EPAM University programs DevOps external course Oleksandr Briukhanov

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).

```
bruh@wibob-X61:~$ sudo find / -perm /6000 -type f -exec ls -ld {} \;>setuid.txt
[sudo] password for bruh:
find: '/run/user/1000/gvfs': Permission denied
find: '/proc/14082/task/14082/fdinfo/5': No such file or directory
find: '/proc/14082/fdinfo/6': No such file or directory
bruh@wibob-X61:~$ cat setuid.txt
-rwsr-xr-x 1 root root 67600 чер 28  2019 /bin/ping
-rwsr-xr-x 1 root root 26012 бер  5 19:23 /bin/umount
-rwsr-xr-x 1 root root 42400 бер  5 19:23 /bin/mount
-rwsr-xr-x 1 root root 30112 сер 11  2016 /bin/fusermount
-rwsr-xr-x 1 root root 43240 бер 22  2019 /bin/su
-rwsr-sr-x 1 root root 117324 лют 12 19:07 /snap/snapd/6439/usr/lib/snapd/snap-confine
-rwsr-xr-x 1 root root 121420 бер 21 20:13 /snap/snapd/6952/usr/lib/snapd/snap-confine
-rwsr-xr-x 1 root root 42400 сер 23  2019 /snap/core18/1289/bin/mount
-rwsr-xr-x 1 root root 67600 чер 28  2019 /snap/core18/1289/bin/ping
-rwsr-xr-x 1 root root 43240 бер 22  2019 /snap/core18/1289/bin/su
-rwsr-xr-x 1 root root 26012 сер 23  2019 /snap/core18/1289/bin/umount
-rwxr-sr-x 1 root shadow 38256 лют 27  2019 /snap/core18/1289/sbin/pam_extrausers_chkpwd
-rwxr-sr-x 1 root shadow 38252 лют 27  2019 /snap/core18/1289/sbin/unix_chkpwd
-rwxr-sr-x 1 root shadow 66076 бер 22  2019 /snap/core18/1289/usr/bin/chage
```

find - is looking for files and compare every file with reference file specified on command line and if file matches do some actions with them.

In our case **find** command options means:

/ - looking whole file system from root

-perm /6000 - looking for objects on file system that contain SUID bit in permissions (/4000 searching pattern in numeric notations) or SGID bit (/2000 searching pattern in numeric notations). /6000 means that all objects contain any of this bits should be used as a true result;

-type f - only regular files should be used;

by default between this operators uses logical operator 'and' hence is - we are looking for files with SGID or SUID bits are set;

with every file matches condition, **find** makes action (operator **-exec** execute command **ls -ld {}** - list in long format and for directory shows only it's properties (but it useless, because we are looking only for files) The string '{}' is replaced by the current file name being processed. ';' - it's the end of argument string and it should be escaped.

> at the end of command redirects standard output to file.

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

## creating directory test in current directory

cd test

## change current directory

touch test1.txt

## creating an empty file named test1.txt in current directory

echo "test1.txt" > test1.txt

## putting string "test1.txt into file named test1.txt

ls -l .

## listing current directory content in "long"format

ln test1.txt test2.txt

## creating hardlink test2.txt to test1.txt

ls -l .

## listing current directory content in "long"format. both hardlinked files are there with quantity of links indicator equivalent 2

echo “test2.txt” > test2.txt

## putting string "test2.txt into file named test2.txt

cat test1.txt test2.txt

## getting on screen file1.txt and file2.txt content (some content, because both hardlinked to each other and to the same place on file system)

rm test1.txt

## deleting file test1.txt

ls -l .

## we can see the file test2.txt, but quantity of links now indicate only 1, because we deleted one of them on previous step

ln -s test2.txt test3.txt

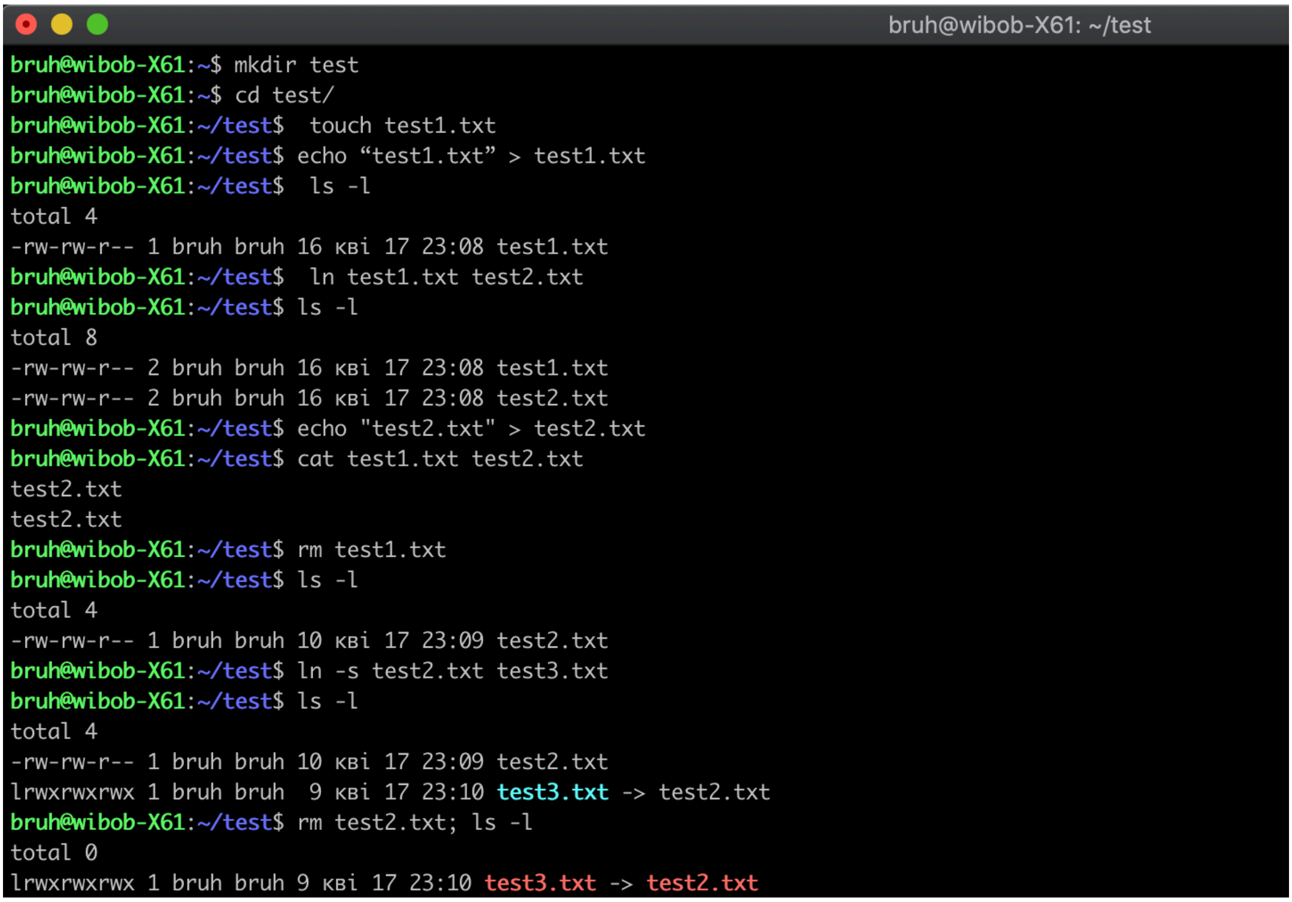
## creating softlink (shortcut) test3.txt on file test2.txt

ls -l .

## we can see source file test2.txt with "1" in link quantity indicator (there is no hardlinks to this file) and shortcut file test3.txt with information about linked source file

rm test2.txt; **ls** -l .

## after deleting source file, we can see name of source file in shortcut (softlinked) file properties despite deleting them (in our case, terminal indicate this linked file with red color)



```
bruh@wibob-X61:~$ mkdir test
bruh@wibob-X61:~$ cd test/
bruh@wibob-X61:~/test$  touch test1.txt
bruh@wibob-X61:~/test$ echo “test1.txt” > test1.txt
bruh@wibob-X61:~/test$  ls -l
total 4
-rw-rw-r-- 1 bruh bruh 16 кві 17 23:08 test1.txt
bruh@wibob-X61:~/test$  ln test1.txt test2.txt
bruh@wibob-X61:~/test$ ls -l
total 8
-rw-rw-r-- 2 bruh bruh 16 кві 17 23:08 test1.txt
-rw-rw-r-- 2 bruh bruh 16 кві 17 23:08 test2.txt
bruh@wibob-X61:~/test$ echo "test2.txt" > test2.txt
bruh@wibob-X61:~/test$ cat test1.txt test2.txt
test2.txt
test2.txt
bruh@wibob-X61:~/test$ rm test1.txt
bruh@wibob-X61:~/test$ ls -l
total 4
-rw-rw-r-- 1 bruh bruh 10 кві 17 23:09 test2.txt
bruh@wibob-X61:~/test$ ln -s test2.txt test3.txt
bruh@wibob-X61:~/test$ ls -l
total 4
-rw-rw-r-- 1 bruh bruh 10 кві 17 23:09 test2.txt
lrwxrwxrwx 1 bruh bruh  9 кві 17 23:10 test3.txt -> test2.txt
bruh@wibob-X61:~/test$ rm test2.txt; ls -l
total 0
lrwxrwxrwx 1 bruh bruh 9 кві 17 23:10 test3.txt -> test2.txt
```

3. I/O redirect.
Execute these commands; comment on the output.
mount

## this command shows all mounted partitions (mounted device, mounting point, filesystem and mounting options)

```
ansible@wsrv-ans:~$ 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=434304k,nr_inodes=108576,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=93148k,mode=755)
/dev/sda2 on / type ext4 (rw,relatime,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs
(rw,relatime,fd=25,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=17231)
mqueue on /dev/mqueue type mqueue (rw,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
configfs on /sys/kernel/config type configfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
/var/lib/snapd/snaps/core_8689.snap on /snap/core/8689 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/snapd/snaps/core_8935.snap on /snap/core/8935 type squashfs (ro,nodev,relatime,x-gdu.hide)
lxcfs on /var/lib/lxcfs type fuse.lxcfs (rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)
tmpfs on /run/user/1001 type tmpfs (rw,nosuid,nodev,relatime,size=93144k,mode=700,uid=1001,gid=1001)
ansible@wsrv-ans:~$
```

blkid

```
ansible@wsrv-ans:~$ blkid
/dev/sda2: UUID="6ae3038c-c752-4242-9aed-8dfba53ba570" TYPE="ext4" PARTUUID="a603be94-
e37c-416c-8c72-52ee389bf991"
ansible@wsrv-ans:~$
```

## shows all block devices with attributes (device name, device id, file system type and id for mounted partitions)

mount | grep sda

```
ansible@wsrv-ans:~$ mount | grep sda
/dev/sda2 on / type ext4 (rw,relatime,data=ordered)
ansible@wsrv-ans:~$
```

## mount output pipelined to grep can be used to show pattern matched output. in our case we can see the only one string wit mounting /dev/sda partition on root (/) using ext4 file system with options:

rw - read/write option

relatime - enable last time access information. Access time is only updated if the previous access time was earlier than the current modify or change time.

data=ordered - all data are forced directly out to the main filesystem prior to its metadata being committed to the journal.

dmesg | grep sda

## this command (display message) display messages from system core to standard out. With our "grep filter" we can see only messages about hard disk sda:

-information about size and blocks size and quantity of blocks
-quantity of bytes in physical bloks
-whether or not write-protected device
-byte mode options for HD controller
-human readable options
-partitions on HD
-mesages about disc connection and filesystem EXT4 mounting mode

```
ansible@wsrv-ans:~$ dmesg | grep sda
[    1.489845] sd 2:0:0:0: [sda] 266338304 512-byte logical blocks: (136 GB/127 GiB)
[    1.490794] sd 2:0:0:0: [sda] 4096-byte physical blocks
[    1.492559] sd 2:0:0:0: [sda] Write Protect is off
[    1.493018] sd 2:0:0:0: [sda] Mode Sense: 0f 00 00 00
[    1.494195] sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[    1.511202]  sda: sda1 sda2
[    1.514452] sd 2:0:0:0: [sda] Attached SCSI disk
[    3.682917] EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
[    6.669853] EXT4-fs (sda2): re-mounted. Opts: (null)
ansible@wsrv-ans:~$
```

sudo grep -R -e "root" /etc > root_entries.txt
(place only a reasonable fragment of root_entries.txt into your report)

## This command browse recursively all files in /etc directory to find strings with 'root' keyword and redirect all results to file root_entries.txt. This file contain a lot of strings with keyword "root" as a part of expressions. Now using additional "filter" we can analyse information about root group/user. Let's see.

```
ansible@wsrv-ans:~$ cat root_entries.txt | grep '/passwd\|/group\|/shadow' | sort
/etc/group-:root:x:0:
/etc/group:root:x:0:
/etc/passwd-:root:x:0:0:root:/root:/bin/bash
/etc/passwd:root:x:0:0:root:/root:/bin/bash
/etc/security/group.conf:# 1. to run an application as root
/etc/shadow-:root:*:18113:0:99999:7:::
/etc/shadow:root:*:18113:0:99999:7:::
ansible@wsrv-ans:~$
```

Using this output we can see:

- *root user has no password and can not be used to direct login;*
- *only one user 'root' is in group 'root'*

To complete security observe for root privileges, we need to analyse sudoers file, passwd file for users with id = 0 and files where SGID and(or) SUID are set.