Linux Commands

- VirtualBox Network Settings and Initial Update
- VM Management - Snapshot, Clone, Export VM
- Terminal - One of the most common ways to interact with linux-based system is via the command-line (SHELL)
    - user $
    - root #
- Shell - terminal/console

```
sh – Bourn shell (Foundation – important
tasks, scripting language)
bash – Bourne-Again shell
ksh – Korn shell (handles loop syntax
better than bash)
zsh – Z shell

To verify current shell execute ps $$
```

- Package Management - apt, dpkg

```
man apt
sudo apt update
sudo apt upgrade metasploit-framework
apt-cache search <packagename>
apt install <packagename>
apt remove --purge <packagename>
apt autoremove
dpkg -i <packagename>
```

- System information

```
uname
hostname
hostname -I
whoami
id
ifconfig
ip addr
ip a
```

- Moving around

```
pwd (Absolute and Relative path)
ls
ls -larti
```

```
cd
whereis
tree
```

- File operations

```
touch
mkdir
cat
cat /etc/os-release
cat /etc/passwd
cat /etc/shadow
cat ~/.bash_history
nano <filename>
leafpad <filename>
mousepad <filename>
pluma <filename>
file
cp
mv
rmdir
rm -rf
```

- cat when used with single redirection operator (>) it will write or replace the existing contents of the

file.

- To append content to the existing file we must use two redirection operators (>>).
- mv command is used to move one file contents to another file or to move a file into existing directory.
- mv command is also used to rename a file or directory. To rename we need to provide file or directory name that exists followed by file or directory name that does not exist.

```
nano <filename>
ctrl+x
y
enter


leafpad <filename>
mousepad <filename>
pluma <filename>
```

To kill process `ctrl+c`

To pause the process `ctrl+z`

- Read the content of /usr/share/wordlists/dirb/small.txt and copy the first 7 lines in to file named as result.txt on users desktop.
    - What is the file size?
    - How many lines contain the letter 'a'?
- Managing Users

```
#Create user with home directory
adduser <username>
```

- Managing Groups
    - Primary group - By default linux will create a group with same username. It is recorded in /etc/passwd
    - Secondary group - The group to which users are added. It is recorded in /etc/group file.

```
#To verify which groups the user belongs to
groups <username>
#To add new group
addgroup <groupname>
```

```
#To add a user to sudo group
usermod -aG <groupname> <username>
usermod -aG sudo <username>
usermod -rG sudo <username>
```

- File permissions
    - To change ownership `chown`
    - To change permissions `chmod`

```
user/owner - u
group - g
others - o

read - r - 4
write - w - 2
execute - x - 1
```

```
#To change file or directory ownership
sudo chown root:kali <file/directoryname>

#To change file or directory permissions
sudo chmod o+w <file/directoryname>
```

```
sudo chmod g-r <file/directoryname>

sudo chmod +x <file/directoryname>

sudo chmod 755 <file/directoryname>
```

- Executing commands as privileged user

```
whoami
sudo whoami

#To verify sudo rights of a user
sudo -l
#To allow current user to run a loginshell as root user
sudo -i
#We can use su to access user account that has been disabled
sudo su
su -
```

- The contents of /etc/sudoers

```
root ALL=(ALL:ALL) ALL
```

The first field indicates the username that the rule will apply to (**root**).

The 1st ALL indicates that this rule applies to **all hosts**.
The 2nd ALL indicates that the **root** user can run commands as **all users**.
The 3rd ALL indicates that the **root** user can run commands as **all groups**.
The last ALL indicates that the rules apply to **all commands**.

```
sudo visudo –f /etc/sudoers

%ceh ALL=ALL, /usr/bin/cat
%sudo ALL=ALL, !/bin/nmap

ben   ALL=(ALL:ALL) /usr/bin/cat
kali  ALL=(root) NOPASSWD: /usr/bin/cat
nik   ALL=(bob) NOPASSWD: /usr/bin/nano
```

- Streams, Redirection and piping

- File streams :
    - standard output - stdout (1)
    - standard input - stdin (0)
    - standard error - stderr (2)
- Output redirects : >, >>
- Input redirects: <
- Piping: |

```
ls -l > file1
ls /etc >> file1
cat file1 file2 file3 > file4

cat < names.txt

ls -l /root 2>error.txt
locate ls 2>error.txt

ls -1 | wc -l
ls -l /etc | more

cp
mv
rmdir
rm -rf
```

```
date
time
cal
```

- Searching files
    - `locate` - uses a prebuilt database, which should be regularly updated - `sudo updatedb`
    - `find` - to recursively search any given path for various files

```
locate <keyword/filename>
locate nc.exe

find . -name "file"
find / -name *.nse
find / -name *.conf 2>errors.txt
find / -name "*.txt" 2>/dev/null
find / ! -user kali -type f
```

2>&1 - Send standard error to where ever standard output is being redirected

- Take backups

```
tar -cf <backup.tar> *
tar -rf <existingarchive> <newfiletoappend>

tar -xvf backup.tar

gzip <file/dir names>
gzip -d <backup.tar.gz>
gunzip <backup.tar.gz>

zip <newfilename.zip> <filestocompress>
unzip <filename.zip>
zip -e <newfilename.zip> <filestocompress>

zip
bzip2
xz
```

- more - view text file one page at a time
- less - same as more with navigation
- head - by default display first 10 lines of file
- tail - display last 10 lines of a file

```
more <filename>
more -10 <filename>
less <filename>
head -15 <filename>
tail <filename>
```

- Text processing
    - cut - to cut parts of lines from specified file
    - grep (global regular expression print)- searches a file for a particular pattern of characters, and displays all lines that contain that pattern.

```
#To display first or specific characters of every line in file
cut -c1 <filename>
cut -c1,2,4 <filename>
cut -c1-5 <filename>
cut -d : -f 1 /etc/passwd
ls -l | cut -c2-4

env | grep SHELL
ls | grep txt
lscpu | grep "model name"
```

```
lscpu | grep -i "model name"
grep "kali" /etc/passwd
grep "/bin/false" /etc/passwd
grep John contacts.txt
grep -w John contacts.txt
grep -wi John contacts.txt
grep ^J contacts.txt
grep .com$ contacts.txt
grep -win -B4 john contacts.txt
grep -win -A4 john contacts.txt
grep -e "Cartel" -e "Hacker" about.txt
grep -E "Naveen|Kumar" names.txt
egrep -i "Naveen|Kumar" names.txt
```