prevent serious data breaches. A penetration test is one of the most effective ways to identify systemic weaknesses and deficiencies in these programs. By attempting to circumvent security controls and bypass security mechanisms, a penetration tester is able to identify ways in which a hacker might be able to compromise an organization's security and damage the organization as a whole.

As you read through this book, remember that you're not necessarily targeting one system or multiple systems. Your goal is to show, in a safe and controlled manner, how an attacker might be able to cause serious harm to an organization and impact its ability to, among other things, generate revenue, maintain its reputation, and protect its customers.

Why Metasploit?

Metasploit isn't just a tool; it's an entire framework that provides the infrastructure needed to automate mundane, routine, and complex tasks. This allows you to concentrate on the unique or specialized aspects of penetration testing and on identifying flaws within your information security program.

As you progress through the chapters in this book and establish a well-rounded methodology, you will begin to see the many ways in which Metasploit can be used in your penetration tests. Metasploit allows you to easily build attack vectors to augment its exploits, payloads, encoders, and more in order to create and execute more advanced attacks. At various points in this book we explain several third-party tools—including some written by the authors of this book—that build on the Metasploit Framework. Our goal is to get you comfortable with the Framework, show you some advanced attacks, and ensure that you can apply these techniques responsibly.its over

```
root@kali:~/Desktop#
```

here the command is also removed from the screen before displaying the output on the screen. Without using -c option the command execution line would appear on the screen and the output would be after it.

ii)-n : to Specify how many lines are printed in the screen for a given file.

Here only 5 lines are displayed .

To control the number of lines displayed on the output we use -n option. here 'n' is replaced by the number of lines we want to display.

Syntax : more -n filename

```
root@kali:~/Desktop# more -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
--More-- (5%)
```

Here only 5 lines from the file are displayed on the screen.

iii)+n : to Starts up the file from the given number.

We can also specify the particular line from where we want to display the output. for this +n option is used where 'n' is replaced by the line number from where we want to display the output.

Syntax : more +n filename

```
root@kali:~/Desktop# more +10 test
targeting one system or multiple systems. Your goal is to show, in a safe and
controlled manner, how an attacker might be able to cause serious harm to
an organization and impact its ability to, among other things, generate reve-
nue, maintain its reputation, and protect its customers.
Why Metasploit?
Metasploit isn't just a tool; it's an entire framework that provides the infra-
structure needed to automate mundane, routine, and complex tasks. This
allows you to concentrate on the unique or specialized aspects of penetration
testing and on identifying flaws within your information security program.
As you progress through the chapters in this book and establish a well-
rounded methodology, you will begin to see the many ways in which Meta-
sploit can be used in your penetration tests. Metasploit allows you to easily
build attack vectors to augment its exploits, payloads, encoders, and more
in order to create and execute more advanced attacks. At various points in
this book we explain several third-party tools—including some written by the
authors of this book—that build on the Metasploit Framework. Our goal is to
get you comfortable with the Framework, show you some advanced attacks,
and ensure that you can apply these techniques responsibly.its over
root@kali:~/Desktop#
```

here file content is displayed from line 10 to the end of file.

Press Enter to display the file line by line.

11) less command :

less" command is used to view files instead of opening the file. This post describes "less" command used in Linux along with usage.

Syntax:

```
less [options] file_name
```

- * Less is a program similar to more but which allows backward movement in the file as well as forward movement.
- * less does not have to read the entire input file before starting, so with large input files it starts up faster than text editors like vi.
- * Less uses termcap (or terminfo on some systems), so it can run on a variety of terminals.

i) To open the file

Syntax : less [filename]

It can be used to open a pre existing file. It can not create a new file.

```
root@kali:~/Desktop# less /etc/passwd
```



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
mysql:x:101:103:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
colord:x:103:107:colord colour management daemon,,,:/var/lib/colord:/bin/false
usbmux:x:104:46:usbmux daemon,,,:/home/usbmux:/bin/false
miredo:x:105:65534::/var/run/miredo:/bin/false
ntp:x:106:113::/home/ntp:/bin/false
Debian-exim:x:107:114::/var/spool/exim4:/bin/false
arpwatch:x:108:117:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh  The quieter you become, the more you are a
avahi:x:109:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
beef-xss:x:110:119::/var/lib/beef-xss:/bin/false
dradis:x:111:121::/var/lib/dradis:/bin/false
pulse:x:112:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
speech-dispatcher:x:113:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
haldaemon:x:114:124:Hardware abstraction layer,,,:/var/run/hald:/bin/false
sshd:x:115:65534::/var/run/sshd:/usr/sbin/nologin
snmp:x:116:126::/var/lib/snmp:/bin/false
:|
```

In order to quit from the prompt press "q" from the keyboard.

ii) To Clear screen before displaying : -c option

If we want to clear the screen before displaying the output.we use -c option. It will clear the screen first and then display the output.

Syntax : less -c [filename]

```
root@kali:~/Desktop# less -c test
```

Why Do a Penetration Test?

Companies invest millions of dollars in security programs to protect critical infrastructures, identify chinks in the armor, and prevent serious data breaches. A penetration test is one of the most effective ways to identify systemic weaknesses and deficiencies in these programs. By attempting to circumvent security controls and bypass security mechanisms, a penetration tester is able to identify ways in which a hacker might be able to compromise an organization's security and damage the organization as a whole.

As you read through this book, remember that you're not necessarily targeting one system or multiple systems. Your goal is to show, in a safe and controlled manner, how an attacker might be able to cause serious harm to an organization and impact its ability to, among other things, generate revenue, maintain its reputation, and protect its customers.

Why Metasploit?

Metasploit isn't just a tool; it's an entire framework that provides the infrastructure needed to automate mundane, routine, and complex tasks. This allows you to concentrate on the unique or specialized aspects of penetration testing and on identifying flaws within your information security program.

As you progress through the chapters in this book and establish a well-rounded methodology, you will begin to see the many ways in which Metasploit can be used in your penetration tests. Metasploit allows you to easily build attack vectors to augment its exploits, payloads, encoders, and more in order to create and execute more advanced attacks. At various points in this book we explain several third-party tools—including some written by the authors of this book—that build on the Metasploit Framework. Our goal is to get you comfortable with the Framework, show you some advanced attacks, and ensure that you can apply these techniques responsibly.its over

~
~
~
~
~
~
~

test (END)

It clears the screen and shown only the contents of the file.

here the command is also removed from the screen before displaying the output on the screen. Without using -c option the command execution line would appear on the screen and the output would be after it.

iii) To see large outputs one-screen-full at a time(using less with pipe)

We can see the output on screen full at a time by using the less command with other commands using pipe.

Syntax : [command to be executed] | less

```
root@kali:~/Desktop# ps aux | less
```

| USER | PID | %CPU | %MEM | VSZ | RSS | TTY | STAT | START | TIME | COMMAND |
|------|-----|------|------|-------|-----|-----|------|-------|------|-----------------|
| root | 1 | 0.0 | 0.0 | 10656 | 816 | ? | Ss | 12:49 | 0:00 | init [2] |
| root | 2 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [kthreadd] |
| root | 3 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:04 | [ksoftirqd/0] |
| root | 5 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [kworker/0:0H] |
| root | 7 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:06 | [rcu_sched] |
| root | 8 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [rcu_bh] |
| root | 9 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [migration/0] |
| root | 10 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [watchdog/0] |
| root | 11 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [watchdog/1] |
| root | 12 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [migration/1] |
| root | 13 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [ksoftirqd/1] |
| root | 15 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [kworker/1:0H] |
| root | 16 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [watchdog/2] |
| root | 17 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [migration/2] |
| root | 18 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [ksoftirqd/2] |
| root | 20 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [kworker/2:0H] |
| root | 21 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [watchdog/3] |
| root | 22 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [migration/3] |
| root | 23 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [ksoftirqd/3] |
| root | 25 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [kworker/3:0H] |
| root | 26 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [khelper] |
| root | 27 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [kdevtmpfs] |
| root | 28 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [netns] |
| root | 29 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [writeback] |
| root | 30 | 0.0 | 0.0 | 0 | 0 | ? | SN | 12:49 | 0:00 | [ksmd] |
| root | 31 | 0.0 | 0.0 | 0 | 0 | ? | SN | 12:49 | 0:00 | [khugepaged] |
| root | 32 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [kintegrityd] |
| root | 33 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [bioset] |
| root | 34 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [kblockd] |
| root | 36 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [khungtaskd] |
| root | 37 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [kswapd0] |
| root | 38 | 0.0 | 0.0 | 0 | 0 | ? | S | 12:49 | 0:00 | [fsnotify_mark] |
| root | 39 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [crypto] |
| root | 44 | 0.0 | 0.0 | 0 | 0 | ? | S< | 12:49 | 0:00 | [kthrotld] |
| | : | | | | | | | | | |

Press q to exit the output screen and go to terminal mode.

here output is displayed on the full screen one at a time.

iv) To Starts up the file from the given number : +n

option

We can also specify the particular line from where we want to display the output. for this +n option is used where 'n' is replaced by the line number from where we want to display the output.

Syntax: less +n filename

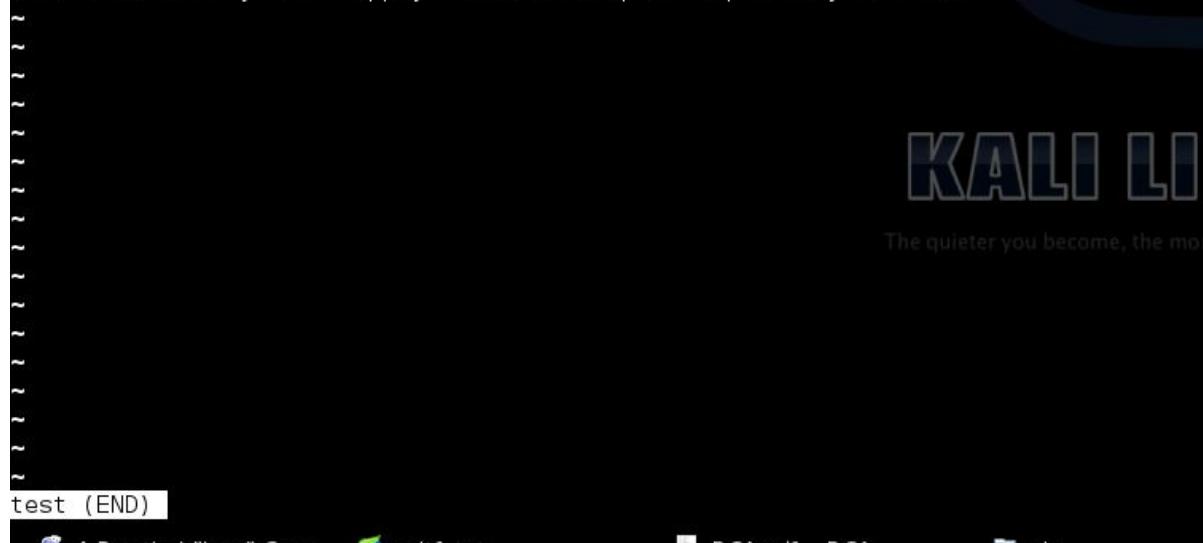
```
root@kali:~/Desktop# less +10 test
```

targeting one system or multiple systems. Your goal is to show, in a safe and controlled manner, how an attacker might be able to cause serious harm to an organization and impact its ability to, among other things, generate revenue, maintain its reputation, and protect its customers.

Why Metasploit?

Metasploit isn't just a tool; it's an entire framework that provides the infrastructure and automation needed to perform penetration testing. This

Metasploit isn't just a tool; it's an entire framework that provides the infrastructure needed to automate mundane, routine, and complex tasks. This allows you to concentrate on the unique or specialized aspects of penetration testing and on identifying flaws within your information security program. As you progress through the chapters in this book and establish a well-rounded methodology, you will begin to see the many ways in which Metasploit can be used in your penetration tests. Metasploit allows you to easily build attack vectors to augment its exploits, payloads, encoders, and more in order to create and execute more advanced attacks. At various points in this book we explain several third-party tools—including some written by the authors of this book—that build on the Metasploit Framework. Our goal is to get you comfortable with the Framework, show you some advanced attacks, and ensure that you can apply these techniques responsibly.



here the output is from the 10 line of file to the end of file.

Navigational Shortcuts of less command :

Forward Search Navigation

- * / - search for a pattern which will take you to the next occurrence.
 - * n - for next match in forward
 - * N - for previous match in backward

Backward Search Navigation

- * ? - search for a pattern which will take you to the previous occurrence.
 - * n - for next match in forward
 - * N - for previous match in backward

Line Navigation

- * j - navigate forward by one line
 - * k - navigate backward by one line

Screen Navigation

- * `CTRL+D` - forward half window
- * `CTRL+U` - backward half window
- * `CTRL+F` - forward one window
- * `CTRL+B` - backward one window

Count numbers as options

- * `5j` - 5 lines forward.
 - * `10k` - 10 lines backward.
-

12) file :

The `file` command tests each argument in an attempt to classify it. There are three sets of tests, performed in this order: filesystem tests, magic tests, and language tests. The first test that succeeds causes the file type to be printed.

The `file` command attempts to classify each filesystem object (i.e., file, directory or link) that is provided to it as an argument (i.e., input). Thus, it can usually provide immediate information as to whether some specified object is, for example, a GIF89a image file, a directory, a GNU tar archive, ASCII English text, a symbolic link, an HTML document, an empty file, bzip2 compressed data, an ELF 32-bit LSB executable, etc.

`file` accomplishes this by probing each object with three types of tests until one succeeds. The first is a filesystem test, which uses the `stat` system call to obtain information from the object's inode (which contains information about a file). A system call is a request in a Unix-like operating system for a service performed by the kernel (i.e., the core of the operating system).

The second test checks to see if there is a magic number, which is a number embedded at or near the beginning of many types of files that indicates the file format (i.e., the type of file).

In the event that the first two tests fail to determine the type of a file, language tests are employed to determine if it is plain text (i.e., composed entirely of human-readable characters), and, if so, what type of plain text, such as HTML (hypertext markup language) or source code (i.e., the original version of a program as written by a human). In this situation, `file` also attempts to determine the natural language (e.g., English, Turkish or Japanese) that is used in the file.

Switches :

i)-v : to get the information about the version of file that is installed.

```
root@kali:~/Desktop# file -v
file-5.11
magic file from /etc/magic:/usr/share/misc/magic
root@kali:~/Desktop#
```

ii) -b : option tells file to not prepend filenames to output lines, which can be useful when compiling statistics about file types. without -b option

```
root@kali:~/Desktop# file test  
test: C source, UTF-8 Unicode text  
root@kali:~/Desktop#
```

file name in starting

with -b option

```
root@kali:~/Desktop# file -b test  
C source, UTF-8 Unicode text  
root@kali:~/Desktop#
```

file name not added.

iii) -k : option tells file to not stop at the first successful test, but to keep going.

The file command attempts to classify each filesystem object (i.e., file, directory or link) that is provided to it as an argument (i.e., input). Thus, it can usually provide immediate information as to whether some specified object is, for example, a GIF89a image file, a directory, a GNU tar archive, ASCII English text, a symbolic link, an HTML document, an empty file, bzip2 compressed data, an ELF 32-bit LSB executable, etc.

file accomplishes this by probing each object with three types of tests until one succeeds. The first is a filesystem test, which uses the stat system call to obtain information from the object's inode (which contains information about a file). A system call is a request in a Unix-like operating system for a service performed by the kernel (i.e., the core of the operating system).

The second test checks to see if there is a magic number, which is a number embedded at or near the beginning of many types of files that indicates the file format (i.e., the type of file).

In the event that the first two tests fail to determine the type of a file, language tests are employed to determine if it is plain text (i.e., composed

entirely of human-readable characters), and, if so, what type of plain text, such as HTML (hypertext markup language) or source code (i.e., the original version of a program as written by a human). In this situation, file also attempts to determine the natural language (e.g., English, Turkish or Japanese) that is used in the file.

Syntax : file [option(s)] object_name(s)

file has several options, but it is most commonly used without any of them. For example, information about a file named file1 that is located in the in the current directory (i.e., the directory in which the user is currently working) could be obtained by merely typing the following and pressing the RETURN key:

```
file file1
```

```
root@kali:~/Desktop# file test
test: C source, UTF-8 Unicode text
root@kali:~/Desktop#
```

Information about the types of all of the files in the current directory can be obtained by using the star wildcard to represent every object in that directory as follows:

```
file *
```

```
root@kali:~/Desktop# file *
598718_3334799490347_667645853_n.jpg: JPEG image data, JFIF standard 1.01
785964.jpg: JPEG image data, JFIF standard 1.01
aaaaaaaaaaaaaaaaaaaaaaa.txt: ASCII text, with very long lines
abc: directory
add: ASCII text
asd.txt: ASCII text, with CRLF line terminators
a.zip: Zip archive data, at least v1.0 to extract
backup.tar: POSIX tar archive (GNU)
bald-rambo.srt: ASCII text, with CRLF line terminators
bill: awk script, ASCII text
dfgfvbhbhbb.png: PNG image data, 1366 x 768, 8-bit/color RGB, non-interlaced
er1-2003.doc: Composite Document File V2 Document, Little Endian, Os: Windows, Version 6.2, Code page: 1252, Author: Aislin, Template: Normal, Last Saved By: Aislin, Revision Number: 2, Name of Creating Application: Microsoft Office Word, Create Time/Date: Sun Apr 12 05:14:00 2015, Last Saved Time/Date: Sun Apr 12 05:14:00 2015, Number of Pages: 1, Number of Words: 0
, Number of Characters: 1, Security: 0
Final-Syllabus-of-Signature-Verification-Training.docx: Microsoft Word 2007+
Ignite JL Annexure.pdf: PDF document, version 1.5
images.jpeg: JPEG image data, JFIF standard 1.01
install_flash_player_11_linux.x86_64.tar.gz: gzip compressed data, was "libflashplayer.so.tar", from Unix, Last modified: Wed Jan 28 21:20:16 2015
Jahreid 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv: EBML file, creator matroska
math: ASCII text
Metasploit - The Penetration Testers Guide - 2011.pdf: PDF document, version 1.6
metasploit: directory
pic: awk script, ASCII text
prinout: PDF document, version 1.3
Recording_peak_time_traffic_in_your_town_1389261322890_(1)_1432268319187.pdf: ASCII text
shiki2: awk script, ASCII text
shiki3: C source, UTF-8 Unicode text
test: awk script, UTF-8 Unicode text
test3: ASCII text
test5: ASCII text
```

Likewise, information about all of the files in another directory can be obtained by using that directory as an argument and following it immediately

by a forward slash and the star wildcard. For example, the following classifies all of the objects in the /boot directory:

```
file /boot/*
```

```
root@kali:~/Desktop# file /boot/*
/boot/config-3.14-kalil-amd64:      ASCII text
/boot/extlinux:
/boot/grub:
/boot/initrd.img-3.14-kalil-amd64: gzip compressed data, from Unix, last modified: Tue Nov 11 10:48:15 2014
/boot/lost+found:
/boot/System.map-3.14-kalil-amd64: ASCII text
/boot/vmlinuz-3.14-kalil-amd64:    Linux kernel x86 boot executable bzImage, version 3.14-kalil-amd64 (debian-kernel@lists.debian.org) #1 SMP Debian,
R0-rootFS, swap_dev 0x2, Normal VGA
root@kali:~/Desktop#
```

The square brackets wildcard can be used together with the star wildcard to show the file types for only those objects whose names begin with specified letters or with a specified range of letters. For example, the following would show only those objects in the current directory whose names begin with letters a through t:

file [a-t] *

```
root@kali:~/Desktop# file [a-t]*  
aaaaaaaaaaaaaaaaaaaaaaa.txt: ASCII text, with very long lines  
abc: directory  
add: ASCII text  
asd.txt: ASCII text, with CRLF line terminators  
a.zip: Zip archive data, at least v1.0 to extract  
bill: POSIX tar archive (GNU)  
dfgfgvbbhb.png: ASCII text, with CRLF line terminators  
aced: awk script, ASCII text  
erl-2003.doc: PNG image data, 1366 x 768, 8-bit/color RGB, non-interlaced  
Final-Syllabus-of-Signature-Verification-Training.docx: Composite Document File V2 Document, Little Endian, Os: Windows, Version 6.2, Code page: 1252, Author: Aislin, Template: Normal, Last Saved By: Aislin, Revision Number: 2, Name of Creating Application: Microsoft Office Word, Create Time/Date: Sun Apr 12 05:14:00 2015, Last Saved Time/Date: Sun Apr 12 05:14:00 2015, Number of Pages: 1, Number of Words: 0  
Final-Syllabus-of-Signature-Verification-Training.docx: , Number of Characters: 1, Security: 0  
Final-Syllabus-of-Signature-Verification-Training.docx: Ignite JL Annexure.pdf:  
Final-Syllabus-of-Signature-Verification-Training.docx: images.jpeg:  
Final-Syllabus-of-Signature-Verification-Training.docx: install_flash_player_11_linux.x86_64.tar.gz:  
Final-Syllabus-of-Signature-Verification-Training.docx: , Unix, last modified: Wed Jan 28 21:20:16 2015  
Final-Syllabus-of-Signature-Verification-Training.docx: Jarhead 2 Field of Fire (2014).BRRip.720p.[Dual-Audio].[Hindi+Eng].700mb.x264.LoneWolf666.mkv: EBM file, creator matroska  
Final-Syllabus-of-Signature-Verification-Training.docx: math:  
Final-Syllabus-of-Signature-Verification-Training.docx: Metasploit - The Penetration Testers Guide - 2011.pdf:  
Final-Syllabus-of-Signature-Verification-Training.docx: pic:  
Final-Syllabus-of-Signature-Verification-Training.docx: printout:  
Recording_peak_time_traffic_in_your_town_1389261322890_(1)_1432268319187.pdf: The quieter you become, the more you are able to hear.  
shiki2:  
shiki3:  
test:  
test3:  
test5:  
testing:  
tst:  
root@kali:~/Desktop#
```

file comparison commands in linux :

1) cmp : The cmp (compare) command compares two files, character (byte) by character, and provides the location of the differences to the screen. Used with no options, cmp will only provide the location of the first discrepancy in the form of "char, line". However, there are options that will provide you with more information.

The cmp command is the best command for simply telling you if two files are different, but unless you want to count each character number, it's not a very good tool for finding the differences.

Example :

following two files are used as sample for comparison.

*)mat1.txt

my name is kali linux.

*)mat2.txt

my name is redhat.

i) Compare file 1 and file without any option :

```
root@kali:~/Desktop# cmp mat1.txt mat2.txt
mat1.txt mat2.txt differ: byte 12, line 1
root@kali:~/Desktop#
```

ii) Skip same number of initial bytes from both input files : - i option

```
root@kali:~/Desktop# cmp -i 3 mat1.txt mat2.txt
mat1.txt mat2.txt differ: byte 9, line 1
root@kali:~/Desktop#
```

we see that the initial 3 bytes were skipped

iii) Display all bytes that differ in both files : -l option

```
root@kali:~/Desktop# cmp -l mat1.txt mat2.txt
12 153 162
13 141 145
14 154 144
15 151 150
16 40 141
17 154 164
18 151 56
19 156 12
cmp: EOF on mat2.txt
root@kali:~/Desktop#
```

2) comm : The comm command can tell you what information is common to two lists and what information appears uniquely in one list or the other.

Example :

following two files are used as sample for comparison.

*file1

abc

def

ghi

```
*file2  
abc  
ghi  
klm  
  
note :  
*)First column displays the lines unique in file1  
*)Second column displays the lines unique in file2  
*)Third column displays the lines that are common in both the files.
```

i) Simple Command Usage

```
root@kali:~/Desktop# comm file1 file2  
      abc  
def  
      ghi  
    klm  
root@kali:~/Desktop# █
```

ii) Suppress a particular column which you don't wish to see .

Syntax : comm -[column no.] file1 file2
here the specified column will not appear in result.

```
root@kali:~/Desktop# comm -2 file1 file2  
      abc  
def  
      ghi  
root@kali:~/Desktop# █
```

Column 2 is suppressed.

3) diff command :

Example:
following two files are used as sample for comparison.

```
*file1  
abc  
def  
ghi
```

```
*file2
```

```
abc
```

```
ghi
```

```
klm
```

i) Simply compare two files

Syntax : diff file1 file2

```
root@kali:~/Desktop# diff file1 file2
2d1
< def
3a3
> klm
root@kali:~/Desktop#
```

Lines from first file are preceded by a less than symbol (<) and lines from second file by a greater than symbol (>). A dashed line (-) is used to separate output from the two files. The letters can be used to convert file1 into file2:

- c Replace lines from file1 with those from file2.
- d Delete lines from file1.
- a Add lines from file2 to file1.

ii) To compare two files while ignoring differences

in the amount of white space, enter: -w option

```
root@kali:~/Desktop# diff -w file1 file2
2d1
< def
3a3
> klm
root@kali:~/Desktop#
```

iii) To compare two different directories

Syntax : diff dir1 dir2

```
root@kali:~/Desktop# diff pic abc
Only in pic: 10.png
Only in pic: 11.png
Only in pic: 12.png
Only in pic: 13.png
Only in pic: 14.png
Only in pic: 15.png
Only in pic: 16.png
Only in pic: 17.png
Only in pic: 18.png
Only in pic: 19.png
Only in pic: 1.png
Only in pic: 20.png
Only in pic: 21.png
Only in pic: 22.png
Only in pic: 23.png
Only in pic: 24.png
Only in pic: 25.png
Only in pic: 26.png
Only in pic: 27.png
Only in pic: 28.png
Only in pic: 29.png
Only in pic: 2.png
Only in pic: 30.png
Only in pic: 31.png
Only in pic: 32.png
Only in pic: 33.png
Only in pic: 34.png
Only in pic: 35.png
Only in pic: 36.png
Only in pic: 37.png
Only in pic: 38.png
Only in pic: 39.png
```

iv) Ignore Case while comparing: -i option

```
root@kali:~/Desktop# cat file
a
B
c
D
root@kali:~/Desktop# cat file3
a
b
e
d
root@kali:~/Desktop# diff -i file file3
3c3
< c
---
> e
root@kali:~/Desktop# █
```

v) Ignore Repeated Blanks : -b option

```
root@kali:~/Desktop# diff -b file1 file2
2d1
< def
3a3
> klm
root@kali:~/Desktop#
```

vi) Output related to file1 is

preceded by stars (****) and file2 by dashes (-). Differences are separated by a long row of stars (*****).

In output the following symbols are used:

- Indicates corresponding lines in the two files that differ.
- + Indicates lines that exist in file2 but not file1.
- Indicates lines that exist in file1 but not file2.

4) sdiff : Compares two files and displays the differences in a side-by-side format.

Example :

i) To print a comparison of two files :

Syntax : sdiff file1 file2

```
root@kali:~/Desktop# sdiff file1 file2
abc
def
ghi
                                abc
<
ghi
> klm
root@kali:~/Desktop#
```

ii) To display only the lines that differ :

```
root@kali:~/Desktop# sdiff -s -w 80 file1 file2
def
                                <
                                > klm
root@kali:~/Desktop#
```

The sdiff command displays the differences. The -w 80 flag and variable sets the page width to 80 columns. The -s flag indicates lines that are identical in both files will not be displayed.

Difference between diff, cmp and comm :

1) cmp :

- *) Compares two files byte by byte and displays the first miss match diff - tells the changes to be made to make the files identical
- *) cmp will list the line and column number that are different between two files.
- *) 'cmp' is used to find the difference between files
- *) we can use directory name in diff, but not in cmp
- *) It can deal with both binary and ASCII file comparisons.

2) diff :

- *) To display all the differences between the files diff command is used. diff compares line by line.
- *) diff gives the text of filename2 which is different from filename1.
- *) diff will list the different files and sub-directories present between two directories.
- *) 'diff' is used to find the difference between directories.
- *) It deals only with ASCII files.

3) comm :

- *) files needs to be sorted where as it is not necessary for diff and cmp.

*) It's output has 3 columns : The first column containing the lines unique to the first file, the second column being the lines common to both files, the third column being the lines unique to the second file.

Command related to disk :

Warning: Don't delete, modify, or add partition, if you don't know what you are doing. You will lose your data!

1) fdisk command :

Using fdisk you can create a new partition, delete an existing partition, or change existing partition.

using fdisk we can create a maximum of four primary partition, and any number of logical partitions, based on the size of the disk. Any single partition requires a minimum size of 40MB.

i) View All Existing Disk Partitions Using fdisk : -l option

To see the all existing Disk partition in our system along with other details we use -l option. It gives all the information about all the available disk partitions.

Syntax: fdisk -l

```
root@kali:~/Desktop# fdisk -l

Disk /dev/sda: 500.1 GB, 500107862016 bytes
255 heads, 63 sectors/track, 60801 cylinders, total 976773168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x8balac96

      Device Boot   Start     End   Blocks  Id  System
/dev/sda1  *    2048  206847  102400    7  HPFS/NTFS/exFAT
/dev/sda2        206848 209932287 104862720    7  HPFS/NTFS/exFAT
/dev/sda4        209934334 976771071 383418369    f  W95 Ext'd (LBA)
/dev/sda5        465545216 721156095 127805440    7  HPFS/NTFS/exFAT
/dev/sda6        721158144 976771071 127806464    7  HPFS/NTFS/exFAT
/dev/sda7        209934336 453877759 121971712   83  Linux
/dev/sda8        453879808 461690879   3905536   82  Linux swap / Solaris
/dev/sda9        461692928 465530879   1918976   83  Linux

Partition table entries are not in disk order
root@kali:~/Desktop#
```

All the disk partition are displayed with all the related information such as size ,name,id etc.

ii) View Partitions of a Specific Hard Disk : -l [hard disk name]

Syntax : fdisk -l [drive name]

```
root@kali:~/Desktop# fdisk -l /dev/sda1

Disk /dev/sda1: 104 MB, 104857600 bytes
255 heads, 63 sectors/track, 12 cylinders, total 204800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x73736572

This doesn't look like a partition table
Probably you selected the wrong device.

      Device Boot   Start     End   Blocks Id System
/dev/sda1p1    1920221984 3736432267 908105142 72 Unknown
/dev/sda1p2    ? 1936028192 3889681299 976826554 6c Unknown
/dev/sda1p3    ?          0          0          0    0 Empty
/dev/sda1p4    27722122 27722568        223+  0 Empty
root@kali:~/Desktop# █
```

iii)View all fdisk Commands :

fdisk command has a lot of sub commands or option related to a particular drive. With the help of these commands it helps us to perform action on that particular drive.

Steps:

First execute the fdisk command with drive name.

Then press m and all list of commands for drive will come

Syntax: fdisk [drive name]

```
root@kali:~/Desktop# fdisk /dev/sda1

Command (m for help): m
Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)

Command (m for help):
```

Using list of commands :

p -> to print all the partition of the drive

To see the all the partition of that drive use p option i.e. press 'p'.

```
a toggle a bootable flag
b edit bsd disklabel
c toggle the dos compatibility flag
d delete a partition
l list known partition types
m print this menu
n add a new partition
o create a new empty DOS partition table
p print the partition table
q quit without saving changes
s create a new empty Sun disklabel
t change a partition's system id
u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)
```

```
Command (m for help): p
```

```
Disk /dev/sda1: 104 MB, 104857600 bytes
255 heads, 63 sectors/track, 12 cylinders, total 204800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x73736572
```

```
This doesn't look like a partition table
Probably you selected the wrong device.
```

| Device | Boot | Start | End | Blocks | Id | System |
|-------------|------|------------|------------|-----------|----|---------|
| /dev/sda1p1 | | 1920221984 | 3736432267 | 908105142 | 72 | Unknown |
| /dev/sda1p2 | ? | 1936028192 | 3889681299 | 976826554 | 6c | Unknown |
| /dev/sda1p3 | ? | 0 | 0 | 0 | 0 | Empty |
| /dev/sda1p4 | | 27722122 | 27722568 | 223+ | 0 | Empty |

```
Command (m for help): █
```

above all the partitions of drive '/dev/sda1' is displayed.

Note : we do not have to write the full command.

d -> to delete a partition

it will give the total no. of partition on that drive. give the drive no. you want to delete.

```
b edit bsd disklabel
c toggle the dos compatibility flag
d delete a partition
l list known partition types
m print this menu
n add a new partition
o create a new empty DOS partition table
p print the partition table
q quit without saving changes
s create a new empty Sun disklabel
t change a partition's system id
u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)
```

```
Command (m for help): p
```

```
Disk /dev/sdal: 104 MB, 104857600 bytes
255 heads, 63 sectors/track, 12 cylinders, total 204800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x73736572
```

This doesn't look like a partition table
Probably you selected the wrong device.

The quieter yo

| Device | Boot | Start | End | Blocks | Id | System |
|-------------|------|------------|------------|-----------|----|---------|
| /dev/sdalp1 | | 1920221984 | 3736432267 | 908105142 | 72 | Unknown |
| /dev/sdalp2 | ? | 1936028192 | 3889681299 | 976826554 | 6c | Unknown |
| /dev/sdalp3 | ? | 0 | 0 | 0 | 0 | Empty |
| /dev/sdalp4 | | 27722122 | 27722568 | 223+ | 0 | Empty |

```
Command (m for help): d
Partition number (1-4): 4
```

n -> to create a new partition.

You have to select the the type of partition whether it is primary or extended.

```
root@kali:~/Desktop# fdisk /dev/sda1

Command (m for help): m
Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)

Command (m for help): n
Partition type:
  p  primary (2 primary, 0 extended, 2 free)
  e  extended
Select (default p): p
All primary partitions have been defined already!

Command (m for help):
```

iv) View the Size of an existing Partition: -s

TO see the size of an particular existing partition we use -s option with name of that partition.

Syntax : fdisk -s [partition]

```
root@kali:~/Desktop# fdisk -s /dev/sda1
102400
root@kali:~/Desktop#
```

here the size of partition '/dev/sda1' is displayed.

v) Fix Partition Table Order : Expert Command f option

When you delete a logical partition, and recreate it again, you might see the "partition out of order" issue. i.e "Partition table entries are not in disk order" error message.

TO remove this error message we execute the expert command f. It comes under the -x command of the list.

```
root@kali:~/Desktop# fdisk /dev/sdal

Command (m for help): m
Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)

Command (m for help): x
Expert command (m for help): f
```

2) df :

The 'df' command stand for "disk filesystem", it is used to get full summary of available and used disk space usage of file system on Linux system.

Using '-h' parameter with (df -h) will shows the file system disk space statistics in "human readable" format, means it gives the details in bytes, mega bytes and gigabyte.

Executing only df command will gives some valuable information on the file systems, their mount points, their memory usage etc.

```
root@kali:~/Desktop# df
Filesystem      1K-blocks   Used Available Use% Mounted on
rootfs          119925964  9212160 104598836  9% /
udev             10240       0    10240   0% /dev
tmpfs            189660     748   188912  1% /run
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad 119925964  9212160 104598836  9% /
tmpfs            5120        4    5116   1% /run/lock
tmpfs            1160420    268   1160152  1% /run/shm
/dev/sda9         1856040   28780   1714928  2% /boot
/dev/sr1           58988     58988       0 100% /media/Idea Net Setter
/dev/sda6         127806460 33903304  93903156 27% /media/98A23980A23963C2
/dev/sda5         127805436 80713780  47091656 64% /media/4474251874250DEC
root@kali:~/Desktop#
```

i) Display Information of all the File Systems : -a option

To see the details of all file systems in our system following command is used.

Syntax: df -a

```
root@kali:~/Desktop# df -a
Filesystem      1K-blocks   Used Available Use% Mounted on
rootfs          119925964  9218984 104592012  9% /
sysfs            0         0     0     0% /sys
proc              0         0     0     0% /proc
udev             10240       0    10240   0% /dev
devpts            0         0     0     0% /dev/pts
tmpfs            189660     760   188900  1% /run
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad 119925964  9218984 104592012  9% /
tmpfs            5120        4    5116   1% /run/lock
tmpfs            1160420    264   1160156  1% /run/shm
/dev/sda9         1856040   28780   1714928  2% /boot
binfmt_misc        0         0     0     0% /proc/sys/fs/binfmt_misc
/dev/sr1           58988     58988       0 100% /media/Idea Net Setter
fusectl            0         0     0     0% /sys/fs/fuse/connections
/dev/sda6         127806460 33903304  93903156 27% /media/98A23980A23963C2
/dev/sda5         127805436 80713780  47091656 64% /media/4474251874250DEC
root@kali:~/Desktop#
```

Details of all the file systems and their memory usage is there.

Following information is given:

- i) file system name.
- ii) total size.
- iii) size used

- iv) free memory.
- v) use detail in %
- vi) point of mounting of file system.

ii) Specify the Memory Block Size : -B option

df command provides an option through which we can change the size of memory block in the output.

Syntax: df -B

```
root@kali:~/Desktop# df -B 100
Filesystem      100B-blocks  Used  Available Use% Mounted on
rootfs          1228041872  94406534 1071018066  9% /
udev             104858       0   104858  0% /dev
tmpfs            1942119    7783   1934336  1% /run
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad 1228041872  94406534 1071018066  9% /
tmpfs            52429        41   52388  1% /run/lock
tmpfs            11882701   2745   11879957  1% /run/shm
/dev/sda9         19005850  294708  17560863  2% /boot
/dev/sr1           604038   604038       0 100% /media/Idea Net Setter
/dev/sda6         1308738151 347169833  961568318  27% /media/98A23980A23963C2
/dev/sda5         1308727665 826509108  482218558  64% /media/4474251874250DEC
root@kali:~/Desktop#
```

we specified a block size of 100 and in the output (second column) block size of 100 is displayed.

iii) Print Human Readable Sizes : -h option

Option -h stands for "human" readable format. As shown in the output below, G is used for gigabytes and M is used for megabytes.

Syntax: df -h

```
root@kali:~/Desktop# df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          115G  8.8G  100G  9% /
udev             10M    0   10M  0% /dev
tmpfs            186M  728K  185M  1% /run
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad 115G  8.8G  100G  9% /
tmpfs            5.0M  4.0K  5.0M  1% /run/lock
tmpfs            1.2G 112K  1.2G  1% /run/shm
/dev/sda9         1.8G  29M  1.7G  2% /boot
/dev/sda6         122G  33G   90G  27% /media/98A23980A23963C2
/dev/sda5         122G  77G   45G  64% /media/4474251874250DEC
root@kali:~/Desktop#
```

iv) Display Grand Total in the Output : --total

We can see the total size of all file system using --total flag.

It will give the sum of all memory of all file system.

Syntax: df --total

```
root@kali:~/Desktop# df -h --total
Filesystem              Size  Used Avail Use% Mounted on
rootfs                  115G   8.8G  100G  9% /
udev                     10M     0   10M  0% /dev
tmpfs                   186M  728K  185M  1% /run
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad  115G   8.8G  100G  9% /
tmpfs                   5.0M  4.0K  5.0M  1% /run/lock
tmpfs                   1.2G  448K  1.2G  1% /run/shm
/dev/sda9                 1.8G   29M   1.7G  2% /boot
/dev/sda6                122G   33G   90G  27% /media/98A23980A23963C2
/dev/sda5                122G   77G   45G  64% /media/4474251874250DEC
total                  476G  127G  337G  28%
root@kali:~/Desktop#
```

a new row 'total' at the end of the output was produced. It is the sum of all file system sizes. -h option is used for displaying it in human readable format.

v) List Inodes (Instead of Block Usage) : -i option

To see the total inodes of each file system instead of size in block we use -i option.

```
root@kali:~/Desktop# df -i
Filesystem              Inodes IUsed  IFree IUse% Mounted on
rootfs                  7626752 352470  7274282  5% /
udev                     234522   505   234017  1% /dev
tmpfs                   237075   462   236613  1% /run
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad  7626752 352470  7274282  5% /
tmpfs                   237075     3   237072  1% /run/lock
tmpfs                   237075     8   237067  1% /run/shm
/dev/sda9                 120000   246   119754  1% /boot
/dev/sda6                93935924 14193  93921731  1% /media/98A23980A23963C2
/dev/sda5                47157192 28525  47128667  1% /media/4474251874250DEC
root@kali:~/Desktop#
```

here inode is displayed in place of size. And inodes used and free is also shown. (like memory block free and use).

vi) Print File System Type : -T option

To know the type of all file system we use the -T option.

Syntax: df -T

It will so the file system type along with all other details.

```
root@kali:~/Desktop# df -T
Filesystem              Type  1K-blocks  Used Available Use% Mounted on
rootfs                  rootfs 119925964 9219004 104591992  9% /
udev                     devtmpfs 10240     0   10240  0% /dev
tmpfs                   tmpfs  189660    728   188932  1% /run
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad ext4  119925964 9219004 104591992  9% /
tmpfs                   tmpfs   5120      4   5116  1% /run/lock
tmpfs                   tmpfs 1160420    116  1160304  1% /run/shm
/dev/sda9                 ext4 1856040   28780 1714928  2% /boot
/dev/sda6                fuseblk 127806460 33903304 93903156 27% /media/98A23980A23963C2
/dev/sda5                fuseblk 127805436 80713780 47091656 64% /media/4474251874250DEC
root@kali:~/Desktop#
```

we can see all the file systems along with their type is displayed.

vii) Include/Exclude Certain File System Type :

While using the df command we can decide which particular file system should be included or excluded in output.

-t to include a particular file system

Syntax: df -t [file system to include]

```
root@kali:~/Desktop# df -t ext4
Filesystem      1K-blocks   Used Available Use% Mounted on
/dev/disk/by-uuid/3d08e1c8-f9ce-4438-8998-ede30cfdccad 119925964 9219144 104591852  9% /
/dev/sda9          1856040   28780    1714928   2% /boot
```

following command shows only ext4 file system details.

-x to exclude a particular file system

Syntax: df -x [file system to be excluded]

The following command displays only ext4 file systems. types.

```
root@kali:~/Desktop# df -x ext4
Filesystem      1K-blocks   Used Available Use% Mounted on
rootfs        119925964 9219144 104591852  9% /
udev            10240       0    10240   0% /dev
tmpfs           189660     728   188932   1% /run
tmpfs            5120       4    5116   1% /run/lock
tmpfs           1160420    116   1160304   1% /run/shm
/dev/sda6      127806460 33903304 93903156 27% /media/98A23980A23963C2
/dev/sda5      127805436 80713780 47091656 64% /media/4474251874250DEC
root@kali:~/Desktop#
```

The following command displays all other file systems except ext4

The quieter you become, the more you are heard.

3) du-

"du" (Disk Usage) is a standard Unix/Linux command, used to check the information of disk usage of files and directories on a machine. It is used for summarizing the disk usage in terms of file size. It can be used with folders to get the total disk usage.

i) A simple du command :

Simple use of du command is as follows:

Syntax: du

```
root@kali:~/Desktop# du
564      ./usr/lib64/kde4
568      ./usr/lib64
472      ./usr/bin
4      ./usr/lib/kde4
8      ./usr/lib
8      ./usr/share/icons/hicolor/22x22/apps
12      ./usr/share/icons/hicolor/22x22
8      ./usr/share/icons/hicolor/24x24/apps
12      ./usr/share/icons/hicolor/24x24
8      ./usr/share/icons/hicolor/32x32/apps
12      ./usr/share/icons/hicolor/32x32
8      ./usr/share/icons/hicolor/48x48/apps
12      ./usr/share/icons/hicolor/48x48
8      ./usr/share/icons/hicolor/16x16/apps
12      ./usr/share/icons/hicolor/16x16
64      ./usr/share/icons/hicolor
68      ./usr/share/icons
12      ./usr/share/kde4/services
16      ./usr/share/kde4
8      ./usr/share/applications
4      ./usr/share/pixmaps
100     ./usr/share
1152    ./usr
2068    ./pic
2448    ./abc/CID Forms
4776    ./abc
749272  .
root@kali:~/Desktop# █
```

only directories that are occupying some disk are listed.

ii) To view disk usage of all files and directories :

-a option

To see the disk usage of all files and directories -a option is used.

Syntax: du -a

For better output it can also be used with 'more' and 'less' command.

```
root@kali:~/Desktop# du -a
4      ./test2
560    ./usr/lib64/kde4/kcm_adobe_flash_player.so
564    ./usr/lib64/kde4
568    ./usr/lib64
468    ./usr/bin/flash-player-properties
472    ./usr/bin
0      ./usr/lib/kde4/kcm_adobe_flash_player.so
4      ./usr/lib/kde4
8      ./usr/lib
4      ./usr/share/icons/hicolor/22x22/apps/flash-player-properties.png
8      ./usr/share/icons/hicolor/22x22/apps
12     ./usr/share/icons/hicolor/22x22
4      ./usr/share/icons/hicolor/24x24/apps/flash-player-properties.png
8      ./usr/share/icons/hicolor/24x24/apps
12     ./usr/share/icons/hicolor/24x24
4      ./usr/share/icons/hicolor/32x32/apps/flash-player-properties.png
8      ./usr/share/icons/hicolor/32x32/apps
12     ./usr/share/icons/hicolor/32x32
4      ./usr/share/icons/hicolor/48x48/apps/flash-player-properties.png
8      ./usr/share/icons/hicolor/48x48/apps
12     ./usr/share/icons/hicolor/48x48
4      ./usr/share/icons/hicolor/16x16/apps/flash-player-properties.png
8      ./usr/share/icons/hicolor/16x16/apps
12     ./usr/share/icons/hicolor/16x16
64    ./usr/share/icons/hicolor
68    ./usr/share/icons
8      ./usr/share/kde4/services/kcm_adobe_flash_player.desktop
12     ./usr/share/kde4/services
16     ./usr/share/kde4
4      ./usr/share/applications/flash-player-properties.desktop
8      ./usr/share/applications
0      ./usr/share/pixmaps/flash-player-properties.png
4      ./usr/share/pixmaps
```

The disk usage of all the files and directories.

iii) Display output in human readable form : -h option

Option -h stands for "human" readable format. As shown in the output below, G is used for gigabytes and M is used for megabytes.

Syntax : du -h

```
root@kali:~/Desktop# du -ah
4.0K    ./test2
560K    ./usr/lib64/Kde4/kcm_adobe_flash_player.so
564K    ./usr/lib64/kde4
568K    ./usr/lib64
468K    ./usr/bin/flash-player-properties
472K    ./usr/bin
0      ./usr/lib/kde4/kcm_adobe_flash_player.so
4.0K    ./usr/lib/kde4
8.0K    ./usr/lib
4.0K    ./usr/share/icons/hicolor/22x22/apps/flash-player-properties.png
8.0K    ./usr/share/icons/hicolor/22x22/apps
12K    ./usr/share/icons/hicolor/22x22
4.0K    ./usr/share/icons/hicolor/24x24/apps/flash-player-properties.png
8.0K    ./usr/share/icons/hicolor/24x24/apps
12K    ./usr/share/icons/hicolor/24x24
4.0K    ./usr/share/icons/hicolor/32x32/apps/flash-player-properties.png
8.0K    ./usr/share/icons/hicolor/32x32/apps
12K    ./usr/share/icons/hicolor/32x32
4.0K    ./usr/share/icons/hicolor/48x48/apps/flash-player-properties.png
8.0K    ./usr/share/icons/hicolor/48x48/apps
12K    ./usr/share/icons/hicolor/48x48
4.0K    ./usr/share/icons/hicolor/16x16/apps/flash-player-properties.png
8.0K    ./usr/share/icons/hicolor/16x16/apps
12K    ./usr/share/icons/hicolor/16x16
64K    ./usr/share/icons/hicolor
68K    ./usr/share/icons
8.0K    ./usr/share/kde4/services/kcm_adobe_flash_player.desktop
12K    ./usr/share/kde4/services
16K    ./usr/share/kde4
4.0K    ./usr/share/applications/flash-player-properties.desktop
8.0K    ./usr/share/applications
0      ./usr/share/pixmaps/flash-player-properties.png
4.0K    ./usr/share/pixmaps
100K   ./usr/share
```

disk usage is listed in terms of 'K'.

Note : - two or more switches can be used together to get more info. here -a and -h are used together.

iv) Display grand total in the output : -c option

TO see the total Disk usage or sum of all disk usage by all files and directories we use -c option.

Syntax : du -c

```
root@kali:~/Desktop# du -hc
564K    ./usr/lib64/kde4
568K    ./usr/lib64
472K    ./usr/bin
4.0K    ./usr/lib/kde4
3.0K    ./usr/lib
3.0K    ./usr/share/icons/hicolor/22x22/apps
12K    ./usr/share/icons/hicolor/22x22
3.0K    ./usr/share/icons/hicolor/24x24/apps
12K    ./usr/share/icons/hicolor/24x24
3.0K    ./usr/share/icons/hicolor/32x32/apps
12K    ./usr/share/icons/hicolor/32x32
3.0K    ./usr/share/icons/hicolor/48x48/apps
12K    ./usr/share/icons/hicolor/48x48
3.0K    ./usr/share/icons/hicolor/16x16/apps
12K    ./usr/share/icons/hicolor/16x16
64K    ./usr/share/icons/hicolor
68K    ./usr/share/icons
12K    ./usr/share/kde4/services
16K    ./usr/share/kde4
3.0K    ./usr/share/applications
4.0K    ./usr/share/pixmaps
100K   ./usr/share
1.2M   ./usr
2.5M   ./pic
2.4M   ./abc/CID Forms
4.7M   ./abc
733M   .
733M   total
root@kali:~/Desktop#
```

A new row total is added at end of list. It displays the sum of all disk usage of all file and directories.

v) Display only the total count : -s option

Rather than seeing all files and directories disk usage if we wish to see only the total disk usage we use -s option.

Syntax: du -s

```
root@kali:~/Desktop# du -sh
733M .
root@kali:~/Desktop#
```

Like -c option , -s option also displays the total disk usage ,only difference is -c option also displays individual disk usage of all files and directories whereas -s option only show total disk usage. -h option os used to show the total size in human readable format.

vi) Display output in bytes : -b option

If we want to see the disk usage in terms of bytes rather than KB, MB or GB we use the -b option.

Syntax: du -b

```
root@kali:~/Desktop# du -b
573944 ./usr/lib64/kde4
578040 ./usr/lib64
480232 ./usr/bin
4137 ./usr/lib/kde4
8233 ./usr/lib
5201 ./usr/share/icons/hicolor/22x22/apps
9297 ./usr/share/icons/hicolor/22x22
4944 ./usr/share/icons/hicolor/24x24/apps
9040 ./usr/share/icons/hicolor/24x24
5240 ./usr/share/icons/hicolor/32x32/apps
9336 ./usr/share/icons/hicolor/32x32
5829 ./usr/share/icons/hicolor/48x48/apps
9925 ./usr/share/icons/hicolor/48x48
4719 ./usr/share/icons/hicolor/16x16/apps
3815 ./usr/share/icons/hicolor/16x16
50509 ./usr/share/icons/hicolor
54605 ./usr/share/icons
8450 ./usr/share/kde4/services
12546 ./usr/share/kde4
5064 ./usr/share/applications
4151 ./usr/share/pixmaps
30462 ./usr/share
1151063 ./usr
2688584 ./pic
2494784 ./abc/CID Forms
4864406 ./abc
767682020 .
root@kali:~/Desktop# █
```

Here the disk usage and total disk usage (at last) is given in bytes..

vii) Display the modification time : --time flag

If we want to see the last modification time of all files and directories we can use --time flag. It will show both day and time last when it was modified.

Syntax: du --time

```
root@kali:~/Desktop# du --time
564    2015-01-28 21:20      ./usr/lib64/kde4
568    2015-01-28 21:20      ./usr/lib64
472    2015-01-28 21:20      ./usr/bin
4      2015-01-28 21:20      ./usr/lib/kde4
8      2015-01-28 21:20      ./usr/lib
8      2015-01-28 21:20      ./usr/share/icons/hicolor/22x22/apps
12     2015-01-28 21:20      ./usr/share/icons/hicolor/22x22
8      2015-01-28 21:20      ./usr/share/icons/hicolor/24x24/apps
12     2015-01-28 21:20      ./usr/share/icons/hicolor/24x24
8      2015-01-28 21:20      ./usr/share/icons/hicolor/32x32/apps
12     2015-01-28 21:20      ./usr/share/icons/hicolor/32x32
8      2015-01-28 21:20      ./usr/share/icons/hicolor/48x48/apps
12     2015-01-28 21:20      ./usr/share/icons/hicolor/48x48
8      2015-01-28 21:20      ./usr/share/icons/hicolor/16x16/apps
12     2015-01-28 21:20      ./usr/share/icons/hicolor/16x16
64     2015-01-28 21:20      ./usr/share/icons/hicolor
68     2015-01-28 21:20      ./usr/share/icons
12     2015-01-28 21:20      ./usr/share/kde4/services
16     2015-01-28 21:20      ./usr/share/kde4
8      2015-01-28 21:20      ./usr/share/applications
4      2015-01-28 21:20      ./usr/share/pixmaps
100    2015-01-28 21:20      ./usr/share
1152   2015-01-28 21:20      ./usr
2804   2015-05-23 13:18      ./pic
2448   2013-07-02 12:10      ./abc/CID Forms
4776   2013-07-02 12:10      ./abc
750008 2015-05-23 13:18      .
root@kali:~/Desktop#
```

Above date and time of last modification is displayed after disk usage column.
