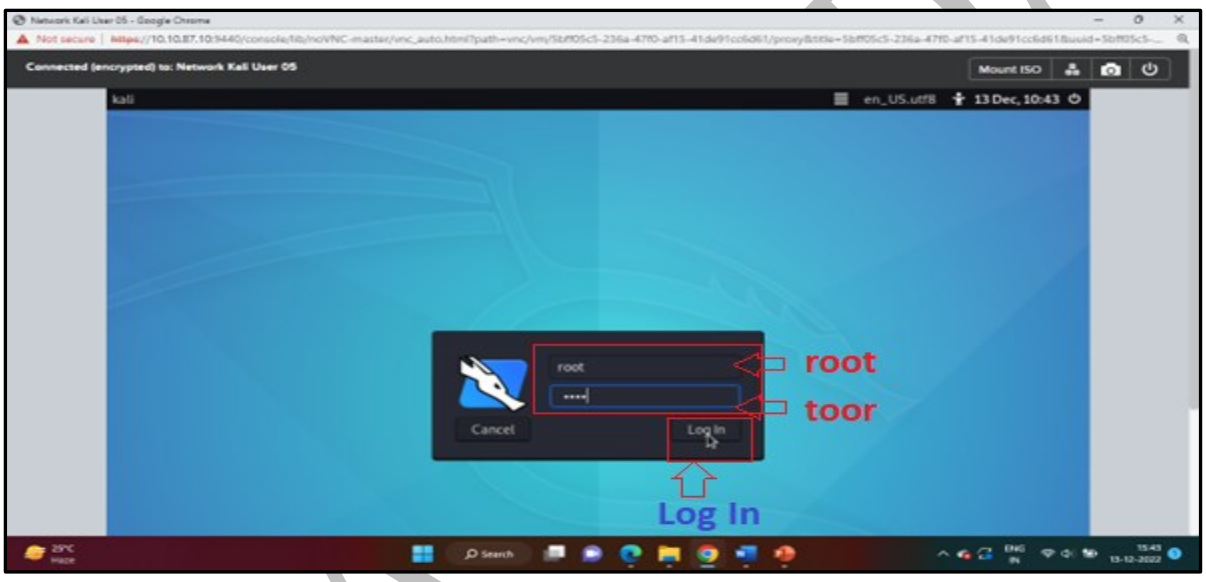
WORKING WITH FILES & DIRECTORIES

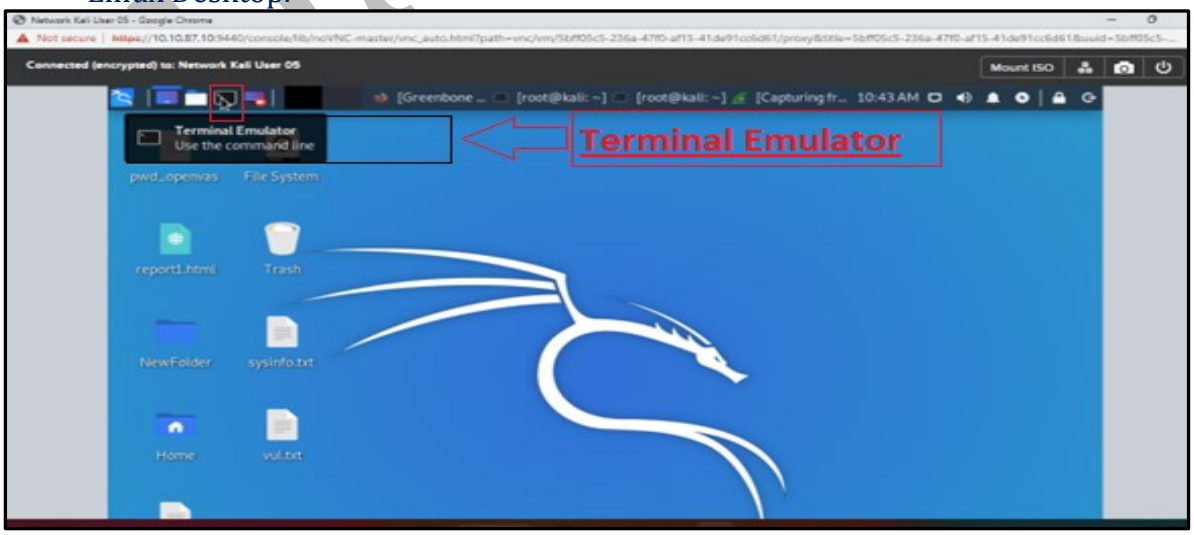
In Linux/Unix operating system everything is considered a file. Even directories are files, files are files, and devices like mice, keyboards, printers, etc are also files. File management commands are used to manage the files. In this lab manual, you will understand some file management commands.

Follow the below-given steps to execute these commands.

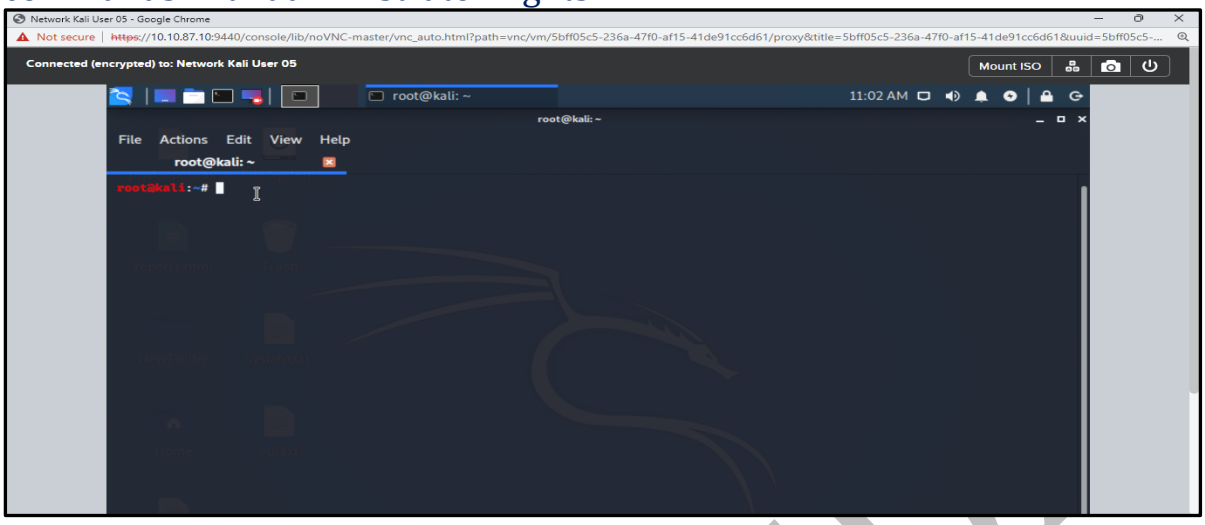
1. Connect to the kali Linux machine, created by you, using the RDP protocol.
2. When prompted for the username and password, enter root as username and toor as password. The root is the administrator user of the machine.



3. click on the black box icon (Terminal Emulator) in the top left corner of the Kali Linux Desktop.



Running the terminal while using the root account, allows you to run various commands with administrator rights.



Let's Understand some navigation file management one by one.

1. LS Command

The 'ls' command is used **to list files and directories**. **The contents of your current working directory**, which is just a technical way of stating the directory that your terminal is presently in, will be listed if you run the "ls" command without any further options.

a) Sort by Last Edited File Using

ls -t

```
┌──(cdac㉿cdac)-[~]
└─$ ls -t
newfile.txt  Desktop  file5.txt  file3.txt  file1.txt  Downloads  Pictures  Templates
vi.txt       fie.zip  file4.txt  file2.txt  Documents  Music      Public    Videos
```

b) Hidden Files Using

ls -a

```
┌──(cdac㉿cdac)-[~]
└─$ ls -a
.                Documents   file4.txt  newfile.txt                .viminfo
..               Downloads   file5.txt  Pictures                   vi.txt
.bash_logout     .face       .java      .profile                   .zsh_history
.bashrc          .face.icon  .lesshst   Public                     .zshrc
.bashrc.original fie.zip     .local     .selected_editor
.cache           file1.txt   .mozilla   .sudo_as_admin_successful
.config          file2.txt   .msf4      Templates
Desktop          file3.txt   Music      Videos
```

c) Display All information Using

ls -l

```
┌──(cdac㉿cdac)-[~]
└─$ ls -l
total 64
drwxr-xr-x 2 cdac cdac 4096 Jun 21 22:56 Desktop
drwxr-xr-x 2 cdac cdac 4096 Jun  6 20:41 Documents
drwxr-xr-x 2 cdac cdac 4096 Jun  6 20:41 Downloads
-rw-r--r-- 1 cdac cdac  757 Jun 21 22:47 fie.zip
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file1.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file2.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file3.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file4.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:47 file5.txt
drwxr-xr-x 2 cdac cdac 4096 Jun  6 20:41 Music
-rw-r--r-- 1 cdac cdac   20 Jun 22 17:49 newfile.txt
drwxr-xr-x 2 cdac cdac 4096 Jun  6 20:41 Pictures
```

d) Display File Size In Human Readable Format using ls -lh

```
┌──(cdac㉿cdac)-[~]
└─$ ls -lh
total 64K
drwxr-xr-x 2 cdac cdac 4.0K Jun 21 22:56 Desktop
drwxr-xr-x 2 cdac cdac 4.0K Jun  6 20:41 Documents
drwxr-xr-x 2 cdac cdac 4.0K Jun  6 20:41 Downloads
-rw-r--r-- 1 cdac cdac  757 Jun 21 22:47 fie.zip
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file1.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file2.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file3.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:46 file4.txt
-rw-r--r-- 1 cdac cdac    1 Jun 21 22:47 file5.txt
drwxr-xr-x 2 cdac cdac 4.0K Jun  6 20:41 Music
```

2. CAT COMMAND

The cat command is a utility command in Linux. One of its most common usages is **to print the content of a file onto the standard output stream**. Other than that, the cat command also allows us to write some texts into a file.

a) To view Single File

cat newfile.txt

```
┌──(cdac㉿cdac)-[~]
└─$ cat newfile.txt
This is nano Editor
```

b) To view Multiple File

Cat newfile.txt vi.txt

```
┌──(cdac㉿cdac)-[~]
└─$ cat newfile.txt vi.txt
This is nano Editor
This is sample
```

c) To view File with Line Numbers

Cat -n newfile.txt

```
┌──(cdac㉿cdac)-[~]
└─$ cat -n newfile.txt
     1  This
     2  is
     3  nano
     4  Editor
```

d) Create a file

Cat createfile

```
┌──(cdac㉿cdac)-[~]
└─$ cat > createfile
This command is used to create file
^C

┌──(cdac㉿cdac)-[~]
└─$ cat createfile
This command is used to create file
```

e) Copy Content of one file to another

Cat newfile.txt > createfile

Cat createfile

```
┌──(cdac㉿cdac)-[~]
└─$ cat newfile.txt > createfile

┌──(cdac㉿cdac)-[~]
└─$ cat createfile
This
is
nano
Editor
```

3. TOUCH COMMAND

The touch command's primary function is to modify a timestamp. Commonly, the utility is used for file creation, although this is not its primary function. The terminal program can change the modification and access time for any given file. The touch command creates a file only if the file doesn't already exist.

a) To Create File only

touch createfile

```
┌──(cdac㉿cdac)-[~/Documents]
└─$ touch createfile
```

b) To check file created or not

touch -c

```
┌──(cdac㉿cdac)-[~/Documents]
└─$ ls
createfile

┌──(cdac㉿cdac)-[~/Documents]
└─$ touch -c createfile

┌──(cdac㉿cdac)-[~/Documents]
└─$ ls
createfile
```

c) To change file Modification time only

touch -m

```
┌──(cdac㉿cdac)-[~/Documents]
└─$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 0 Dec 16  2022 createfile

┌──(cdac㉿cdac)-[~/Documents]
└─$ touch -m createfile

┌──(cdac㉿cdac)-[~/Documents]
└─$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 0 Jun 22 21:58 createfile

┌──(cdac㉿cdac)-[~/Documents]
└─$
```

d) **4.** To update access and modification time

touch -c -d

```
┌──(cdac㉿cdac)-[~/Documents]
└─$ stat createfile
  File: createfile
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d      Inode: 3951297     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/    cdac)   Gid: ( 1000/    cdac)
Access: 2022-06-22 21:55:09.008000000 +0530
Modify: 2022-06-22 21:55:09.008000000 +0530
Change: 2022-06-22 21:55:09.008000000 +0530
 Birth: 2022-06-22 21:54:56.992000000 +0530

┌──(cdac㉿cdac)-[~/Documents]
└─$ touch -c -d '16 Dec' createfile

┌──(cdac㉿cdac)-[~/Documents]
└─$ stat createfile
  File: createfile
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d      Inode: 3951297     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/    cdac)   Gid: ( 1000/    cdac)
Access: 2022-12-16 00:00:00.000000000 +0530
Modify: 2022-12-16 00:00:00.000000000 +0530
Change: 2022-06-22 21:56:41.720000000 +0530
 Birth: 2022-06-22 21:54:56.992000000 +0530
```

4. MKDIR COMMAND

The mkdir command in Linux/Unix allows users to create or make new directories. mkdir stands for “make directory.” With mkdir, you can also set permissions, create multiple directories (folders) at once, and much more

a) To Create Single Directory

Mkdir directory name

```
┌──(cdac㉿cdac)-[~/Documents]
└─$ ls
createfile

┌──(cdac㉿cdac)-[~/Documents]
└─$ mkdir first

┌──(cdac㉿cdac)-[~/Documents]
└─$ ls
createfile  first
```

b) To Create Multiple Directories

Mkdir -p directory name

```
┌──(cdac㉿cdac)-[~/Documents]
└─$ tree
.
├── createfile
└── first

1 directory, 1 file

┌──(cdac㉿cdac)-[~/Documents]
└─$ mkdir -p first/second/third/

┌──(cdac㉿cdac)-[~/Documents]
└─$ tree
.
├── createfile
└── first
    └── second
        └── third

3 directories, 1 file
```

5. RM COMMAND

Use the rm command **to remove files you no longer need**. The rm command removes the entries for a specified file, group of files, or certain select files from a list within a directory. User confirmation, read permission, and write permission are not required before a file is removed when you use the rm command.

a) To remove file

rm filename

```
┌──(cdac㉿cdac)-[~/Public]
└─$ ls
a.txt  b.txt  c.txt  d.txt

┌──(cdac㉿cdac)-[~/Public]
└─$ rm a.txt

┌──(cdac㉿cdac)-[~/Public]
└─$ ls
b.txt  c.txt  d.txt

┌──(cdac㉿cdac)-[~/Public]
└─$ rm b.txt

┌──(cdac㉿cdac)-[~/Public]
└─$ ls
c.txt  d.txt
```

b) To recursively delete a directory and all its contents
rm -r

```
┌──(cdac㉿cdac)-[~/Public]
└─$ tree
.
├── A
│   └── B
│       └── C
│           └── D
│               └── E
├── c.txt
└── d.txt

5 directories, 2 files

┌──(cdac㉿cdac)-[~/Public]
└─$ rm -r A

┌──(cdac㉿cdac)-[~/Public]
└─$ tree
.
├── c.txt
└── d.txt

0 directories, 2 files
```

6. CP COMMAND

You can use the cp command for copying files from one location to another. This command can also copy directories (folders). [file/directory-sources] specifies the sources of the files or directories you want to copy. And the [destination] argument specifies the location you want to copy the file to.

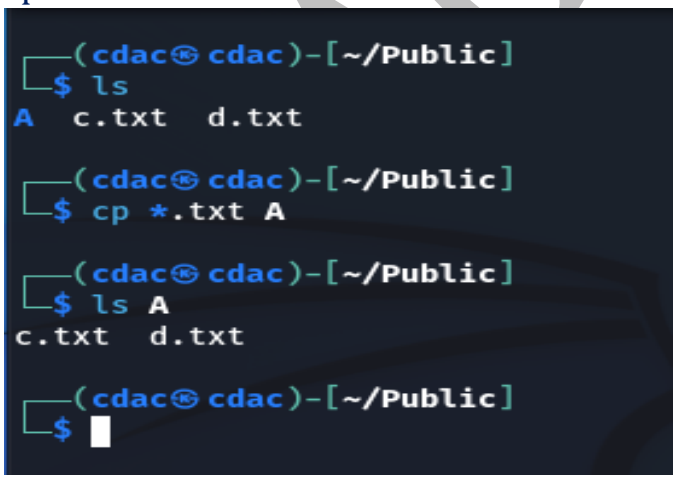
cp [OPTION] Source Destination
cp [OPTION] Source Directory
cp [OPTION] Source-1 Source-2 Source-3 Source-n Directory

a) To recursively copy a directory and all its contents

cp-r

b) To copy all files and directories

cp * dir

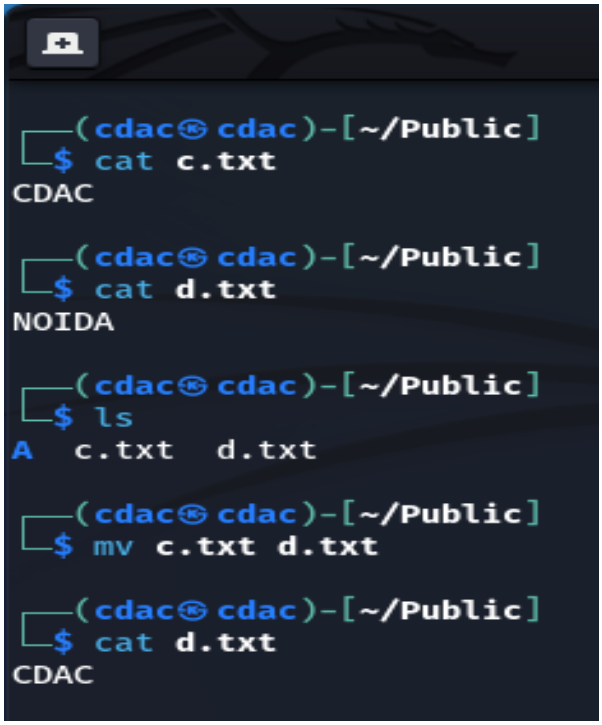


7. MV COMMAND

Use the mv command to move files and directories from one directory to another or to rename a file or directory.

a) To move one or more files or directories from one place to another in a file system like UNIX.

mv c.txt d.txt



b) To force mv by overwriting destination file without prompt

mv –f c.txt d.txt