

Cyber Security And Ethical Hacking

Assignment-2: Bash Shell Basics

Name:P.Harish Chowdary

Reg.No:20BCD7052

1. File and Directory Manipulation

1. Create a directory called "my_directory".
2. Navigate into the "my_directory".
3. Create an empty file called "my_file.txt".
4. List all the files and directories in the current directory.
5. Rename "my_file.txt" to "new_file.txt".
6. Display the content of "new_file.txt" using a pager tool of your choice.
7. Append the text "Hello, World!" to "new_file.txt".
8. Create a new directory called "backup" within "my_directory".
9. Move "new_file.txt" to the "backup" directory.
10. Verify that "new_file.txt" is now located in the "backup" directory.
11. Delete the "backup" directory and all its contents.

All the above tasks are performed and the screenshots of the same are attached below and the screenshots itself are self explanatory.

Commands used to complete the above tasks are

1. **mkdir** my_directory
2. **cd** my_directory
3. **touch** my_file.txt
4. **ls**
5. **mv** my_file.txt new_file.txt
6. **cat** new_file.txt
7. **nano** new_file.txt
8. **cat** new_file.txt
9. **mkdir** backup
10. **mv** new_file.txt backup
11. **cd** backup
12. **ls**
13. **rm -r** backup

Output:

```
(harish@kali)-[~]
$ cd Documents

(harish@kali)-[~/Documents]
$ mkdir my_directory

(harish@kali)-[~/Documents]
$ ls
my_directory

(harish@kali)-[~/Documents]
$ cd my_directory

(harish@kali)-[~/Documents/my_directory]
$ touch my_file.txt

(harish@kali)-[~/Documents/my_directory]
$ ls
my_file.txt

(harish@kali)-[~/Documents/my_directory]
$ mv my_file.txt new_file.txt

(harish@kali)-[~/Documents/my_directory]
$ ls
new_file.txt

(harish@kali)-[~/Documents/my_directory]
$ nano new_file.txt

(harish@kali)-[~/Documents/my_directory]
$ cat new_file.txt
Hello, World!

(harish@kali)-[~/Documents/my_directory]
$ ls
new_file.txt

(harish@kali)-[~/Documents/my_directory]
$ mkdir backup

(harish@kali)-[~/Documents/my_directory]
$ ls
backup new_file.txt

(harish@kali)-[~/Documents/my_directory]
$ mv new_file.txt backup

(harish@kali)-[~/Documents/my_directory]
$ ls
backup

(harish@kali)-[~/Documents/my_directory]
$ cd backup

(harish@kali)-[~/Documents/my_directory/backup]
$ ls
new_file.txt

(harish@kali)-[~/Documents/my_directory/backup]
$ cd -
/home/harish/Documents/my_directory
```

```
(harish@kali) - [~/Documents/my_directory]
$ ls
backup
(harish@kali) - [~/Documents/my_directory]
$ rm -r backup
(harish@kali) - [~/Documents/my_directory]
$ ls
(harish@kali) - [~/Documents/my_directory]
$
```

2. Permissions and Scripting

- Create a new file called "my_script.sh".
- Edit "my_script.sh" using a text editor of your choice and add the following lines:

bash

#!/bin/bash

echo "Welcome to my script!"

echo "Today's date is \$(date)."

Save and exit the file.

- Make "my_script.sh" executable.
- Run "my_script.sh" and verify that the output matches the expected result.

All the above tasks are performed and the screenshots of the same are attached below and the screenshots itself are self explanatory.

Commands used to complete the above tasks are

1. **touch my_script.sh**
2. **nano my_script.sh**
3. **ls -l**
4. **chmod 700 my_script.sh**
5. **ls -l**
6. **./my_script.sh**

Output:

```
(harish@kali)-[~/Documents/my_directory]
$ touch my_script.sh

(harish@kali)-[~/Documents/my_directory]
$ nano my_script.sh

(harish@kali)-[~/Documents/my_directory]
$ ls -l
total 4
-rw-r--r-- 1 harish harish 73 May 30 15:44 my_script.sh

(harish@kali)-[~/Documents/my_directory]
$ chmod 700 my_script.sh

(harish@kali)-[~/Documents/my_directory]
$ ls -l
total 4
-rwx----- 1 harish harish 73 May 30 15:44 my_script.sh

(harish@kali)-[~/Documents/my_directory]
$ my_script.sh
my_script.sh: command not found

(harish@kali)-[~/Documents/my_directory]
$ ./my_script.sh
Welcome to my script!
Today's date is Tue May 30 03:46:15 PM IST 2023.

(harish@kali)-[~/Documents/my_directory]
$ nano my_script.sh
```

File Actions Edit View Help

GNU nano 7.2

my_script.sh

#!/bin/bash

echo "Welcome to my script!"

echo "Today's date is \$(date)."

0 bytes from www.google.com: icmp_seq=1 ttl=116 time=11.6
0 bytes from www.google.com: icmp_seq=2 ttl=116 time=17.2
0 bytes from www.google.com: icmp_seq=3 ttl=116 time=17.2
0 bytes from www.google.com: icmp_seq=4 ttl=116 time=17.2
0 bytes from www.google.com: icmp_seq=5 ttl=116 time=17.2
0 bytes from www.google.com: icmp_seq=6 ttl=116 time=17.2

--- google.com ping statistics ---

6 packets transmitted, 6 received, 0% packet loss, time 5025ms

rtt min/avg/max/mdev = 15.022/17.196/17.942/1.023 ms

~

^C sudo su

/home/harish

3. Command Execution and Pipelines

- List all the processes running on your system using the "ps" command.
- Use the "grep" command to filter the processes list and display only the processes with "bash" in their name.
- Use the "wc" command to count the number of lines in the filtered output.

All the above tasks are performed and the screenshots of the same are attached below and the screenshots itself are self explanatory.

Commands used to complete the above tasks are

1. **ps -ef**
2. **ps aux | grep bash**
3. **ps aux | grep bash | wc -l**

Output:

```
(harish@kali)-[~/Documents/my_directory]
$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root         1        0  0 12:41 ?           00:00:01 /sbin/init splash
root         2        0  0 12:41 ?           00:00:00 [kthreadd]
root         3        2  0 12:41 ?           00:00:00 [rcu_gp]
root         4        2  0 12:41 ?           00:00:00 [rcu_par_gp]
root         5        2  0 12:41 ?           00:00:00 [slub_flushwq]
root         6        2  0 12:41 ?           00:00:00 [netns]
root         8        2  0 12:41 ?           00:00:00 [kworker/0:0H-events_highpri]
root        10        2  0 12:41 ?           00:00:00 [mm_percpu_wq]
root        11        2  0 12:41 ?           00:00:00 [rcu_tasks_kthread]
root        12        2  0 12:41 ?           00:00:00 [rcu_tasks_rude_kthread]
root        13        2  0 12:41 ?           00:00:00 [rcu_tasks_trace_kthread]
root        14        2  0 12:41 ?           00:00:01 [ksoftirqd/0]
root        15        2  0 12:41 ?           00:00:11 [rcu_preempt]
root        16        2  0 12:41 ?           00:00:00 [migration/0]
root        18        2  0 12:41 ?           00:00:00 [cpuhp/0]
root        19        2  0 12:41 ?           00:00:00 [cpuhp/1]
root        20        2  0 12:41 ?           00:00:01 [migration/1]
root        21        2  0 12:41 ?           00:00:07 [ksoftirqd/1]
root        23        2  0 12:41 ?           00:00:02 [kworker/1:0H-kblockd]
root        26        2  0 12:41 ?           00:00:00 [kdevtmpfs]
root        27        2  0 12:41 ?           00:00:00 [inet_frag_wq]
root        28        2  0 12:41 ?           00:00:00 [kauditd]
root        29        2  0 12:41 ?           00:00:00 [khungtaskd]
root        30        2  0 12:41 ?           00:00:00 [oom_reaper]
root        32        2  0 12:41 ?           00:00:00 [writeback]
root        33        2  0 12:41 ?           00:00:01 [kcompactd0]
root        34        2  0 12:41 ?           00:00:00 [ksmd]
root        35        2  0 12:41 ?           00:00:01 [khugepaged]
root        36        2  0 12:41 ?           00:00:00 [kintegrityd]
root        37        2  0 12:41 ?           00:00:00 [kblockd]
root        38        2  0 12:41 ?           00:00:00 [blkcg_punt_bio]
root        39        2  0 12:41 ?           00:00:00 [tpm_dev_wq]
root        40        2  0 12:41 ?           00:00:00 [edac-poller]
root        41        2  0 12:41 ?           00:00:00 [devfreq_wq]
root        42        2  0 12:41 ?           00:00:04 [kworker/0:1H-kblockd]
root        43        2  0 12:41 ?           00:00:00 [kswapd0]
root        51        2  0 12:41 ?           00:00:00 [kthrotld]
root        53        2  0 12:41 ?           00:00:00 [acpi_thermal_pm]
root        54        2  0 12:41 ?           00:00:00 [xenbus_probe]
root        55        2  0 12:41 ?           00:00:00 [mld]
root        57        2  0 12:41 ?           00:00:00 [ipv6_addrconf]
root        62        2  0 12:41 ?           00:00:00 [kstrp]
root        67        2  0 12:41 ?           00:00:00 [zswap-shrink]
root        68        2  0 12:41 ?           00:00:00 [kworker/u5:0]
root       138        2  0 12:41 ?           00:00:00 [cryptd]
root       169        2  0 12:41 ?           00:00:00 [ata_sff]
root       170        2  0 12:41 ?           00:00:00 [scsi_eh_0]

(harish@kali)-[~/Documents/my_directory]
$ ps aux | grep bash
harish    15794  0.0  0.1  8248  5036 pts/0    Ss   13:12   0:00 /bin/bash
root      36261  0.0  0.1  8108  4644 pts/1    Ss+  13:56   0:00 /bin/bash
harish    39084  0.0  0.1  9524  5676 pts/2    S    14:02   0:00 /bin/bash -i
harish    97749  0.0  0.0  6332  2120 pts/2    S+   16:07   0:00 grep --color=auto bash

(harish@kali)-[~/Documents/my_directory]
$ ps aux | grep bash | wc -l
4
```