A Sendmail megerősítése

Mick megvizsgálja a Sendmail biztonsági hiányosságait, és felépít egy internetes leveleket kezelő SMTP-átjárót.

Igen, ó igen, a Sendmail. Rajonghatunk sokoldalúságáért és elterjedtségéért, vagy éppen utálhatjuk nagyságáért, bonyolultságáért és biztonsági hibáiért. Az is megeshet, hogy a levélkiszolgálók világában új fiúként egyszerűen csak adunk egy esélyt a Sendmailnek (végül is a Sendmail kétségtelenül a legnépszerűbb nyílt forráskódú programcsomag az Interneten).
Nos, ha szóba kerül a biztonság, a divatos szemlélettel ellentétben a Sendmailt nem kell teljesen leírni, és arra sincs szükség, hogy a ***sendmail.cf*** ősi írásmódját elsajátítsuk (de tényleg nem árt, ha a keményvonalas Sendmail-guruk ismerik). Ebben a hónapban ezeket és más Sendmail biztonsági kérdéseket fogunk boncolgatni, és a Sendmail hasznos m4 makróit felhasználva gyorsan felépítünk egy biztonságos, de működőképes Simple Mail Transport Protocol (SMTP, azaz egyszerű levéltovábbító szabvány) átjárót az internetes levelek kezeléséhez.

Miért (vagy miért ne) használjunk Sendmailt?

A Sendmail az egyik legősibb internetes programcsomag, amit még mindig széles körben használnak. Elsőként a BSD UNIX 4.1c változatában jelent meg (1983 áprilisában), és egészen napjainkig saját kategóriájának legkedveltebb alkalmazása maradt. Az üzenettovábbító ügynökprogramok (MTA-k) között a Sendmail az Internet fő igavonója, ami a leveleket a hálózatok között megfelelően és (a végfelhasználó számára) átlátszóan közvetíti. Csakhogy a Sendmail sem mentes a hátrányoktól. Kedvező jellemzői közé sorolható: a Sendmail hatalmas felhasználói közösséggel bír, ennek eredményeképpen igen könnyű mind kereskedelmi forgalomban beszerezhető, mind ingyenes támogatást találni hozzá, nem is beszélve a gazdag elektronikus és nyomtatott leírásokról. Kiforrottságának hála elég megbízható és kiszámítható.
Kedvezőtlen jellemzői közé tartozik: a Sendmail hosszú története során elég sok „cruft”-ot (régi kódot) gyűjtött össze, így aztán lassan biztonsági hibáiról és „elhízottságáról” is elhíresült. Természetesen mindkét vád vitatható. Az évek során számos jelentős sérülékenységére derült fény, ugyanakkor napvilágra kerülésük után igen gyorsan ki is javították őket. A terebélyesség vádpontját illetően tény, hogy a Sendmail kódbázisa sokkal nagyobb, mint más MTA-ké (mondjuk a Qmailé vagy a Postfixé), és memóriafoglalása is kétségtelenül méretesebb – ennek azonban legalább annyira a monolitikusság (egyetlen végrehajtható állomány teszi elérhetővé a legtöbb Sendmail-képességet) az oka, mint a felgyülemlett régi kódsorok. Ha jobban belegondolunk, a hosszú évek során a Sendmail forrását már oly sok programozó vizsgálta át tüzetesen, hogy nehezen képzelhető, hogy az elmúlt húsz évet túl sok kizárólag csak történelmi jelentőségű és elavult kód élte volna túl érintetlenül.
Sokkal hasznosabb a monolitikusság kérdését vizsgálni. A Sendmailnek bizonyos feladatok ellátásához néha rendszergazdai jogosultsággal kell futnia, például ha több különböző felhasználó saját könyvtárába ír leveleket. Emiatt aztán a Sendmail az olyan rendszereken, ahol kizárólag levéltovábbító (e-mail relay) vagy átjárófeladatokat lát el, csakis különleges jogosultságok nélküli (unprivileged) felhasználóként futhat.
A Sendmailt összetettségéért is bírálják. Beállításfájljának, a ***sendmail.cf***-nek szövevényessége nem éppen ösztönző, hogy mást ne mondjak – véleményem szerint valahol a C programnyelv és a szabványos kifejezések közti nehezen behatárolható helyen helyezkedik el. Mindezek oka természetesen a Sendmail különleges hatékonysága (bár sokan szeretnénk, ha a Sendmail inkább a C-t, a szabványos kifejezéseket vagy más, kicsit szabványosabb beállításnyelvet használna a ***sendmail.cf***-ben – legalább ennyire bonyolult, de egyedi saját nyelve helyett). Manapság már ez a pont is erősen vitatható. A Sendmail jelenlegi változatait már m4 makrókon keresztül állíthatjuk be, amelyek sokkal kevesebb felhasználóellenes élményt okoznak, mint a ***sendmail.cf*** kézi szerkesztése.
Egyes felhasználók véleményétől függetlenül a Sendmail megkérdőjelezhetetlenül hatékony és jól támogatott program. Ha a Sendmail előnyeit többre tartjuk a hátrányainál, akkor jó csapatban vagyunk. Azonban még ennél is jobb csapatba kerülhetünk, ha a Sendmail biztonságos futtatását is elsajátítjuk.

A Sendmail felépítése

Mint korábbiakban is említettük, a Sendmail monolitikus felépítésű a tekintetben, hogy minden tényleges munkát egyetlen végrehajtható állomány (maga a Sendmail) végez. A Sendmailnek két működési módja létezik: meghívható igény szerint, ebben az esetben feldolgozza a várakozó leveleket, majd kilép; avagy állandóan futó háttérdémonmódban is elindíthatjuk.
A démonmód csak akkor szükséges, ha a kívülről jövő levelek fogadása is a Sendmail feladatai közé tartozik; amennyiben viszont kizárólag levélküldésre használjuk, nem kell démonként futtatnunk, sőt, tulajdonképpen akár itt abba is hagyhatnánk az olvasást, hiszen a Sendmailnek ehhez semmi szüksége további beállításokra – hacsak nem akarjuk chrootolva futtatni.
Az, hogy a Sendmail miképpen működik, erősen attól függ, hogyan indítottuk el. Ha démonként (azaz a `-bd` kapcsolóval) futtatjuk, a 25-ös TCP-kapun figyeli a bejövő SMTP-kapcsolatokat, és időnként megkísérli elküldeni a ***/var/spool/mqueue*** nevű kimenősor könyvtárában összegyűlt leveleket. Ha csak úgy meghívtuk, akkor azt a kimenő levelet próbálja meg kézbesíteni, amiért meghívtuk, illetve a ***/var/spool/mqueue*** könyvtárat ellenőrzi egyéb, esetleg még várakozó kimenő leveleket keresve.

A feladat

Mielőtt továbblépnénk, szeretném egyértelműsíteni, mit is akarunk felépíteni. Példának az SMTP-átjárót választottam, mivel erre a feladatra egyrészt gyakran használják a Sendmailt, másrészt ennél a szerepnél igen sokat számít a biztonságosság (a legtöbb szervezetnél a nyilvánosság számára is elérhető levélkiszolgálók sokkal komolyabban fenyegetettségnek vannak kitéve, mint a belső levélkiszolgálók).
Az SMTP-átjárók esetében általában különös figyelmet kell

fordítani a jogosultsági szintekre, a fájljogosultságokra, és általában csak annyi szolgáltatást szabad engedélyezni, amennyi a levél célbajuttatásához valóban szükséges. Egy ilyen kiszolgálón a Sendmailnek lehetőleg jogosulatlan felhasználóként kell futnia; és csakis végső esetben – tehát amikor fájlokat kell írnia – szükséges chrootolni (a / egy alhalmazán), ugyanakkor úgy kell beállítanunk, hogy a leveleket csak a saját szervezetünknek továbbítsa, a levélszemétküldőkét (spammer) ne.
Red Hat 7 alatt nem túl sok trükk kívánatos a Sendmail SMTP-átjáró megerősítéséhez, illetve alig valamivel több lépés szükséges a SuSE vagy más terjesztések esetében.

A Sendmail beszerzése és telepítése

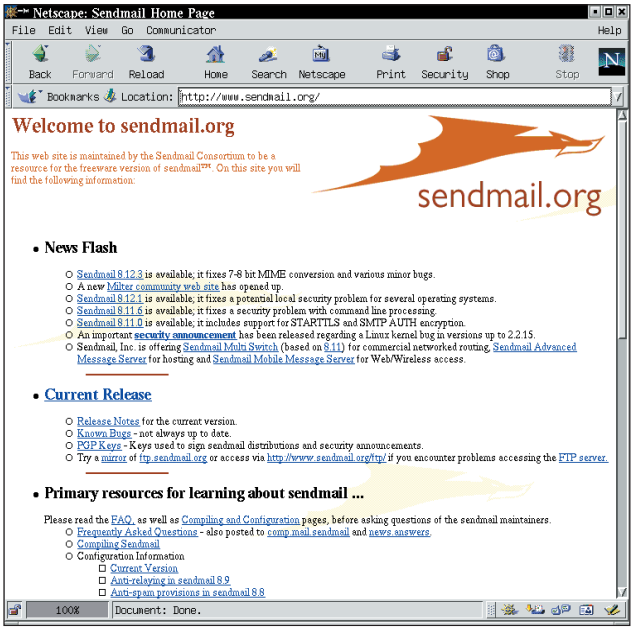
Teljes biztonsággal állíthatom, hogy az általunk kiválasztott Linux-terjesztés egy vagy több Sendmail-csomagot is tartalmaz. Természetesen az, hogy tényleg fel van-e telepítve a rendszerre, és hogy az általunk használni kívánt megfelelő változatról van-e szó, már más kérdés.
Ha rpm-alapú terjesztést használunk (Red Hat, Mandrake, SuSE stb.), a következő parancs kiadásával nézhetjük meg, hogy valamilyen változatú Sendmail fel van-e már telepítve:
`rpm -qv sendmail`
(a Debian-felhasználóknak a következőt kell beírniuk: `dpkg -s sendmail` – a fordító). A Red Hat és leszármazottai a Sendmailt három csomagra bontják: `sendmail`, `sendmail-cf` és `sendmail-doc`. A SuSE ellenben egyetlen `sendmail` nevű csomagot használ.
Tehát melyik változatot is futtatjuk? Ez idő tájt, amikor e sorokat írom, a legújabb Sendmail-változat a 8.12.2. A Red Hat 7 és a SuSE 7 viszont még mindig a Sendmail 8.11 különböző változatait támogatja. Amennyire tudom, semmi gondunk nem lesz, ha a terjesztésünk által támogatott Sendmail-változatnál maradunk, feltéve, hogy az legalább 8.11.0 vagy magasabb változatszámú. A 8.10-es és 8.11-es változatokban nem volt nagyobb biztonsági hiba; a 8.11, tulajdonképpen „bővített" kiadás volt: nem biztonsági lyukak foltozását tartalmazta, hanem azért adták ki, mert a Sendmail-csapat ekkor adta az SMTP-hez a TLS-titkosítást és az SMTP AUTH-továbbfejlesztéseket.
Természetesen amennyiben akad némi időnk és kedvünk, soha nem árt, ha a legfrissebb üzembiztos változatot fordítjuk és telepítjük. A józan ész kedvéért írásunk további részeiben feltételezem, hogy a Sendmail 8.10.0 vagy újabb változatát használjuk (az ettől eltérőt külön megemlítem).

Megjegyzés Debian-felhasználók számára

A Debian GNU/Linux v2.2 (Potato) még mindig a Sendmail v.8.9.3-at használja. Annak ellenére, hogy megbízható és viszonylag biztonságos kiadás, mostanra már két fő változattal is lemaradt (már amennyiben valaki, mint például én is, a második számot tekinti fő változatnak, hiszen az első szám már egy fél évtizede a nyolcas). Továbbá a 8.9.3 nem támogatja a TLS- és az SMTP AUTH-lehetőségeket (hamarosan, várhatóan május elején megérkezik a Woody Debian GNU/Linux v3.0 – a fordító).
Amennyiben TLS-t vagy SMTP AUTH-ot szeretnénk használni, vagy egyszerűen csak nem kívánunk régi változatot futtatni, még mindig eltávolíthatjuk a csomagot, letölthetjük a legfrissebb forráskódcsomagot a ➲ http://www.sendmail.org-ról, majd a Sendmailt forráskódból lefordíthatjuk és telepíthetjük. A forráskódcsomag jól leírt és Linux alatt könnyedén lefordul, feltételezve természetesen, hogy működőképes `gcc`-telepítéssel rendelkezünk.
Miután a Sendmailt akár bináris csomagként a terjesztésből, akár forráskódból fordítottuk és telepítettük, akad még pár feladat, amit nem árt elvégezni, mielőtt a Sendmail végrehajtható állományát démonként kezdenénk futtatni.

A SuSE Sendmail előkészítése

Ha SuSE-t használunk – amennyiben eddig még nem tettük volna meg –, váltsunk rendszergazdai jogosultságra. Nyissuk meg a */etc/rc.config* fájlt a kedvenc szövegszerkesztőnkkel, és az SMTP-változót állítsuk „yes"-re. Ez feltétlenül szükséges, ha azt szeretnénk, hogy a Sendmail */etc/init.d* könyvtárban található indítóparancsfájlja a rendszer indulásakor lefusson.
Továbbá a */etc/rc.config.d/sendmail.rc.config* fájlt át kell szerkesztenünk, hogy a `SENDMAIL_TYPE` változó „no"-ra legyen állítva. Ezáltal tulajdonképpen azt gátoltuk meg, hogy a *SuSEconfig* felhasználja a */etc/rc.config.d/sendmail.rc.config*-ot,



amely egyéb esetekben önműködően egy egyszerű Sendmail-beállítást hozna létre. Mi azonban most egy SMTP-átjárót, illetve továbbítórendszert szeretnénk beüzemelni, ami igencsak túlmutat a *sendmail.rc.config* képességein. Ha gépünk csak egyszerű SMTP-kiszolgálóként fog működni a saját helyi felhasználóihoz, valószínűleg ezt az egy fájlt elég átszerkesztenünk (ehhez előbb a `SENDMAIL_TYPE` változót „yes"-re kell állítanunk); ha ezt választanánk, a *sendmail.rc.config* teljes leírását megtaláljuk a */etc/mail/README* fájlban.
Miután az *rc.config* és a *sendmail.rc.config* állományokat átszerkesztettük, futtassuk le a *SuSEconfig*-ot. Ezzel érvényt szerezhetünk az *rc.config* és *sendmail.rc.config* fájlokban imént elvégzett változtatásoknak. A démon elindításához begépelhetjük a `/init.d/sendmail start` parancsot, azonban én azt javaslom, előbb inkább várjuk meg, amíg a Sendmailt teljesen beállítjuk.

A Red Hat Sendmail előkészítése

A Red Hat-felhasználóknak a Sendmail beállítása előtt mindössze egyetlen lépést kell elvégezniük: át kell szerkeszteni a */etc/sysconfig/sendmail* fájlt, hogy a `DAEMON` változó értéke „yes" legyen. Ez mondja meg ugyanis a */etc/init.d/sendmail* indítóparancsfájlnak, hogy rendszerindításkor a Sendmailt démonként kell futtatnia.

A Sendmail beállítása

Végre-valahára nekikezdhetünk a Sendmail beállításának tartományunk SMTP-átjárójaként. A következőkben leírtak a 8.9-es felett a Sendmail bármely változatára érvényesek (semmilyen körülmények közt ne futtassunk a 8.8-as változatot!).
A Sendmail beállításfájl (***sendmail.cf*** és a hozzá tartozó fájlok) egyszerűsített változatának előállításához a következő lépéseket kell megtennünk:

1. A ***sendmail.mc***-ben engedélyezzük a szükséges képességeket.
2. Ha szükséges, a ***sendmail.mc***-ben állítsuk be a tartománynév-álcázást (domain-name masquerading).
3. Futtassuk le az m4-et, amely a ***sendmail.mc***-ből létrehozza a ***generate.cf*** fájlt.
4. A ***mailertable*** szerkesztésével állítsuk be a kézbesítési (delivery) szabályokat.
5. Az ***access*** szerkesztésével állítsuk be a továbbítási (relay) szabályokat.
6. Az ***aliases***-ben állítsuk be a helyi felhasználói álneveket.
7. A ***mailertable***, ***access*** és ***aliases*** állományokat alakítsuk adatbázissá.
8. Az összes helyi gépnevet adjuk meg a ***local-host-names***-ben.
9. Indítsuk (újra) a Sendmailt.

A „sendmail.mc" érdekesebb beállításai

Az első és valószínűleg legidőigényesebb feladat az SMTP-átjáró felállítása során a ***/etc/sendmail.cf*** előállítása. Ezt legkönnyebben a ***/etc/mail/sendmail.mc*** átírásával tehetjük meg (SuSE-rendszereken e fájl neve ***/etc/mail/linux.mc*** – más terjesztések alatt eltérő is lehet).
A használt Linux-terjesztéstől függően a ***sendmail.mc*** beállítási adatait a ***/usr/share/doc/sendmail/README.cf*** (Red Hat és társai) a ***/usr/share/sendmail/README*** (SuSE) vagy valamilyen más állományban találjuk. Nincs elég hely arra, hogy e fájl számos beállítási lehetőségét részletekbe menően ismertessem. Megvizsgálom viszont azokat, amelyek biztonság tekintetében hasznosak lehetnek vagy beállításainkat modularizálják.
A Sendmail beállítását magán a ***sendmail.cf***-en kívül, külső fájlokból beolvasott adatokkal is megoldhatjuk. Ez két okból is célszerű: egyrészt a ***sendmail.cf*** közvetlen szerkesztése elég kényelmetlen, a ***sendmail.mc***-ből történő újralétrehozása pedig nem mindig kívánatos. Másrészt amennyiben SMTP-átjárónkon több különböző jogosultsággal rendelkező rendszergazda is lesz, jól jöhet, ha a ***sendmail.cf*** fájlt zárva tartjuk, ugyanakkor a többi rendszergazdának megengedjük az álnevek és a levéltovábbítási szabályok szerkesztését (vagyis a ***/etc/mail/access*** és ***/etc/mail/mailertable*** fájlok módosítását).
A leghasznosabb külső beállításfájlok, amelyeket érdemes engedélyezi:

- a ***mailertable***, amely a helyi kézbesítési szabályokat írja elő;
- a ***virtusertable***, ez virtuális tartománymegfeleltetéseket ír le felhasználónkénti és tartományonkénti bontásban;
- az ***access***, ami meghatározza, hogy mely gépek használhatják a kiszolgálót SMTP-továbbítóként.

A fenti fájlokat engedélyező ***sendmail.mc***-utasítások a következők:

```
FEATURE('mailertable',
➥'hash -o /etc/mail/mailertable.db·)dnl
FEATURE('virtusertable',
➥'hash -o /etc/mail/virtusertable.db·)dnl
FEATURE('access_db','hash -o
➥/etc/mail/access.db·)dnl
```

(A ***mailertable*** és `access_db` képességek Red Hat alatt alapértelmezetten érvényesek, viszont a ***virtusertable*** részt kézzel kell hozzáadni.)
Minden sor arra utasítja a Sendmailt, hogy az adott fájlra hivatkozást készítsen (bár az elérési adatbázist ***access***-nek neveztük `access_db` helyett), és annak hash-adatbázisát, illetve elérési útját használja. Hamarosan megismerhetjük, hogy hogyan használjuk fel ezeket a fájlokat, előbb viszont végre kell még néhány dolgot hajtanunk a ***sendmail.mc***-ben.
Ha felhasználóink elektronikus címei tartományunk szerintiek és a gép szerint, ahová bejelentkeztek, nem változnak – azaz ***mick@polkatistas.org*** formátumúak ***mick@myron.polkatistas.org*** formátum helyett, akkor kimenő leveleik ***From:*** mezőjét valószínűleg ennek megfelelően érdemes megváltoztatni (az ilyen általános címeken fogadott levelek felhasználói álneveket igényelnek – lásd később).
A következő sorok olyan ***sendmail.mc***-beállításokat mutatnak be, melyek arra utasítják példa-SMTP-átjárónkat, hogy a ***polkatistas.org*** felhasználóitól érkező levelek ***From:*** mezőjét az előbbieknek megfelelően írja át. A lenti példa összes sorát be kell szúrni, vagy a megjegyzésből ki kell szedni:

```
MASQUERADE_AS('polkatistas.org·)dnl
MASQUERADE_DOMAIN('.polkatistas.org·)dnl
EXPOSED_USER('root·)dnl
FEATURE('masquerade_entire_domain·)dnl
FEATURE('masquerade_envelope·)dnl
```

- A `MASQUERADE_AS` direktíva azt a teljes értékű tartománynevet adja meg, amit a megfelelő ***From:*** címekben szeretnénk látni.
- A `MASQUERADE_DOMAIN` direktíva adja meg azt a gépet, amire a `MASQUERADE_AS` vonatkozik. A ***polkatistas.org*** előtt álló "." azt jelenti, hogy az összes ebbe a tartományba tartozó gépnevet álcázni kell.
- Az `EXPOSED_USER` azt a felhasználónevet adja meg, amelynek ***From:*** címét nem szabad álcázni. A rendszergazda gyakori vendég e mezőben, mivel a tőle érkező levél sokszor figyelmeztetéseket és riasztásokat tartalmaz – ha ilyet kapunk, általában azt is tudni szeretnénk, hogy melyik géptől érkezett.
- A `masquerade_entire_domain` képesség azt jelenti, hogy a `MASQUERADE_DOMAIN` teljes tartományként és nem gépnévként értelmezendő; a `masquerade_envelope` eredményeképpen az álcázás nemcsak az SMTP-fejlécre vonatkozik, hanem a borítékra is.

Négy másik direktívát – egy logisztikait és három biztonsági jellegűt – találunk az *1. listában*.

- Az `always_add_domain` képesség Red Hat és SuSE alatt alapértelmezetten be van kapcsolva; az `use_cw_file` és `smrsh` Red Hat alatt érvényes, a SuSE alatt viszont nem; a `confSAFE_FILE_ENV` beállítást pedig minden esetben nekünk kell megadnunk.

1. lista Néhány további sendmail.mc-képesség

```
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE('smrsh·,'/usr/sbin/smrsh·)dnl
define('confSAFE_FILE_ENV·,
'/var/mailjail·)dnl
```

2. lista A /var/mailjail tartalma

```
/var/mailjail:
total 1
drwx------   5 mail    mail    1024 Jan 22 17:09 var

/var/mailjail/var:
total 3
drwx------   4 mail    mail    1024 Jan 22 17:07 spool

/var/mailjail/var/spool:
total 2
drwx------   2 mail    mail    1024 Jan 22 17:06 mail
drwx------   2 mail    mail    1024 Jan 22 17:06 mqueue

/var/mailjail/var/spool/mail:
total 98
-rwx------   1 mail    mail    48528 Jan 22 17:06 bobo
-rwx------   1 mail    mail    47627 Jan 22 17:06 root

/var/mailjail/var/spool/mqueue:
total 0
```

- Az `always_add_domain` képesség a gép tartománynevét egyszerűen minden olyan levélhez kötelezően hozzáadja, amely magát tartománynév nélkül azonosító gépről érkezik. Például ha az SMTP-átjáró levelet kap ***bobo*** felhasználótól egy olyan gépről, amely magát csak ***whoopeejohn*** néven azonosította, a Sendmail a ***From:*** mezőt az eredeti ***bobo@whoopeejohn*** helyett ***bobo@whoopeejohn.polkatistas.org*** formátumúra írja át (de az álcázási direktívák itt is érvényesek).
- A `use_cw_file` képesség használata arra utasítja a Sendmailt, hogy a Sendmail által helyinek értékelt gépek listáját a */etc/mail/local-host-names* fájlból vegye. A */etc/mail/local-host-names* egyszerű szöveges állomány, amely soronként egyetlen gépnevet tartalmaz. Tegyük fel, hogy példa-SMTP-átjárónk nemcsak a ***polkatistas.org*** tartományból kap levelet, hanem a ***tubascoundrels.net***-ről is. Ha az átjáró neve `mail`, a ***local-host-names*** fájlja a következőképpen fog kinézni:

```
localhost.localdomain
mail.polkatistas.org
mail.tubascoundrels.net
```

Az 1. listában megadott harmadik képesség az `smrsh`, azaz a Sendmail korlátozott héjprogram. Ez nagyon fontos biztonsági lehetőség, amely képes korlátozni felhasználók ***.forward*** fájljából végrehajtható parancsokat.
Az 1. lista negyedik sora azt mondja meg a Sendmailnek, hogy a ***sendmail.cf*** `SafeFileEnvironment` változóját állítsa be – ahogy már biztosan ki is találták – a saját könyvtár (/) egy olyan alkönyvtárára, ahová a Sendmail chrootolni fog (mármint amelyik így lett beállítva). Jelenleg ez csak akkor következik be, amikor a Sendmail fájlokat ír. Ha meggondoljuk, ennek az ötvenszázaléknyi chrootolásnak is van értelme: pontosan a fájlírások azok, amelyek miatt a leginkább aggódnunk kell, és egy ilyen chrootkörnyezet kialakítása sokkal egyszerűbb, mintha a sok helyen használt `chroot jail`-t választanánk (abban az esetben ugyanis a chrootolt program által igényelt minden fájlszerkezetből, fájlból, végrehajtható állományból és eszközből másolatot kell tartani).
A 2. *lista* az én példa-`SafeFileEnvironment`-em */var/mailjail* könyvtárának teljes listáját (`ls -lR`) mutatja be. A */var/mailjail/var/spool/mqueue/bobo* és */.../root* fájlokat a Sendmail hozta létre. Ez előtt az egész `chroot jail`-környezetet mindössze négy paranccsal hoztam létre:

```
mkdir -p /var/mailjail/var/spool/mail/var/
➥mailjail/var/spool/mqueue
cd /var/mailjail
chown -R mail:mail *
chmod -R 700 *
```

Ha valakit a kéretlen kereskedelmi levelek témaköre érdekelne, számára is akad néhány jó hírem: a Sendmail alapértelmezés

Mick széljegyzete

Nem kell túl nagy jelentőséget tulajdonítani neki, de jómagam Postfix-rajongó vagyok. Postfixet és nem Sendmailt futtatok a saját tartományom SMTP-átjárójaként (igaz, saját hálózatomon helyi levéltovábbításhoz Sendmailt használok). Így aztán az e cikkben leírtakból, ideértve egyáltalán a létezését is, senki ne következtessen arra, hogy úgy gondolom, a Sendmail a legjobb választás, ha valakinek MTA-ra van szüksége – ezt mindenkinek magának kell eldöntenie.
Megkockáztatva, hogy kétértelműnek tűnök, azt kell mondanom, az elmúlt években meglehetősen sok időt fordítottam a Sendmailre és sokat segítettem másokat is a Sendmail használatában. Úgy vélem, hogy sokkal jobb, mint amennyire némelyek becsülik. Tapasztalataim szerint egyáltalán nem az a döcögő, cammogó, törékeny szörny, mint amilyennek néhány kritikusa beállítja. Valójában a Sendmailt igen megbízhatónak és hatékonynak tartom, mégha kicsit ijesztő is a bonyolultsága. Továbbá a legutóbbi, 1997-es CERT-tanácsadó óta (number CA-1997-05), amely a Sendmail biztonsági hibáját is tartalmazta, egyszerűen nem látom bizonyítottnak, hogy a Sendmail örökletesen ne lenne biztonságossá tehető. (A Sendmail átvizsgálása nyilván nem lett kevésbé szigorú az elmúlt öt évben, mint korábban.) Ezért azt hiszem, hogy bár más MTA-k (köztük a Postfix, Qmail és az Exim) egyértelmű előnyökkel rendelkeznek a Sendmaillel szemben a teljesítmény és a biztonság terén, egyben úgy vélem, a Sendmail elég jó minőségű ahhoz, hogy megkapja a Paranoid Pingvin minősítést (emellett az MTA-k „királyi családjából” származik: a beltenyészet miatt ugyan aggódhatok egy kicsit, de ettől még tisztelettel tartozom neki).

szerint nem engedélyezi az SMTP-továbbítást (relaying), ezt a levélszemétküldők által általánosan használt módszert. A szolgáltatást ugyan a ***sendmail.mc***-ben ki lehet kapcsolni, de tőlem ugyan meg nem tudod, hogyan. Inkább hagyjuk úgy. Továbbá a Sendmailt oly módon is beállíthatjuk, hogy minden ismert levélszemétforrásból érkező levelet utasítson el, amely a Realtime Blackhole List (RBL) feketelistáján szerepel. A következő sort kell csupán beszúrunk, illetve a megjegyzésből kiszednünk:

```
FEATURE('dnsbl')
```

Ahhoz azonban, hogy ez tényleg működjön, előbb fel kell iratkozni az RBL-re – honlapjuk címét megadtuk a ***Kapcsolódó címek*** között. A weblapon feliratkozási és felhasználási tanácsokat olvashatunk, illetve néhány fontos nyilatkozatot találunk (nem árt tudni, hogy míg a RBL-feliratkozás az egyéni, illetve hobbihelyek számára (Individual/Hobby Sites) ingyenes, ennek a szolgáltatásnak díjszabása is van). Az RBL felhasználásával a jogosult felhasználók leveleit éppúgy megállíthatjuk, mint a levélszemetelőkét, ezért kezeljük óvatosan.
Ha Red Hat 7.1-es vagy 7.2-es változatot használunk, létezik még egy ***sendmail.mc*** lehetőség, amit érdemes megnézni – ezúttal egy olyan, amit megjegyzésbe kell tenni. Amennyiben a ***/etc/mail/sendmail.mc*** fájlunk egy ilyen sort tartalmaz:

```
DAEMON_OPTIONS('Port=smtp,Addr=127.0.0.1,
➥Name=MTA·)
```

Tegyük megjegyzésbe, úgy, hogy a `dnl` karaktereket a sor elejére írjuk. Ameddig működik, ez a sor megakadályozza, hogy a Sendmail saját hurokeszközének (loopback) csatolófelületén kívül bármilyen más hálózatról érkező kapcsolatot fogadjon. Mondanunk sem kell, hogy egy SMTP-átjáró esetében ez nemkívánatos (bár kétségtelenül növeli a biztonságot). Ezek voltak a számunkra legfontosabb ***sendmail.mc***-beállítások. Léteznek természetesen más, biztonsági szempontból fontos beállítási lehetőségek is, különös tekintettel a nem átjárószabályokra (helyi kézbesítés stb.). További tájékoztatásért olvassuk el a ***README.cf*** vagy a README fájlokat, amelyeket e rész elején említettem.
Macro-beállítófájlunk ***sendmail.cf***-fé alakításához a következő parancsot használjuk:

```
m4 /etc/mail/sendmail.mc > /etc/sendmail.cf
```

Amennyiben macro-beállításfájlunk neve nem ***sendmail.mc***, helyettesítsük be a ***linux.mc*** vagy az éppen használt macro-beállítófájlnévvel. A Sendmail azt várja, hogy beállításfájlját ***sendmail.cf***-nek nevezzék; ráadásul mindig a ***/etc*** könyvtárban keresi, így aztán a parancs e része terjesztéstől, sőt, Sendmail-változattól függetlenül mindig így néz ki.

A kézbesítési szabályok beállítása

A nehezén túl vagyunk, most már csak azt kell a Sendmailnek megmondanunk, hogy mit csináljon a beérkezett levelekkel, milyen helyi gépnevek fogadhatók el, és milyen felhasználók, hálózatok és tartományok használhatják az SMTP-átjárót nem helyi célú levelek küldésére.
A ***mailertable***-t a kézbesítési szabályok megadására használjuk. Egyszerű az írásmódja, és terjesztéstől függően a ***/usr/share/doc/sendmail/README.cf*** vagy a ***/usr/share/sendmail/README*** fájlban találhatjuk meg. Dióhéjban: minden sor két részből áll, a célazonosítóból és a műveletből. A célazonosítónak kell megegyeznie a cél címével vagy annak egy részével; a művelet azt mondja meg, hogy a Sendmailnek mit kell tennie azokkal az üzenetekkel, amelyek célja megegyezett az azonosítóval. Ha az azonosító „.”-tal kezdődik, akkor minden levélforráscím, amely a pont után megadott részre végződik, találatnak számít. Ha nem, a „@” jelet követő összes karakternek meg kell egyeznie. A ***bobo@weird-al.polkatistas.org*** nem fog egyezni a ***polkatistas.org*** címmel, de egyezni fog a ***.polkatistas.org***-gal. A művelet ***ügynök:cselekmény*** alakú, ahol az ügynök lehet a levelező (a mailer, amit a ***sendmail.mc/linux.mc MAILER()*** pontjában állíthatunk be), vagy a beépített ***local*** ügynök, illetve ***error***. A ***local*** ügynök természetesen azt feltételezi, hogy a levelet valamilyen helyi felhasználónak küldjük, amit a kettőspont után adunk meg (ha a kettőspontot semmi sem követi, akkor magában az üzenetben megadott felhasználót fogja használni). Alább egy ***mailertable*** látható két különböző cselekménnyel:

```
polkatistas.org
➥smtp:internalmail.polkatistas.org
mail.polkatistas.org  local:postmaster
```

A kézbesítési szabályokon felül a Sendmailnek tudnia kell, hogy mely elektronikus levélcélokat kell a helyi (az SMTP-átjáró) gépnév rokonának tekintenie. Ezeket a ***/etc/mail/local-host-names*** fájlban adhatjuk meg, soronként egyet:

```
mail.polkatistas.org
weird-al.polkatistas.org
1.23.234.2
```

Végül azoknak a listáját kell megadnunk, akiknek a továbbítást a ***/etc/mail/access*** átszerkesztésével engedélyezzük. Az írásmód nagyon egyszerű: minden sor egy forrásnevet vagy -címet tartalmaz, ami után egy cselekmény áll (a részletekért ismét a ***README.cf***-et vagy az ennek megfelelő állományt nézzük át rendszerünkön). A cselekmény lehet `RELAY` (továbbít), `REJECT` (elutasít), `DISCARD` (elvet), `OK` vagy `ERROR` (hiba). A gyakorlatban ezek között a `RELAY` a leghasznosabb cselekmény, mivel alapesetben minden más továbbítás elutasításra kerül. A `REJECT` és a `DISCARD` cselekményeknek csak akkor van haszna, ha egy adott `RELAY`-szabály alól akarunk kivételeket megadni. Íme, egy egyszerű ***access*** fájl:

```
localhost.localdomain      RELAY
localhost     RELAY
127.0.0.1     RELAY
192.168       RELAY
```

Ugye, észrevettük a valódi gépnevek hiányát a fenti példában? Ebben a példában az SMTP-átjáró mindössze kimenő továbbítást enged; a bemenő levelek kizárólag helyi címekre jöhetnek, és a kimenő továbbításoknak is olyan gépekről kell érkezniük, amelyek IP-címe a `192.168` számokkal kezdődik (ami az Interneten nyilvánvalóan nem megadható címtartomány). Kedvelem ezt a módszert (az IP-cím használatot), hiszen így az IP-címálcázást tűzfalszabályaimmal megakadályozhatom, igaz, nem tudom meggátolni a hamis ***From:*** levélcímek átadását (természetesen a te igényeid mások is lehetnek):

```
access
local-host-names
mailertable
```

Kifinomultabb Sendmail biztonsági eljárások

Az SMTP AUTH (a Sendmail 8.10-es változatától fölfelé) már azonosítási lehetőséget hozott az SMTP-műveletek világába, azaz képes megállapítani, hogy engedélyezheti-e a továbbítást. Ez különösen akkor hasznos, amikor a rendszerek vagy a felhasználók nem futtatnak saját MTA-t, mégis szeretnének leveleket küldeni, azaz a kifelé menő leveleket egy központi átjárón át muszáj küldeni.

Ha olyan SMTP-kiszolgálót futtatunk, amely más tartományokból érkező leveleket is továbbít, nem árt, ha megismerkedünk ezzel a képességgel, mivel igen fontos védelem a kéretlen kereskedelmi levelek ellen, amelyek elkövetői jelentős részben az SMTP-továbbításokban bíznak.

Már csak egyetlen fájl maradt, amin finomítani lehetne: ez az ***aliases***. Ez a fájl tartalmazza a felhasználók elektronikus leveleinek álnévlistáját. Általában egy SMTP-átjárónak nincs szüksége túlságosan részletes ***alias***-adatbázisra; egész tartományok (vagy virtuális tartományok) levélcímeihez jobb, ha inkább a felhasználói adatbázist használjuk (ezt azonban hely hiányában sajnos nem áll módomban leírni). Szerencsére eléggé magától értetődő, így nyugodtan szerkesszük át, ha szükséges.

A tárgyalt négy fájlból három: a ***mailertable***, ***access*** és ***aliases*** állományok közvetlenül nem használhatók fel a Sendmailhez, először adatbázissá kell alakítanunk őket. A */etc/mail* könyvtár egy hasznos kis `Makefile`t tartalmaz e célra. Használatához egyszerűen csak váltsunk a */etc/mail* könyvtárba, és gépeljük be a következő parancsot:

```
Make access.db mailertable.db
```

A fenti parancs az ***aliases*** fájlhoz nem lesz jó, mivel ennek saját eszköze van: a `newaliases`. Futtassuk le minden kapcsoló nélkül a `newaliases`-t, és megváltoztatott */etc/aliases* fájlunk önműködően */etc/aliases.db* fájllá alakul.

Egyelőre ennyi. Sok mindent nem sikerült elmondanom: külön kiemelném közülük az `smrsh` héjprogramot (amit főként a helyi levélkézbesítéshez lehet felhasználni és nem az átjárókhoz). Remélem, hogy azért sikerült néhány hasznos tippet adnom és útmutatást szolgáltatnom néhány teljesebb információforráshoz. Sok szerencsét!

Linux Journal március, 95. szám

Mick Bauer (mick@visi.com) hálózati biztonsággal foglalkozó szaktanácsadó. 1995 óta a Linux elkötelezett híve, 1997 óta pedig OpenBSD prófétaként tevékenykedik. Mick minden kérdést és megjegyzést szívesen fogad.

Kapcsolódó címek

- http://www.sendmail.net/000705securitygeneral.shtml
- http://www.sendmail.net/000710securitytaxonomy.shtml
- http:// www.itworld.com/Net/3314/swol-0699-security
- http://www.sendmail.net/810usingantispam.shtml
- http://www.sendmail.net/usingsmtpauth.shtml