

# Zabrana korištenja stare zaporkke

Zdravko Rašić | 15 travnja, 2016

Naši korisnici uvijek traže što jednostavnije načine upotrebe resursa koji su im dodijeljeni, ne vole učiti i ne žele pamtiti previše stvari. Tako ne vole ni mijenjati zaporku (Čemu, kad je i stara zaporka dobra?). Vaš poslužitelj, iz sigurnosnih razloga, možete podesiti tako da korisniku nakon nekoliko mjeseci istekne zaporka, pa ga sustav prisiljava da je promjeni, ali korisnici kao korisnici, najčešće opet utipkaju staru zaporku.

No, postoji način kako onemogućiti recikliranje starih zaporki. Da bi korisnika prisilili da prilikom izmjene zaporkke odabere posve novu zaporku, na sistemu moramo napraviti nekoliko jednostavnih izmjena u "pam" (Pluggable Authentication Modules) konfiguraciji za autentikaciju.

Prvi korak je instalacija "cracklib" modula koji se koristi za provjeru i usporedbu jakosti odabrane zaporkke.

```
debian: /etc/pam.d
7638D0442B90D010
W: There is no public key available for the following key IDs:
7638D0442B90D010
W: There is no public key available for the following key IDs:
9D6D8F6BC857C906
root@debian      :/etc/pam.d# apt-get install libpam-cracklib
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  cracklib-runtime libcrack2
The following NEW packages will be installed:
  cracklib-runtime libcrack2 libpam-cracklib
0 upgraded, 3 newly installed, 0 to remove and 165 not upgraded.
Need to get 327 kB of archives.
After this operation, 1,245 kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://ftp.hr.debian.org/debian/ wheezy/main libcrack2 1386 2.8.19-3 [59.9 kB]
Get:2 http://ftp.hr.debian.org/debian/ wheezy/main cracklib-runtime 1386 2.8.19-3 [184 kB]
Get:3 http://ftp.hr.debian.org/debian/ wheezy/main libpam-cracklib 1386 1.1.3-7.1 [83.7 kB]
Fetched 327 kB in 0s (1,892 kB/s)
Selecting previously unselected package libcrack2.
(Reading database ... 69596 files and directories currently installed.)
Unpacking libcrack2 (from ../libcrack2_2.8.19-3_1386.deb) ...
```

Nakon uspješne instalacije slijedi podešavanje `/etc/pam.d/common-password` datoteke. Potražimo redak:

```
IW common-password (Modified) Row 26 Col 44 8:24 Ctrl-K H for help
# login.defs.
#
# See the pam_unix manpage for other options.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.
# here are the per-package modules (the "Primary" block)
password requisite pam_cracklib.so retry=3 minlen=8 difok=3
password [success=1 default=ignore] pam_unix.so obscure use_authok try_first_pass sha512
# here's the fallback if no module succeeds
password requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password required pam_permit.so

# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

Na sam kraj linije, točnije iza enkripcije sha512, dodamo "**remember=broj**" gdje broj označava koliko dugo želimo čuvati stare zaporke.

Ako upišemo primjerice broj 2, sistem će čuvati zadnje dvije zaporke i neće dopustiti da se one ponove, ako upišemo 5 čuva zadnjih pet zaporki i tako dalje. Ovaj "povijesni" dio nalazi se u `/etc/oshadow` datoteci.

```
IW common-password Row 26 Col 1 8:26 Ctrl-K H for help
#
# See the pam_unix manpage for other options.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.
# here are the per-package modules (the "Primary" block)
password requisite pam_cracklib.so retry=3 minlen=8 difok=3
password [success=1 default=ignore] pam_unix.so obscure use_authok try_first_pass sha512 remember=2
# here's the fallback if no module succeeds
password requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password required pam_permit.so

# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

Snimimo izmjene i napravimo testiranje.

```
debian: /etc/pam.d
```

```
Changing password for pero.  
(current) UNIX password:  
New password:  
Retype new password:  
Password has been already used. Choose another.  
passwd: Authentication token manipulation error  
passwd: password unchanged  
debian:/etc/pam.d$
```

Kao što se vidi iz primjera prilikom izmjene zaporke i pokušaja upisa iste kojom smo se prijavili dobijemo obavijest da je zaporka već bila korištena te da se odabere nova.

Ono što bi još trebalo postaviti je minimalna duljina zaporke. Ponovo editirajmo datoteku `/etc/pam/common-passwd` i pronađimo liniju `passwd`.

Tu upišemo `"minlen=8"`, što govori da je minimalna duljina zaporke 8 znakova:

Još jedan primjer je prisiljavanje korisnika da prilikom promjene zaporke mora upisati minimalno 3 broja (da nisu svi znakovi slova):

```
password requisite pam_cracklib.so retry=3 minlen=8 dcredit=-2
```

Naravno da možete uključiti i neke druge uvjete, poput uporabe velikih i malih slova te ostalih znakova.

U gornjoj liniji samo još dodajte: