Védekezés levélkiszolgáló túlterhelése ellen

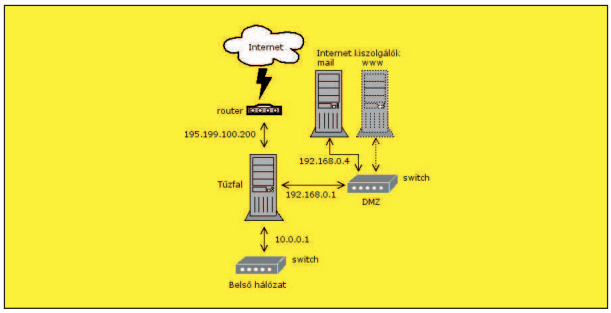
Egy győri szakközépiskola tanár-rendszergazdájaként linuxos kiszolgálókat üzemeltetek. A múlt év végétől, egyre gyakrabban tapasztaltuk azt a hibajelenséget, hogy a munkaállomásokon működő ügyfélprogramok hosszasan próbálkoztak egy-egy e-mail elküldésével, majd „Kapcsolat megszakadt” hibaüzenettel leálltak...

Mivel eddig nem tapasztaltam ilyen rendellenességet, több dologra is gyanakodtam. Végül levélkiszolgálónk forgalmának és terheltségének elemzésével világossá vált a valódi ok, és hamarosan a megoldást is sikerült megtalálnom.
A levelező kiszolgálónkon ***Debian Linux*** alatt ***Postfix MTA*** üzemel, amely a ***/var/log/mail.log*** fájlba naplóz. A naplóbejegyzéseket root-ként egyszerűen a ***Midnight Commander*** segítségével néztem meg. Itt csak 4 jellegzetes sort mutatok meg, amelyek egyetlen küldési kísérlethez tartoznak. Kiderült, hogy kiszolgálónkhoz gyors egymásutánban több IP-ről nagyszámú e-mail érkezett, mégpedig ismeretlen felhasználók címére. Ahogy a 3. sorban látható, az ismeretlen felhasználónak küldött levél a 450-es számú `User unknown` hibaüzenetet váltja ki és a bejegyzésbe a `reject` (elutasítva) kifejezés kerül:

```
1: Feb  6 06:51:22 mail postfix/smtpd[17399]:
➥connect from
➥mx6a.uniserve.ca[216.113.192.92]
2: Feb  6 06:51:23 mail postfix/smtpd[17399]:
➥06616662CA:
➥client=mx6a.uniserve.ca[216.113.192.92]
3: Feb  6 06:51:23 mail postfix/smtpd[17399]:
➥reject: RCPT from
➥mx6a.uniserve.ca[216.113.192.92]: 450
➥<estela.rudolph68@pattantyus-gyor.sulinet.hu>:
➥User unknown; from=<>
➥to=<estela.rudolph68@pattantyus-gyor.sulinet.hu>
4: Feb  6 06:51:28 mail postfix/smtpd[17399]:
➥disconnect from mx6a.uniserve.ca[216.113.192.92]
```

A ***Postfix*** csak akkor utasítja el az ismeretlen felhasználóknak küldött leveleket, ha megfelelően van beállítva. Ehhez az ***/etc/postfix/main.cf*** fájlban, a következő soroknak kell szerepelni:

```
1: alias_maps = hash:/etc/aliases
2: alias_database = hash:/etc/aliases
3: local_recipient_maps = $alias_maps
➥nis:passwd.byname
```



1. ábra Az iskola hálózatának felépítése

Az 1. és 2. sor az ***alias fájlok*** helyét adja meg. Ha például valakinek `horvathl` a felhasználói neve, de a `horvath.laszlo@domain` címmel szeretne levelezni, akkor ezt az alias fájlban kell megadni. Azt is itt lehet megadni, hogy a `titkarsag@domain` címre küldött e-mail melyik felhasználónak, vagy felhasználóknak a postafiókjába kerüljön.
A 3. sor arra utasítja a ***Postfixet***, hogy csak azon felhasználók címére fogadjon el e-mailt, akik vagy szerepelnek az alias-ban, vagy a ***NIS*** adatbázisban léteznek (a kiszolgálóinkon jelenleg ***NIS*** azonosítás működik).
Az elutasításnak természetesen más okai is lehetnek. Ha például a küldő ***IP***-nek nincs ***DNS***-bejegyzése, vagy ha továbbítónak (relay) akarja használni a kiszolgálónkat. Ilyenkor persze másféle hibaüzenetek keletkeznek.
Először arra gyanakodtam, hogy a rengeteg téves próbálkozás valamilyen e-mail küldő vírus tevékenységének a nyoma. A következő módon megvizsgált levelek tartalma azonban azt bizonyította, hogy valójában reklámküldési kísérletekről van szó. A ***/etc/postfix/main.cf*** fájlon a következő változásokat hajtottam végre:

```
1: #local_recipient_maps = $alias_maps
➥nis:passwd.byname
2: luser_relay =
➥felhasznalonev@pattantyus-gyor.sulinet.hu
```

Az 1. sor a postafiók meglétének ellenőrzését kikapcsolja (megjegyzésjelet tettem a sor elejére). A 2. sor hatására

a rendszer az ismeretlen felhasználóknak küldött leveleket az itt megadott postafiókba továbbítja.
A módosítás után a Postfixet újra kell indítani:
`/etc/init.d/postfix restart`

Vigyázat!!! Ha valahova olyan sok reklámlevél érkezik mint a mi címünkre, rövid idő alatt is rengeteg e-mail kerülhet a megadott fiókba. Ezért néhány perc után a */etc/postfix/main.cf* fájlt helyreállítva a ***Postfixet*** újra kell indítani.

A kiszolgáló forgalmi statisztikája és terheltségének vizsgálata

A napló tartalmának megismerése után kíváncsi voltam a szerver forgalmára. A ***Postfix*** esetében ehhez csak a ***pflogsumm*** csomagot kell telepíteni. Statisztika készítése a naplófájlban lévő valamennyi nap adataival:

```
pflogsumm.pl /var/log/mail.log > /root/statisztika
```

Csak az aktuális napi forgalom statisztikája:

```
pflogsumm.pl -d today /var/log/mail.log >
➥/root/statisztika
```

A `today` helyett `yesterday`-t írva pedig az előző napi forgalom statisztikáját készíti el.
Az elkészített statisztikából látszik, hogy egy nap alatt 40483 e-mailt fogadott (`received`), s ebből mindössze 591-et továbbított (`delivered`), 40028-at pedig elutasított (`rejected`):

```
Postfix log summaries for Feb 6

Grand Totals

messages

  40483   received
    591   delivered
      1   forwarded
     11   deferred  (33  deferrals)
      0   bounced
  40028   rejected
```

Lejjebb látható, hogy miért, illetve melyik küldőtől hány e-mailt utasított el a rendszer. Ebből az derült ki, hogy az elutasítások túlnyomó többsége azért történt, mert ismeretlen felhasználónak (`User unknown`) címezték a levelet.

```
message reject detail
---------------------
  RCPT
    Sender address rejected: Access denied
          26   spamblocker-
               ➥challenge@bounce.earthlink.net
          14   molnarlouis@bellsouth.net
          13   ErnoHoka@aol.com
    Sender address rejected: Domain not found
          20   AntiVirus@axxis.local
          14   Symantec_Mail_Security_for_
               ➥SMTP@MSL.COM
           3   MAILER-DAEMON@gcs.gateway
           1   admin@system.mail
    User unknown
         437   cox.net
         256   rr.com
         244   swip.net
         199   siteprotect.com
         154   netvigator.com
        .
        .
```

Ha a levelezőkiszolgáló elutasítja a nemkívánatos e-maileket, akkor miért terhelődik túl? A ***Postfix*** az internetről és a belső hálózatról is, az ***smtpd*** programjával fogadja az e-maileket, amely alapértelmezés szerint egyszerre legfeljebb 50 kapcsolatot tud kezelni. A ***Postfix*** 2.2-es változata óta lehetőség van ennek az értéknek a megemelésére, de a hardver gyorsasága és főleg a sávszélesség által szabott korlátok miatt nem biztos hogy ez célszerű.
Az adott pillanatban élő ***smtpd***-kapcsolatok számát a következő paranccsal néztem meg:

```
ps ax | grep smtpd | wc -l
```

Ennek értéke a fent bemutatott terheltség mellett legtöbbször 50 volt így nem tudott új kapcsolatot felépíteni a ***Postfix***. A levelező ügyfelek meghatározott ideig próbálkoztak, majd időtúllépés miatt írták ki a *„Kapcsolat megszakadt”* hibaüzenetet.

A bejövő smtp kapcsolatok időben történő korlátozása

Az első ötletem az volt, hogy csomagszűrő segítségével korlátozom az internet felől felépíthető kapcsolatok számát a levélkiszolgáló felé. Mivel intézményünkben a levelező kiszolgáló tűzfal mögött működik, a tűzfalgép szűrési szabályaiba építettem be a korlátozást:

```
1:  eth_kulso='eth0'
2:  IPTABLES=/sbin/iptables

4:  $IPTABLES -t nat -N pre_smtp
5:  $IPTABLES -t nat -A pre_smtp -m limit --limit
    ➥20/minute -j RETURN
6:  $IPTABLES -t nat -A pre_smtp -j DROP
7:  $IPTABLES -t nat -A PREROUTING  -i $eth_kulso
    ➥-d 195.199.100.200 -p tcp --dport smtp --syn
    ➥-j pre_smtp

9:   $IPTABLES -t nat -A PREROUTING -p tcp -d
     ➥195.199.100.200 --dport smtp -j DNAT --to
     ➥192.168.0.4:25
10:  $IPTABLES -t nat -A POSTROUTING -p tcp -s
     ➥192.168.0.4 --sport smtp -j SNAT --to
     ➥195.199.100.200:25
```

Itt a tűzfal külső (internetre csatlakozó) hálózati kártyájának IP címe `195.199.100.200`, míg a levélkiszolgáló címe `192.168.0.4`.

A beállítások jelentése a következő:

- 1. sor: A tűzfalgép külső (internetre csatlakozó) ethernet kártyájának megadása.
- 2. sor: Az `iptables` parancs elérési útja, hogy később ne kelljen mindig megadni.

- 4. sor: Új `pre_smtp` nevű lánc létrehozása
- 5. sor: Ha a kapcsolatok száma nem több percenként 20-nál, akkor visszatér a láncból, vagyis elfogadja (az értéket kísérletileg kell megállapítani, a szerver feldolgozóképességétől függ)
- 6. sor: Ha az előbbi sorban megadottnál több kapcsolat akar felépülni, akkor eldobja azokat.
- 7. sor: Ha a külső kártyán keresztül (internet felől) a tűzfalgép külső IP címére a TCP protokollon a 25-ös (smtp) portra kapcsolatteremtő csomag érkezik, ugrás a létrehozott `pre_smtp` láncra
- 9. sor: Ha TCP protokollon a tűzfal gép külső címén a 25-ös (smtp) portra érkezik csomag, átirányítja a levelező szerverre.
- 10. sor: Ha TCP protokollon a levelező szerver 25-ös (smtp) portjáról érkezik csomag, átirányítja a tűzfal gép külső kártyájára.

A fenti pár soros szkriptet természetesen módosítani kell, ha a levélkiszolgáló közvetlenül csatlakozik az internetre. Ekkor a `-t nat` részeket el kell hagyni, a `PREROUTING` bejegyzést `INPUT`-ra kell cserélni, a 9. és 10. sorok pedig természetesen elmaradnak.
A fenti korlátozás csökkentette a terhelést, a belső levelezést zavartalanná tette, de sajnos a kívülről érkező legális leveleket is várakozásra kényszerítette és időtúllépés miatt nem mindegyik jutott el a címzetthez. Olyan megoldást kellett ezért keresnem, ami lehetőleg csak a kéretlen levelek küldőit korlátozza.

Az „ellenséges” IP címek szelektív tiltása

Ötletem a következő volt. A levélkiszolgáló naplóállományát periodikusan átvizsgálva, ki lehet keresni azokat a címeket, melyekről megadott számú elutasított e-mail érkezik egy megadott idő alatt és ezeket egy szintén megadott időre ki kell tiltani. Természetesen nem lehet örök időre kitiltani egyetlen IP címet sem és nem is érdemes, hiszen az IP-k változnak. Azt is figyelembe kellett vennem, hogy olyan címek ne legyenek kitiltva, ahonnan hivatalos levél is érkezhet.
A kitiltásokat és engedélyezéseket naplózni kell, hogy bármikor meg lehessen nézni a tiltásokat és engedélyezéseket.
A feladat megoldására héjprogramot készítettem, mely a tiltást csomagszűréssel valósítja meg. Gondolom néhányan csóválják a fejüket, miért nem Perl-ben vagy esetleg C-ben írtam. Eredetileg az egyszerűség miatt döntöttem a héjprogram mellett, a gyakorlat azonban bebizonyította, hogy a héj és a különböző szűrőparancsok minden szükséges eszközt biztosítanak egy ilyen feladat hatékony megoldásához. Itt csak az érdekesebb sorokat mutatom be. A teljes program letölthető a ***Linuxvilág magazin*** webhelyéről (***http://www.linuxvilag.hu***).

```
1: #!/bin/sh
5: # Keszitette: Jaszberenyi Jozsef, 2005

21: MAX_REJECT=10
24: TILTOTT_ORAK=36
27: MAX_LOG=5000

30: LOGFAJL="/var/log/mail.log"
33: TILTOTTIPDIR="/var/log/ipdisable"
36: TILTOTTIPFAJL="tiltottIP.log"
39: NAPLO="naplo.log"
42: NEMTILTHATOIP="nemtilthatoIP"
45: TILTOTTIPFAJL="$TILTOTTIPDIR/$TILTOTTIPFAJL"
46: NAPLO="$TILTOTTIPDIR/$NAPLO"
47: NEMTILTHATOIP="$TILTOTTIPDIR/$NEMTILTHATOIP"
        .
50: ECHO=/bin/echo
51: [ -x $ECHO ] || exit 5
        .
        .
126: trap '$RM -f /tmp/$$_?.tmp; exit 2' 1 2 3 9 15

129: TMP1="/tmp/$$_1.tmp"
130: TMP2="/tmp/$$_2.tmp"
131: TMP_DOMAIN="/tmp/$$_3.tmp"
132: TMP_IPCIM="/tmp/$$_4.tmp"
133: TMP_TILTANDO="/tmp/$$_5.tmp"

137: if ! [ -f $NEMTILTHATOIP ]
138: then
139:   $ECHO "Nem talalom a $NEMTILTHATOIP fajl-t!"
140:   exit 5
141: fi

153: if ! [ -f $TILTOTTIPFAJL ]
154: then
155:     $TOUCH $TILTOTTIPFAJL
156: fi

162: HONAP=`$DATE +%b`
163: HONAPOK=`$ECHO $HONAP | $CUT -c 1 | $TR a-z A-Z`
164: HONAPOK=$HONAPOK`$ECHO $HONAP | $CUT -c 2- | $TR
➥A-Z a-z`
167: NAP=`$DATE +%e | $TR -d ' ``
170: TILTASI_IDO=$[`$DATE +%s` / 3600]

177: NAPLOARCHIV=$NAPLO.tgz
180: if [ -f $NAPLO ]
181: then
183:   MAX_LOG=$[$MAX_LOG * 1024]
185:   FILEMERET=`$LS -l $NAPLO | $AWK ' {print $5} '`
187:   if [ $FILEMERET -ge $MAX_LOG ]
188:   then
190:     $RM -f $NAPLOARCHIV
192:     $TAR cfz $NAPLOARCHIV $NAPLO
194:     $RM -f $NAPLO
195:   fi
196: fi

201: $ECHO "#------------------------------" >> $NAPLO
202: $ECHO "Start: "`$DATE` >> $NAPLO

204: # Kigyujti az aktualis napon keletkezett,
➥"reject:" kifejezest tartalmazo sorokat 9. mezojet
207: $GREP "reject:" $LOGFAJL | $AWK -v H=$HONAPOK -v
➥N=$NAP '$1 ~ H && $2 ~ N {print $9}' > $TMP1

209: # Kigyujti a domain neveket egy kulon listaba
```

```
210: $AWK -F[ '{print $1}' $TMP1 > $TMP_DOMAIN

212: # Kigyujti az IP cimeket egy kulon listaba
215: $SED s/^.*\\[// < $TMP1 | $SED s/].*$// >
➥$TMP_IPCIM

217: # Elollitja az uj listat soronkent egy IP-cim,
➥majd szokozzel a domain nev
218: $PASTE $TMP_IPCIM $TMP_DOMAIN > $TMP1

220: # Eltavolitja a nem tilthato IP-ket
221: for cim in `$CAT $NEMTILTHATOIP`
222: do
223:     $GREP -v ^$cim $TMP1 > $TMP2
224:     $CP -f $TMP2 $TMP1
225: done

227: # Sorba rendezi a sorokat IP szerint es meg-
➥szamolja melyik IP hanyszor fordult elo a listaban
228: $SORT < $TMP1 | $UNIQ -c > $TMP2

230: # Kivalogatja azokat az IP-ket, melyekrol a meg-
➥adott szamot
231: #  meghalado elutasitas tortent es kiirja
➥a tiltando IP-ket tarolo fajlba
232: $AWK -v DARAB=$MAX_REJECT '$1>=DARAB {print
➥$2":"$3}' < $TMP2 > $TMP_TILTANDO

235: # Kitiltja a tiltando cimeket ha nem szerepelnek
➥meg a tiltottIP fajlban
236: #  es listazza a tiltottIP fajlba idobelyeggel es
➥domain nevvel egyutt.
239: for sor in `cat $TMP_TILTANDO`; do
240:      cim=`$ECHO $sor | $AWK -F: ' {print $1} ' `
241:      domain=`$ECHO $sor | $AWK -F: ' {print $2} ' `
242:      if ! $GREP -s $cim $TILTOTTIPFAJL >>
➥/dev/null 2>&1
243:      then
244:         $IPTABLES -I INPUT -i eth0 -p tcp --dport
➥smtp -s $cim/32 -j REJECT
245:    $ECHO -n $cim >> $TILTOTTIPFAJL
246:    $ECHO -n : >> $TILTOTTIPFAJL
247:    $ECHO -n $TILTASI_IDO >> $TILTOTTIPFAJL
248:    $ECHO -n : >> $TILTOTTIPFAJL
249:    $ECHO $domain  >> $TILTOTTIPFAJL
250:    $ECHO "Tiltva:   $cim    $domain" >> $NAPLO
251:     fi
252: done

255: # Ha van olyan IP a tiltottIP fajlban ami a meg-
➥adott idonel regebben lett tiltva,
256: # akkor torli a tiltast, torli a tiltottIP
➥fajlbol a bejegyzest es naplozza
258: for sor in `cat $TILTOTTIPFAJL`; do
259:      # Kulonvalasztja a mezoket
260:      cim=`$ECHO $sor | $AWK -F: ' {print $1} ' `
261:      ERTEK=`$ECHO $sor | $AWK -F: ' {print $2} ' `
262:      domain=`$ECHO $sor | $AWK -F: ' {print $3} ' `
263:
264:     if [ $ERTEK -le $[$TILTASI_IDO-$TILTOTT_ORAK] ]
265:     then
266:         $IPTABLES -D INPUT -i eth0 -p tcp --dport
➥smtp -s $cim/32 -j REJECT
267:            $GREP -v $sor $TILTOTTIPFAJL > $TMP1
268:    $MV -f $TMP1 $TILTOTTIPFAJL
269:    $ECHO "Engedelyezve: $cim $domain" >> $NAPLO
270:     fi
271: done

274: $RM -f /tmp/$$_?.tmp

276: #Script futas befejezesi idopontjanak naplozasa
277: $ECHO "End: "`$DATE` >> $NAPLO
278: $ECHO

280: exit 0
```

A programfájlt természetesen futtathatóvá kell tenni: `chmod 755 ipdisable`

Elemzés

- A 21. sorban adjuk meg, hány elutasított e-mail után tiltjuk ki a küldő IP címét. Ezt az értéket kísérletileg kell meghatározni a terheltség függvényében. A program alapértelmezésként csak az aznapi bejegyzéseket vizsgálja.
- A 24. sor tartalmazza, hogy hány órára lesz kitiltva az IP cím. Célszerű ezt az értéket 24-nél nagyobbra állítani, mert az elutasított e-mailek jelentős része nem levélkiszolgálón keresztül érkezik, hanem közvetlenül internetre csatlakozott gépekről, melyek IP címe általában 24 óra után változik meg (a szolgáltató bontja a kapcsolatot és új IP-t oszt ki)
- A 27. sorban a naplófájl legnagyobb méretét adjuk meg (KB-ban), míg a 30-47 sorok bizonyos elérési utakat és fájlneveket adnak meg.
- A `nemtilthatoIP` nevű fájlban (42. sor) kell megadni azokat az IP címeket, illetve tartományokat, melyeket a programnak soha nem szabad kitiltania. Itt mindenek előtt célszerű megadni a belső hálózat IP tartományát. A fájlban megadható teljes IP cím (`193.233.21.14`), vagy IP tartomány is (`10.0.`).
- Az 50-123 sorokban megadjuk bizonyos ***UNIX*** parancsok elérési útját, illetve ellenőrizzük meglétüket. A 126. sor felhasználói megszakítás, vagy a gép újraindulása esetén gondoskodik az átmeneti fájlok törléséről, amelyek helyét a 129-133 sorokban adjuk meg.
- A 137-156 sorokban egyes fájlok meglétét ellenőrizzük. A `nemtilthatoIP` fájlnak létezni kell a `TILTOTTIPDIR` változóban megadott könyvtárban, különben a program futása befejeződik. A többi fájlt a program létrehozza, ha még nem léteznek.
- 162-164 sorokban az aktuális dátum hónap nevét kérdezzük le, majd konvertáljuk ugyan olyan formátumra, mint ahogy a ***Postfix*** naplófájlban szerepel. A 167. sorban a dátum napját kérdezzük le és töröljük a felesleges szóközt, ami egyjegyű szám esetén szükséges.
- 170. sorban időbélyeget állítunk elő 1 órás felbontással. Az időbélyeg segítségével tudjuk megállapítani, mennyi idő óta van egy IP kitiltva.

- 177-196 sorokban a naplófájl maximális méretét korlátozzuk. Ha van naplófájl és a hossza eléri a megadott max. méretet, akkor tömörítjük, majd töröljük a már betömörített eredeti naplót. Ha volt már betömörített naplófájl, akkor azt töröljük.
- 201-202 sorokban kezdjük a naplózást, a program futás kezdeti időpontjának kiírásával.
- 207. sorban a ***Postfix*** naplófájlból kigyűjtjük egy átmeneti fájlba az aktuális napon keletkezett, `reject:` kifejezést tartalmazó soroknak a 9. mezőjét. Az átmeneti fájlban a domain után szögletes zárójelek között lesz az IP cím. Ennek a formátumnak a kezelése nem lenne egyszerű, ezért át kell alakítani.
- 210-218 sorokban olyan formátumú fájlt hozunk létre, melyet a későbbiekben könnyen tudunk feldolgozni. A fájl minden sorában egy ***IP*** címet írunk, majd szóközzel a hozzátartozó domaint. A 210. sorban létrehozzuk a domaineket tartalmazó ideiglenes fájlt. Mivel a domain után szóköz nélkül egy kezdő szögletes zárójel van, az `awk` programmal könnyen szét tudjuk választani az utána lévőktől. A 215. sorban az ***IP***-ket tartalmazó ideiglenes fájlt hozzuk létreolyan módon, hogy `sed` parancs segítségével töröljük a szögletes zárójelek előtti és utáni karaktereket, beleértve a zárójeleket is. A 218. sorban egyesítjük az ***IP***-ket és a domaineket tartalmazó fájlokat.
- 221-225 sorokban töröljük a fájlból a nem tiltható ***IP***-ket olyan módon, hogy csak azokat a sorokat másoljuk át, melyek nem úgy kezdődnek, mint a `nemtilthatoIP` sorai.
- 228. sorban megszámoljuk, hogy melyik sor hányszor fordul elő. A rendezésre azért van szűkség, mert az `uniq` csak így működik helyesen.
- 232. sorban kiválogatjuk azokat a sorokat, melyek legalább megadott darabszámban fordulnak elő. Ezeket az ***IP***-ket fogjuk kitiltani, ha eddig nem lettek (nem szerepelnek még a `tiltottIP.log` fájlban).
- 239-252 sorokban tiltjuk ki az „ellenséges” ***IP***-ket: ehhez sorra vesszük a kitiltandó ***IP***-ket és ha még nincsenek kitiltva, akkor kitiltjuk azokat. A tiltást naplózzuk a `tiltottIP.log` fájlba „IP:időbélyeg:domain” formában, valamint a `naplo.log` fájlba. A 242. sor végére azért írtam a `>> /dev/null 2>&1` részt, mert különben a `grep` kimenetéről minden futás után e-mailt küld.
- 258-271 sorokban engedélyezzük azokat az IP-ket, melyek már elég ideig voltak tiltva. Az engedélyezett IP bejegyzését töröljük a `tiltottIP.log` fájlból és az engedélyezést naplózzuk a `naplo.log` fájlba.
- 274. sorban töröljük az ideiglenes fájlokat.
- 277-278 sorokban fejezzük be a naplózást. A program futásának befejezési idejét a `naplo.log` fájlba írjuk.

Időzítés és a működés ellenőrzése

A programot rendszeres időközönként kell futtatni, mivel óránként jelentkeznek újabb kalóz IP-k. Ez legegyszerűbben a `crontab`-bal oldható meg. Egy tetszőleges nevű fájlba kell a `crontab` utasítást írni, majd rootként bejelentkezve `crontab fájlnév` paranccsal hozzáadni a root nevében végrehajtandó `crontab` listához. Nálam a program 15 percenként fut, így a `crontab` utasítás a következőképpen néz ki:

```
5-50/15 * * * * /eleresi_ut/ipdisable >/dev/null
2>&1
```

A `>/dev/null 2>&1` részre azért van szükség, hogy a program futásáról ne küldjön e-mailt.
A program minden egyes futásakor ír a naplóba, ezért legegyszerűbben a naplófájl (***/var/log/ipdisable/naplo.log***) tartalmának megjelenítésével lehet ellenőrizni a működését:

```
#-----------------------------------------------
Start: Tue Feb 15 16:02:01 CET 2005
Tiltva: 159.53.206.179  smtpext15.bankone.com
        .
        .
Engedelyezve: 162.33.130.251 unknown
        .
        .
End: Tue Feb 15 16:02:19 CET 2005
```

Ha nem történt sem tiltás sem engedélyezés, akkor csak a programfutás indulásának és befejezésének időpontja látszik. A naplót érdemes rendszeresen átnézni, és megvizsgálni, hogy nem tiltott-e ki olyan IP-t, ahonnan fontos levél is jöhet.
A ***tiltottIP.log*** fájlba az éppen kitiltott IP-k találhatók `IP:időbélyeg:tartomány` formában:

```
193.229.0.43:307876:fep34-0.kolumbus.fi
204.140.14.154:307876:kitty.warnerbros.com
64.243.89.147:307876:unknown
        .
        .
206.168.3.177:307876:unknown
```

Az `iptables` paranccsal is lehet ellenőrizni, hogy milyen IP címek vannak kitiltva:

```
iptables -L INPUT -n | grep REJECT
```

A következő parancs az éppen kitiltott IP-k számát írja ki:

```
iptables -L INPUT -n | grep REJECT | wc -l
```

Tapasztalat

A program több hete működik. Az általunk használt gépen körülbelül 30 másodperc alatt lefut. Beüzemelése óta kollégáim nem panaszkodtak, hogy nem tudnak levelet küldeni, kézbesítetlen e-mail pedig csak az első napokban volt, amíg a `nemtilthatoIP` fájlban meg nem adtuk az összes szükséges helyet.

Jászberényi József
(jaszberenyij@pattantyus-gyor.sulinet.hu)
Szeret biciklizni, kirándulni, olvasni, sörözni és szabadban főzni. A stratégiai játékoktól a műszaki CAD programokig sok minden érdekli. Legtöbbet szerverprogramokkal foglalkozik és néha mérgelődik.