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

*Mariia Markina*

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).
We are finding files (-type f) in a root directory (/), that have no permissions to read, write and execute either for user, or for group user, or other (only for owner or owner`s group) (-perm /6000). Then we are executing command large ls for found directory (-exec ls -ld {}, where {} – found directory). After that, writing the results into setuid.txt (>setuid.txt).

```
mariia@mariia-VirtualBox:~$ sudo find / -perm /6000 -type f -exec ls -ld {} \; >setuid.txt
find: ‘/run/user/1000/gvfs’: Permission denied
find: ‘/proc/9968/task/9968/fdinfo/6’: No such file or directory
find: ‘/proc/9968/fdinfo/5’: No such file or directory
```

```
mariia@mariia-VirtualBox:~$ cat setuid.txt
-rwsr-xr-x 1 root root 40152 січ 27 16:28 /snap/core/8689/bin/mount
-rwsr-xr-x 1 root root 44168 тра  7  2014 /snap/core/8689/bin/ping
-rwsr-xr-x 1 root root 44680 тра  7  2014 /snap/core/8689/bin/ping6
-rwsr-xr-x 1 root root 40128 бер 25  2019 /snap/core/8689/bin/su
-rwsr-xr-x 1 root root 27608 січ 27 16:28 /snap/core/8689/bin/umount
-rwxr-sr-x 1 root shadow 35632 кві  9  2018 /snap/core/8689/sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root shadow 35600 кві  9  2018 /snap/core/8689/sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 62336 бер 25  2019 /snap/core/8689/usr/bin/chage
-rwsr-xr-x 1 root root 71824 бер 25  2019 /snap/core/8689/usr/bin/chfn
-rwsr-xr-x 1 root root 40432 бер 25  2019 /snap/core/8689/usr/bin/chsh
-rwxr-sr-x 1 root systemd-network 36080 кві  6  2016 /snap/core/8689/usr/bin/crontab
-rwxr-sr-x 1 root mail 14856 гру  7  2013 /snap/core/8689/usr/bin/dotlockfile
-rwxr-sr-x 1 root shadow 22768 бер 25  2019 /snap/core/8689/usr/bin/expiry
-rwsr-xr-x 1 root root 75304 бер 25  2019 /snap/core/8689/usr/bin/gpasswd
-rwxr-sr-x 3 root mail 14592 гру  4  2012 /snap/core/8689/usr/bin/mail-lock
-rwxr-sr-x 3 root mail 14592 гру  4  2012 /snap/core/8689/usr/bin/mail-touchlock
-rwxr-sr-x 3 root mail 14592 гру  4  2012 /snap/core/8689/usr/bin/mail-unlock
-rwsr-xr-x 1 root root 39904 бер 25  2019 /snap/core/8689/usr/bin/newgrp
-rwsr-xr-x 1 root root 54256 бер 25  2019 /snap/core/8689/usr/bin/passwd
-rwxr-sr-x 1 root crontab 358624 бер  4  2019 /snap/core/8689/usr/bin/ssh-agent
-rwsr-xr-x 1 root root 136808 січ 31 20:37 /snap/core/8689/usr/bin/sudo
-rwxr-sr-x 1 root tty 27368 січ 27 16:28 /snap/core/8689/usr/bin/wall
-rwsr-xr-- 1 root systemd-resolve 42992 лис 29 14:40 /snap/core/8689/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

2. Discovering soft and hard links.
Comment on results of these commands (place the output into your report):
**cd**
**mkdir** test

**cd** test
**touch** test1.txt
**echo** “test1.txt” > test1.txt
**ls** -l .
*(a hard link)*
**ln** test1.txt test2.txt
**ls** -l .
cd takes us to the home directory, then we are making there a new directory test and creating there test1.txt file. Then, writing out “test1.txt” in this file. Now, there is this file in test directory. After this, creating a new hard link test2.txt (which refers to test1.), as we can see, it looks exactly like the original file, number of links – 2 for both files.

```
mariia@mariia-VirtualBox:~$ mkdir test
mariia@mariia-VirtualBox:~$ cd test
mariia@mariia-VirtualBox:~/test$ touch test1.txt
mariia@mariia-VirtualBox:~/test$ echo "test1.txt">test1.txt
mariia@mariia-VirtualBox:~/test$ ls -l .
total 4
-rw-r--r-- 1 mariia mariia 10 кві 16 14:36 test1.txt
mariia@mariia-VirtualBox:~/test$ ln test1.txt text2.txt
mariia@mariia-VirtualBox:~/test$ ls -l .
total 8
-rw-r--r-- 2 mariia mariia 10 кві 16 14:36 test1.txt
-rw-r--r-- 2 mariia mariia 10 кві 16 14:36 text2.txt
```

*(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
**ls** -l .
*(now a soft link)*
**ln** -s test2.txt test3.txt
**ls** -l .
*(pay attention to the number of links to the created files)*
**rm** test2.txt; **ls** -l .
Writing “test2.txt” to the test2.txt file and now, as we can see, in both files “test2.txt” is written (so, file changes if it`s hard link file is changed) and removing first file. Creating a soft link file test3.txt, which refers to test2.txt. Number of files is 1 for both files. After removal of the original file in ls we can see that soft link is still there, but something is wrong with it and it cannot be read (because file is deleted).

```
mariia@mariia-VirtualBox:~/test$ echo "test2.txt">test2.txt
mariia@mariia-VirtualBox:~/test$ cat test1.txt test2.txt
test2.txt
test2.txt
mariia@mariia-VirtualBox:~/test$ rm test1.txt
mariia@mariia-VirtualBox:~/test$ ls -l
total 4
-rw-r--r-- 1 mariia mariia 10 кві 16 16:18 test2.txt
mariia@mariia-VirtualBox:~/test$ ln -s test2.txt test3.txt
mariia@mariia-VirtualBox:~/test$ ls -l
total 4
-rw-r--r-- 1 mariia mariia 10 кві 16 16:18 test2.txt
lrwxrwxrwx 1 mariia mariia  9 кві 16 16:20 test3.txt -> test2.txt
mariia@mariia-VirtualBox:~/test$ rm test2.txt
mariia@mariia-VirtualBox:~/test$ ls -l
total 0
lrwxrwxrwx 1 mariia mariia 9 кві 16 16:20 test3.txt -> test2.txt
mariia@mariia-VirtualBox:~/test$
```

3. I/O redirect.

Execute these commands; comment on the output.

**mount –** shows data about file system.

**blkid** – shows information about devices.

**mount** | **grep** sda – shows only that information about file system, which refers to sda (hard disk).

**dmesg** | **grep** sda -shows messages about devices, detected by kernel, in this case messages about sda (grep sda).

```
mariia@mariia-VirtualBox:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1991108k,nr_inodes=497777,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=403092k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
```

```
mariia@mariia-VirtualBox:~$ blkid
/dev/sr0: UUID="2020-02-18-17-20-05-35" LABEL="VBox_GAs_6.1.4" TYPE="iso9660"
mariia@mariia-VirtualBox:~$ mount | grep sda
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
mariia@mariia-VirtualBox:~$ dmesg | grep sda
[    1.740355] sd 2:0:0:0: [sda] 20971520 512-byte logical blocks: (10.7 GB/10.0 GiB)
[    1.740361] sd 2:0:0:0: [sda] Write Protect is off
[    1.740362] sd 2:0:0:0: [sda] Mode Sense: 00 3a 00 00
[    1.740371] sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[    1.745020]  sda: sda1
[    1.745171] sd 2:0:0:0: [sda] Attached SCSI disk
[    3.403993] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts: (null)
[    7.595129] EXT4-fs (sda1): re-mounted. Opts: errors=remount-ro
mariia@mariia-VirtualBox:~$ sudo grep -R -e "root"/etc >root_entries.txt
```

**sudo grep** -R -e “root” /etc > root_entries.txt – shows all entries of text “root” in directory /etc, checking also symlinks(-R) and putting output in root_entries.txt file.

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

```
/etc/security/access.conf:# Disallow non-root logins on tty1
/etc/security/access.conf:#-:ALL EXCEPT root:tty1
/etc/security/access.conf:# User "root" should be allowed to get access via cron .. tty5 tty6.
/etc/security/access.conf:#+ : root : cron crond :0 tty1 tty2 tty3 tty4 tty5 tty6
/etc/security/access.conf:# User "root" should be allowed to get access from hosts with ip addresses.
/etc/security/access.conf:#+ : root : 192.168.200.1 192.168.200.4 192.168.200.9
/etc/security/access.conf:#+ : root : 127.0.0.1
/etc/security/access.conf:# User "root" should get access from network 192.168.201.
/etc/security/access.conf:#+ : root : 192.168.201.
/etc/security/access.conf:# User "root" should be able to have access from domain.
/etc/security/access.conf:#+ : root : .foo.bar.org
/etc/security/access.conf:# User "root" should be denied to get access from all other sources.
```

Here we can see permissions and privileges of the root user, for example, what tty can be used.