Završna laboratorijska vježba

Napredno korištenje operacijskog sustava Linux

3. lipnja 2017.

Marcus želi konačno znati podići svoj server. Ali ne samo lokalno, za suho isprobavanje komunikacijskih protokola za koje je čuo na fakultetu, već želi nešto što nudi potpunije iskustvo s radom na serveru. Čuo je od kolege Rona da mu studentski status na FER-u nudi brojne pogodnosti što se tiče besplatnog *hostinga* servera, pa je silno poželio naučiti uzeti svoj server, kako bi na njemu mogao pokrenuti različite *dockere* u kojima bi vrtio aplikacije koje razvija.
Kako priča ne bi stala tu, Marcus je zamislio i stvaranje korisničkih računa za svoje prijatelje, kako bi i oni mogli razvijati svoje web aplikacije. Ipak, budući da je njegov najam servera, bio bi red da uvede i neki sustav kontrole korištenja serverskog diskovnog prostora. Sjetio se davnog razgovora sa svojim prijateljem *sistemašem* Jeremyjem, koji mu je spominjao kvote u UNIX sustavima, pa bi svakako htio napraviti nešto takvo i ostvariti jedno realno i uređeno višekorisničko okruženje.
Da bi cijela stvar bila koliko-toliko *user-friendly*, a opet jednostavno konfigurabilna i sigurna, Marcus je naumio koristiti NginX, budući da s njim već ima prethodnog iskustva. Međutim, ono što svakako želi jest voditi dnevnike pristupa dijelovima njegovog servera. Zato će pokušati upregnuti svoje poznavanje logova na UNIX sustavima. Te logove potom želi prebaciti na udaljeni server, a za to je naumio koristiti *rsync* protokol, koji mu je jako nahvalila kolegica Saša u pauzi između predavanja na prošlogodišnjoj DORS/CLUC konferenciji. Konačno, budući da se već pošteno bacio u održavanje svog servera, poželio je dio zadataka automatizirati, pa se sjetio sustava cron, kao najzastupljenijeg sustava za automatizaciju periodički ponovljenih zadataka.

Nužno je imati na umu da je Marcus ipak relativno "zelen" u UNIX okruženjima, pa ne čudi da vas je lijepo zamolio da mu pomognete s njegovim ne-baš-bezazlenim projektom.

Napomena: nije zgorega prije nastavka rješavanja vježbe prvo pročitati završne napomene koje se nalaze na zadnjoj stranici teksta vježbe.

Zadatak 1 - nabavka servera

Rasprava o radu na serveru nije moguća bez servera na kojem bismo radili. Prema tome, prvo je potrebno pronaći neki *hosting* servis koji nudi servere. U nastavku su dane neke opcije koje nude besplatan ili probni hosting, koji bi bio dovoljan za naše potrebe.

- **GitHub Education Pack** studentima besplatno nudi brojne alate za kojekakve programske zadatke, a između ostalog i nekoliko *cloud hosting* opcija koje će nam poslužiti - ovdje je možda najjednostavnija takva opcija ***DigitalOcean***.
- **Amazon Web Services Free Tier** je skup besplatnih razvojnih alata koje - uz, naravno, neka ograničenja - nudi Amazon. U *GitHub Education Packu* postoji i opcija uzimanja AWS-ovog servera.
- **Google Cloud Platform Free Tier** je, pak, skup Googleovih razvojnih alata.
- Virtualni stroj[1].

NAPOMENE:

- Svaka od ovih metoda nabavke servera ima svojih poteškoća - primjerice, DigitalOcean iz sigurnosnih razloga traži i valjanu bankovnu karticu, ali Visa kartice iz nekih logističkih razloga u pravilu odbija, pa je prije iskorištavanja promo kôda potrebno podnijeti zahtjev za korisničkom podrškom (engl. *ticket*), dok AWS ima relativno nezgrapan i neintuitivan registracijski proces, pa vas, za Marcusovo dobro, molimo da (barem ovaj) zadatak **počnete raditi na vrijeme, a ne noć prije predaje!** U slučaju nemogućnosti dolaska do servera, javite nam se što je ranije moguće.
- Što se operacijskog sustava tiče, važno je samo da je riječ o nekoj Linux distribuciji koja koristi ***systemd***. Radi popularnosti (kako globalnoj, tako i na vještini) i jednostavnosti, preporučamo serversku inačicu distribucije Ubuntu.
- Nijedan pošteni serverski OS **nema GUI** (*Graphical User Interface* - grafičko korisničko sučelje), pa vas je Marcus pomalo drsko zamolio da ga ne sramotite tako da mu na server instalirate GUI. Naime: GUI u serverskom okruženju ne donosi gotovo nikakve korisne mogućnosti, jer se većina konfiguriranja ionako svodi na rad kroz terminal i uređivanje tekstualnih konfiguracijskih datoteka. S druge strane, GUI zauzima neke resurse na serverskom računalu, i predstavlja samo još jednu komponentu koja se može pokvariti. Također, zbog arhitekture X sustava, otvara se i mogućnost sigurnosnih propusta, što nikako nije poželjno[2].

[1] Ovo je daleko najmanje impresivna opcija i zbilja bi trebala biti razmotrena tek kao zadnja opcija. S virtualnim ste se strojevima sigurno susreli na NKOSL-u, a vrlo vjerojatno i na OKOSL-u i na još nekoliko predmeta na fakultetu, pa u tom pogledu **nećete ništa novo naučiti.**

[2] Više o argumentaciji zaoštrenih odnosa između servera i GUI-ja možete pročitati na linku.

Zadatak 2 - dodavanje korisnika; kvote

Nakon posljednjeg predavanja NKOSL-a, Marcus se s nekoliko kolega upustio u žustru diskusiju o serverima. Vrlo zadovoljan i motiviran raspravom, odlučio je na svom serveru napraviti i jedan korisnički račun za kolege s NKOSL-a, kako bi oni mogli staviti svoje web-stranice ili neke datoteke na njegov server. Ipak, budući da je svjestan vrijednosti resursa svog servera, želi ograničiti njihovo korištenje diskovnog prostora, i to koristeći kvote (engl. *quotas*.

Dakle, nakon što smo osigurali neku inačicu servera, moramo pomoći Marcusu da ostvari željenu višekorisničku konfiguraciju, koja treba sadržavati barem sljedeće:

- Uz *root* i *default* korisnika (koji bi se mogao zvati *marcus*), potrebno je dodati i korisnika s imenom *nkosl*.
- Korisnik *marcus* mora imati sudo ovlasti, tj. mora se nalaziti u datoteci */etc/sudoers*.
- Korisnik *nkosl* ne smije imati sudo ovlasti.

Što se kvota tiče, nužno je da postoje sljedeće komponente:

- *Soft limit*
- *Hard limit*
- *Grace period*

Budući da ne zna puno o kvotama, kolega nam je ostavio nešto slobode što se tiče izbora vrijednosti. Važno je da su *soft limit* i *hard limit* različiti, te da postoji grace period. Ipak, nastojte te vrijednosti svesti na nešto što je lako demonstrirati (čak i ako se time malo narušava vjerodostojnost sustava). Primjer konfiguracije:

- SL = 10 MB
- HL = 15 MB
- GP od nekoliko sekundi

Također, na vama je i da smislite adekvatnu demonstraciju kao dokaz da su kvote ispravno konfigurirane.

NAPOMENE:

- Ako se ukaže potreba za uređivanjem datoteke */etc/sudoers*, za to **obavezno koristiti naredbu *visudo*!** U protivnom je moguće, primjerice, ukloniti sudo prava svim korisnicima, što nikako ne bi bilo dobro.
- Za brže shvaćanje kvota, dobre se upute nalaze na linku.

Zadatak 3 - korisnički pristup: SSH, lozinka

Marcusu se jako sviđa dosad obavljeno, ali prelijen je da bi svaki put upisivao lozinku, pa se sjetio SSH ključeva i činjenice da se SSH ključevi koriste za autentikaciju korisnika. Prema tome, Marcus vas je zamolio da **omogućite** SSH login i **zabranite** login lozinkom svim korisnicima.

S tim željama na umu, vaš se zadatak ugrubo može podijeliti na sljedeće dijelove:

- Dignite SSH server na portu 31415.
- Omogućite login pomoću RSA keyeva (za marcusa te za korisnika *nkosl*, za kojeg ćemo vam mi dati ključ)
- O**ne**mogućite login koristeći password.
- Onemogućite login korisniku root[3].
- Kreirajte korisnika *papiga* (nominativ) te kopirajte njegov *.bashrc* u direktorij kojeg poslužuje NginX iz sljedećeg zadatka.
- Uredite *welcome message* tako da u toj poruci piše kad i odakle se korisnik prethodni put spojio.

BONUS: Za jednostavniji rad i demonstraciju pokušajte napraviti alias za login na server naredbom "ssh srvr", tj. da ne morate pisati ni IP adresu servera ni username. Hint: koristiti *ssh_profile*, ne *bashrc*.

[3] Ovo je česta i dobra praksa

Zadatak 4 - NginX

Odlučili smo se malo poigrati sa stvaranjem svoje male web stranice uz pomoć Nginx-a, kako bismo olakšali pristup važnim datotekama s našeg servera. Pritom, naravno, ne želimo da tome svatko može pristupiti. Zato smo ga odlučili koliko-toliko zaštititi jednostavnim metodama.

Za početak želimo:

- Preko URL-a *server_IP/mydistro* pristupiti web stranici svoje omiljene Linux distribucije
- Preko URL-a *server_IP/files* pristupiti direktoriju u kojem se nalaze datoteke i moći pregledavati/preuzimati svaku od njih

Nakon ove kratke konfiguracije, pobrinut ćemo se da Marcusova stranica bude osigurana. Da bismo to ostvarili, potrebno je:

- Promijeniti port na kojem Nginx sluša zahtjeve na 8080.
- Omogućiti pristup našem web serveru samo sa CARNet-ovih IP adresa:
  - 161.53.0.0/16
  - 193.198.0.0/16
  - 82.132.0.0/17
  - 31.147.0.0/16

Napomena: za potrebe testiranja možete dodati još neki pogodan raspon IP adresa (lokalne adrese (192.168.0.0/16), adrese vašeg providera i sl.). Kompletan popis adresa možete pronaći na linku.

Puno smo napravili, i Marcus je zadovoljan trudom i predanošću, ali sa sigurnosne mu strane ovo nije dovoljno, pa nas je zamolio da mu pomognemo prebaciti stranicu na HTTPS te zaštititi naše datoteke korisničkim imenom i lozinkom. Za to je potrebno:

- stvoriti *self-signed* SSL certifikat s RSA ključem za potrebe HTTPS-a (hint: *openssl*)
  - napomena: dodatni podaci koje budete unosili nisu važni
- konfigurirati web server da:
  - koristi HTTPS i da sluša na portu 3648,
  - sve HTTP zahtjeve preusmjerava na HTTPS,
  - kod pristupanja datotekama zatraži korisničko ime i lozinku,
  - pritom imati na umu sljedeće naputke:
    - treba stvoriti datoteku s popisom korisnika,
    - *openssl* je dovoljan.

Napomena: kod prvog pristupa HTTPS verziji stranice preglednik bi trebao javiti poruku da veza nije sigurna (*connection insecure*). To je normalna pojava u ovom slučaju.

Zadatak 5 - logovi

Sada je Marcus zadovoljan postavkama NginX-a. I, naravno, beskrajno zahvalan. Sljedeće što ga zanima je vođenje logova o radu servera. Ranije spomenuta kolegica Saša rekla mu je za *systemd*-ov alat *journalctl*, pa je naumio njega koristiti. Saša je KSET-ovka, pa mu je rekla da, ako želi, može radi vježbe logove prosljeđivati na KSET-ov server, koristeći protokol *rsync*. On je tu ponudu objeručke prihvatio, pa je naumio napraviti sljedeće:

- Otkriti način kako, koristeći naredbu *journalctl*, doći do logova vezanih za NginX.
  - Obavijestite Marcusa da mu tu možda može pomoći naredba *systemctl list-units*.
- Ograničiti se na logove samo od trenutnog dana.
- Tako profiltrirane logove zapisati u datoteku *logNginx.txt*.
- Konačno, tako zapisane logove poslati, koristeći naredbu *rsync*, na adresu *hactar.kset.org*, u udaljeni direktorij *nkosl/log*.

Zadatak 6 - cron

Nakon dobro obavljenog posla, Marcus je lokalno pokrenuo naredbu *ping -D google.hr*, i, uvjerivši se da server radi kako treba, odvojio nekoliko trenutaka da se nasloni i zadovoljno protrlja rukama. Ali - dogodila se nezgoda: dok se Marcus divio (više) vašim i (manje) vlastitim *sysadmin* postignućima, mačka mu je skočila na *router* i pritom ga isključila iz struje! Nakon što je Marcus preneraženo gledao nekoliko propalih *pingova*, primijetio je da je *router* isključen, i ponovno ga uključio u struju.

Ta ga je nezgoda nagnala na razmišljanje, i ustanovio je da želi voditi nekakvu evidenciju o dostupnosti mreže. Budući da mu se na ekranu još uvijek vrtila naredba *ping*, i da je iz nje bilo očito kada je došlo do prekida veze, a kada je veza ponovno uspostavljena, zaključio je da bi jedan grubi način vođenja te evidencije mogao biti periodičko pinganje i, ako je neki paket "propao", evidentiranje toga u neku datoteku.

Marcus se sjetio svog cimera sa Šare, Malog Pere, koji je svojevremeno "rasturao" Linux, i sjetio se da je Pero za štošta koristio sustav za raspoređivanje i automatsko periodično obavljanje zadataka *cron*. Shodno tome, nazvao je Peru i pitao ga može li mu pomoći sa sastavljanjem *cronjoba* koji bi periodički, svaka 4 sata, pokrenuo ping naredbu i zapisao grešku u log u slučaju gubitka ping paketa.

Pero mu je, naravno, odgovorio potvrdno i rado pomogao. Rekao mu je da mora napraviti sljedeće:

- smišljenu ping naredbu koja šalje 8 paketa, a njen ispis na *stderr stream* preusmjeriti u log datoteku proizvoljnog naziva[4],
- pokrenuti naredbu *crontab -e*,[5]
- cijelu ranije smišljenu ping naredbu staviti na adekvatno mjesto u crontab,
- konfigurirati cron tako da pokreće tu naredbu svaki dan, svaka četiri sata.

Marcus je drage volje poslušao svog prijatelja i učinio kako mu je ovaj rekao. Nakon toga mu je palo na pamet da je mogao upregnuti *cron* i za prethodni zadatak i slanje logova. Pomognite mu da još to učini (na analogan način) i steknite njegovu vječnu zahvalnost!

[4] Konvencija je da nazivi log datoteke završe s ".log".

[5] To je naredba namijenjena postavljanju crontab datoteke, preko koje se konfiguriraju cronjobovi za pojedinog korisnika. U njoj bi trebali biti zadani neki primjeri korištenja. Trebali bi biti dovoljni za shvaćanje sintakse crona, ali svakako dodatno istražite tu sintaksu ako vam nije jasna.

Bonus zadaci, naputci i napomene

Bonus zadaci:

1. Ako želite, pokušajte napisati i upregnuti skriptu koja bi omogućila proizvoljan ispis u log datoteku, i koja bi prekinula izvođenje ping naredbe odmah nakon prvog pucanja veze.

2. Pošaljite i log pingnja na rsync server kao u petom zadatku.

3. Napravite cronjob koji jednom dnevno, u podne, pokreće naredbu *traceroute*, koja prati[6] put ping-paketa od izvora do odredišta, i taj zapis prebacuje (nevažno koristi li se *scp* ili *rsync*) na isti server kao u petom zadatku.

Naputci i napomene:

- Važno je još jednom istaknuti da barem prvi zadatak **počnete raditi na vrijeme!** Čak i u slučaju da se ne odlučite koristiti neki cloud hosting servis za nabavku servera, nego se opredijelite za virtualni stroj, greške i problemi vrebaju!

- Neke komponente ove vježbe (poput *crona*, *rsynca* ili *scp*-a) namjerno nisu detaljno objašnjene, kako vas ne bismo previše "držali za ruku" - u cilju nam je pokušati vas navesti da se snađete pred problemom. Prema tome, cijenimo svaki pokušaj rješavanja problema, pa makar se on pokazao neuspješnim.

- *man* stranice za mrežne protokole često znaju biti prevelike, nepregledne i nimalo *user-friendly*, pogotovo ako ne znate što tražiti. Zato postoji projekt *tldr*, koji koristi vezu s internetom kako bi dohvatio sažetak *man* stranica koji je čitak i koristan. Projekt se može dohvatiti u ovom git repozitoriju, a njegova bi se instalacija nekome tko nikad nije koristio git za dohvat programa mogla shvatiti kao još jedan bonus zadatak. Ali bez brige - nije teško, a iznimno je korisno.

Sretno!

[6] Uvjetno rečeno "prati" - više detalja o ovome možete čuti, primjerice, na Komunikacijskim mrežama.