EPAM University Programs

DevOps external course

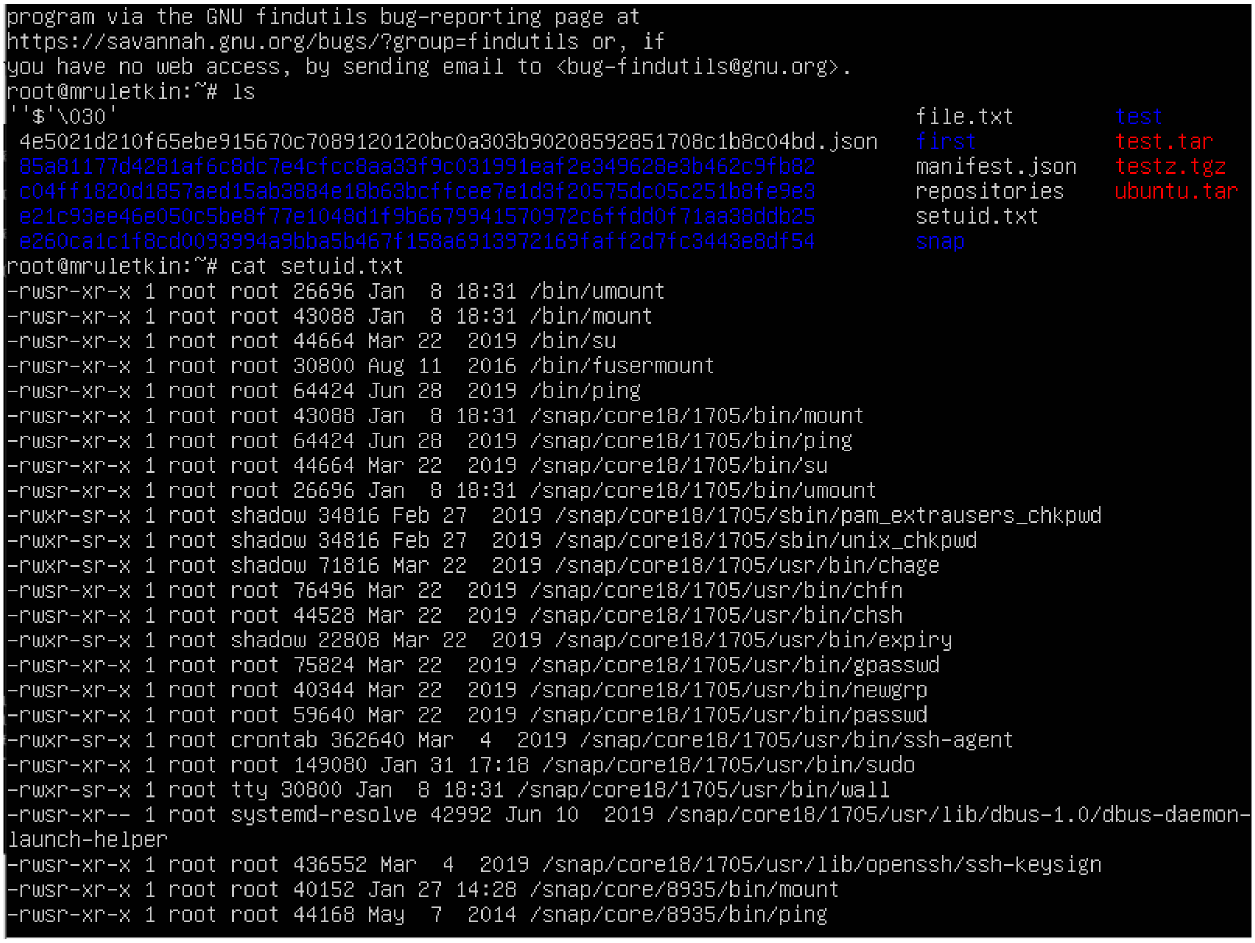
Module 4 Linux & Bash Essentials

TASK 4.5

1. To discover files with active sticky bits, use the following version of the **find** command:

sudo find / -perm /6000 -type f -exec ls -ld {} \;>setuid.txt

Put into your report a fragment of setuid.txt file. Explain meaning of parameters of the above **find** command (hint: use find’s man page).



```
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>.
root@mruletkin:~# ls
''$'\030'                                                            file.txt       test
 4e5021d210f65ebe915670c7089120120bc0a303b90208592851708c1b8c04bd.json  first          test.tar
 85a81177d4281af6c8dc7e4cfcc8aa33f9c031991eaf2e349628e3b462c9fb82       manifest.json  testz.tgz
 c04ff1820d1857aed15ab3884e18b63bcffcee7e1d3f20575dc05c251b8fe9e3       repositories   ubuntu.tar
 e21c93ee46e050c5be8f77e1048d1f9b66799415709 72c6ffdd0f71aa38ddb25       setuid.txt
 e260ca1c1f8cd0093994a9bba5b467f158a6913972169faff2d7fc3443e8df54       snap
root@mruletkin:~# cat setuid.txt
-rwsr-xr-x 1 root root 26696 Jan  8 18:31 /bin/umount
-rwsr-xr-x 1 root root 43088 Jan  8 18:31 /bin/mount
-rwsr-xr-x 1 root root 44664 Mar 22  2019 /bin/su
-rwsr-xr-x 1 root root 30800 Aug 11  2016 /bin/fusermount
-rwsr-xr-x 1 root root 64424 Jun 28  2019 /bin/ping
-rwsr-xr-x 1 root root 43088 Jan  8 18:31 /snap/core18/1705/bin/mount
-rwsr-xr-x 1 root root 64424 Jun 28  2019 /snap/core18/1705/bin/ping
-rwsr-xr-x 1 root root 44664 Mar 22  2019 /snap/core18/1705/bin/su
-rwsr-xr-x 1 root root 26696 Jan  8 18:31 /snap/core18/1705/bin/umount
-rwxr-sr-x 1 root shadow 34816 Feb 27  2019 /snap/core18/1705/sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root shadow 34816 Feb 27  2019 /snap/core18/1705/sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 71816 Mar 22  2019 /snap/core18/1705/usr/bin/chage
-rwsr-xr-x 1 root root 76496 Mar 22  2019 /snap/core18/1705/usr/bin/chfn
-rwsr-xr-x 1 root root 44528 Mar 22  2019 /snap/core18/1705/usr/bin/chsh
-rwxr-sr-x 1 root shadow 22808 Mar 22  2019 /snap/core18/1705/usr/bin/expiry
-rwsr-xr-x 1 root root 75824 Mar 22  2019 /snap/core18/1705/usr/bin/gpasswd
-rwsr-xr-x 1 root root 40344 Mar 22  2019 /snap/core18/1705/usr/bin/newgrp
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /snap/core18/1705/usr/bin/passwd
-rwxr-sr-x 1 root crontab 362640 Mar  4  2019 /snap/core18/1705/usr/bin/ssh-agent
-rwsr-xr-x 1 root root 149080 Jan 31 17:18 /snap/core18/1705/usr/bin/sudo
-rwxr-sr-x 1 root tty 30800 Jan  8 18:31 /snap/core18/1705/usr/bin/wall
-rwsr-xr-- 1 root systemd-resolve 42992 Jun 10  2019 /snap/core18/1705/usr/lib/dbus-1.0/dbus-daemon-
launch-helper
-rwsr-xr-x 1 root root 436552 Mar  4  2019 /snap/core18/1705/usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 40152 Jan 27 14:28 /snap/core/8935/bin/mount
-rwsr-xr-x 1 root root 44168 May  7  2014 /snap/core/8935/bin/ping
```

Parameters of **find** command:

- **-exec ls -ld {} \;** - run **ls** to get details about files
- **/ -perm /6000 –** searching for files with permission 6000
- **-type f –** searching for only files
- **>setuid.txt** – directing output to **setuid.txt**

2. Discovering soft and hard links.

Comment on results of these commands (place the output into your report):

```
cd #enter home directory
mkdir test #make directory “test”
cd test #enter “test”
touch test1.txt #create file “test1.txt”
echo “test1.txt” > test1.txt # redirecting output to the file.
ls -l . #output information about content in long format  ls
```

(a hard link)

```
ln test1.txt test2.txt #create hard link between test1.txt and test2.txt
ls -l .
```

(pay attention to the number of links to test1.txt and test2.txt)

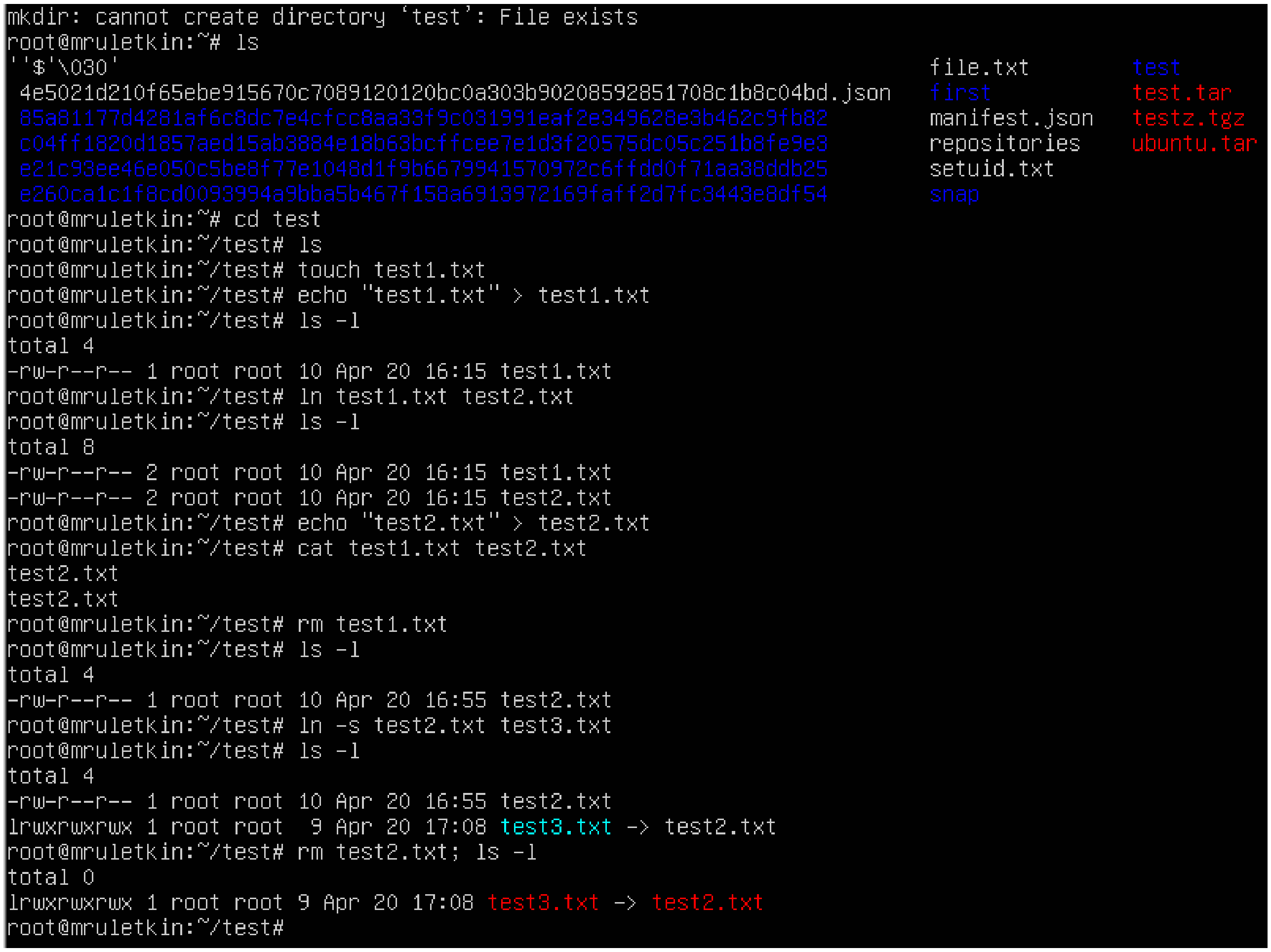
```
echo “test2.txt” > test2.txt
cat test1.txt test2.txt
rm test1.txt #remove file
ls -l .
```

(now a soft link)

```
ln -s test2.txt test3.txt # create soft link between test2.txt and test3.txt
ls -l .
```

(pay attention to the number of links to the created files)

```
rm test2.txt; ls -l . #remove file and sequential execution of ls -l
```



```
mkdir: cannot create directory ‘test’: File exists
root@mruletkin:~# ls
''$'\030'                                                                     file.txt        test
 4e5021d210f65ebe915670c7089120120bc0a303b90208592851708c1b8c04bd.json        first           test.tar
 85a81177d4281af6c8dc7e4cfcc8aa33f9c031991eaf2e349628e3b462c9fb82             manifest.json   testz.tgz
 c04ff1820d1857aed15ab3884e18b63bcffcee7e1d3f20575dc05c251b8fe9e3             repositories    ubuntu.tar
 e21c93ee46e050c5be8f77e1048d1f9b66799415709 72c6ffdd0f71aa38ddb25            setuid.txt
 e260ca1c1f8cd0093994a9bba5b467f158a6913972169faff2d7fc3443e8df54             snap
root@mruletkin:~# cd test
root@mruletkin:~/test# ls
root@mruletkin:~/test# touch test1.txt
root@mruletkin:~/test# echo "test1.txt" > test1.txt
root@mruletkin:~/test# ls -l
total 4
-rw-r--r-- 1 root root 10 Apr 20 16:15 test1.txt
root@mruletkin:~/test# ln test1.txt test2.txt
root@mruletkin:~/test# ls -l
total 8
-rw-r--r-- 2 root root 10 Apr 20 16:15 test1.txt
-rw-r--r-- 2 root root 10 Apr 20 16:15 test2.txt
root@mruletkin:~/test# echo "test2.txt" > test2.txt
root@mruletkin:~/test# cat test1.txt test2.txt
test2.txt
test2.txt
root@mruletkin:~/test# rm test1.txt
root@mruletkin:~/test# ls -l
total 4
-rw-r--r-- 1 root root 10 Apr 20 16:55 test2.txt
root@mruletkin:~/test# ln -s test2.txt test3.txt
root@mruletkin:~/test# ls -l
total 4
-rw-r--r-- 1 root root 10 Apr 20 16:55 test2.txt
lrwxrwxrwx 1 root root  9 Apr 20 17:08 test3.txt -> test2.txt
root@mruletkin:~/test# rm test2.txt; ls -l
total 0
lrwxrwxrwx 1 root root 9 Apr 20 17:08 test3.txt -> test2.txt
root@mruletkin:~/test#
```

3. I/O redirect.

Execute these commands; comment on the output.

mount #display all currently attached file systems

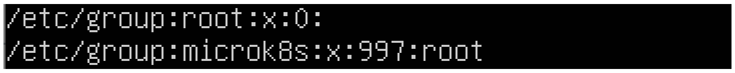
blkid #print block device attributes

mount | **grep** sda #**grep** uses output of **mount** to search files in text by template

dmesg | **grep** sda #**dsmesg** list all hardware detected by the kernel and **grep** uses its output

sudo grep -R -e “root” /etc > root_entries.txt #grep makes recursive search of files in the directory and redirect output to the file

(place only a reasonable fragment of root_entries.txt into your report)



```
/etc/group:root:x:0:
/etc/group:microk8s:x:997:root
```

It lets see if the user is a member of the group