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 /600 -type f -exec ls -ld {} \;>setuid.txt

-perm /mode — any of the permission bits mode are set for the file. Symbolic modes are accepted in this form. You must specify `u', `g' or `o' if you use a symbolic mode. If no permission bits in mode are set, this test matches any file (the idea here is to be consistent with the behaviour of -perm -000).

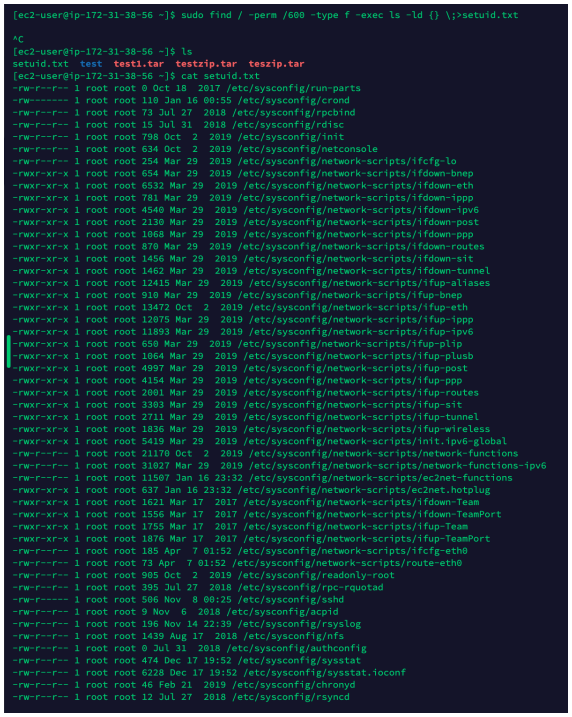
-type f — File is of type f. F — regular file.

-exec command {} + — this variant of the -exec action runs the specified command on the selected files, but the command line is built by appending each selected file name at the end; the total number of invocations of the command will be much less than the number of matched files.

ls -ld — long listing of directories

\; — path to file

> setuid.txt — write out to 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).

2. Discovering soft and hard links.

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

cd — change directory to home

mkdir test — create directory "test"

cd test — change directory to test

touch test1.txt — create file test1.txt

echo "test1.txt" > test1.txt — display "test1.txt" and write to test1.txt

ls -l . — long listing of files

(a hard link)

ln test1.txt test2.txt — create hard link

ls -l . — long listing of files

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

echo "test2.txt" > test2.txt — display "test2.txt" and write to test2.txt

cat test1.txt test2.txt — stdout what it files

rm test1.txt — delete file test1.txt

ls -l . — long listing of files

(now a soft link)

ln -s test2.txt test3.txt — create soft link

ls -l . — long listing of files

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

rm test2.txt;**ls** -l .— long listing of files only after delete file test2.txt(show change)

```
[ec2-user@ip-172-31-38-56 test]$ cd
[ec2-user@ip-172-31-38-56 ~]$ mkdir test
mkdir: cannot create directory 'test': File exists
[ec2-user@ip-172-31-38-56 ~]$ cd test
[ec2-user@ip-172-31-38-56 test]$ touch test1.txt
[ec2-user@ip-172-31-38-56 test]$ echo "test1.txt" > test1.txt
[ec2-user@ip-172-31-38-56 test]$ ls -l .
total 16
-rw-rw-r-- 1 ec2-user ec2-user     0 Apr 18 08:13 test1
-rw-rw-r-- 1 ec2-user ec2-user    16 Apr 22 14:53 test1.txt
-rw-rw-r-- 1 ec2-user ec2-user 10240 Apr 18 08:13 v
[ec2-user@ip-172-31-38-56 test]$ ln test1.txt test2.txt
[ec2-user@ip-172-31-38-56 test]$ ls -l .
total 20
-rw-rw-r-- 1 ec2-user ec2-user     0 Apr 18 08:13 test1
-rw-rw-r-- 2 ec2-user ec2-user    16 Apr 22 14:53 test1.txt
-rw-rw-r-- 2 ec2-user ec2-user    16 Apr 22 14:53 test2.txt
-rw-rw-r-- 1 ec2-user ec2-user 10240 Apr 18 08:13 v
[ec2-user@ip-172-31-38-56 test]$ echo "test2.txt" > test2.txt
[ec2-user@ip-172-31-38-56 test]$ cat test1.txt test2.txt
"test2.txt"
"test2.txt"
[ec2-user@ip-172-31-38-56 test]$ rm test1.txt
[ec2-user@ip-172-31-38-56 test]$ ls -l .
total 16
-rw-rw-r-- 1 ec2-user ec2-user     0 Apr 18 08:13 test1
-rw-rw-r-- 1 ec2-user ec2-user    16 Apr 22 14:54 test2.txt
-rw-rw-r-- 1 ec2-user ec2-user 10240 Apr 18 08:13 v
[ec2-user@ip-172-31-38-56 test]$ ln -s test2.txt test3.txt
[ec2-user@ip-172-31-38-56 test]$ ls -l .
total 16
-rw-rw-r-- 1 ec2-user ec2-user     0 Apr 18 08:13 test1
-rw-rw-r-- 1 ec2-user ec2-user    16 Apr 22 14:54 test2.txt
lrwxrwxrwx 1 ec2-user ec2-user     9 Apr 22 14:54 test3.txt -> test2.txt
-rw-rw-r-- 1 ec2-user ec2-user 10240 Apr 18 08:13 v
[ec2-user@ip-172-31-38-56 test]$ rm test2.txt; ls -l
total 12
-rw-rw-r-- 1 ec2-user ec2-user     0 Apr 18 08:13 test1
lrwxrwxrwx 1 ec2-user ec2-user     9 Apr 22 14:54 test3.txt -> test2.txt
-rw-rw-r-- 1 ec2-user ec2-user 10240 Apr 18 08:13 v
[ec2-user@ip-172-31-38-56 test]$
```

3. I/O redirect.

Execute these commands; comment on the output.

mount — used to mount the filesystem found on a device to big tree structure(Linux filesystem) rooted at '/'

```
[ec2-user@ip-172-31-38-56 test]$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=485468k,nr_inodes=121367,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
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/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
/dev/xvda1 on / type xfs (rw,noatime,attr2,inode64,noquota)
mqueue on /dev/mqueue type mqueue (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=31,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=14275)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=100696k,mode=700,uid=1000,gid=1000)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,relatime)
tracefs on /sys/kernel/debug/tracing type tracefs (rw,relatime)
[ec2-user@ip-172-31-38-56 test]$
```

blkid — a command-line utility that displays information about available block devices.

```
[ec2-user@ip-172-31-38-56 test]$ blkid
[ec2-user@ip-172-31-38-56 test]$  blkid /dev/sda1
[ec2-user@ip-172-31-38-56 test]$ blkid -h

Usage:
 blkid --label <label> | --uuid <uuid>

 blkid [--cache-file <file>] [-ghlLv] [--output <format>] [--match-tag <tag>]
       [--match-token <token>] [<dev> ...]

 blkid -p [--match-tag <tag>] [--offset <offset>] [--size <size>]
       [--output <format>] <dev> ...

 blkid -i [--match-tag <tag>] [--output <format>] <dev> ...

Options:
 -c, --cache-file <file>    read from <file> instead of reading from the default
                              cache file (-c /dev/null means no cache)
 -d, --no-encoding          don't encode non-printing characters
 -g, --garbage-collect      garbage collect the blkid cache
 -o, --output <format>      output format; can be one of:
                              value, device, export or full; (default: full)
 -k, --list-filesystems     list all known filesystems/RAIDs and exit
 -s, --match-tag <tag>      show specified tag(s) (default show all tags)
 -t, --match-token <token>  find device with a specific token (NAME=value pair)
 -l, --list-one             look up only first device with token specified by -t
 -L, --label <label>        convert LABEL to device name
 -U, --uuid <uuid>          convert UUID to device name
 <dev>                      specify device(s) to probe (default: all devices)

Low-level probing options:
 -p, --probe                low-level superblocks probing (bypass cache)
 -i, --info                 gather information about I/O limits
 -S, --size <size>          overwrite device size
 -O, --offset <offset>      probe at the given offset
 -u, --usages <list>        filter by "usage" (e.g. -u filesystem,raid)
 -n, --match-types <list>   filter by filesystem type (e.g. -n vfat,ext3)

 -h, --help     display this help and exit
 -V, --version  output version information and exit

For more details see blkid(8).
[ec2-user@ip-172-31-38-56 test]$
```

mount | **grep sda** — pipe of mounting all which contains sda

dmesg | **grep sda** — pipe of display message or display drive all which contains sda

```
[ec2-user@ip-172-31-38-56 test]$ mount | grep sda
[ec2-user@ip-172-31-38-56 test]$ dmesg | grep sda
[ec2-user@ip-172-31-38-56 test]$
```

```
[ec2-user@ip-172-31-38-56 test]$ dmesg
[    0.000000] Linux version 4.14.173-137.229.amzn2.x86_64 (mockbuild@ip-10-0-1-143) (gcc version 7.3.1 20180712 (Red Hat 7.3.1-6) (GCC)) #1 SMP Wed Apr 1 18:06:08
UTC 2020
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.14.173-137.229.amzn2.x86_64 root=UUID=55da5202-8008-43e8-8ade-2572319d9185 ro console=tty0 console=ttyS0,115
200n8 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.shell=0
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
[    0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[    0.000000] e820: BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009dfff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009e000-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000e0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x00000000bfffffff] usable
[    0.000000] BIOS-e820: [mem 0x00000000fc000000-0x00000000ffffffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] SMBIOS 2.7 present.
[    0.000000] DMI: Xen HVM domU, BIOS 4.2.amazon 08/24/2006
[    0.000000] Hypervisor detected: Xen HVM
[    0.000000] Xen version 4.2.
[    0.000000] Xen Platform PCI: I/O protocol version 1
[    0.000000] Netfront and the Xen platform PCI driver have been compiled for this kernel: unplug emulated NICs.
[    0.000000] Blkfront and the Xen platform PCI driver have been compiled for this kernel: unplug emulated disks.
               You might have to change the root device
               from /dev/hd[a-d] to /dev/xvd[a-d]
               in your root= kernel command line option
[    0.000000] HVMOP_pagetable_dying not supported
[    0.000000] tsc: Fast TSC calibration using PIT
[    0.000000] e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
[    0.000000] e820: remove [mem 0x000a0000-0x000fffff] usable
[    0.000000] e820: last_pfn = 0x40000 max_arch_pfn = 0x400000000
[    0.000000] MTRR default type: write-back
[    0.000000] MTRR fixed ranges enabled:
[    0.000000]   00000-9FFFF write-back
[    0.000000]   A0000-BFFFF write-combining
[    0.000000]   C0000-FFFFF write-back
[    0.000000] MTRR variable ranges enabled:
[    0.000000]   0 base 0000F0000000 mask 3FFFF8000000 uncachable
```

sudo grep -R -e “root” /etc > root_entries.txt — “grep” search for PATTERN in each FILE or standard input “-R” likewise, but follow all symlinks “-e” use PATTERN for matching with word “root” in /etc directory and write to file root_entries.txt

```
[ec2-user@ip-172-31-38-56 ~]$ sudo grep -R -e "test" /etc > root_entries.txt
[ec2-user@ip-172-31-38-56 ~]$ cat root_entries.txt
/etc/sysconfig/network-scripts/ifdown-ipv6:# IPv6 test, no module loaded, exit if system is not IPv6-ready
/etc/sysconfig/network-scripts/ifdown-ipv6:ipv6_test testonly || exit 0
/etc/sysconfig/network-scripts/ifdown-ipv6:ipv6_test_device_status $DEVICE
/etc/sysconfig/network-scripts/ifdown-ipv6:ipv6_test_device_status tun6to4
/etc/sysconfig/network-scripts/ifdown-sit:# IPv6 test, no module loaded, exit if system is not IPv6-ready
/etc/sysconfig/network-scripts/ifdown-sit:ipv6_test testonly || exit 0
/etc/sysconfig/network-scripts/ifdown-sit:ipv6_test_device_status $DEVICE
/etc/sysconfig/network-scripts/ifup:        if test -z "$PHYSDEV"; then
/etc/sysconfig/network-scripts/ifup:          if test -z "$PHYSDEV"; then
/etc/sysconfig/network-scripts/ifup-aliases:# My test setup of four class C address blocks on a P166 took 25 seconds of
/etc/sysconfig/network-scripts/ifup-ipv6:# IPv6 test, module loaded, exit if system is not IPv6-ready
/etc/sysconfig/network-scripts/ifup-ipv6:ipv6_test || exit 1
/etc/sysconfig/network-scripts/ifup-ipv6:ipv6_test_device_status $DEVICE
/etc/sysconfig/network-scripts/ifup-ipv6:    ipv6_test_device_status tun6to4
/etc/sysconfig/network-scripts/ifup-ipv6:        if ! ipv6_test_ipv4_addr_global_usable $ipv4addr; then
/etc/sysconfig/network-scripts/ifup-post:    # Determine what regexp we should use (for testing below):
/etc/sysconfig/network-scripts/ifup-sit:# IPv6 test, module loaded, exit if system is not IPv6-ready
/etc/sysconfig/network-scripts/ifup-sit:ipv6_test || exit 1
/etc/sysconfig/network-scripts/ifup-sit:ipv6_test_device_status $DEVICE
/etc/sysconfig/network-scripts/init.ipv6-global:        # IPv6 test, module loaded, exit if system is not IPv6-ready
/etc/sysconfig/network-scripts/init.ipv6-global:        ipv6_test || exit 1
/etc/sysconfig/network-scripts/init.ipv6-global:        # IPv6 test, module loaded, exit if system is not IPv6-ready
/etc/sysconfig/network-scripts/init.ipv6-global:        ipv6_test || exit 1
/etc/sysconfig/network-scripts/init.ipv6-global:        # Unreachable IPv6: 6bone test addresses
/etc/sysconfig/network-scripts/init.ipv6-global:        # IPv6 test, no module loaded, exit if system is not IPv6-ready
/etc/sysconfig/network-scripts/init.ipv6-global:        ipv6_test testonly || exit 0
/etc/sysconfig/network-scripts/init.ipv6-global:        if ipv6_test_device_status sit0 ; then
/etc/sysconfig/network-scripts/network-functions-ipv6:# $1: (optional) testflag: currently supported: "testonly" (do not load a module)
/etc/sysconfig/network-scripts/network-functions-ipv6:# return code: 0=ok 2=IPv6 test fails
/etc/sysconfig/network-scripts/network-functions-ipv6:ipv6_test() {
/etc/sysconfig/network-scripts/network-functions-ipv6:    local fn="ipv6_test"
/etc/sysconfig/network-scripts/network-functions-ipv6:    local testflag=$1
/etc/sysconfig/network-scripts/network-functions-ipv6:        if [ "$testflag" = "testonly" ]; then
/etc/sysconfig/network-scripts/network-functions-ipv6:# return code: 0=ok 1=argument error 2=IPv6 test fails 3=major problem adding route
/etc/sysconfig/network-scripts/network-functions-ipv6:    ipv6_test || return 2
/etc/sysconfig/network-scripts/network-functions-ipv6:    ipv6_test_ipv6_addr_valid $networkipv6 || return 2
/etc/sysconfig/network-scripts/network-functions-ipv6:    ipv6_test_ipv6_addr_valid $gatewayipv6 || return 2
/etc/sysconfig/network-scripts/network-functions-ipv6:# return code: 0=ok 2=IPv6 test fails 3=major problem
/etc/sysconfig/network-scripts/network-functions-ipv6:    ipv6_test || return 2
/etc/sysconfig/network-scripts/network-functions-ipv6:    if ipv6_test_device_status sit0; then
/etc/sysconfig/network-scripts/network-functions-ipv6:          if ! ipv6_test_device_status sit0; then
```