

6. laboratorijska vježba

Linux permissions and ACLs

U okviru ove vježbe upoznali smo se s osnovnim postupkom upravljanja korisničkim računima na Linux OS-u. Pri tome će se poseban naglasak staviti na **kontrolu pristupa (eng. access control)** datotekama, programima i drugim resursima Linux sustava.

A. Kreiranje novog korisničkog računa

U Linux-u svaka datoteka ili program ima vlasnika. Svakom korisniku pridjeljen je jedinstveni identifikator *User ID (UID)*. Svaki korisnik mora pripadati barem jednoj grupi, pri čemu više korisnika može dijeliti istu grupu. Linux grupe također imaju jedinstvene identifikatore *Group ID (GID)*.

Prvo smo provjerili UID i GID našeg računala pomoću naredbe `id` te je on ispao sljedeći. Također vidimo i grupe kojima pripadamo (najbitnije da pripadamo administratorskoj grupi `sudo`)

```
id
uid=1000(student) gid=1000(student)
groups=1000(student),4(adm),20(dialout),24(cdrom),
25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),114(netdev),1001(docker)
```

Sljedeći korak je bio dodavanje korisnika "alice" i "bob" pomoću naredbe (nama su nazvani `alice5` i `bob5` jer su prijašnje grupe već radile to). Bitno napomenuti da korisnike jedino možemo dodavati kada smo admin tj. `student`.

```
sudo adduser bob5
```

```
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ sudo adduser bob5
Adding user `bob5' ...
Adding new group `bob5' (1010) ...
Adding new user `bob5' (1008) with group `bob5' ...
Creating home directory `/home/bob5' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bob5
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ |
```

Onda smo se logirali kao novi korisnik `su -alice` te provjerili id tog korisnika.

```
alice5@DESKTOP-7Q0BASR:~$ id
uid=1007(alice5) gid=1009(alice5) groups=1009(alice5)
alice5@DESKTOP-7Q0BASR:~$ pwd
/home/alice5
alice5@DESKTOP-7Q0BASR:~$ cd
alice5@DESKTOP-7Q0BASR:~$ pwd
/home/alice5
alice5@DESKTOP-7Q0BASR:~$ mkdir srp
alice5@DESKTOP-7Q0BASR:~$ cd srp/
alice5@DESKTOP-7Q0BASR:~/srp$ dir
alice5@DESKTOP-7Q0BASR:~/srp$ echo Hello world > security.txt
alice5@DESKTOP-7Q0BASR:~/srp$ ls
security.txt
alice5@DESKTOP-7Q0BASR:~/srp$ cat security.txt
Hello world
alice5@DESKTOP-7Q0BASR:~/srp$ |
```

B. Standardna prava pristupa datotekama

Kreirali smo novi direktorij ali kao korisnik alice5 tj. prvo na home direktoriju napravili srp direktorij i onda u njemu security.txt (taj postupak je prikazan na slici poviše)

```
# create a new directory
mkdir

# create a file with text
echo "Hello world" > security.txt

# print file content
cat security.txt
```

Da bi izlistali informacije o novom direktoriju koristili smo naredbe `ls -l` ili `getfacl`

```
ls -ls
-rw-rw-r-- 1 alice5 alice5 12 Jan 18 11:14 security.txt (koja prava ima lasnik(rw), grupa(rw) i svi ostali)
```

Tada smo pokušali oduzeti pravo pristupa datoteci security.txt vlasniku datoteke tj. alice5. Za tu promjenu koristili smo naredbu `chmod`:

```
alice5@DESKTOP-7Q0BASR:~/srp$ chmod u-r security.txt
alice5@DESKTOP-7Q0BASR:~/srp$ cat security.txt
cat: security.txt: Permission denied
alice5@DESKTOP-7Q0BASR:~/srp$ |
```

“u” - označava user, a -r znači oduzimanje prava čitanja datoteke

I vidimo da je oduzimanje prava uspjelo.

Za dodat prava nazad samo umjesto “-” stavili smo “+”.

Sljedeće smo provjerili da li bob5 ima pravo čitanja datoteke security.txt:

```
C:\Users\A507>wsl
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ su - bob5
Password:
bob5@DESKTOP-7Q0BASR:~$ cat /home/alice5/srp/security.txt
Hello world
bob5@DESKTOP-7Q0BASR:~$ id
uid=1008(bob5) gid=1010(bob5) groups=1010(bob5)
bob5@DESKTOP-7Q0BASR:~$
```

Bob5 će moći pristupiti security.txt jer je stavljeno da ostali mogu čitati datoteku pri stvaranju txt-a.

Da mu maknemo pristup koristimo naredbu `chmod o-r security.txt`, tako smo onemogućili pristup svima koji pripadaju other. Ovu naredbu treba napisati kao alice5.

Da dodamo boba u grupu moramo se odlogirat od alice5 sa `exit` i napisat `sudo usermod -aG alice5 bob5`

Promjenu ćemo tek vidjeti kada se odlogiramo i ponovno logiramo u bob5.

Onda smo maknuli boba iz grupe alice5 `sudo gpasswd -d bob alice`.

C. Kontrola pristupa korištenjem *Access Control Lists (ACL)*

Da bi boba bas dodali u ACL(prije smo ga baš dodali u grupu) napišemo `sudo setfacl -m u:bob5:r /home/alice5/srp/security.txt`

```
getfacl: Removing leading '/' from absolute path names
# file: home/alice5/srp/security.txt
# owner: alice5
# group: alice5
user::rw-
user:bob5:r--
group::rw-
mask::rw-
other:---
bob5@DESKTOP-7Q0BASR:~$
```

Pomoću `getfacl` vidimo da bob nije dodan u neku grupu nego je dodan u ACL.

Može se i napraviti skroz nova grupa samo za čitanje

`sudo setfacl -m g:alice_reading_group5:r /home/alice5/srp/security.txt`

(g: - dio koji označava da je grupa a ne korisnik)

```
getfacl: Removing leading '/' from
# file: home/alice5/srp/security.t
# owner: alice5
# group: alice5
user::rw-
user:bob5:r--
group::rw-
group:alice_reading_group5:r--
mask::rw-
other::---
```

D. Linux procesi i kontrola pristupa

Svaki linux proces u izvršavanju ima svoj jedinstveni identifikator, *process identifier* PID. Osim toga, svaki proces ima i id vlasnika, UID. Na temelju UID-ja Kernel će odlučivati ima li proces pristup određenim resursima ili ne.

Trenutno aktivne procese možete izlistati korištenjem naredbe `ps -ef`.

```
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ ps -ef
UID      PID  PPID  C  STIME TTY          TIME CMD
root         1      0  0  10:54 ?            00:00:00 /init
root       112      1  0  10:54 ?            00:00:00 /init
root       113     112  0  10:54 ?            00:00:00 /init
root       114     113  0  10:54 pts/0        00:00:00 /mnt/wsl/docker-desktop/doc
root       123     112  0  10:54 ?            00:00:00 /init
student    124     123  0  10:54 pts/1        00:00:01 docker serve --address unix
root       140      1  0  11:04 ?            00:00:00 /init
root       141     140  0  11:04 ?            00:00:00 /init
student    142     141  0  11:04 pts/2        00:00:00 -bash
root       266      1  0  11:11 ?            00:00:00 /init
root       267     266  0  11:11 ?            00:00:00 /init
student    268     267  0  11:11 pts/3        00:00:00 -bash
root       421     268  0  11:38 pts/3        00:00:00 su - bob5
bob5       422     421  0  11:38 pts/3        00:00:00 -su
student    460     142  0  11:49 pts/2        00:00:00 ps -ef
```

Da bi makli bob5 i ACL mičemo cijelu ACL `sudo setfacl -b /home/alice5/srp/security.txt`

Sljedeće smo otvorili WSL shell i u tekućem direktoriju napravili Python skriptu koja sadrži sljedeće:

```
import os

print('Real (R), effective (E) and saved (S) UIDs:')
```

```
print(os.getresuid())

with open('/home/alice5/srp/security.txt', 'r') as f:
    print(f.read())

#želimo procitati id usera koji pokrece process, tj. skriptu
#vratit će nam R, E , S UIDove korisnika
```

```
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ python lab6_g5.py
Real (R), effective (E) and saved (S) UIDs:
(1000, 1000, 1000)
Traceback (most recent call last):
  File "lab6_g5.py", line 6, in <module>
    with open('/home/alice5/srp/security.txt', 'r') as f:
IOError: [Errno 13] Permission denied: '/home/alice5/srp/security.txt'
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ id
```

Izbacuje error jer student pripada grupi other pa on nema pravo čitanja i pristupa, ali ako stavimo sudo python lab6.. onda će pokrenuti jer je to super user.

Ako probamo to pokrenuti sa boba ni on neće imati pravo pristupa.

Opcionalni zadatak

Ako pokusamo kod boba promijenit šifru sa `passwd`, mi to ne bi trebali moć izvest jer kod `/etc/shadow` other nemaju ovlasti pisanja. (Shadow je mjesto gdje se spremaju lozinke)

Ali ipak možemo promijeniti šifru, zašto?

`getfacl $(which passwd)` gledamo kome pripada passwd

```
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ getfacl $(which passwd)
getfacl: Removing leading '/' from absolute path names
# file: usr/bin/passwd
# owner: root
# group: root
# flags: s--
user::rwx
group::r-x
other::r-x
```

Vidimo da passwd process pripada vlasniku root koji ima ID 0, dok bob ima 1008.

To smo testirali na sljedeći način:

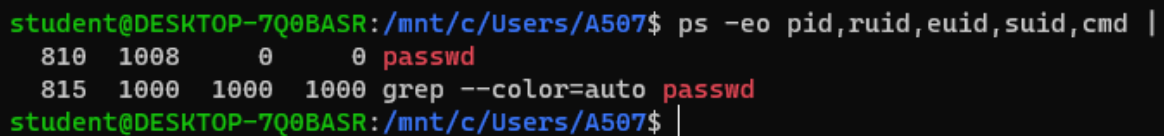
1. izvršite naredbu `passwd` (kao neprivilagirani korisnik).

```
passwd
Changing password for alice.
(current) UNIX password:
# !!! NEMOJTE UNOSITI NIKAKVU LOZINKU !!!

#ostavili smo da program "visi"
```

2. U drugom terminalu izvršite sljedeću naredbu (koja će vam ispisati tekuće procese sa njihovim stvarnim i efektivnim vlasnicima):

```
ps -eo pid,ruid,euid,suid,cmd
```



```
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ ps -eo pid,ruid,euid,suid,cmd |
 810  1008    0    0 passwd
 815  1000  1000  1000 grep --color=auto passwd
student@DESKTOP-7Q0BASR:/mnt/c/Users/A507$ |
```

1008 real id od boba 0 efektivni id od roota

Vidimo da kod procesa 810 a to je naš nedovršeni proces passwd, RUID je jednak 1008, do je EUID i SUID jednak 0 tj. IDu roota koji je vlasnik passwd processa. Zbog toga korisnik bob5 može promijeniti svoju šifru.