Detektiranje roota iz ljuske

Jeste li ikada pišući neku skriptu (s kojom ste htjeli automatizirati neke stvari na sustavu) zažalili što ste je uopće krenuli pisati? Ovdje prvenstveno mislimo na činjenicu da je pomoću loše napisane skripte moguće napraviti manju ili veću štetu na sustavu, primjerice obrisati dio korisnika, dio vlastitog weba i slično.

Ovo se može dogoditi. Ukoliko ne uključite dodatne opcije, skripte ljuske (shell skripte) jednostavno nastavljaju izvršavanje, bez obzira je li se pojavila neka greška ili ne. Iako ljuska (u većini slučajeva Bash) ima mogućnosti usporedive s nekim programskim jezikom, obično se u skriptama pokreću uglavnom vanjske naredbe i programi. Ukoliko nismo poduzeli dostatne mjere provjere, greška u izvršavanju neke vanjske naredbe neće zaustaviti izvršavanje skripte. Ovdje ne ubrajamo sintaktičke pogreške unutar samog "jezika" bash, no kako su skripte interpetirane, a ne kompiliraju se, zaustavit će se tek u trenutku kada interpreter naiđe na neispravan dio koda. Dio ispred toga će biti uredno izvršen, što može dovesti do polu-ispravne konzistencije sustava.

Za učenje je najbolje navesti primjer. Evo primjera skripte koja provjerava izvršava li se skripta s povećanim privilegijama ("kao root") ili ne:

```
#!/bin/sh
# Jesmo li root?
if [ `id -u` != "0" ]; then
   echo "Ova se skripta mora izvršiti pod korisnikom root"
   exit 1
fi
# daljnje naredbe...
#...
```

Možda ste se zapitali zašto provjeravamo pod kojim se korisničkim računom izvršava skripta, kad je svejedno namjeravamo koristiti za administrativne poslove (a tu se izvršavanje pod korisnikom root gotovo podrazumijeva)?

Naime, ponekad je potrebno osigurati da se sve operacije zaista mogu izvršiti, te na taj način osigurati da će zadatak biti obavljen. U suprotnom, možete napraviti neodređenu štetu na sustavu jer se dio skripte mogao izvršiti pod nekim neprivilegiranim korisničkim računom, a dio nije. U ovisnosti o kakvoj se skripti radi, i jeste li koristili provjere u skripti, možete očekivati probleme od benignog "access denied" do brisanja podataka s kojima radite. Ovakve situacije nisu česte, ali s provjerom ste ih u potpunosti izbjegli.

Možete iskoristiti i suprotnu situaciju, tako da se skripta izvršava samo ako ste je pokrenuli kao neprivilegirani, obični, korisnik. Samo promijenite znak nejednakosti "**!=**" u "**=**", pa će se skripta izvršiti samo ako **niste root**.

Bash nudi još jedan način provjere. Nakon prijave na sustav, jedna od varijabli koja se postavlja u vašem okolišu (*environmentu*) je $UID (*User IDentification*). Ona sadržava broj, koji je numerička oznaka vašeg korisničkog računa:

```
$ declare | grep UID
```

```
EUID=1008
UID=1008
```

Vidimo da postoji i varijabla $EUID. Ona predstavlja vaš trenutni efektivni UID, a to je UID koji se postavlja nakon što svom računu povećamo privilegije putem naredbe su ili sudo:

```
$ su
Password:
# declare | grep UID
EUID=0
UID=0
```

Skripta tada može glasiti ovako:

```
#!/bin/sh
# Jesmo li root?
if [ "$EUID" -ne 0 ]; then
   echo "Ova se naredba mora izvršiti pod korisnikom root"
   exit 1
fi
```

Skripte su pisane za klasični Bourne shell (/bin/sh), a standardni Bash (kojeg imate pod imenom /bin/bash i obično linkan na /bin/sh) to čak može riješiti jednim retkom:

```
(( EUID )) && echo "Izvršiti pod korisnikom root" && exit 1
```

U ovom obliku, ne treba navoditi znak '$' u imenu varijable.

čet, 2013-03-28 20:06 - Marko Jukić**Kuharice:** Linux [1]
Kategorije: Operacijski sustavi [2]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/1236

Links
[1] https://sysportal.carnet.hr/taxonomy/term/17
[2] https://sysportal.carnet.hr/taxonomy/term/26