SSH tuneliranje

SSH, osim za sigurno spajanje na udaljene poslužitelje u svrhu rada u naredbenoj liniji, nudi i nekoliko dodatnih korisnih opcija. Jedna od njih je tuneliranje mrežnog prometa sigurnim, odnosno enkriptiranim kanalom. Ovo može poslužiti i za izbjegavanje vatrozidova, kada promet možete preumjeriti preko otvorenih portova na one na koje nemate pristup iz vanjske mreže. Ovaj postupak kreira svojevrsni "poor man's" VPN - sve to bez ikakvog posebnog konfiguriranja. Bit ce dovoljan samo odgovarajući OpenSSH klijent.

Tuneliranje internet prometa preko SSH protokola vam može pomoći u mnogim situacijama, a obradit ćemo tri najčešća slučaja (zasada ćemo se ograničiti na LocalForward nčin). Nećemo zaboraviti niti Windows korinsike, jer mnogi SSH klijenti na Windowsima također imaju ovu mogućnost. Za primjere ćemo uzeti najpopularniji, Putty.

U prvom primjeru pokazat ćemo kako na udaljenom poslužitelju "otvoriti" port koji je inače dostupan samo iz lokalne mreže. Pretpostavimo da se radi o internom mail poslužitelju i da ne želite imati otvoren port 25 prema cijelom svijetu. Na tom poslužitelju (nazovimo ga server.domena.hr) morate imati korisnički račun kojem možete pristupiti preko SSH protokola. Na naredbenoj liniji otkucajte:

```
# ssh -f korisnik@server.domena.hr -L 3333:server.domena.hr:25 -N
```

Iako je sintaksa pomalo čudna i možda malo teško pamtljiva, nakon nekog vremena će vam vjerojatno biti logičnija. Ovom konstrukcijom otvarate port 3333 na lokalnom računalu i povezujete ga s portom 25 (za SMTP) na udaljenom računalu. Ovo znači dvije stvari: svoje lokalne mail klijente morate podesiti tako da poštu šalju na localhost port 3333, a ne kao prije na server.domena.hr port 25. Druga stvar je što podaci od vašeg računala putuju kriptirano i ne morate podešavati dodatne SMTP mehanizme u ovu svrhu. Ovo je odlično, jer nikada niste sigurni koji će portovi biti otvoreni na mjestu odakle se spajate (možda je blokiran promet prema portu 25 iz anti-spam razloga).

Što se tiče upotrijebljenih opcija, "-f" određuje da se ssh proces povuče u pozadinu i ne ostane aktivan "u foregroundu" (očigledno je opcija "-b" već bila zauzeta). Na taj način nam neće smetati, i možete nastaviti rabiti shell u tom terminalskom prozoru.

Opcija "-N" govori da se neće izvršiti nikakva naredba na udaljenom poslužitelju, te da je jedina svrha tuneliranje. Opciju "-L" smo već opisali, a navest ćemo još jednom njenu sintaksu:

```
-L lokalni_port:posluzitelj:udaljeni_port
```

Drugi primjer će svakako dobro doći na konferencijama, raznim hotspotovima. Iako postoji enkripcija, najčešće na takvim mjestima nije uključena kako korisnici ne bi imali problema sa spajanjem. Ukoliko napravite ovo:

```
# ssh -f korisnik@server.domena.hr -L 8000:server.domena.hr:80 -N
```

i konfigurirate vaš web browser da rabi localhost:8000 surfat ćete sigurno, bez obzira postoji li enkripcija na bežičnoj mreži ili ne. Isto tako, ako se spajate preko Etherneta, otežat ćete prikupljanje potencijalno osjetljivih podataka koje vaš browser šalje i prima.

VAŽNO: surfanje je sigurno samo od vašeg računala do poslužitelja server.domena.hr. Od tamo na dalje promet nije enkriptiran. Tunelom želimo izbjeći nepoznati dio mreže, a pretpostavka je da je server.domena.hr "pouzdan" poslužitelj, kad već tamo imate korisnički račun.

Za treći, zadnji, zadnji primjer smo odabrali jednostavno zaobilaženje vrlo restriktivnog vatrozida, ali ovaj put unutar lokalne mreže. Recimo da je zabranjen port 6881. Ukoliko upotrijebimo sljedeću sintaksu:

```
# ssh -f -L 5000:www.negdje.com:6881 korisnik@server.domena.hr -N
```

Analizirajmo. Ukoliko podesite svoj klijent da se spaja na localhost:5000, konekcija će ići preko server.domena.hr na port 6881. Jedino što trebate je pronaći slobodan port. Port 80 nije dobar odabir jer ćete dobiti poruku:

```
bind: Address already in use
channel_setup_fwd_listener: cannot listen to port: 80
Could not request local forwarding.
```

Kako provjeriti je li port slobodan? možemo rabiti netcat, ili obični telnet:

```
$ telnet localhost 5000
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
```

ili

```
$ nc localhost 5000
localhost [127.0.0.1] 5000 (x11) : Connection refused
```

Dakle, možemo rabiti port 5000.

Članak je napisan s dobrim namjerama u vidu, te ukoliko rabeći ove savjete prekršite pravil institucije čije resurse rabite, odgovornost je samo na vama. Raspitajte se kod ovlaštene osobe prije pokušavanja zaobilaženja bilo kakvih ograničenja.

Želimo vam ugodan i siguran rad.

- Logirajte [1] se za dodavanje komentara

sub, 2009-10-31 19:13 - Željko Borš**Kuharice:** Linux [2]
Kategorije: Software [3]
Servisi [4]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/658

Links
[1] https://sysportal.carnet.hr/sysportallogin
[2] https://sysportal.carnet.hr/taxonomy/term/17
[3] https://sysportal.carnet.hr/taxonomy/term/25
[4] https://sysportal.carnet.hr/taxonomy/term/28