OpenSSH savjeti i trikovi

OpenSSH je slobodna, odnosno otvorena implementacija "Secure Shell" specifikacije IETF-a. OpenSSH omogućava da se na udaljene poslužitelje spajamo na siguran način, rabeći različite načine enkripcije, te je *de facto* postao industrijski standard tamo gdje je potrebna i minimalna razina zaštite.

Osim toga, OpenSSH omogućava i spajanje na poslužitelj bez potrebe kucanja zaporke uz pomoć pomoćnih programa ssh-add i ssh-agent, te pravljenje tunela (u smislu VPN tunela) kroz različite sustave zaštite vatrozidom.

OpenSSH je zamišljen kao zamjena za stare protokole, rlogin [1], rsh i rcp, no zapravo se najčešće rabi kao sigurna zamjena za telnet [2].

Najpoznatiji klijent na operativnom sustavu Windows s kojim se možemo spojiti na (Open)SSH poslužitelj je Putty, s kojim je također moguće praviti tunele između vašeg PC-a pod Windowsom i udaljenog poslužitelja.

- Logirajte [3] se za dodavanje komentara

Knocking on Heaven's Door, 1. dio

Poetski naslov ovog članka, kojeg ćemo zbog duljine objaviti u dva dijela, inspiriran je načinom na koji se pokušava riješiti problem napada, najčešće preko SSH protokola. Naime, u zadnje vrijeme je primjećen povećan broj pokušaja upada na poslužitelje preko tog protokola. Iako su uspješni upadi i bez dodatnih zaštita relativno rijetki, ne treba ih nikada u potpunosti zanemariti. Već smo opisali nekoliko metoda zaštite, fail2ban [4]i dali recepte za uporabu ipt_recent [5]modula u iptablesima, no napadači konstantno mijenjanju načine napada i treba ih spremno dočekati.

Sustavi poput fail2bana uspješno zaustavljaju napade koje dolaze s iste adrese, no ne mogu zaustaviti napade koje dolaze s mnogo adresa, i to namjerno usporenim ritmom i korisničkim imenom. U logovima (/var/log/auth.log) to izgleda otprilike ovako:

```
Jan 17 18:34:28 srv sshd[12635]: Failed keyboard-interactive/pam for invalid
user simon from 200.40.XX.6 port 12601 ssh2
Jan 17 18:36:24 srv sshd[12949]: Failed keyboard-interactive/pam for invalid
user sistemas from 113.17.XX.199 port 41059 ssh2
Jan 17 18:36:35 srv sshd[12953]: Failed keyboard-interactive/pam for invalid
user site from 193.62.XX.134 port 44625 ssh2
```

Nadajući se da će proći "ispod radara", napadač ne ponavlja napad s istog računala (nego vjerojatno

iz nekog BotNeta [6]), ne napada istog korisnika, niti pokušava pogoditi zaporku previše puta u minuti. Sve ovo čini u nadi kako će izbjeći IDS [7](Intrusion Detection) sustave, što mu, očigledno, može uspjeti.

Naravno, još uvijek valja pogoditi ispravnu kombinaciju korisnika i zaporke, no kad jednom napadači uđu u sustav, sve je daleko lakše. Mnoge ranjivosti, koje inače ne mogu iskoristiti udaljeno, sada su im na dohvat ruke.

Najjednostavnija obrana, barem što se SSH protokola tiče, je promijeniti osnovni port na kojem servis "sluša". Mnogi su to već odavno učinili, no iako to u ovom slučaju može pomoći, nije nikakva zaštita od "pravih" hackera, kojima neće biti nikakav problem prepoznati koji servis sluša na nekom neuobičajenom portu.

Kao vatrogasnu mjeru, dakle, možemo jednostavno promijeniti port na kojem sluša SSH daemon. Što treba promijeniti? Treba promijeniti direktivu "Port" u datoteci /etc/ssh/sshd_config i restartati sshd. Port je proizvoljan (ako nije zauzet), no morate obavijestiti sve korisnike da se port promijenio. Ako nitko osim vas ne rabi SSH, tim bolje - ne morate nikoga obavještavati.

U /etc/ssh/sshd_config upišite:

```
#Port 22
Port 12345
```

Nakon toga, kao što već vjerojatno znate, treba restartati servis:

```
# /etc/init.d/ssh restart
```

Mnogima će i ovakvo rješenje biti zadovoljavajuće, no može se mnogo više.

Ovdje dolazimo otkud nam naslov članka: od pojma "port knocking". Radi se o načinu zaštite (neki bi rekli da je to ipak samo jedan način "security through obscurity [8]"), koji otvara SSH port (zapravo, može i bilo koji drugi) samo pod određenim uvjetima. Ti uvjeti su doslovce "kucanje na portove", odnosno morate se pokušati spojiti na unaprijed predoodređen niz portova unutar vremena, prije nego se otvori port 22.

Na taj način napadač ne može znati da postoji SSH servis, jer port 22 nije otvoren. Ni skeniranjem ostalih portova neće ništa saznati, jer samo određeni niz otvara port 22 (npr. mora se proći niz portova 7777, 1234, 8888, 9876). Dakle, slučajnim skeniranjem se ne može otkriti niz koji otvara port 22 (portove nemojte odabrati u nizu, 7777, 7778, 7779 itd).

Portovi ne moraju biti TCP, nego mogu biti i UDP, što dodatno komplicira napad. Možda najveća sigurnosna prednost je mogućnost da poslani paketi moraju sadržavati točno određene TCP zastavice (SYN, FIN, URG...), inače knock niz neće biti uspješan.

Prednosti Port knockinga se otprilike ovdje, nažalost, završavaju.

Negativne strane Port knockinga su:

- unosi dodatno kašnjenje prilikom spajanja na poslužitelj jer korisnik mora napraviti dodatne predradnje

- moguće je, uz određene tehničke pretpostavke i znanje, presresti sekvencu portova, i na taj način zaobići zaštitu

- morate otvoriti te knock portove na vatrozidu, te je sposobnom hackeru moguće nizom pokušaja

doći do prave kombinacije

- ne podržava enkripciju, barem u osnovnoj izvedbi

Postoje druge implementacije Port knockinga, koje pokušavaju umanjiti ove nedostatke, ali u svrhu dodatne zaštite i "obični" knocking je dobar.

Više o Port knockingu pročitajte na http://www.portknocking.org, ili na drugim web sjedištima gdje se objašnjava princip rada.

U sljedećem nastavku ćemo vam pokazati kako upogoniti cijeli sustav i unijeti dodatnu razinu zaštite na vaš poslužitelj.

- Logirajte [3] se za dodavanje komentara

pon, 2010-01-18 12:07 - Željko Borošt**Kuharice:** Linux [9]
Kategorije: Servisi [10]
Vote: 4

Vaša ocjena: Nema Average: 4 (1 vote)

Knocking on Heaven's Door, 2. dio

 U prošlom članku smo opisalo što je "knocking" sustav, no što nam je za njegovu implementaciju sustava potrebno? Osim, naravno, iptables vatrozida, potreban je **knockd**, Port knocking daemon, te ponešto konfiguracije. Prvo instalirajmo knockd na standardan način:

```
# apt-get install knockd
```

U /etc/default/knockd promijenite varijablu START_KNOCKD=0 u START_KNOCKD=1.

```
# vim /etc/default/knockd
...
```

Glavna konfiguracijska datoteka je /etc/knockd.conf. Prvo što trebamo promijeniti je sekvenca koja otvara port:

```
[openSSH]
  sequence    = 7777,1234,8888,9876
  seq_timeout = 5
  command     = <iptables naredba>
```

Jasno je da ne smijete rabiti niti default portove koji dolaze uz paket, kao ni portove iz ovog članka. Ne trebamo objašnjavati zašto?

Uzmite 3 ili 4 različita porta, ne manje. Nemojte ni pretjerivati. Također, potrudite se izabrati niz koji će koliko-toliko biti pamtljiv, a opet nepredvidljiv napadačima. Nije preporučljivo uzeti niže portove, jer su to obično Well-known [11] portovi, pa će napadači tražiti slabosti u tim servisima (iako samog servisa nema).

U datoteci knockd.conf pod sekcijom [closeSSH] upišite isti niz, obrnuti ili pak nešto treće. Ovaj će niz zatvoriti port 22.

```
[closeSSH]
  sequence    = 9876,8888,1234,7777
  seq_timeout = 5
  command     = <iptables naredba>
```

Direktiva "**seq_timeout**" označava koliko će dugo knockd čekati da se cijeli niz izvrši i može biti u bilo kojoj sekciji. Kad to vrijeme istekne, niz morate ponoviti od početka.

Direktiva "**command**" je ona koja obavlja "pravi" posao, ali ju je potrebno izmijeniti u ovisnosti o konfiguraciji vašeg vatrozida. Osnovna je:

```
/sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
```

za otvaranje porta, odnosno

```
/sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
```

za zatvaranje porta.

U dokumentaciji knockda se spominje "-A" ("append") umjesto "-I" ("insert"), no to ovisi o vašoj konfiguraciji vatrozida. Pravilo iz knockda mora doći prije završnog DENY pravila, pa je možda sigurnije ubaciti to pravilo na
početak svih pravila, pogotovo ako nemate vrlo složen vatrozid.

Pogledajte dokumentaciju iptablesa ukoliko vam je potrebno više informacija.

Ako ste možda pomislili da bi bilo zgodno da se port sam zatvori nakon što ste završili rad na poslužitelju, knock i to podržava. Umjesto "[openSSH]" i "[closeSSH]", upotrijebite "[opencloseSSH]":

```
[opencloseSSH]
        sequence      = 7777,1234,8888,9876
        seq_timeout   = 5
        tcpflags      = syn,ack
        start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --syn -j ACCEPT
        cmd_timeout   = 15
        stop_command  = /sbin/iptables -D INPUT -s %IP% -p tcp --syn -j ACCEPT
```

Direktiva "**cmd_command**" određuje koliko imate vremena spojiti se na port 22. Nakon 15 sekundi (ili koliko odredite), pravilo se briše.

Bitna napomena, kako bi ovaj način radio, vaš vatrozid mora biti konfiguriran uporabom ESTABLISHED konekcija, u suprotnom će vam novouspostavljenu konekciju knockd jednostavno - prekinuti. Evo kako to primjerice može izgledati:

```
ACCEPT     tcp  --  161.53.XXX.0/26      0.0.0.0/0           tcp dpt:22
ACCEPT     tcp  --  193.198.YYY.128/25      0.0.0.0/0           tcp dpt:22
ACCEPT     tcp  --  127.0.0.0/24         0.0.0.0/0           tcp dpt:22
DROP       tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22 state INVALID
,NEW,RELATED,UNTRACKED
```

Dodavanje zadnje direktive radite ovako:

```
# iptables -A INPUT -s <RANGE> -p tcp --dport 22 -m state \! --state ESTABLISHED -j D
ROP
```

Na kraju, ukoliko daemon nije već startan, pokrenite ga:

```
# /etc/init.d/knockd start
Starting Port-knock daemon: knockd.
```

Kako bi testirali, odnosno mogli rabiti cijeli sustav, s paketom osim daemon dolazi i klijent, "knock". Njegova sintaksa je jednostavna, otkucajte na sustavu s kojeg se želite spojiti:

```
$ knock -v knockd_posluzitelj 7777 1234 8888 9876
```

Nakon toga se možete spojiti uobičajeno:

```
$ ssh korisnik@knocd_posluzitelj
```

Imajte na umu da u direktivu "command" možete upisati i bilo koju drugu naredbu, pa tako možete s određenim nizom ugasiti poslužitelj, rebootati ga,poslati predefinirani mail, upaliti ili ugasiti druge servise i slično.

Za Windows operativni sustav, postoje mnogi klijenti, a jedan je "knock.exe" http://www.zeroflux.org/proj/knock/files/knock-win32.zip [12]. Njih možete staviti u batch ili cmd skriptu koju ćete napisati i pokretati svoj SSH klijent nakon toga.

Umjesto knock.exe, možete rabiti bilo koji drugi program (netcat, nmap, pa čak i obični "telnet" program). Svi su oni dostupni i na linuxu i na Windowsu.

Više informacije možete naći na http://www.portknocking.org.

- Logirajte [3] se za dodavanje komentara

sri, 2010-01-20 12:23 - Željko Boroš**Kuharice:** Linux [9]

Kategorije: Servisi [10]
Vote: 0

No votes yet

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 [3] se za dodavanje komentara

sub, 2009-10-31 19:13 - Željko Borš**Kuharice:** Linux [9]
Kategorije: Software [13]
Servisi [10]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr./node/691

Links
[1] http://en.wikipedia.org/wiki/Rlogin
[2] http://en.wikipedia.org/wiki/Telnet
[3] https://sysportal.carnet.hr./sysportallogin
[4] https://sysportal.carnet.hr./node/542
[5] https://sysportal.carnet.hr./node/71
[6] http://en.wikipedia.org/wiki/Botnet
[7] http://en.wikipedia.org/wiki/Intrusion_detection_system
[8] http://en.wikipedia.org/wiki/Security_through_obscurity
[9] https://sysportal.carnet.hr./taxonomy/term/17
[10] https://sysportal.carnet.hr./taxonomy/term/28
[11] http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
[12] http://www.zeroflux.org/proj/knock/files/knock-win32.zip
[13] https://sysportal.carnet.hr./taxonomy/term/25