Rsync - lagano do backupa

Rsync je odličan alat za kopiranje datoteka, kako na lokalno računalo, tako i na udaljeno. Kopiranje na lokalno računalo ovog nas časa ne zanima, pa ćemo se pozabaviti prijenosom datoteka preko mreže, tj. izradom sigurnosne kopije.

Najbolja odlika rsync alata jest brzina, a nju postiže prijenosom samo razlika između datoteka na lokalnom i udaljenom računalu (korištenjem tzv. *delta-transfer* algoritma). Rsync se uglavnom koristi za izradu sigurnosnih kopija i "*mirroring*", a neke od karakteristika alata su podrška za kopiranje linkova, uređaja, vlasnika, grupa i dozvola pristupa, isključivanje pojedinih datoteka i/ili direktorija uključujući CVS način rada, te upotreba ssh protokola za prijenos podataka. Kako je najbolji način za upoznavanje s alatom kroz primjere, mi ćemo kreirati automatizirani proces kopiranja */var* direktorija klijenta s adresom 192.168.1.2 na backup poslužitelj s adresom 192.168.1.1. Proces ćemo inicirati s klijenta i to kao *root* korisnik, a razlog tome je što jedino *root* ima pristup svim datotekama na računalu (primjerice */var/lib/mysql/** ili */var/mail/**).

Da bi stvar proradila, rsync je potreban na oba računala. Sama instalacija je jednostavna:

```
root@stroj:~# apt-get update
root@stroj:~# apt-get install rsync
```

Nakon instalacije rsync daemon se aktivira tako da u datoteci */etc/default/rsync* redak *RSYNC_ENABLE=false* promijenimo u *RSYNC_ENABLE=true* te ponovo pokrenemo rsync daemon.

Budući je prijenos podataka preko mreže osjetljiv, iz sigurnosnih razloga koristi ćemo ssh protokol. To bi se moglo smatrati i obavezom ukoliko se udaljeno računalo nalazi u drugoj mreži. Međutim, prilikom spajanja ssh-om potrebno je svaki put ukucati i lozinku korisnika koji se spaja na udaljeno računalo. Ako se radi automatiziranoj izradi sigurnosne kopije, taj nam scenarij ne odgovara pa ćemo se poslužiti spajanjem bez lozinke. To se postiže korištenjem para javnog i privatnog ključa bez lozinki pa ih kreirajmo na klijentu:

```
root@client:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
03:6c:94:51:11:87:6e:51:d8:19:38:53:94:df:a4:6f root@client
```

Privatni ključ zaštititimo tako da bude čitljiv samo *root* korisniku (*chmod 600*). Sad trebamo prebaciti javni ključ na poslužitelj naredbom:

```
root@client:~# scp .ssh/id_rsa.pub root@192.168.1.1:client_key.pub
```

Sustav će pitati za lozinku i prebaciti javni ključ na poslužitelj unutar */root* direktorija uz promjenu

imena ključa kako bi ga lakše razlikovali od postojećih ključeva. Da bi omogućili spajanje ssh protokolom bez unosa lozinke, potrebno je taj ključ uključiti u datoteku *authorized_keys* unutar */root/.ssh* direktorija te izolirati tu datoteku od ostalih korisnika na poslužitelju:

```
root@server:~# cat client_key.pub >> $HOME/.ssh/authorized_keys
root@server:~# chmod 600 $HOME/.ssh/authorized_keys
```

Sama datoteka *authorized_keys* sadrži javne ključeve za autentikaciju, a njeno korištenje proširuju dodatni parametri koje ćemo iskoristiti kasnije. Detaljnije objašnjenje ove datoteke se može pogledati s naredbom *man sshd*. Nakon ovog možemo isprobati spajanje ssh-om s klijenta na server i pri tome nas sustav ne bi trebao pitati za lozinku.

Korištenje rsync alata se svodi na jednostavnu naredbu oblika

```
rsync [OPCIJE] izvorište odredište
```

Opcija ima mnogo, nabrojit ćemo samo važnije:

- -d - delete files that don't exist on the sending side
- -e - alternative remove shell
- -g - preserve group
- -l - copy symlinks as symlinks
- -o - preserve owner (super-user only)
- -p - preserve permissions
- -r - recurse into directories
- -t - preserve modification times
- -v - verbose
- -z - compress

Od verzije 2.6 defaultni *remote shell* jest upravo ssh pa u tom slučaju opcija -e nije potrebna. Moguće je, međutim, specificirati neku drugu ljusku poput rsh. Ukoliko želimo na brzinu koristiti rekurzivni pristup direktorijima i sačuvati (skoro) sve što se može sačuvati, možemo koristiti opciju *-a* koja zamjenjuje redom opcije *-rlptgoD*. Tako ćemo za naš slučaj na klijentu kreirati izvršnu skriptu pod imenom *rsync_backup* koja će sadržavati sljedeće:

```
#bin/sh
/usr/bin/rsync -avz /var root@192.168.1.1:/backup/client/
```

Treba pripaziti na završne znakove */* jer se rsync tu ponaša malo neobično. Na primjer, ako u izvorištu specificiramo */var/*, onda će se na udaljeno računalo prebaciti samo sadržaj */var* direktorija. Ako za izvorište stavimo */var*, onda će se kopirati taj direktorij i sav njegov sadržaj. Skriptu možemo staviti u, primjerice */root/bin*, ali je svakako treba osigurati pravima pristupa isključivo *root* korisniku (*chmod 700*).

Već ste primijetili da se na udaljeno računalo spajamo kao *root* korisnik, što znači da je na tom udaljenom računalu prekršeno "pravilo" zabrane *root* ssh pristupa. To je, složit ćete se, priličan sigurnosni propust, pogotovo ako na računalu nemamo instalirani neki sustav za obranu od *brute-force* napada (primjerice, iptables). Naime, obično je unutar datoteke */etc/ssh/sshd.conf* zabranjen *root* ssh pristup linijom *PermitRootLogin no*. S druge strane, kako želimo pristup svim datotekama koje želimo pohraniti na udaljeno računalo, potreban su nam prava *root* korisnika. Isto tako, primijetimo da opcija *-o* "radi" samo ukoliko naredbu izvršava korisnik s *root ovlastima*. Iz svega ovog slijedi da nam treba *root* ssh pristup, samo ga trebamo dodatno osigurati. Prvi dio ovog postupka jest kreiranje skripte koju ćemo nazvati *secure_rsync* i smjestiti je unutar */root* direktorija na serveru:

```
#!/bin/sh

case "$SSH_ORIGINAL_COMMAND" in
*\&*)
   echo "Rejected"
   ;;
*\(*)
   echo "Rejected"
   ;;
*\{*)
   echo "Rejected"
   ;;
*\;*)
   echo "Rejected"
   ;;
*\<*)
   echo "Rejected"
   ;;
*\`*)
   echo "Rejected"
   ;;
*\|*)
   echo "Rejected"
   ;;
rsync\ --server*)
   $SSH_ORIGINAL_COMMAND
   ;;
*)
   echo "Rejected"
   ;;
esac
```

Ovom skriptom omogućujemo ssh pristup isključivo rsync naredbi, dok ostale pokušaje pristupa odbijamo. Da bi ovo sve proradilo, iskoristit ćemo opcije već spomenute datoteke *authorized_keys* i to na sljedeći način: na njen početak ćemo ubaciti dvije opcije, *command* i *from*. Tako naša datoteka *authorized_keys* nakon uređivanja izgleda ovako:

```
from="192.168.1.2", command="/root/secure_rsync", ssh-rsa AAAB3NzaC1yEAAABU3NC2…
```

Drugi dio osiguravanja *root* ssh pristupa poslužitelju jest promjena već spomenute datoteke */etc/ssh/sshd.conf* i njenog parametra *PermitRootLogin* u vrijednost *forced-commands-only*. Time smo kompletirali ograničenje kojim smo odredili s kojeg se računala može pristupiti ssh-om i to isključivo rsync naredbom. Opcija *from* može biti i FQDN klijenta, a ako klijent nema fiksnu adresu, može se i izostaviti.

Još je ostalo podešavanje cron servisa, kako bi proces izrade sigurnosne kopije bio potpuno automatiziran. Pokrećemo naredbu *crontab -e* i ubacujemo novu liniju:

```
0 3 * * * /root/bin/rsync_backup
```

Ovako smo definirali izvršavanje skripte svaki dan u 03:00. Naravno, moguće je napraviti dodatne modifikacije skripti poput dodavanja datoteka koje želimo isključiti, dodati više klijenata u datoteku *authorized_keys* ili kreirati rotirajuće dnevne sigurnosne kopije.

Rsync se može koristiti kao dopuna usluge sys.backup koji pruža CARNet ili kao njen nadomjestak

ukoliko postoje razlozi zbog koji ovu uslugu nije moguće koristiti. Tako je moguće za relativno malen trošak nabave novih diskova napraviti sasvim pristojan backup poslužitelj.

pet, 2012-04-27 19:37 - Mirko Lovričević**Kuharice:** Linux [1]
Kategorije: Software [2]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/999?page=0

Links
[1] https://sysportal.carnet.hr/taxonomy/term/17
[2] https://sysportal.carnet.hr/taxonomy/term/25