Šesti izvještaj

Linux permissions and ACLs

U ovoj vježbi se upoznajemo s osnovnim postupkom upravljanja korisničkim računima na Linux OS-u. Naglasak se stavlja na kontrolu pristupa resursima Linux sustava.

A. Kreiranje korisničkog računa

Svaka datoteka ili programa u Linuxu ima svog vlasnika. Svaki korisnik ima **User ID**, mora pripadati bar jednoj grupi koje također imaju **Group ID**, a jednoj grupi može pripadati više korisnika.

Spajamo se na Linux preko Terminala korištenjem naredbe:

```
code
```

Identifikatore i pripadnost grupama provjeravamo naredbom:

```
id
```

Provjeravamo pripadamo li grupi sudo, tj. imamo li administratorske ovlasti jednostavno:

```
groups
```

Kreiramo novog korisnika:

```
sudo adduser alice4
```

Za *logiranje* kao novi korisnik koristimo naredbu:

```
su - alice4
```

Program nas pita lozinku koju smo koristili pri kreiranju novoga korisnika. Kako bi kreirali novog korisnika, moramo se vratiti u *shell* korisnika koji ima administratorske ovlasti naredbom:

```
exit
```

Korisnika bob4 smo kreirali na isti način kao i alice4.

B. Standardna prava pristupa datotekama

Logirali smo se u sustav kao alice4 i u korisnikovom home direktoriju kreirali novi direktorij srp, a u navedenom datoteku security.txt.

```
mkdir srp

cd srp

echo "Hello world" > security.txt

cat security.txt
```

echo - koristimo za unos teksta u datoteku

cat - ispis sadržaja datoteke

Prava i ovlasti definirana za novu datoteku i direktorij provjeravamo s naredbom **getfacl**.

```
getfacl security.txt

getfacl .
```

Najveća prava imaju vlasnik i njegova grupa, a ostali imaju samo pravo čitati (r).

Oduzimamo pa dodajemo pravo čitanja datoteke alice4 (vlasniku) , zatim joj oduzimamo pa dodajemo pravo izvršavanja direktorija te na kraju dajemo korisniku bob4 (others) pravo čitanja datoteke.

```
#oduzimanje prava čitanja datoteke
chmod u - r security.txt

#davanje prava čitanja datoteke
chmod u + r security.txt

#oduzimanje prava izvršavanja direktorija
chmod u-x .

#davanje prava izvršavanja direktorija
chmod u+x /srp

#davanje prava čitanja ostalima
chmod o-r security.txt
```

Sada smo trebali vratiti pravo čitanja korisniku bob4 tako da ga dodamo u grupu alice4. Vraćamo se u *shell* korisnika s administratorskim pravima te dodajemo korisnika u grupu s naredbom:

```
usermod -aG alice4 bob4
```

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

Za inspekciju i modifikaciju ACL-ova resursa, tj. datoteka i direktorija koristimo naredbe **getfacl** i **setfacl**.

Korisniku bob4 dodajemo prava čitanja korištenjem naredbe **setfacl**:

```
setfacl -m u:bob4:r security.txt
```

Korisniku bob4 također možemo pridijeliti prava čitanja tako da kreiramo novu grupu kojoj će biti omogućeno pravo čitanja. Kreiramo grupu:

```
groupadd alice_reading_group_4
```

Prije nego bob4 dodamo u grupu, moramo mu oduzeti dosadašnja prava, dodati prava u grupu i onda tek bob4 u grupu:

```
setfacl -r u:bob4 security.txt

setfacl g:alice_reading_group_4:r security.txt

usermod -aG alice_reading_group_4 bob4
```

D. Linux procesi i kontrola pristupa

Linux procesi su programi koji se izvršavaju trenutno u odgovarajućem adresnom prostoru. Svaki proces ima vlasnika (**UID**) i jedinstveni identifikator procesa (**PID**).

Korisnika bob4 uklanjamo iz grupe:

```
gpasswd -d bob4 alice_reading_group
```

U direktoriju otvaramo VSCode te kreiramo Python skriptu pomoću koje ćemo pregledavati user ID-eve.

```
import os

print('Real (R), effective (E) and saved (S) UIDs:')
print(os.getresuid())

with open('/home/alice4/srp/security.txt', 'r') as f:
    print(f.read())
```

Kad izvršimo skriptu kao administrator ili alice4 možemo otvoriti datoteku, a kao bob4 ne.

Dodatni zadatak

Pristup /etc/shadow file-u, u kojem se nalaze hashevi passworda, ima samo korisnik koji se nalazi u grupi root.

Kada korisnik preda zahtjev za promjenu lozinke privremeno preuzme ID od root korisnika i može modificirati pristup i shadow-file.

```
passwd
```