Name : Shriram Karpoora Sundara Pandian

Course : CSEC 600 Introduction to Cyber Security

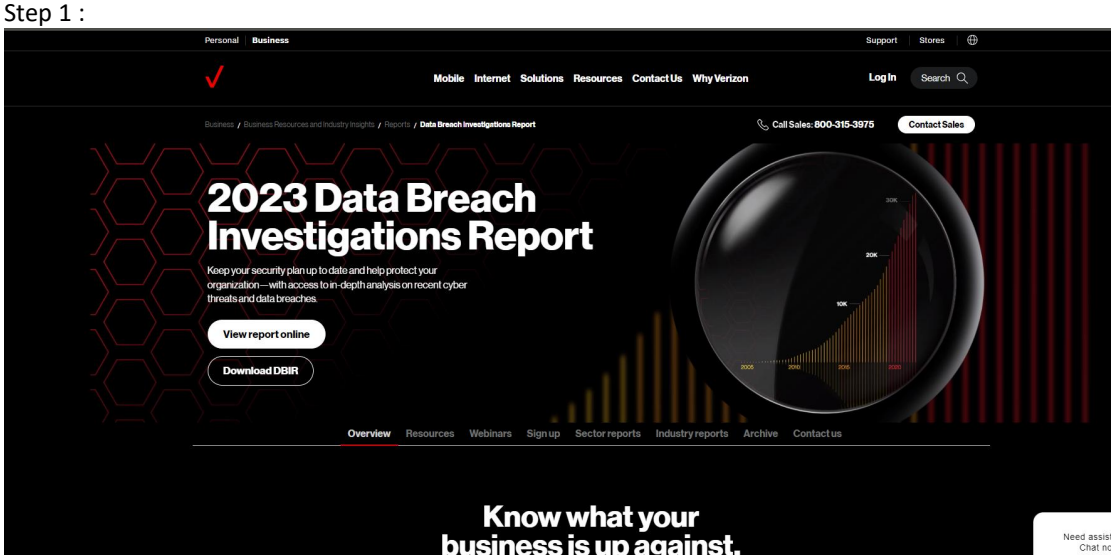
Title : System Administration 1

Lab : 8

Chapter : 2 (General Security Concepts)

Exercise 2. 01 :

Step 1 :



Step 2 :
The three takeaways from the project :
1. DoS and Ransomware attacks are getting more common and increasing year by year which makes lot of challenges to scale the networks and making backups critical for saving the files by digital forensics.
2. Having a Multifactor authentication is very important for securing the accounts and adding up another layer of security for the file or system.
3. Enhancing the security policy and access controls are crucial to avoid the threats through human error. Giving proper training to employees are also mandatory.

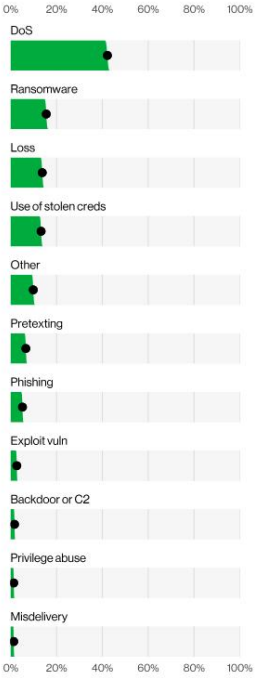


Figure 15. Top Action varieties in incidents (n=14,829)

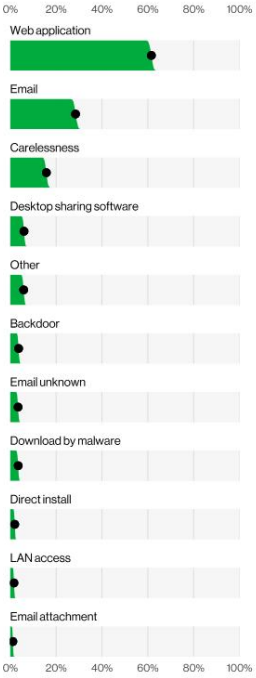


Figure 16. Top Action vectors in breaches (n=3,194)

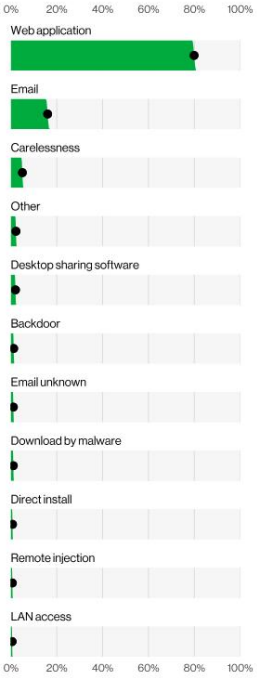
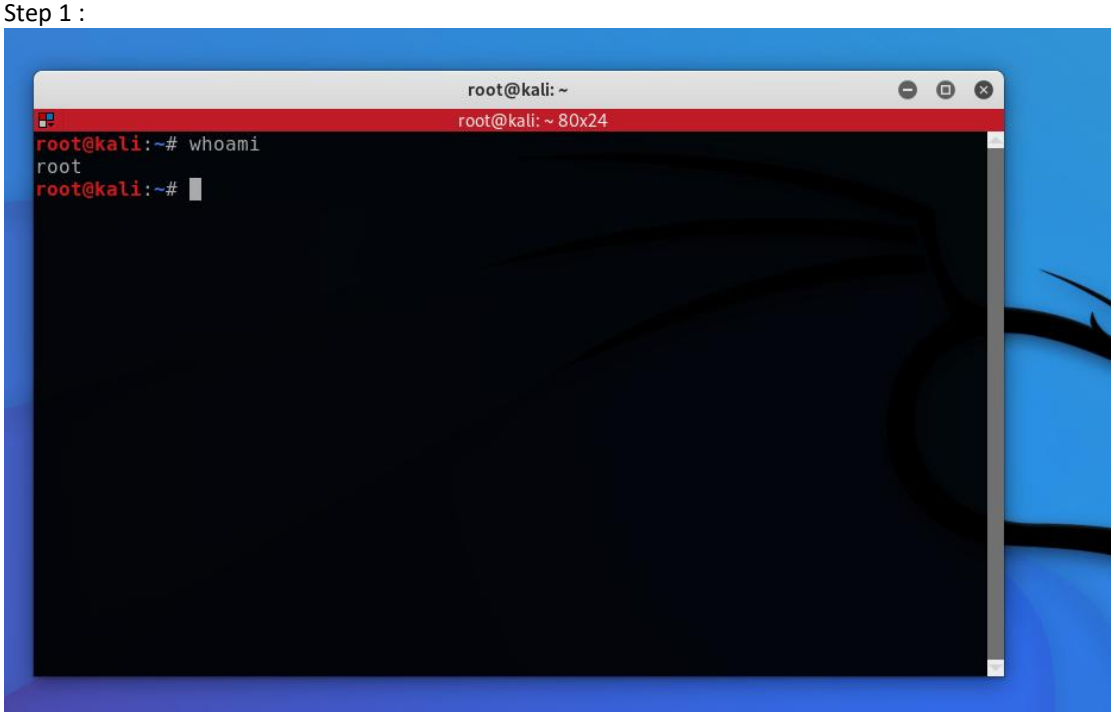


Figure 17. Top Action vectors in incidents (n=10,502)

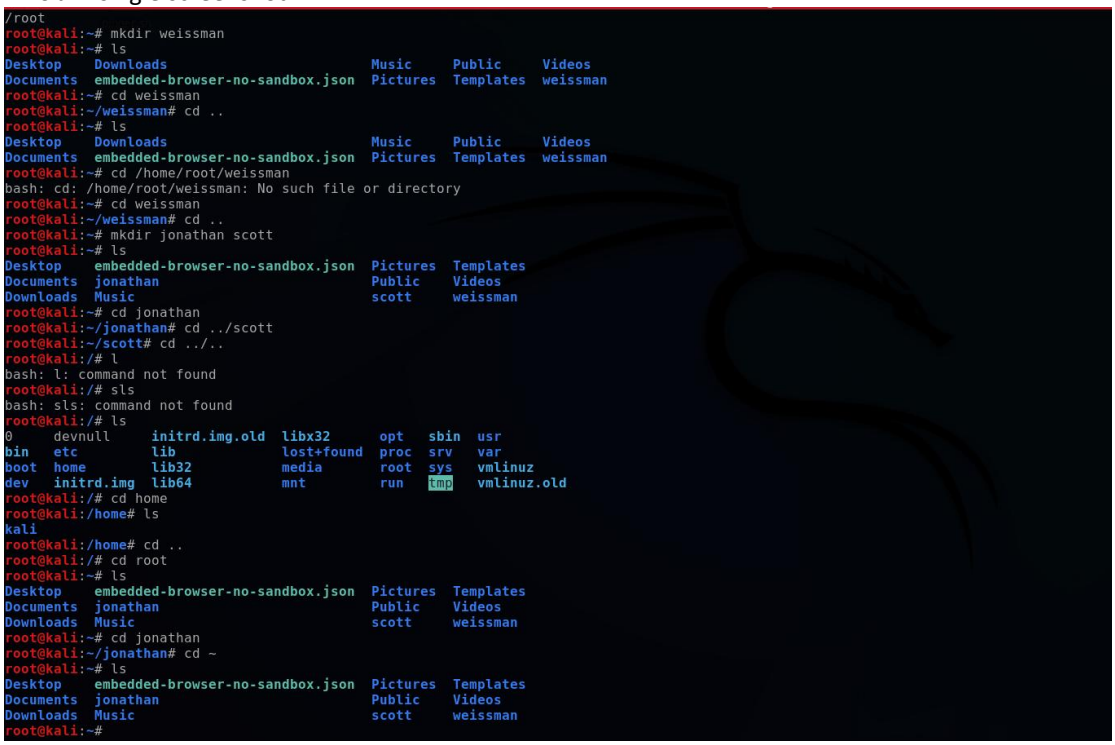
Step 3 :
DOS attack gives breach in availability.
Ransomware attack breach in availability
Backdoor attacks and vulnerabilities leads to breach in confidentiality.
Stolen credentials gives and challenges integrity.

Exercise : 2. 02 :
All lab activity by default I am using with root user privilege.
Step 1 :



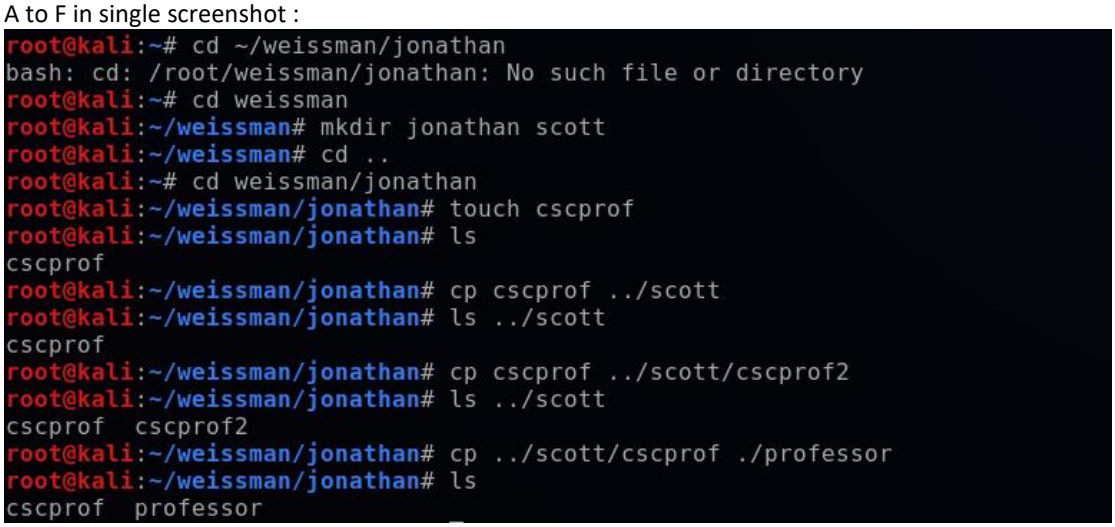
Step 2 :

A To J in single screenshot :



Step 3 :

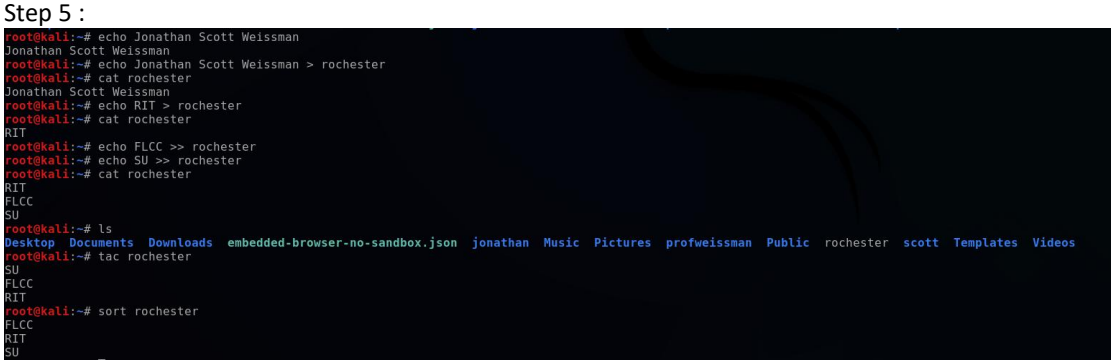
A to F in single screenshot :



Step 4 :

A to E screenshot :

Step 5 :



Exercise 2. 03 :

Step 1 :

```
root@kali:~# adduser jsw
Adding user `jsw' ...
Adding new group `jsw' (1001) ...
Adding new user `jsw' (1001) with group `jsw' ...
Creating home directory `/home/jsw' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for jsw
Enter the new value, or press ENTER for the default
        Full Name []: jsw
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@kali:~# passwd jsw
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
root@kali:~# passwd jsw
New password:
Retype new password:
passwd: password updated successfully
root@kali:~# su jsw
┌──(jsw㉿kali)-[/root]
└─$ whoami
jsw

┌──(jsw㉿kali)-[/root]
└─$ exit
exit
root@kali:~# whoami
root
```

Step 2 :

A. Vim bob :

```
root@kali: ~ 190x47
10/20/2023
~
~
~
"bob" 1L, 11B
```

B to F Screenshot :

```
root@kali:~# vim bob
root@kali:~# bob
bash: bob: command not found
root@kali:~# ./bob
bash: ./bob: Permission denied
root@kali:~# ls -l bob
-rw-r--r-- 1 root root 11 Oct 20 23:07 bob
root@kali:~# chmod 744 bob
root@kali:~# ls -l bob
-rwxr--r-- 1 root root 11 Oct 20 23:07 bob
root@kali:~# ./bob
./bob: line 1: 10/20/2023: No such file or directory
root@kali:~# ls
bob  Desktop  Documents  Downloads  embedded-browser-no-sandbox.json  jonathan  Music  Pictures  profweissman  Public  rochester  scott  Templates  Videos
root@kali:~# ./bob
./bob: line 1: 10/20/2023: No such file or directory
```

Step 3 :
A to u in single screenshot :

```
root@kali:~# mkdir monroe
root@kali:~# ls -l | grep monroe
drwxr-xr-x 2 root root 4096 Oct 20 23:13 monroe
root@kali:~# chmod 754 monroe
root@kali:~# ls -l | grep monroe
drwxr-xr-- 2 root root 4096 Oct 20 23:13 monroe
root@kali:~# cd monroe
root@kali:~/monroe# echo meadowbrook > brighton
root@kali:~/monroe# cat brighton
meadowbrook
root@kali:~/monroe# ls -l brighton
-rw-r--r-- 1 root root 12 Oct 20 23:15 brighton
root@kali:~/monroe# su jsw
┌──(jsw㉿kali)-[/root/monroe]
└─$ ls -l
ls: cannot open directory '.': Permission denied

┌──(jsw㉿kali)-[/root/monroe]
└─$ exit
exit
root@kali:~/monroe# cd ..
root@kali:~# chomod 755 monroe
bash: chomod: command not found
root@kali:~# chmod 755 monroe
root@kali:~# ls -l | grep monroe
drwxr-xr-x 2 root root 4096 Oct 20 23:15 monroe
root@kali:~# cd monroe
root@kali:~/monroe# su jsw
┌──(jsw㉿kali)-[/root/monroe]
└─$ ls -l
total 4
-rw-r--r-- 1 root root 12 Oct 20 23:15 brighton

┌──(jsw㉿kali)-[/root/monroe]
└─$ cat brighton
meadowbrook

┌──(jsw㉿kali)-[/root/monroe]
└─$ echo upstate > newyork
bash: newyork: Permission denied

┌──(jsw㉿kali)-[/root/monroe]
└─$ echo hi >> brighton
bash: brighton: Permission denied
```

Step 4 :
A to L screenshot :

```
root@kali:~/monroe# cd ..
root@kali:~# chmod 777 monroe
root@kali:~# ls -l | grep monroe
drwxrwxrwx 2 root root 4096 Oct 20 23:15 monroe
root@kali:~# cd monroe
root@kali:~/monroe# chmod 777 brighton
root@kali:~/monroe# ls -l | brighton
bash: brighton: command not found
root@kali:~/monroe# ls
brighton
root@kali:~/monroe# ls -l | brighton
bash: brighton: command not found
root@kali:~/monroe# ls -l
total 4
-rwxrwxrwx 1 root root 12 Oct 20 23:15 brighton
root@kali:~/monroe# su jsw
┌──(jsw㉿kali)-[/root/monroe]
└─$ echo upstate > newyork

┌──(jsw㉿kali)-[/root/monroe]
└─$ echo hi >> brighton

┌──(jsw㉿kali)-[/root/monroe]
└─$ cat newyork brighton
upstate
meadowbrook
hi

┌──(jsw㉿kali)-[/root/monroe]
└─$ exit
exit
root@kali:~/monroe# cd ..
```

Step 5 :
A to Y screenshot :

```
root@kali:~# mkdir stickybit
root@kali:~# chomod 1777 stickybit
bash: chomod: command not found
root@kali:~# chmod 1777 stickybit
root@kali:~# ls -l | grep stickybit
drwxrwxrwt 2 root root 4096 Oct 20 23:35 stickybit
root@kali:~# cd stickbit
bash: cd: stickbit: No such file or directory
root@kali:~# cd stickybit
root@kali:~/stickybit# echo hi > file1
root@kali:~/stickybit# su jsw
┌──(jsw㉿kali)-[/root/stickybit]
└─$ echo hello > file2

┌──(jsw㉿kali)-[/root/stickybit]
└─$ cat file2
hello

┌──(jsw㉿kali)-[/root/stickybit]
└─$ cat file1
hi

┌──(jsw㉿kali)-[/root/stickybit]
└─$ echo more >> file2

┌──(jsw㉿kali)-[/root/stickybit]
└─$ cat file2
hello
more

┌──(jsw㉿kali)-[/root/stickybit]
└─$ echo more >> file1
bash: file1: Permission denied

┌──(jsw㉿kali)-[/root/stickybit]
└─$ rm file2

┌──(jsw㉿kali)-[/root/stickybit]
└─$ rm file1
rm: remove write-protected regular file 'file1'? y
rm: cannot remove 'file1': Operation not permitted

┌──(jsw㉿kali)-[/root/stickybit]
└─$ exit
exit
root@kali:~/stickybit# cd ..
```

Step 6 :

```
root@kali:~# echo date > pizza
root@kali:~# ls -l pizza
-rw-r--r-- 1 root root 5 Oct 20 23:39 pizza
root@kali:~# chomod 744 pizza
bash: chomod: command not found
root@kali:~# chmod 744 pizza
root@kali:~# ls -l pizza
-rwxr--r-- 1 root root 5 Oct 20 23:39 pizza
root@kali:~# sudo shown jsw pizza
sudo: shown: command not found
root@kali:~# sudo chown jsw pizza
root@kali:~# sudo chgrp jsw pizza
root@kali:~# ls -l pizza
-rwxr--r-- 1 jsw jsw 5 Oct 20 23:39 pizza
root@kali:~# su jsw
┌──(jsw㉿kali)-[/root]
└─$ ./pizza
bash: ./pizza: Permission denied

┌──(jsw㉿kali)-[/root]
└─$ exit
exit
root@kali:~# sudo chown jonathan:jonathan pizza
chown: invalid user: ‘jonathan:jonathan’
root@kali:~# sudo chown root:root pizza
root@kali:~# ls -l
total 72
-rwxr--r-- 1 root root   11 Oct 20 23:07 bob
drwxr-xr-x 2 root root 4096 Sep 23 18:15 Desktop
drwxr-xr-x 2 root root 4096 Sep 21  2022 Documents
drwxr-xr-x 2 root root 4096 Sep 26  2022 Downloads
-rwxr-xr-x 1 root root  160 Sep 16  2021 embedded-browser-no-sandbox.json
drwxr-xr-x 2 root root 4096 Oct 20 22:23 jonathan
drwxrwxrwx 2 root root 4096 Oct 20 23:28 monroe
drwxr-xr-x 2 root root 4096 Sep 21  2022 Music
drwxr-xr-x 2 root root 4096 Sep 21  2022 Pictures
-rwxr--r-- 1 root root    5 Oct 20 23:39 pizza
drwxr-xr-x 3 root root 4096 Oct 20 22:38 profweissman
drwxr-xr-x 2 root root 4096 Sep 21  2022 Public
-rw-r--r-- 1 root root   12 Oct 20 22:43 rochester
drwxr-xr-x 2 root root 4096 Oct 20 22:23 scott
drwxrwsrwx 2 root root 4096 Oct 20 23:31 sgid
drwxrwxrwt 2 root root 4096 Oct 20 23:37 stickybit
drwxr-xr-x 2 root root 4096 Sep 21  2022 Templates
drwxr-xr-x 2 root root 4096 Sep 21  2022 Videos
root@kali:~# su jsw
┌──(jsw㉿kali)-[/root]
└─$ ./pizza
```

Permission is denied for ./pizza.

Step 7 :

A to H screenshot :

```
root@kali:~# addgroup pentesters1
Adding group `pentesters1' (GID 1002) ...
Done.
root@kali:~# addgroup pentesters2
Adding group `pentesters2' (GID 1003) ...
Done.
root@kali:~# usermod -a -G pentesters1,pentesters2 jsw
root@kali:~# grep pentesters /etc/group
pentesters1:x:1002:jsw
pentesters2:x:1003:jsw
root@kali:~# groups jsw
jsw : jsw pentesters1 pentesters2
root@kali:~# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:kali
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
```

```
Debian-gdm:x:137:
kali:x:1000:
kaboxer:x:138:kali
beef-xss:x:139:
jsw:x:1001:
pentesters1:x:1002:jsw
pentesters2:x:1003:jsw
root@kali:~# usermod -G pentesters1 jsw
root@kali:~# groups jsw
jsw : jsw pentesters1
root@kali:~# usermod -a -G pentesters1,pentesters2 jsw
root@kali:~# gpasswd -d jsw pentesters2
Removing user jsw from group pentesters2
root@kali:~# group jsw
bash: group: command not found
root@kali:~# groups jsw
jsw : jsw pentesters1
root@kali:~# adduser alice
Adding user `alice' ...
Adding new group `alice' (1004) ...
Adding new user `alice' (1002) with group `alice' ...
Creating home directory `/home/alice' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alice
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
root@kali:~# deluser alice
Removing crontab ...
Removing user `alice' ...
Done.
root@kali:~# addgroup cryptographers
Adding group `cryptographers' (GID 1004) ...
Done.
root@kali:~# delcroup cryptographers
bash: delcroup: command not found
root@kali:~# delgroup cryptographers
Removing group `cryptographers' ...
Done.
```

Step 8 :

Screenshots :

```
root@kali:~# cd /
root@kali:/# ls
0    boot  devnull  home        initrd.img.old  lib32  libx32      media  opt   root  sbin  sys  usr  vmlinuz
bin  dev   etc      initrd.img  lib             lib64  lost+found  mnt    proc  run   srv   tmp  var  vmlinuz.old
root@kali:/# ls -a
.   0    boot   dev      etc   initrd.img      lib    lib64   lost+found  mnt  proc  run   srv  tmp  var      vmlinuz.old
..  bin  .cache  devnull  home  initrd.img.old  lib32  libx32  media       opt  root  sbin  sys  usr  vmlinuz
root@kali:/# ls -A
0    boot   dev      etc   initrd.img      lib    lib64   lost+found  mnt  proc  run   srv  tmp  var      vmlinuz.old
bin  .cache  devnull  home  initrd.img.old  lib32  libx32  media       opt  root  sbin  sys  usr  vmlinuz
root@kali:/# ls -F
0    boot/  devnull  home/        initrd.img.old@  lib32@  libx32@      media/  opt/   root/  sbin@  sys/  usr/  vmlinuz@
bin@  dev/   etc/    initrd.img@  lib@             lib64@  lost+found/  mnt/    proc/  run/   srv/   tmp/  var/  vmlinuz.old@
root@kali:/# ls -F /usr/bin/ping
/usr/bin/ping*
root@kali:/# ls -aF
./   0    boot/   dev/    etc/   initrd.img@      lib@    lib64@   lost+found/  mnt/  proc/  run/   srv/  tmp/  var/      vmlinuz.old@
../  bin@  .cache/  devnull  home/  initrd.img.old@  lib32@  libx32@  media/       opt/  root/  sbin@  sys/  usr/  vmlinuz@
root@kali:/# ls -Fa
./   0    boot/   dev/    etc/   initrd.img@      lib@    lib64@   lost+found/  mnt/  proc/  run/   srv/  tmp/  var/      vmlinuz.old@
../  bin@  .cache/  devnull  home/  initrd.img.old@  lib32@  libx32@  media/       opt/  root/  sbin@  sys/  usr/  vmlinuz@
root@kali:/# ls -R
.:
0    boot  devnull  home        initrd.img.old  lib32  libx32      media  opt   root  sbin  sys  usr  vmlinuz
bin  dev   etc      initrd.img  lib             lib64  lost+found  mnt    proc  run   srv   tmp  var  vmlinuz.old

./boot:
config-5.18.0-kali7-amd64  grub  initrd.img-5.18.0-kali7-amd64  System.map-5.18.0-kali7-amd64  vmlinuz-5.18.0-kali7-amd64

./boot/grub:
fonts  grub.cfg  grubenv  i386-pc  locale  themes  unicode.pf2

./boot/grub/fonts:
unicode.pf2

./boot/grub/i386-pc:
915resolution.mod     cmosdump.mod    exfat.mod           gcry_twofish.mod         linux16.mod     mpi.mod          pata.mod          setjmp.mod          uhci.mod
acpi.mod              cmostest.mod    exfctest.mod        gcry_whirlpool.mod       linux.mod       msdospart.mod    pbkdf2.mod        setjmp_test.mod     usb_keyboard.mod
adler32.mod           cmp.mod         ext2.mod            gdb.mod                  loadenv.mod     mul_test.mod     pbkdf2_test.mod   setpci.mod          usb.mod
affs.mod              cmp_test.mod    extcmd.mod          geli.mod                 loopback.mod    multiboot2.mod   pcidump.mod       sfs.mod             usbms.mod
afs.mod               command.lst     f2fs.mod            gettext.mod              lsacpi.mod      multiboot.mod    pci.mod           shift_test.mod      usbserial_common.mod
afsplitter.mod        configfile.mod  fat.mod             gfxmenu.mod              lsapm.mod       nativedisk.mod   pgp.mod           signature_test.mod  usbserial_ftdi.mod
ahci.mod              core.img        file.mod            gfxterm_background.mod   lsmmap.mod      net.mod          plan9.mod         sleep.mod           usbserial_pl2303.mod
all_video.mod         cpio_be.mod     font.mod            gfxterm_menu.mod         ls.mod          newc.mod         play.mod          sleep_test.mod      usbserial_usbdebug.mod
aout.mod              cpio.mod        freedos.mod         gfxterm.mod              lspci.mod       nilfs2.mod       png.mod           smbios.mod          usbtest.mod
archelp.mod           cpuid.mod       fshelp.mod          gptsync.mod              luks2.mod       normal.mod       priority_queue.mod  spkmodem.mod        vbe.mod
ata.mod               crc64.mod       fs.lst              gzio.mod                 luks.mod        ntfscomp.mod     probe.mod         squash4.mod         verifiers.mod
at_keyboard.mod       cryptodisk.mod  functional_test.mod  halt.mod                lvm.mod         ntfs.mod         procfs.mod        strtoull_test.mod   vga.mod
```

```
grub.cfg
grubenv
i386-pc
locale
themes
unicode.pf2

root@kali:/# ls -R | less
```

```
root@kali:/# find / -name ping 2>&1 | grep -v "Permission denied"
/usr/bin/ping
/usr/share/bash-completion/completions/ping
/usr/lib/python3/dist-packages/faraday_plugins/plugins/repo/ping
```

```
FIND(1)                                    General Commands Manual                                    FIND(1)

NAME
       find - search for files in a directory hierarchy

SYNOPSIS
       find [-H] [-L] [-P] [-D debugopts] [-Olevel] [starting-point...] [expression]

DESCRIPTION
       This  manual  page  documents  the  GNU version of find.  GNU find searches the directory tree rooted at each given starting-point by evaluating the given expression from left to
       right, according to the rules of precedence (see section OPERATORS), until the outcome is known (the left hand side is false for and operations, true for or), at which point find
       moves on to the next file name.  If no starting-point is specified, `.' is assumed.

       If  you are using find in an environment where security is important (for example if you are using it to search directories that are writable by other users), you should read the
       `Security Considerations' chapter of the findutils documentation, which is called Finding Files and comes with findutils.  That document also includes a lot more detail and  dis-
       cussion than this manual page, so you may find it a more useful source of information.

OPTIONS
       The -H, -L and -P options control the treatment of symbolic links.  Command-line arguments following these are taken to be names of files or directories to be examined, up to the
       first argument that begins with `-', or the argument `(' or `!'.  That argument and any following arguments are taken to be the expression describing what is to be searched  for.
       If no paths are given, the current directory is used.  If no expression is given, the expression -print is used (but you should probably consider using -print0 instead, anyway).

       This  manual page talks about `options' within the expression list.  These options control the behaviour of find but are specified immediately after the last path name.  The five
       `real' options -H, -L, -P, -D and -O must appear before the first path name, if at all.  A double dash -- could theoretically be used to signal that any remaining  arguments  are
       not  options,  but  this  does  not really work due to the way find determines the end of the following path arguments: it does that by reading until an expression argument comes
       (which also starts with a `-').  Now, if a path argument would start with a `-', then find would treat it as expression argument instead.  Thus, to ensure that all  start  points
       are  taken  as  such,  and especially to prevent that wildcard patterns expanded by the calling shell are not mistakenly treated as expression arguments, it is generally safer to
       prefix wildcards or dubious path names with either `./' or to use absolute path names starting with '/'.  Alternatively, it is generally safe though non-portable to use  the  GNU
       option -files0-from to pass arbitrary starting points to find.

       -P     Never  follow  symbolic links.  This is the default behaviour.  When find examines or prints information about files, and the file is a symbolic link, the information used
              shall be taken from the properties of the symbolic link itself.

       -L     Follow symbolic links.  When find examines or prints information about files, the information used shall be taken from the properties of the file to which the link points,
              not from the link itself (unless it is a broken symbolic link or find is unable to examine the file to which the link points).  Use of this option implies -noleaf.  If you
              later use the -P option, -noleaf will still be in effect.  If -L is in effect and find discovers a symbolic link to a subdirectory  during  its  search,  the  subdirectory
              pointed to by the symbolic link will be searched.

              When  the -L option is in effect, the -type predicate will always match against the type of the file that a symbolic link points to rather than the link itself (unless the
              symbolic link is broken).  Actions that can cause symbolic links to become broken while find is executing (for example -delete) can give rise to confusing behaviour.  Us-
              ing -L causes the -lname and -ilname predicates always to return false.

       -H     Do  not  follow  symbolic  links,  except while processing the command line arguments.  When find examines or prints information about files, the information used shall be
              taken from the properties of the symbolic link itself.  The only exception to this behaviour is when a file specified on the command line is a symbolic link, and the  link
              can  be  resolved.  For that situation, the information used is taken from whatever the link points to (that is, the link is followed).  The information about the link it-
              self is used as a fallback if the file pointed to by the symbolic link cannot be examined.  If -H is in effect and one of the paths specified on the command line is a sym-
 Manual page find(1) line 1 (press h for help or q to quit)
```

```
root@kali:/# vim months
root@kali:/# head months
January
February
March
April
June
July
August
September
October
November
root@kali:/# head -7 months
January
February
March
April
June
July
August
root@kali:/# head -n 7 months
January
February
March
April
June
July
August
root@kali:/# head -n -3 months
January
February
March
April
June
July
August
September
```

```
root@kali:/# tail months
February
March
April
June
July
August
September
October
November
December
root@kali:/# tail -3 months
October
November
December
root@kali:/# tail -n -3 months
October
November
December
root@kali:/# tail +3 months
March
April
June
July
August
September
October
November
December
root@kali:/# tail -n +3 months
March
April
June
July
August
September
October
November
December
```

Exercise 2. 04 :
My kali is configured by default with root user.
Step 1 :

```
┌──(kp㉿kali)-[~]
└─$ sudo passwd root
New password:
Retype new password:
passwd: password updated successfully

┌──(kp㉿kali)-[~]
└─$ sudo passwd -l root
passwd: password changed.

┌──(kp㉿kali)-[~]
└─$ sudo passwd -u root
passwd: password changed.
```

Step 2 :
A and B :

```
┌──(kp㉿kali)-[~]
└─$ su
Password:
root@kali:/home/kp# exit
exit

┌──(kp㉿kali)-[~]
└─$ su -
Password:
root@kali:~# exit
logout
```

C.

```
┌──(kp㉿kali)-[~]
└─$ cat /etc/shadow
cat: /etc/shadow: Permission denied

┌──(kp㉿kali)-[~]
└─$ su -c 'cat /etc/shadow'
Password:
root:$y$j9T$IM6lmhwaNKhLJ/3PXos2r.$y0sKlvuoWee3HfYzZrlS8oY8.C0XijWl5j8Nn/tdhf7:1
9651:0:99999:7:::
daemon:*:19257:0:99999:7:::
bin:*:19257:0:99999:7:::
sys:*:19257:0:99999:7:::
sync:*:19257:0:99999:7:::
games:*:19257:0:99999:7:::
man:*:19257:0:99999:7:::
lp:*:19257:0:99999:7:::
mail:*:19257:0:99999:7:::
news:*:19257:0:99999:7:::
uucp:*:19257:0:99999:7:::
proxy:*:19257:0:99999:7:::
www-data:*:19257:0:99999:7:::
backup:*:19257:0:99999:7:::
```

At first command permission is denied .

But on the second command we able to run it well.

D to F.

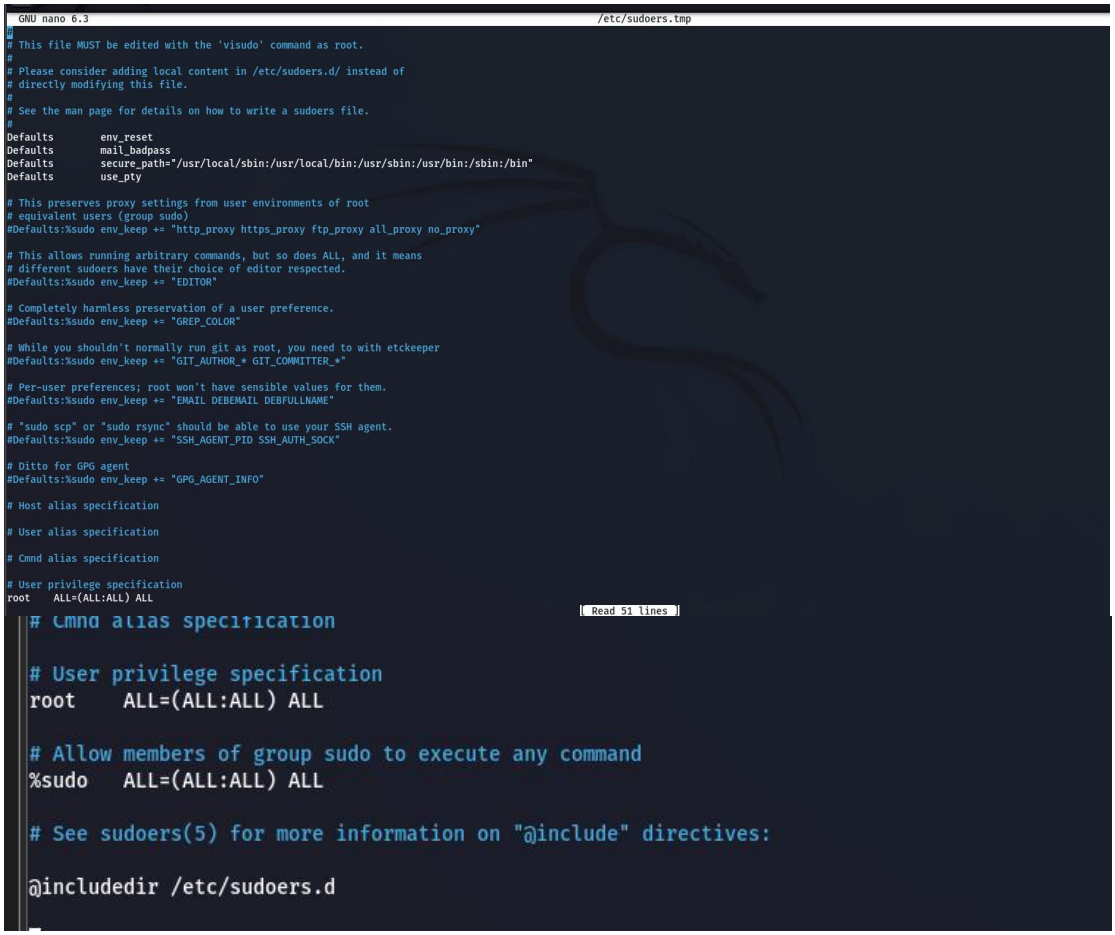
```
┌──(kp㉿kali)-[~]
└─$ sudo passwd -l root
[sudo] password for kp:
passwd: password changed.

┌──(kp㉿kali)-[~]
└─$ su -c 'cat /etc/shadow'
Password:
su: Authentication failure

┌──(kp㉿kali)-[~]
└─$ sudo su
root@kali:/home/kp# exit
exit
```

Step 3 :

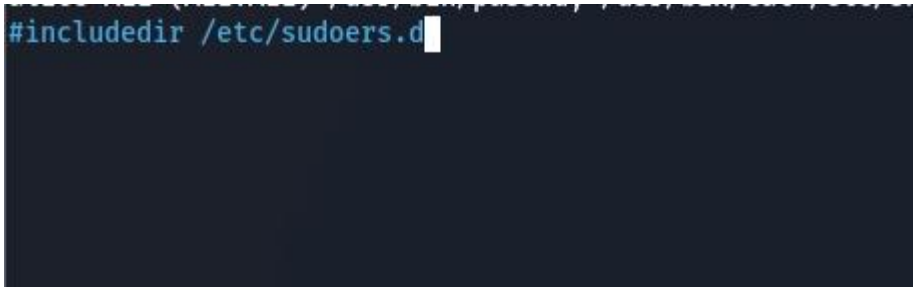
A


GNU nano 6.3 /etc/sudoers.tmp
#
This file MUST be edited with the 'visudo' command as root.
#
Please consider adding local content in /etc/sudoers.d/ instead of
directly modifying this file.
#
See the man page for details on how to write a sudoers file.
#
Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
Defaults use_pty
This preserves proxy settings from user environments of root
equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"
This allows running arbitrary commands, but so does ALL, and it means
different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"
Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"
While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"
Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"
"sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"
Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"
Host alias specification
User alias specification
Cmnd alias specification
User privilege specification
root ALL=(ALL:ALL) ALL
[Read 51 lines]
Cmnd alias specification
User privilege specification
root ALL=(ALL:ALL) ALL
Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL
See sudoers(5) for more information on "@include" directives:
@includedir /etc/sudoers.d

C.


kp@kali: ~
GNU nano 6.3 /etc/sudoers.d/alice.tmp *
alice ALL=(ALL:ALL) /usr/bin/passwd, /usr/bin/cat /etc/shadow

D.


#includedir /etc/sudoers.d

B, E :

```
┌──(kp㉿kali)-[~]
└─$ sudo visudo
visudo: /etc/sudoers.tmp unchanged

┌──(kp㉿kali)-[~]
└─$ sudo visudo -f /etc/sudoers.d/alice
[sudo] password for kp:
visudo: /etc/sudoers.d: too many levels of includes
What now?
Options are:
  (e)dit sudoers file again
  e(x)it without saving changes to sudoers file
  (Q)uit and save changes to sudoers file (DANGER!)

What now?
Options are:
  (e)dit sudoers file again
  e(x)it without saving changes to sudoers file
  (Q)uit and save changes to sudoers file (DANGER!)

What now? x

┌──(kp㉿kali)-[~]
└─$
```

Step 4 :

A.

```
┌──(kp㉿kali)-[~]
└─$ sudo adduser alice
[sudo] password for kp:
Sorry, try again.
[sudo] password for kp:
Adding user `alice' ...
Adding new group `alice' (1002) ...
Adding new user `alice' (1002) with group `alice' ...
Creating home directory `/home/alice' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] ^[[A^[[A^[[A^[[B^[[By
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for alice
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y

┌──(kp㉿kali)-[~]
└─$ sudo adduser bob
Adding user `bob' ...
Adding new group `bob' (1003) ...
Adding new user `bob' (1003) with group `bob' ...
Creating home directory `/home/bob' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for bob
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
```

```
┌──(kp㉿kali)-[~]
└─$ sudo passwd bob
New password:
Retype new password:
passwd: password updated successfully

┌──(kp㉿kali)-[~]
└─$ sudo passwd eve
New password:
Retype new password:
passwd: password updated successfully

┌──(kp㉿kali)-[~]
└─$ s
```

B and C :

```
  GNU nano 6.3                                                    /etc/sudoers.
alice ALL=(ALL:ALL) /usr/bin/passwd, !/usr/bin/passwd bob, /usr/bin/cat /etc/shadow
```

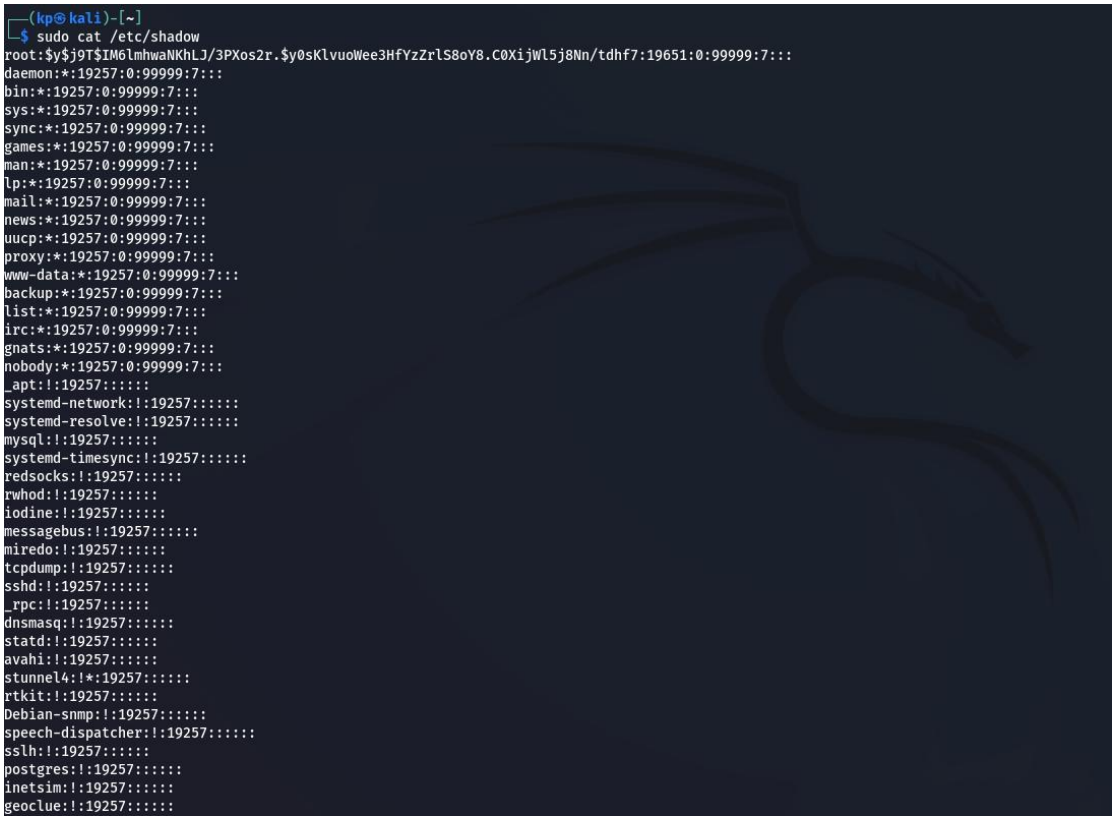
D.

```
┌──(kp㉿kali)-[~]
└─$ sudo visudo -f /etc/sudoers.d/defaults
```

E.

```
  GNU nano 6.3                                                  /etc/sudoers.d/def
Defautls insults, !lecture, passwd_timeout=1, passwd_tries=5, timestamp_timeout=10
```

F.



Lab Analysis :

1. I think because it wants to showcase its security analysis and wants promote its service. To give others the insights of cyber security measures and how threats and their actors are getting innovated each year.
They are showcasing how threats can be prevented and how good their team is at analyzing it.

2. Because it completely change the contents by overwriting of the file, it may result in data loss without any proper awareness.
3. No its not secure to use chmod 777 because it gives privilege to others to modify the file as well which makes it security risk. Any user can make use of that command, so avoiding the 777 is best option.
4. Because it wont give the direct access to anybody to make changes on kernal or or shell as they desire, only people with knowing the password can access it and while non working on root directory it will save system from other risk unexpectedly which user may do for the system, which makes switching to user good option than being in root. Its like editing the photo on photoshop with copied photo rather than working on original one.

Quiz :

1. execute
2. Breach
3. visudo
4. Command.