Bez zaporke uz pomoć ssh-agenta

Koliko ste samo puta prilikom ukucavanja zaporke pomislili "pa zašto sam odabrao baš tu, za kucanje tešku zaporku". Ili, možda administririrate više poslužitelja i na dan rabite različite zaporke na desetke puta. Zar ne bi bilo jednostavnije kada bi se to moglo izbjeći?

Naravno da rješenja postoje. Najjednostavnije je držati otvorene sesije unutar screena ili tmuxa, poznatih multipleksera korisničkih ljuski. Ovo uistinu jest nekakvo rješenje, ali zaporku morate ukucati barem jednom, što omogućava napadaču da je presretne. Postoji i drugi način, a to je **ssh-agen**t.

Pravo je čudo da program ssh-agent već nismo opisali. Ukratko, ssh-agent čuva vaš "identitet" unutar vlastitog poslužitelja (daemona), koji pri uspostavljanju veze s udaljenim računalom putem protokola SSH automatski šalje vaše podatke poslužitelju. Na ovaj način je moguće prijaviti se na udaljeni poslužitelj bez unosa zaporke. Kako zaporka nije ni otkucana, njeno presretanje ni nije moguće.

Što je potrebno da se ovakav scenarij odigra? Prvo je potrebno napraviti vlastiti identitet, odnosno ključ na stroju sa kojeg se želite spajati na druge strojeve bez zaporke:

```
$ ssh-keygen -t dsa -f ~/.ssh/id_dsa -C "korisnik@domena.hr"
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/korisnik/.ssh/id_dsa.
Your public key has been saved in /home/korisnik/.ssh/id_dsa.pub.
The key fingerprint is:
eb:42:ca:2c:cb:d1:13:12:bb:50:10:cf:c6:31:eb:e8 korisnik@domena.hr
The key's randomart image is:
+--[ DSA 1024]----+
|o.o              |
| = +             |
|  O              |
| = o             |
|o + .   S        |
|.. + ..  .       |
| Eoooo  .        |
| ...+...         |
|  oo   ..        |
+-----------------+
```

Naredbom ssh-keygen postavit ćemo ključ u datoteku **id_rsa** u korisničkom direktoriju (/home/korisnik ili jednostavnije $HOME), odnosno njegovom poddirektoriju **.ssh**. Ključ je u 1024-bitnom DSA formatu, a sa opcijom -C odaberite riječ (zapravo, komentar) po kojem ćete znati o kojem se ključu radi.

Prilikom generiranja ključa bit ćete upitani za ***passphrase***. Za razliku od passworda, passphrase je dulji, obično se koristi cijela rečenica. Sigurnost ključa je veća što je passphrase dulji.

Sljedeća akcija je prebacivanje vašeg ključa na drugi poslužitelj, odnosno na sve poslužitelje na koje se imate namjeru spajati. Obično je bolje samo nadodati ključ na kraju datoteke *authorized_hosts*, jer se u toj datoteci mogu nalaziti i drugi ključevi koje ne želimo izgubiti. Možemo to učiniti *onelinerom*:

```
cat ~/.ssh/id_dsa.pub | ssh korisnik@druga.domena.hr 'cat - >> ~/.ssh/authorized_keys
'
```

Stigli smo gotovo do kraja ali moramo učiniti još nekoliko koraka. Ukoliko bismo se sada pokušali spojiti na drugi poslužitelj, dobili bismo sljedeću situaciju:

```
$ ssh korisnik@druga.domena.hr
Enter passphrase for key '/home/korisnik/.ssh/id_dsa':
```

Dakle sustav bi nas upitao za passphrase koji smo definirali maločas. Kako je passphrase još dulji nego password, nismo dobili nikavo poboljšanje i očigledno se mora još nešto učiniti. Moramo pokrenuti sam daemon, ali na ovakav način:

```
$ eval `ssh-agent`
SSH_AUTH_SOCK=/tmp/ssh-HpOPm15648/agent.15648; export SSH_AUTH_SOCK;
SSH_AGENT_PID=15649; export SSH_AGENT_PID;
echo Agent pid 15649;
```

Da nismo rabili naredbu eval, morali bismo napraviti copy/paste ova tri retka koja su bila ispisana te ih takve izvršiti unutar trenutne ljuske.

No, još nismo spremni za rad. Moramo dati generirane ključeve daemonu ssh-agent, a to se može napraviti pomoću naredbe **ssh-add**:

```
$ ssh-add .ssh/id_dsa
Enter passphrase for .ssh/id_dsa:
Identity added: .ssh/id_dsa (.ssh/id_dsa)
$ ssh-add -l1024 eb:42:ca:2c:cb:d1:13:12:bb:50:10:cf:c6:31:eb:e8 .ssh/id_dsa (DSA)
```

Daemon može sadržavati više ključeva koje možemo vidjeti s opcijom "-l".

Sada se konačno možemo prijaviti bez uporbe zaporke:

$ ssh korisnik@druga.domena.hr
Linux po 2.6.32-5-686-bigmem #1 SMP Fri Sep 9 21:28:24 UTC 2011 i686
Agent pid 22657

Odjavom sa sustava prestaje raditi na ovaj način pokrenut daemon ssh-agent. Odnosno, trebao bi, prestati raditi. Da smanjimo rizik od toga da proces ostane aktivan i tako omogući zloporabu, valjalo bi ga ugasiti prilikom odjave, tako da u datoteku .logout upišemo:

```
kill $SSH_AGENT_PID
```

Ssh-agent možemo pokrenuti automatski prilikom prijave na sustav, tako da u datoteku .bash_profile upišemo:

```
SSH_ENV="$HOME/.ssh/environment"
```

```
function start_agent {
     echo "Initialising new SSH agent..."
     /usr/bin/ssh-agent | sed 's/^echo/#echo/' > "${SSH_ENV}"
     echo succeeded
     chmod 600 "${SSH_ENV}"
     . "${SSH_ENV}" > /dev/null
     /usr/bin/ssh-add;
}

# Source SSH settings, if applicable

if [ -f "${SSH_ENV}" ]; then
     . "${SSH_ENV}" > /dev/null
     ps -ef | grep ${SSH_AGENT_PID} | grep ssh-agent$ > /dev/null || {
         start_agent;
     }
else
     start_agent;
fi
```

Iako ova skripta brine i o zaostalim ssh-agentima, i dalje će kod prvog pokretanja biti potrebno dodati ključeve i ukucati passphrase, pa to imajte na umu.

Naravno, sasvim je legitimna uporaba da način da ssh-agent pokrećete samo kad za ovakvu *passwordless* autentikaciju imate potrebu, te ga nakon završetka posla ugasite.

Ugodan rad!

pet, 2012-10-26 15:02 - Željko Borš**Kuharice:** Linux [1]
Vote: 0

No votes yet

Source URL: https://sysportal.carnet.hr/node/1124

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