

EPAM University Programs
DevOps external course
Module 4 Linux & Bash Essentials
TASK 4.5

Danylenko Homework

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).

```
danylenko@VM2:~$ cat setuid.txt
-rwxr-sr-x 1 root tty 10232 cep  5  2017 /var/lib/docker/overlay2/4368a95af7c7b
02cef87994031805fb5bea51351fb3d46a6ca3afd5496c779b3/diff/usr/lib/mc/cons.saver
-rwxr-sr-x 1 root tty 10232 cep  5  2017 /var/lib/docker/overlay2/83ddbdb337212
9db57204167a19dc0aafaeb7c9ed1cabea5df9d5c53b9b4cab4/diff/usr/lib/mc/cons.saver
-rwsr-xr-x 1 root root 26696 ci4  8 20:31 /var/lib/docker/overlay2/592dbfa02c2e
276195ef5b5b7fee496195c88666606934c802248639433bc009/diff/bin/umount
-rwsr-xr-x 1 root root 44664 bep 22  2019 /var/lib/docker/overlay2/592dbfa02c2e
276195ef5b5b7fee496195c88666606934c802248639433bc009/diff/bin/su
-rwsr-xr-x 1 root root 43088 ci4  8 20:31 /var/lib/docker/overlay2/592dbfa02c2e
276195ef5b5b7fee496195c88666606934c802248639433bc009/diff/bin/mount
-rwxr-sr-x 1 root shadow 34816 лют 27  2019 /var/lib/docker/overlay2/592dbfa02c
2e276195ef5b5b7fee496195c88666606934c802248639433bc009/diff/sbin/pam_extrausers
_chkpwd
-rwxr-sr-x 1 root shadow 34816 лют 27  2019 /var/lib/docker/overlay2/592dbfa02c
2e276195ef5b5b7fee496195c88666606934c802248639433bc009/diff/sbin/unix_chkpwd
```

Result: Command to find all files, starting from root dir, with any of setuid or setgid special modes set in permissions. List each file permissions and properties, and save output to file:

«**-type f**»

Search only regular files

“**-perm /6000**»

Entries with any permission bytes in 6000> “---S--S---” setuid or setgid

-exec command {} \; execute following command on each result,

{} each result entry as argument in executed command

; end of exec command line, escaped with \

ls -ld {} run ls -ld on results. -d argument doesn't do anything, and not required since we search only files.

>setuid.txt save to file

2. Discovering soft and hard links.

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

cd change directory to default - \$HOME

mkdir test	make dir test
cd test	change dir to test
touch test1.txt	update mod date of test1.txt, create if none
echo "test1.txt" > test1.txt	print text "test1.txt" into file test1.txt
ls -l .	Detailed listing of current dir
<i>(a hard link)</i>	
ln test1.txt test2.txt	create link to test1.txt with name test2.txt
ls -l .	listing of dir, both files are equal and have 2 links
<i>(pay attention to the number of links to test1.txt and test2.txt)</i>	
echo "test2.txt" > test2.txt	print text "test2.txt" into file test2.txt
cat test1.txt test2.txt	both files contain "test2" text because all changes are written into both hardlinked test2.txt file and test1.txt equally.
rm test1.txt	remove original linked file test1.txt
ls -l .	File test2.txt, that was hard-linked with test1.txt, is still in directory with all content as standalone file without links.

Simple recreation of file2.txt doesn't restore link, link needs to be created again.

```
danylenko@VM2:~/tmp$ cd
danylenko@VM2:~$ cd tmp
danylenko@VM2:~/tmp$ cd $HOME
danylenko@VM2:~$ mkdir test
danylenko@VM2:~$ cd test
danylenko@VM2:~/test$ touch test1.txt
danylenko@VM2:~/test$ echo

danylenko@VM2:~/test$ echo "test.txt"> test1.txt
danylenko@VM2:~/test$ cat test1.txt
test.txt
danylenko@VM2:~/test$ ls -l
total 4
-rw-r--r-- 1 danylenko danylenko 9 KiB 21 17:22 test1.txt
danylenko@VM2:~/test$ ln test1.txt test2.txt
danylenko@VM2:~/test$ ls -l
total 8
-rw-r--r-- 2 danylenko danylenko 9 KiB 21 17:22 test1.txt
-rw-r--r-- 2 danylenko danylenko 9 KiB 21 17:22 test2.txt
danylenko@VM2:~/test$ echo "test2.txt" > test2.txt
danylenko@VM2:~/test$ cat test1.txt test2.txt
test2.txt
test2.txt
danylenko@VM2:~/test$ rm test1.txt
danylenko@VM2:~/test$ ls -l
total 4
-rw-r--r-- 1 danylenko danylenko 10 KiB 21 17:29 test2.txt
```

(now a soft link)

ln -s test2.txt test3.txt	create softlink test3.txt to file test2.txt
ls -l .	Listing of directory, files are not equal, have 1 hard link each, and test3.txt is showed as soft link to test2.txt

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

rm test2.txt; ls -l . Remove original file test2.txt, listing shows file3.txt as broken link to test2.txt. Content of test2.txt is lost.
Recreation of file test2.txt makes soft link test2.txt to work again, but it links to new file with new content.

```
danylenko@VM2:~/test$ ln -s test2.txt test3.txt
danylenko@VM2:~/test$ ls -l
total 4
-rw-r--r-- 1 danylenko danylenko 10 Kbi 21 17:29 test2.txt
lrwxrwxrwx 1 danylenko danylenko 9 Kbi 21 17:37 test3.txt -> test2.txt
danylenko@VM2:~/test$ rm test2.txt; ls -l
total 0
lrwxrwxrwx 1 danylenko danylenko 9 Kbi 21 17:37 test3.txt -> test2.txt
danylenko@VM2:~/test$ cat test3.txt
cat: test3.txt: No such file or directory
```

3. I/O redirect.

Execute these commands; comment on the output.

mount	list all mounted file systems
blkid	list all block device files in /dev
mount grep sda	filter output of mount, show lines with "sda" pattern phrase
dmesg grep sda	filter diagnostic msgs of kernel, show lines with "sda" phrase

sudo grep -R -e "root" /etc > root_entries.txt

-R recursive reading and filtering of /etc folder content, filters all lines in files that contain regexp (-e) "root", results are placed into .txt file

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

```
danylenko@VM2:~$ sudo grep -R -e "root" /etc > root_entries.txt
danylenko@VM2:~$ cat root_entries.txt
/etc/services:rootd 1094/tcp
/etc/services:rootd 1094/udp
/etc/skel/.bashrc:# set variable identifying the chroot you work in (used in the prompt below)
/etc/skel/.bashrc:if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
/etc/skel/.bashrc:    debian_chroot=$(cat /etc/debian_chroot)
/etc/skel/.bashrc:    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
/etc/skel/.bashrc:    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
/etc/skel/.bashrc:    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
/etc/cups/cups-files.conf:# User that is substituted for unauthenticated (remote) root access
```