EPAM University Programs

DevOps external course

Module 4 Linux & Bash Essentials

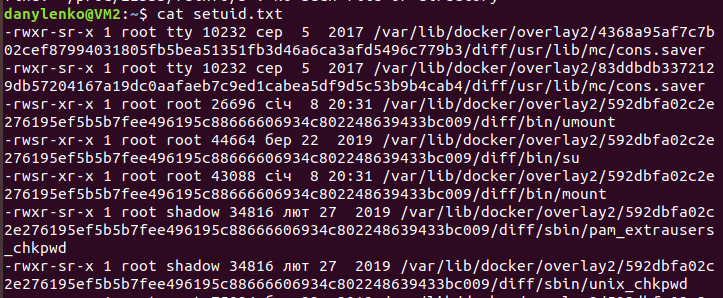
TASK 4.5

**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).



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 сhange 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 “text1.txt” into file test1.txt

ls -l . Detailed listing of current dir

(a hard link)

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

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

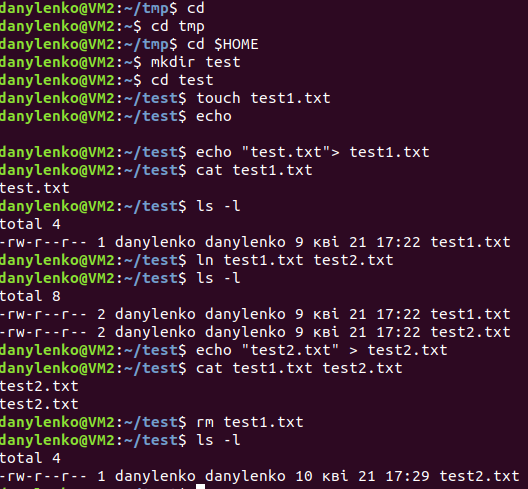
echo “test2.txt” > test2.txt print text “text2.txt” into file text2.txt

cat test1.txt test2.txt both files contain “test2” text because all changes are written into both hardlinked text2.txt file and text1.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.



(now a soft link)

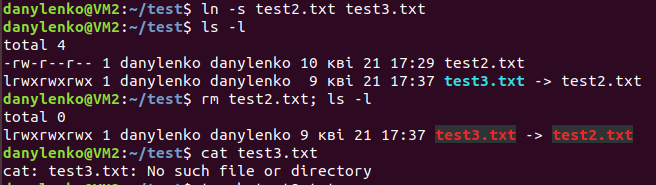
ln -s test2.txt test3.txt create softlink test3.txt to file test2.txt

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.



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)

