Linux kiterjesztett ACL - jogosultság kezelés. Set UID, Set GID, Sticky bit

Varga Tibi 2021

Hasznos parancsok az aktuális ACL meghatározásához

- Az id parancs megmutatja felhasználó csoportjait, a felhasználóval a megadott felhasználóét : **# id <felhasználó>**

ACL - Access Control List

- A POSIX ACL (hozzáférés-vezérlési listák) a hagyományos unixos fájlrendszer-jogosultságrendszer kiterjesztése. Olyan jogosultság-konstrukciók is leírhatók a segítségével, amelyek a hagyományos jogosultságrendszerrel nem: pl. hogy egy adott fájlt egy csoport tagjai írhassák és olvashassák, egy másik csoport tagjai csak olvashassák, a többiek pedig se ne írhassák, se ne olvashassák.
- Amint már tanultuk a három jogosultsághalmaz három-három bitből áll: ezek az **olvasás** (r), **az írás** (w) és **végrehajtás (futtathatóság** (x) jogát reprezentálják. Az így kapott jogokat, minden **fáj**l illetve **könyvtár** esetében, kilenc biten tudjuk tárolni.
- **Megadható ezen felül a felhasználói azonosító beállítása (set user id), a csoportazonosító beállítása (set group id) és a ragadós (sticky) bit is.**

Kiterjesztett ACL

Egy kiterjesztett ACL-bejegyzés- az eddig tanult három alappal szemben 6 jogosultsági entitáshoz, három (r,w,x) információt tartalmaz:

- <acl típusa>:<név>:<jogok>

ACL-bejegyzéstípusok

Típus	Kód	Szöveges forma	
tulajdonos	ACL_USER_OBJ	user::rwx	• Alap
konkrét felhasználó	ACL_USER	user:név:rwx	• Kiterjesztett
tulajdonoscsoport	ACL_GROUP_OBJ	group::rwx	• Alap
konkrét csoport	ACL_GROUP	group:név:rwx	• Kiterjesztett
maszk	ACL_MASK	mask::rwx	• Kiterjesztett
egyéb	ACL_OTHER	other::rwx	• Alap

Kiterjesztett vagy bővített ACL-

- Az előzőtáblázatban látható hogy egy könyvtárhoz és fájlhoz alapokon túl megadható másik konkrét felhasználó illetve csoport aki a bejegyzésben megadott jogosultságokkal rendelkezhet az állományon.
- A kiterjesztett ACL-ek tartalmaznak egy mask bejegyzést illetve megnevezett felhasználókat és csoportokat is. Más szavakkal: **ha egy ACL-ben szerepel GROUP vagy USER típusú (tehát konkrét felhasználóra vagy konkrét csoportra vonatkozó) bejegyzés**, akkor kötelezően **tartalmaz egy MASK** típusú bejegyzést is, a MASK bejegyzés minden egyéb esetben opcionális.

A MASZK

A maszk a jogosultságok elbírálásakor jut szerephez. Ha egy processz megpróbál megnyitni egy olyan fájlt, amelyhez bővített ACL tartozik, egy jogosultságvizsgáló algoritmus fut le, amelynek a működése a következőképpen foglalható össze:

- Ha a processz effektív UID-ja megegyezik a fájl tulajdonosáéval, akkor a USER_OBJ bejegyzésben megadott jogosultságbitek az érvényesek.
- GID esetén is az irányadó
- Ha nem egyezik, és van a processz effektív UID-jára vonatkozó USER típusú ACL-bejegyzés, az ebben szereplő **jogosultságbitek és a maszk ÉS kapcsolata** az irányadó, kivéve, **ha a maszk üres vagy hiányzik** ("---"), mert akkor az **OTHER bejegyzés az irányadó.**

Példa látható hogy a konkrét csoport és a maszk esetén a tényleges jogosultság az ÉS kapcsolat alapján a mérvadó

Típus	Szöveges forma	Jogok
konkrét csoport	group:név:r-x	r-x
mask	mask::rw-	rw-
Effektív jogok		r--

Kiterjesztett ACL beállítás ellenőrzése

- ls –l listázás parancs kimenete beállított kiterjesztett ACL esetén a megjelenik a jogosultságok után a + jel:

```
root@kali:~# ls -l test.py
-rw-rwxr--+ 1 root root 0 jan   30 09:32 test.py
```

- Beállítások részletes lekérdezése : **$ getfacl <fáljnév>**

Nem beállított ACL esetén

```
root@kali:~# getfacl test.py
# file: test.py
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Beállított ACL esetén

```
root@kali:~# getfacl test.py
# file: test.py
# owner: root
# group: root
user::rw-
user:gabi:rwx
group::r--
mask::rwx
other::r--
```

Kiterjesztett ACL beállítása vagy módosítása

Kiterjesztett ACL beállítás :

- **# setfacl –m u:<usernév:<jogosulság> <fájl vagy könyvtár>**

vagy csoport esetén

- **# setfacl –m g:<csoportnév>:<jogosulság> <fájl vagy könyvtár>**

```
root@kali:~# setfacl -m u:gabi:rwx -m g:portas:rw test.py
```

A maszk változtatás

- **# setfacl –m mask::<jogosulság> <fájl vagy könyvtár>**

- **Ilyenkor az ellenőrzéskor megjelennek az effektív (tényleges) jogosultságok is!**

```
root@kali:~# setfacl -m mask::-x test.py
root@kali:~# getfacl test.py
# file: test.py
# owner: root
# group: root
user::rw-
user:gabi:rwx                   #effective:--x
group::r--                      #effective:---
group:portas:rw-                #effective:---
mask::--x
other::r--
```

Kiterjesztett alapértelmezett ACL

- A hozzáférési ACL-eken túlmenően ún. alapértelmezett ACL-eket is használhatunk. Ha egy könyvtárhoz tartozik alapértelmezett ACL, akkor az adott könyvtárban újonnan létrehozott valamennyi fájlrendszer-objektum hozzáférési ACL-ként ezt az alapértelmezett ACL-t örökli (az új alkönyvtárak alapértelmezett ACL-ként is).

 # setfacl -d -m g:takarito:r-x <könyvtar név>

 # setfacl -m d:g:takarito:r-x <könyvtar név>

- Az beállított kiterjesztet ACL törlése: **# setfacl –x u:<felhasználó> <fájl>**

```
root@kali:~# setfacl -x u:gabi test.py
```

- Teljes kiterjesztett ACL törlése:

 # setfacl –b <fájl név>

```
root@kali:~# setfacl -b  test.py
root@kali:~# getfacl   test.py
# file: test.py
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Az extra jogok,
Setuid, setgid, sticky-bit

- Setuid bit
- Setgid bit

Ha engedélyezzük a fenti jogot egy futtatható állományra, akkor az a tulajdonos nevében fog futni és nem annak a nevében aki futtatja.

Csak bináris futtatható fájlokra van hatással!

A bitek hatása könyvtár esetén:

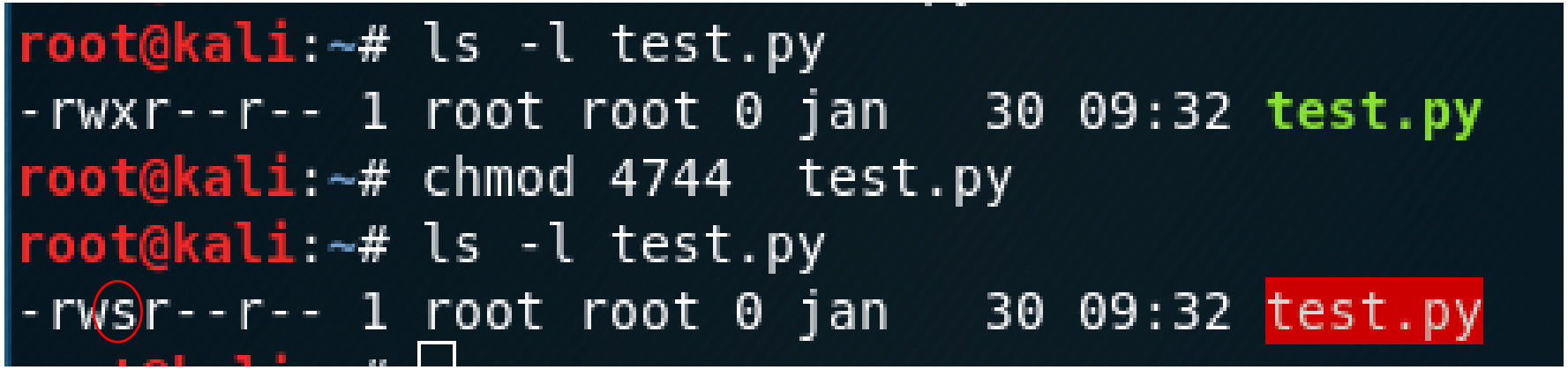
- Setuid – a könyvtárban létrejövő fájlok és könyvtárak a könyvtár tulajának tulajdonába kerülnek.
- Setgid – a könyvtárban létrejövő fájlok és könyvtárak a könyvtár csoportjának csoportjába kerülnek.

Setuid

A setuid (set user ID, "felhasználói azonosító beállítása") egy olyan speciális fájlattribútum, amelyik arra utasítja a rendszert, hogy az így megjelölt programokat egy meghatározott felhasználói azonosító (UID) nevében hajtsa végre. Vegyük példának a passwd parancsot:

-rwsr-xr-x 1 root shadow 80036 2004-10-02 11:08 /usr/bin/passwd

A felhasználói jogosultságok között látható a setuid bitet jelző s karakter. A setuid bit beállítása miatt a passwd programot minden felhasználó a root nevében futtatja. Beállítás: chmod 4722 <fájl>



Setgid

Beállítás: chmod 2722 <fájl>

```
root@kali:~# ls -l test.py
-rwxr--r-- 1 root root 0 jan   30 09:32 test.py
root@kali:~# chmod 2774   test.py
root@kali:~# ls -l test.py
-rwxrwsr-- 1 root root 0 jan   30 09:32 test.py
```

Beállítások betűkkel.

- Setuid: **# chmod u+s <fájlnév>**

```
-r-xr-xr-- 1 root root 0 jan   30 09:32 test.py
root@kali:~# chmod u+s  test.py
root@kali:~# ls -l test.py
-r-sr-xr-- 1 root root 0 jan   30 09:32 test.py
```

- Setgid: **# chmod g+s <fájlnév>**

```
-r-xr-xr-- 1 root root 0 jan   30 09:32 test.py
root@kali:~# chmod g+s  test.py
root@kali:~# ls -l test.py
-r-xr-sr-- 1 root root 0 jan   30 09:32 test.py
```

Setgid

A csoportokra a setgid bit vonatkozik. Egy olyan program, amelyhez ezt a bitet beállítják, azon csoport azonosítója (GID) alatt fog futni, amelyikkel elmentették, függetlenül attól, hogy melyik felhasználó indítja el. Éppen ezért egy olyan könyvtárban, amelynek be van állítva a setgid bitje, az összes újonnan létrehozott fájl és alkönyvtár ahhoz a csoporthoz lesz rendelve, amelyhez a könyvtár maga is tartozik. Vegyük a következő példát:

drwxrw**s**--- 2 tux archive 48 Nov 20 17:12 mydream.log

Az s karakter jelzi, hogy a setgid bit be lett állítva a csoportjogosultságokhoz. A könyvtárat a könyvtár tulajdonosa és az archive csoport tagjai érhetik el. Azok a felhasználók, akik nem a csoport tagjai, „leképződnek” a megfelelő csoportra. Az összes kiírt fájl tényleges csoportazonosítója az archive lesz. Például egy mentést végző program, amely az archive csoportazonosító nevében fut, elérheti ezt a könyvtárat root jogosultságok nélkül is.

Fontos!

Az előbb tárgyalt példákkal kapcsolatban fontos megemlítenünk, hogy habár a szkriptek is végrehajtható állományok (mellyek a !#/../.. sorral kezdődnek), **nem fognak a valóditól eltérő effektív felhasználói azonosítóval futni**. Ennek oka abban keresendő, hogy a parancssori szkriptek **nem hívják** a setuid(2) rendszerhívást.

A különbség a Setuid és Setgid esetén a „s” és „S” között:

Ha **nagy „S”** van a jogosultságban akkor be van állítva a setuid és setgid, **de** az eredeti tulajdonos vagy csoportnak **nincs futtatás joga** tehát, jó eséllyel az ennek nevében meghívó felhasználó vagy csoport sem tudja futtatni az állományt, ellentétben a **kis „s”** jogosultsággal amikor **rendelkezésre áll a futás jog**!

Ez a két speciális engedély (a setuid és a setgid) a programhoz tartozó engedélyek kiterjesztésével **csökkenthetjük rendszerünk biztonságát!**

Sticky (ragadós)bit

- A sticky bit beállításával megakadályozható, hogy egy állomány törlésre kerüljön idegen felhasználó által még akkor is, ha a tartalmazó könyvtár ezt lehetővé teszi.
- **Törölni ilyen esetben csak a tulajdonos illetve a root tud, gyakorlatban a /tmp könyvtárakban használatos.**
- Beállítása: **# chmod 1777 <könyvtár>** vagy **# chmod +t könyvtárnév**
- Fájl esetén a ragadós bittel ellátott fájlt a RAM-ban maradásra kényszerítjük, A kis „t” esetén futtatható a fájl nagy „T” esetén nem futtatható az állomány

```
root@kali:~# ls -ld tmp
drwxr-xr-x 2 root root 4096 jan   30 12:52 tmp
root@kali:~# chmod +t tmp
root@kali:~# ls -ld tmp
drwxr-xr-t 2 root root 4096 jan   30 12:52 tmp
```

Setuid, setgid, sticky-bit könyvtárak

A bitek hatása könyvtár esetén:

- Setuid – a könyvtárban létrejövő fájlok és könyvtárak a könyvtár tulajának tulajdonába kerülnek.

- Setgid – a könyvtárban létrejövő fájlok és könyvtárak a könyvtár csoportjának csoportjába kerülnek.

- Sticky – a könyvtárban mindenki csak a saját fájljait képes törölni (és természetesen a root).

UID, GID, és Sticky bit beállítás számokkal

A speciális balról első bitek a következők:

0 – semmi
1 – Sticky-bit (T-bit)
2 – SGID (setgid)
3 – Sticky-bit + SGID
4 – SUID (setuid)
5 – Sticky-bit + SUID
6 – SGID + SUID
7 – Sticky-bit + SGID + SUID