



PRELIMINARY TASK - 5

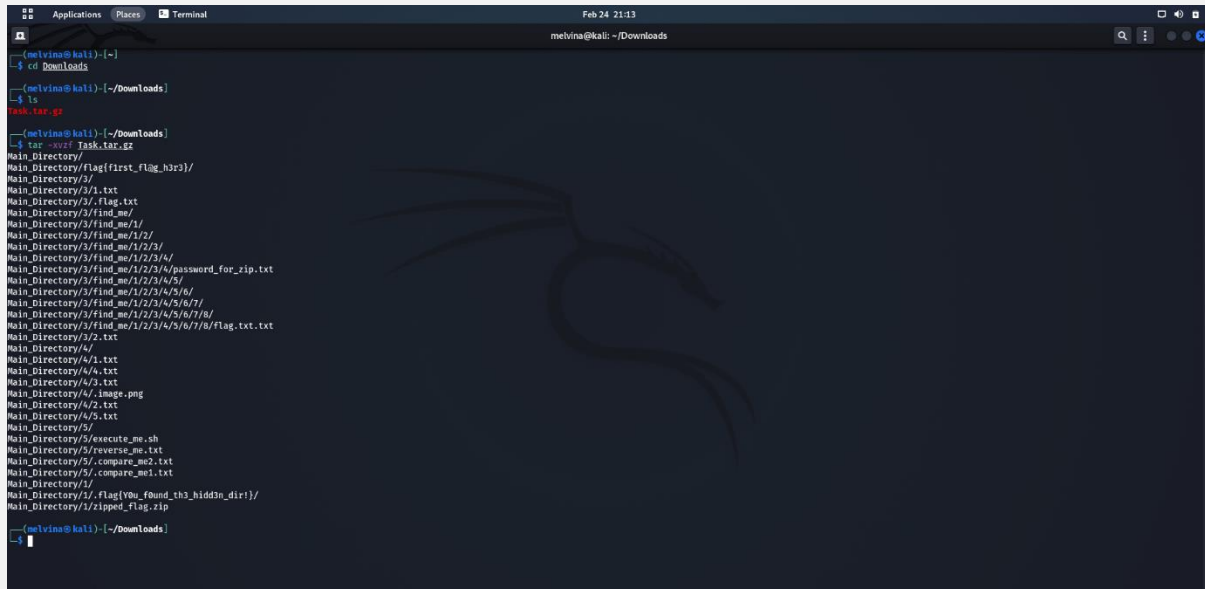
CYS BASICS

SUBMITTED BY :

Melvina Jose
1st year CYS

Part 1 :

Extracting the files of Task.tar.gz to a new directory



```
(melvina@kali):~$ cd Downloads
(melvina@kali):~/Downloads$ ls
Task.tar.gz
(melvina@kali):~/Downloads$ tar -xvzf Task.tar.gz
Main_Directory/
Main_Directory/flag{f1rst_fl@g_h3r3}/
Main_Directory/3/
Main_Directory/3/1.txt
Main_Directory/3/.flag.txt
Main_Directory/3/find_me/
Main_Directory/3/find_me/1/
Main_Directory/3/find_me/1/2/
Main_Directory/3/find_me/1/2/3/
Main_Directory/3/find_me/1/2/3/4/
Main_Directory/3/find_me/1/2/3/4/password_for_zip.txt
Main_Directory/3/find_me/1/2/3/4/5/
Main_Directory/3/find_me/1/2/3/4/5/6/
Main_Directory/3/find_me/1/2/3/4/5/6/7/
Main_Directory/3/find_me/1/2/3/4/5/6/7/8/
Main_Directory/3/find_me/1/2/3/4/5/6/7/8/flag.txt.txt
Main_Directory/3/2.txt
Main_Directory/4/
Main_Directory/4/1.txt
Main_Directory/4/4.txt
Main_Directory/4/3.txt
Main_Directory/4/.image.png
Main_Directory/4/2.txt
Main_Directory/4/5.txt
Main_Directory/5/
Main_Directory/5/execute_me.sh
Main_Directory/5/reverse_me.txt
Main_Directory/5/compare_me2.txt
Main_Directory/5/compare_me1.txt
Main_Directory/1/
Main_Directory/1/.flag[you_found_the_hidden_dir!]/
Main_Directory/1/zipper_flag.zip
```

Commands used :

tar - used to create Archive and extract the Archive files.

options used -

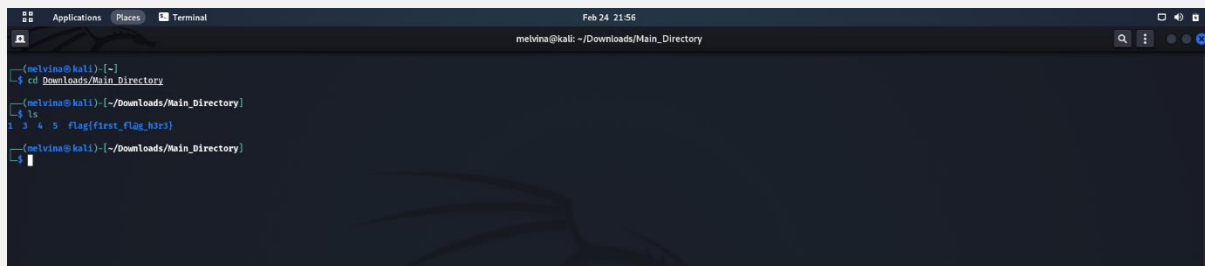
-x : Extract the archive

-v : Displays Verbose Information

-z : zip, tells tar command that creates tar file using gzip

-f : creates archive with given filename

Flag 1 :



```
(melvina@kali):~$ cd Downloads/Main_Directory
(melvina@kali):~/Downloads/Main_Directory$ ls
1 3 4 5 flag{f1rst_fl@g_h3r3}
```

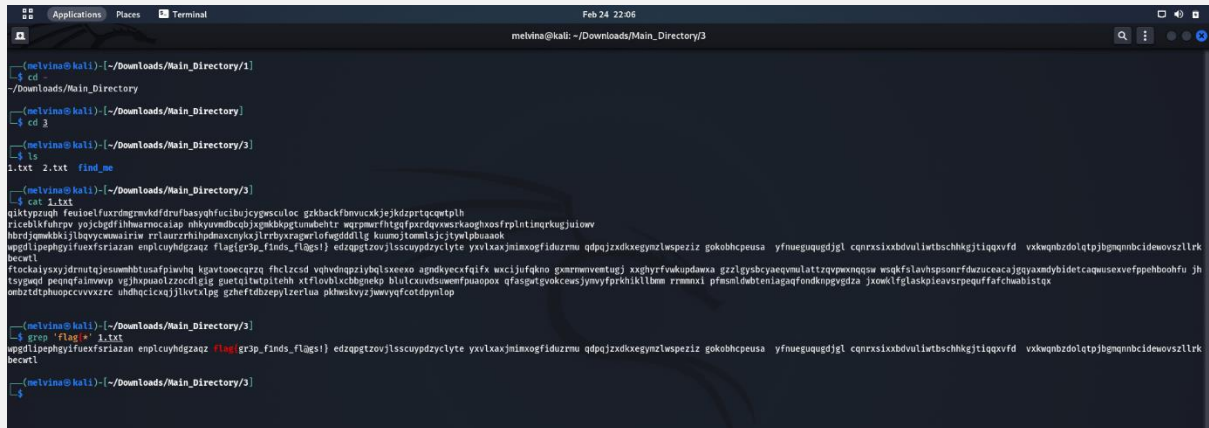
Flag : flag{f1rst_fl@g_h3r3}

Commands used :

cd - change directory

ls - list files

Flag 2 :



```
melvina@kali: ~/Downloads/Main_Directory/3
$ cd -
~/Downloads/Main_Directory
$ cd 3
$ ls
1.txt 2.txt find_me
$ cat 1.txt
qikTyzpZqH feuiSeIfuxzdgmrwddfrubasyghfucibujcypeculoc gzbachfawucckjshjdzprtcqwtph
ricbikfufurpy yojcqdGfihwamociaap nhkyuvmdbcojxgmkkpgtumebetr wqrmerfhtgfprrdqyvwskagthxosfrplntimrkugjuow
hbrdjmwbkijlbqyvwmaairw rrlaurzrhphdmxcnyxjlrbyxragwlofugddllg kuonojtomlsjcytwlpbuaaak
apgdilpephgyifuexfriaan enpicuyhdgzaqz flag{gr3p_f1nds_fl@gsl} edzqpgtzovjlsscuydpdzcyte yxvlxaxjnimxogfiduzrmu qdpqjzdxkxegmzlwspziz gokobhcepsa yfnueguqugdjgl cqnrxsxxbdvulwtbschhkgjtiqkvfd vxkqnbzdoletpjbmqnbcidewovszllrk
becwtl
ftckaiysxyjdrnutqjesummbtusafpiwuh kvavtoecqrzq fhclzcsd vqhdnqprizyblsxexeo agndkyecxfqfx wxcijufqkno gxmrmvmentugj xxghyffwakupdaaxa gzzlgsybcaeqvmulattzyvpxnqqs wsgkflavhspsonrfdzuceacajgqyxandybidetcaqusexvfppehboohfu jh
tsywgqd peqnfaimwvp vghxpmualzccdlsg guetqitwtpthh xflvoblxcbgnekp blulcuvsdswenfpuaopox qfasatgvokceusjnyvfpkrhikllbm rrmnxi pfmsnlwbtenlagaqfondkngvda jxowkifglaskpieavsrpequffafchwabistqx
omztdtphuoocvzxrcr uhdmqicakjllivxalp gzhftdbzpylzerlaa pkheskvyizjwvyqfctddynlop
$ grep 'flag+' 1.txt
apgdilpephgyifuexfriaan enpicuyhdgzaqz flag{gr3p_f1nds_fl@gsl} edzqpgtzovjlsscuydpdzcyte yxvlxaxjnimxogfiduzrmu qdpqjzdxkxegmzlwspziz gokobhcepsa yfnueguqugdjgl cqnrxsxxbdvulwtbschhkgjtiqkvfd vxkqnbzdoletpjbmqnbcidewovszllrk
becwtl
$
```

Flag : flag{gr3p_f1nds_fl@gsl}

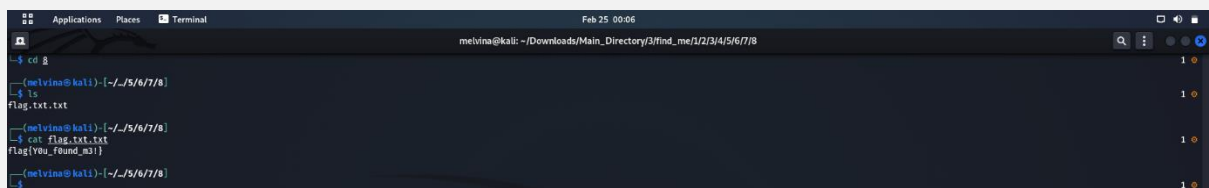
Commands used :

'cd -' - go to previous directory

cat - used to print the content of a file onto the standard output stream.

grep - used to search for a string of characters in a specified file.

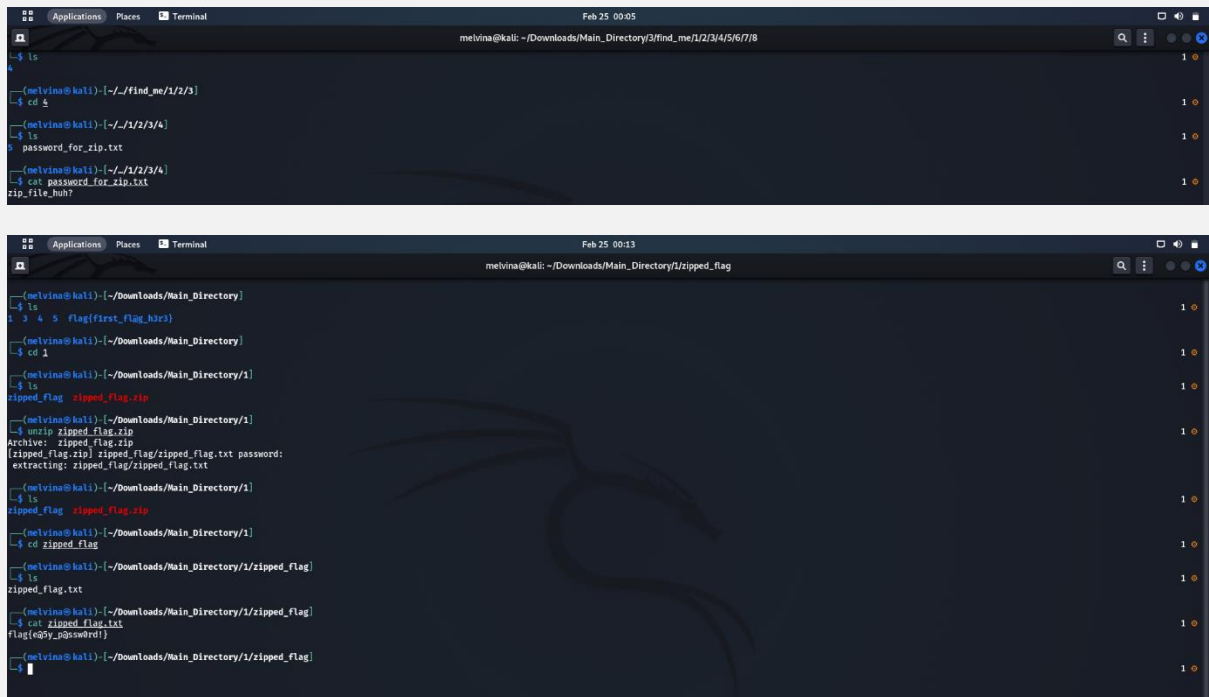
Flag 3 :



```
melvina@kali: ~/Downloads/Main_Directory/3/find_me/1/2/3/4/5/6/7/8
$ cd g
$ ls
flag.txt.txt
$ cat flag.txt.txt
flag{Y0u_f0und_m3!}
$
```

Flag : flag{Y0u_f0und_m3!}

Flag 4 :



The first terminal screenshot shows a user navigating through a directory structure to find a zip file. The user starts in the root directory, then goes to /Downloads/Main_Directory/3/find_me/1/2/3/4/5/6/7/8. They then navigate to /1/2/3/4 and find a file named password_for_zip.txt. The second terminal screenshot shows the user navigating to /Downloads/Main_Directory/1 and finding a file named zipped_flag.zip. They then use the unzip command to extract the file, which creates a file named zipped_flag.txt. Finally, they use the cat command to display the contents of the file, which is flag{e@5y_p@ssw0rd!}.

```
melvina@kali: ~/Downloads/Main_Directory/3/find_me/1/2/3/4/5/6/7/8
$ ls
4
(melvina@kali): ~/find_me/1/2/3/4
$ cd 4
(melvina@kali): ~/1/2/3/4
$ ls
password_for_zip.txt
(melvina@kali): ~/1/2/3/4
$ cat password_for_zip.txt
zip_file_huh?

melvina@kali: ~/Downloads/Main_Directory/1/zipped_flag
$ ls
1 3 4 5 flag(first_flag_h3r3)
(melvina@kali): ~/Downloads/Main_Directory
$ cd 1
(melvina@kali): ~/Downloads/Main_Directory/1
$ ls
zipped_flag zipped_flag.zip
(melvina@kali): ~/Downloads/Main_Directory/1
$ unzip zipped_flag.zip
Archive:  zipped_flag.zip
  zipped_flag.zip
  extracting: zipped_flag/zipped_flag.txt
(melvina@kali): ~/Downloads/Main_Directory/1
$ ls
zipped_flag zipped_flag.zip
(melvina@kali): ~/Downloads/Main_Directory/1
$ cd zipped_flag
(melvina@kali): ~/Downloads/Main_Directory/1/zipped_flag
$ ls
zipped_flag.txt
(melvina@kali): ~/Downloads/Main_Directory/1/zipped_flag
$ cat zipped_flag.txt
flag{e@5y_p@ssw0rd!}
(melvina@kali): ~/Downloads/Main_Directory/1/zipped_flag
$
```

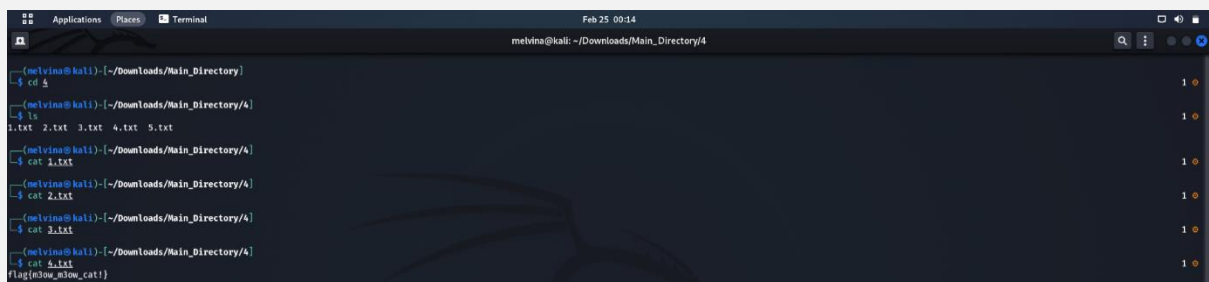
Password for text file : zip_file_huh?

Flag : flag{e@5y_p@ssw0rd!}

Commands used :

unzip - used to extract all files from the specified ZIP Archive

Flag 5 :

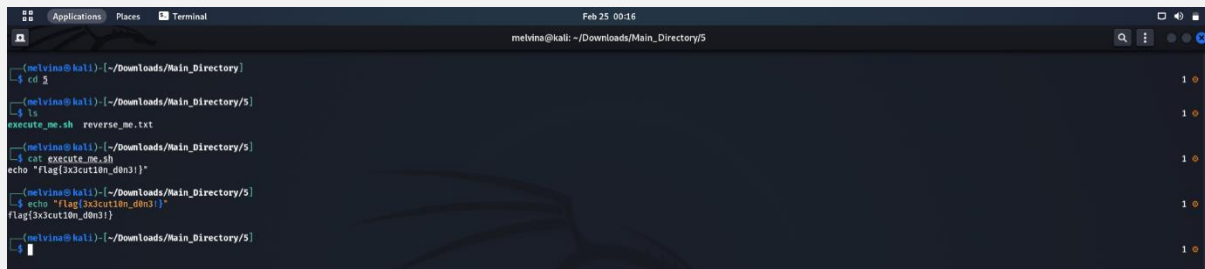


The terminal screenshot shows a user navigating to the directory /Downloads/Main_Directory/4. They then use the ls command to list the files in the directory, which are 1.txt, 2.txt, 3.txt, 4.txt, and 5.txt. They then use the cat command to display the contents of each file. The contents of 1.txt, 2.txt, and 3.txt are empty. The contents of 4.txt are flag{m3ow_m3ow_cat!}. The contents of 5.txt are empty.

```
melvina@kali: ~/Downloads/Main_Directory/4
$ cd 4
(melvina@kali): ~/Downloads/Main_Directory/4
$ ls
1.txt 2.txt 3.txt 4.txt 5.txt
(melvina@kali): ~/Downloads/Main_Directory/4
$ cat 1.txt
(melvina@kali): ~/Downloads/Main_Directory/4
$ cat 2.txt
(melvina@kali): ~/Downloads/Main_Directory/4
$ cat 3.txt
(melvina@kali): ~/Downloads/Main_Directory/4
$ cat 4.txt
flag{m3ow_m3ow_cat!}
(melvina@kali): ~/Downloads/Main_Directory/4
$ cat 5.txt
```

Flag : flag{m3ow_m3ow_cat!}

Flag 6 :

A terminal window showing the process of finding Flag 6. The user is in the directory ~/Downloads/Main_Directory/S. They run 'ls' and see 'execute_me.sh' and 'reverse_me.txt'. They run 'cat execute_me.sh' and see 'echo "flag{3x3cut10n_d0n3!}"'. They then run 'echo "flag{3x3cut10n_d0n3!}"' and 'flag{3x3cut10n_d0n3!}' is printed. The terminal has a Kali Linux logo watermark.

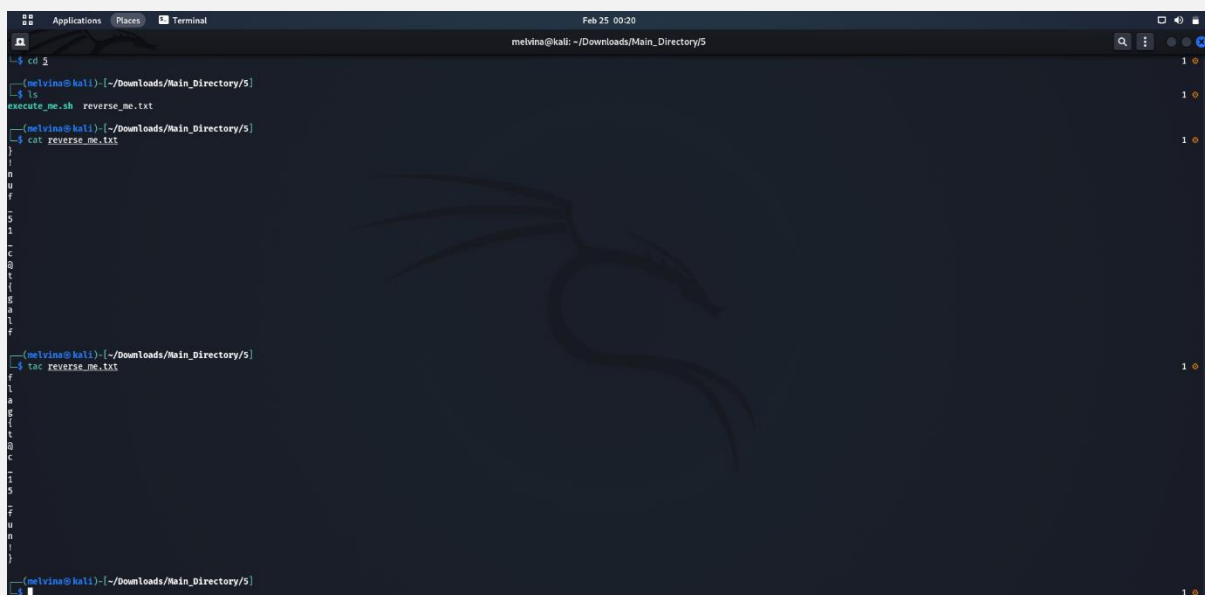
```
(melvina@kali)~/Downloads/Main_Directory/$ cd /
(melvina@kali)~/Downloads/Main_Directory/$ ls
execute_me.sh  reverse_me.txt
(melvina@kali)~/Downloads/Main_Directory/$ cat execute_me.sh
echo "flag{3x3cut10n_d0n3!}"
(melvina@kali)~/Downloads/Main_Directory/$ echo "flag{3x3cut10n_d0n3!}"
flag{3x3cut10n_d0n3!}
(melvina@kali)~/Downloads/Main_Directory/$
```

Flag : flag{3x3cut10n_d0n3!}

Commands used :

echo - used to display line of text/string that are passed as an argument .

Flag 7 :

A terminal window showing the process of finding Flag 7. The user is in the directory ~/Downloads/Main_Directory/S. They run 'ls' and see 'execute_me.sh' and 'reverse_me.txt'. They run 'cat reverse_me.txt' and see a long string of characters. They then run 'tac reverse_me.txt' and the string is printed in reverse order. The terminal has a Kali Linux logo watermark.

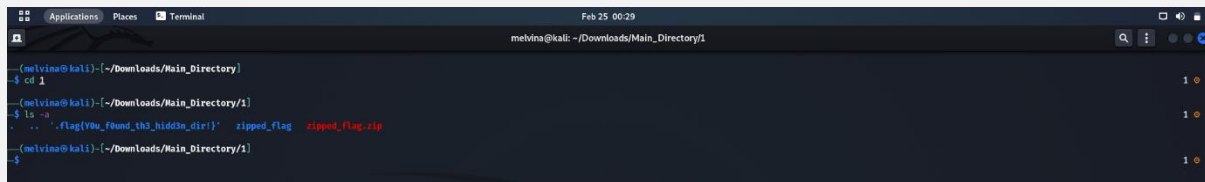
```
(melvina@kali)~/Downloads/Main_Directory/$ cd /
(melvina@kali)~/Downloads/Main_Directory/$ ls
execute_me.sh  reverse_me.txt
(melvina@kali)~/Downloads/Main_Directory/$ cat reverse_me.txt
f
l
a
g
{
t
@
c
_
1
5
_
f
u
n
!
}
(melvina@kali)~/Downloads/Main_Directory/$ tac reverse_me.txt
f
l
a
g
{
t
@
c
_
1
5
_
f
u
n
!
}
(melvina@kali)~/Downloads/Main_Directory/$
```

Flag : flag{t@c_15_fun!}

Commands used :

tac - used to display the file content in reverse order.

Flag 8 :

A terminal window titled 'Applications Places Terminal' with the date 'Feb 25 00:29'. The user 'melvina' is at the prompt 'melvina@kali: ~/Downloads/Main_Directory/1'. The commands and output are:

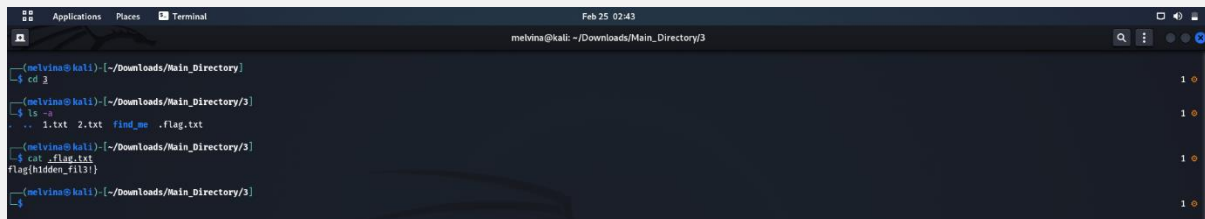
```
(melvina@kali)~/Downloads/Main_Directory/1
$ cd 1
(melvina@kali)~/Downloads/Main_Directory/1
$ ls -la
.. '.flag{Y0u_f0und_th3_hidd3n_dir!}' zipped_flag zipped_flag.zip
(melvina@kali)~/Downloads/Main_Directory/1
$
```

Flag : flag{Y0u_f0und_th3_hidd3n_dir!}

Commands used :

'ls -a' - used to display all the files including the hidden ones

Flag 9 :

A terminal window titled 'Applications Places Terminal' with the date 'Feb 25 02:43'. The user 'melvina' is at the prompt 'melvina@kali: ~/Downloads/Main_Directory/3'. The commands and output are:

```
(melvina@kali)~/Downloads/Main_Directory/3
$ cd 3
(melvina@kali)~/Downloads/Main_Directory/3
$ ls -la
.. 1.txt 2.txt find_me .flag.txt
(melvina@kali)~/Downloads/Main_Directory/3
$ cat .flag.txt
flag{hidden_fil3}
(melvina@kali)~/Downloads/Main_Directory/3
$
```

Flag : flag{h1dden_fil3!}

Flag 10 :

A terminal window titled 'Applications Places Terminal' with the date 'Feb 25 00:24'. The user 'melvina' is at the prompt 'melvina@kali: ~/Downloads/Main_Directory/4'. The commands and output are:

```
(melvina@kali)~/Downloads/Main_Directory/4
$ cd 4
(melvina@kali)~/Downloads/Main_Directory/4
$ ls -la
.. 1.txt 2.txt 3.txt 4.txt 5.txt .image.png
(melvina@kali)~/Downloads/Main_Directory/4
$ cat .image.png
flag{t3xt_15_n0t_h1dd3n!}
(melvina@kali)~/Downloads/Main_Directory/4
$
```

Flag : flag{t3xt_15_n0t_h1dd3n!}

Flag 11 :

```
Applications Places Terminal Feb 25 02:38
melvina@kali: ~/Downloads/Main_Directory/5

(melvina@kali)~/Downloads/Main_Directory/5
$ cd 5
(melvina@kali)~/Downloads/Main_Directory/5
$ ls -la
.. .compare_me1.txt .compare_me2.txt execute_me.sh reverse_me.txt
(melvina@kali)~/Downloads/Main_Directory/5
$ cp .compare_me1.txt file1.txt
(melvina@kali)~/Downloads/Main_Directory/5
$ cp .compare_me2.txt file2.txt
(melvina@kali)~/Downloads/Main_Directory/5
$ ls
execute_me.sh file1.txt file2.txt reverse_me.txt
(melvina@kali)~/Downloads/Main_Directory/5
$ diff file1.txt file2.txt
3d3
---
2
21c173
---
2
40c340
---
2
21c431
---
2
85c585
---
2
01d0d0
---
65c664,665
---
2
2
69c769
```

Flag : `flag{d1ff_15_u53ful!}`

Commands used :

`diff` - allows you to compare two files line by line.

Linux Commands - mkdir, mv, rm

```
Applications Places Terminal Feb 25 02:48
melvina@kali: ~/Downloads/Main_Directory/temp

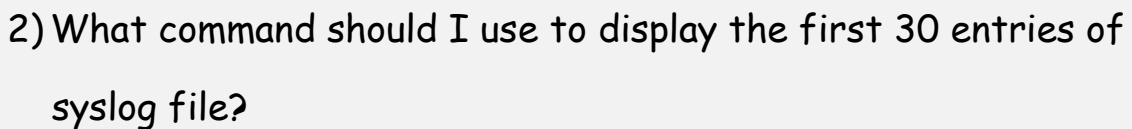
(melvina@kali)~/Downloads/Main_Directory/
$ mkdir temp
(melvina@kali)~/Downloads/Main_Directory/
$ ls
3 4 5 flag{first_flag_h3r3} temp
(melvina@kali)~/Downloads/Main_Directory/
$ echo "New text file" > new.txt
(melvina@kali)~/Downloads/Main_Directory/
$ ls
3 4 5 flag{first_flag_h3r3} new.txt temp
(melvina@kali)~/Downloads/Main_Directory/
$ cat new.txt
New text file
(melvina@kali)~/Downloads/Main_Directory/
$ mv new.txt temp
(melvina@kali)~/Downloads/Main_Directory/
$ ls
3 4 5 flag{first_flag_h3r3} temp
(melvina@kali)~/Downloads/Main_Directory/
$ cd temp
(melvina@kali)~/Downloads/Main_Directory/temp
$ ls -la
.. new.txt
(melvina@kali)~/Downloads/Main_Directory/temp
$ rm new.txt
(melvina@kali)~/Downloads/Main_Directory/temp
$ ls
(melvina@kali)~/Downloads/Main_Directory/temp
$
```

`mkdir` - make a directory

`mv` - move a file

`rm` - remove a file

1) Write a bash script to echo your name 25 times



3) What command should I use to display the last 30 entries of syslog file?

```
$ head -30 /var/log/syslog
```


4) What command should I use to arrange the entries of a file

- Alphabetically
`$ sort <filename.txt>`
- Reverse order
`$ sort -r <filename.txt>`
- Numerical order
`$ sort -n <filename.txt>`

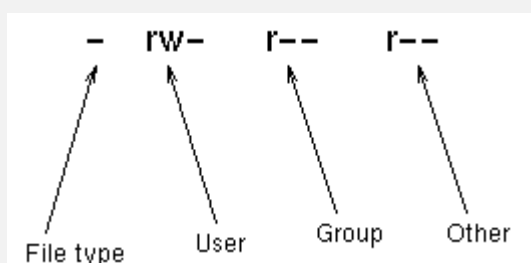
5) Copee is a hard-working cop. He found a case and almost at the verge of cracking it. It could be his best breakthrough. He has the list of criminals but lots of duplicates are there. He needs to find the only one that is different. He sought your help. How will you sort this issue?

`$ uniq <filename.txt>`

6) What are the four parts of file's permission?

The three types of file permissions are **Read**, **Write**, and **Execute**.

Parts of a file's permission



The first character indicates the **file type**. Here it is a dash because it shows the permission of an ordinary file. It could also be a `d` for a directory, or various other letters for more obscure types of file.

Example : `drwxr-xr-x`
`-rw-r--r--`

The next 9 characters are divided into three sets of three.

Each set represents the corresponding **access rights** of the **user who owns the file, the group which owns the file, and all other users.**

'r' - read; 'w' - write; 'x' - execute; '-' - permission not granted

Read - Can list the files of a directory and read them

Write - Can create, modify and delete the files in a directory

Execute - Can change to the directory and run the files as a program

x-----x