Utazás a Postfix körül (1. rész)

Kezdetben vala a Sendmail. Majd érkezett a Qmail. És lám, itt van a Postfix...

A két korábbi rendszer már régen ismert, kiforrott, a Postfixre azonban most kezdenek felfigyelni. A cikksorozat első részében bemutatjuk az írót és a program felépítését.

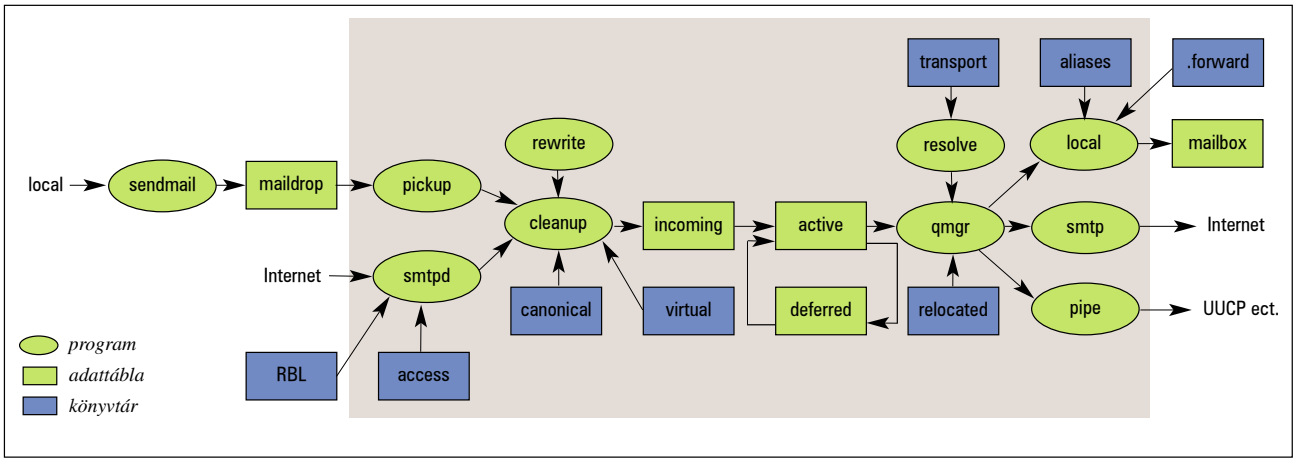
Wietse Venema: a Programozó

Mikor Isten a programozót teremtette, valószínűleg Venemára gondolt. A kódok tiszták, a programok felépítése átgondolt és logikus. Tekintsük át, milyen programok fűződnek a nevéhez. A TCT, azaz a The Coroner's Toolkit. E programon *Dan Farmer*-rel együtt dolgoztak. A programmal betörés után kísérelhetjük meg felderíteni annak menetét. Használata nem egyszerű, de megéri megtanulni. A következő a Satan rendszerátvizsgáló eszköz, mely nemcsak a hibát mutatja meg nekünk, hanem elhárítására megoldást is javasol. Ma már kicsit idejétmúlt, a Nessus nagyobb támogatottságot élvez ➲ http://www.nessus.org. Azután a TCP Wrapper, mely minden Linux-változatban megtalálható program, segítségével szabályozhatjuk a hálózati szolgáltatások elérését csomagszűrő vagy alkalmazásszintű tűzfal nélkül. És a rengeteg kis – főleg Solaris-rendszerre írt – programon kívül a `postfix`.
A felsorolt programok természetéből látható, hogy az író nem sorolható a kezdők közé, és rendkívül foglalkoztatja a biztonság témaköre.

A Postfix gondolatvilága

Amikor Venema elhatározta, hogy létrehozza a szerinte lehető legbiztonságosabb és leggyorsabb levelezőkiszolgálót, végignézte az addig megírt kiszolgálóprogramokat, majd összesítette azok előnyeit és hátrányait. A következő célt tűzte ki: az új kiszolgálónak legyen olyan sok szolgáltatása, mint a Sendmailnek, de túlbonyolított beállítások nélkül. Továbbá bizonyuljon legalább olyan biztonságosnak, mint a Qmail. És végül legyen a leggyorsabb. Elmondhatjuk, hogy egyedülálló mű született. Felépítését tekintve a Postfix moduláris, minden szolgáltatást külön programrész hajt végre, a jogosultságok pedig a lehető legalacsonyabb szintre kerültek. A rendszert két fő beállítófájlon keresztül tudjuk módosítani, felépítésüket nagyon könnyű megérteni. A programváltozat a következő állapotokat ismeri: a pillanatfelvételt (snapshot) és a teljes kiadást (release). A pillanatfelvételek olyan programrészleteket és programokat tartalmaznak, melyeket a szerző még nem tart véglegesnek, míg a kiadás már a hivatalos változatot, ebből hiányoznak a kísérleti jellegű vagy a nem százszázalékosan kidolgozott programok. A pillanatfelvétel-változatokról egy rövid megjegyzés: Venema a saját levelezőkiszolgálójára is ezeket telepíti. A programok ezen állapotbeli üzembiztonságát sok másik program megirigyelhetné.
A rendszert úgy tervezték, hogy a távoli hálózatokkal a lehető legkevesebb modul érintkezzen, azok pedig a legkisebb jogosultsági szinttel rendelkezzenek. Aki megpróbál egy Postfix-alapú levelezőrendszert feltörni, annak több szinten kell keresztülverekednie magát, mire egy olyan programhoz jut, ami rendszergazdai jogosultságokkal fut. A programokat és a rendszer felépítését részletesen az *1. ábra* mutatja be, ennek segítségével kövessük egy levél fogadását.

- `sendmail`
  Ez a kis program gyakorlatilag egy burkoló (wrapper), a sendmailt helyettesíti, hogy az ahhoz megírt programok helyi kéréseit átfordítsa a `postfix` által is értelmezhető hívásokká, és letegye azt az előírt könyvtárba. Ha a könyvtár mindenki által írható (world writeable), akkor egyedül is meg tudja ezt tenni, ha azonban csak egy adott csoport számára elérhető – az alapértelmezett csoport a `maildrop` – akkor a `postdrop` programot használja erre a célra, ami szintén a `postfix` rendszer része.
- `pickup`
  Begyűjti a leveleket a `maildrop` könyvtárból és átadja őket a `cleanup` programnak (magyarázatát lásd később).
- `smtpd`
  Fogadja a belső hálózaton vagy az Interneten keresztül érkező leveleket és különböző ellenőrzések után átadja azt a `cleanup` programnak. Itt tudjuk előírni, hogy vizsgálja meg: a küldő fél nyitott továbbító-e (`open-relay` – mindenki tud rajta keresztül levelet küldeni akárhova, a levélszemetelők kedvenc célpontja), vagy sem.



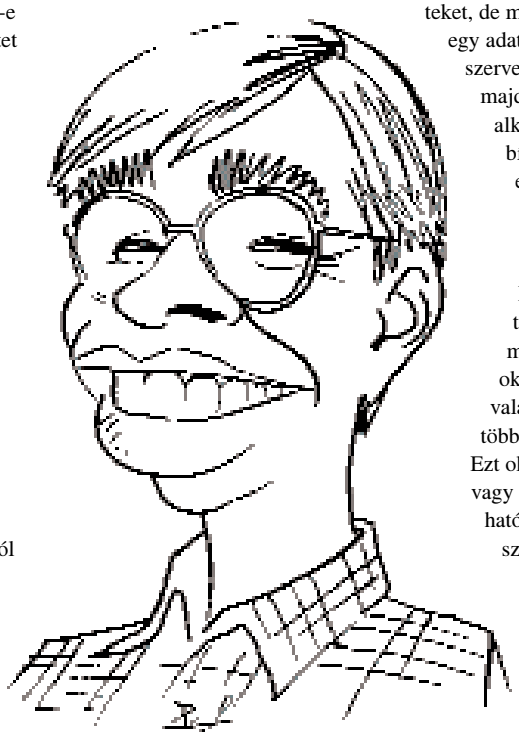
Bejövő levelek ellenőrzése

Ellenőrzi, hogy nem tiltottuk-e ki a küldőt más módon, például olyan táblázatokat használva, amelyek a küldőre vonatkoznak, akár hálózati cím, egész tartománynév vagy küldő személy szerint. De itt akár engedélyezhetjük azt is, hogy elfogadjuk a levelet egy olyan levelezőkiszolgálótól, amelyik benne van a nyitott továbbítókat nyilvántartó adatbázisban.

- `cleanup`
  Ellenőrzi a levél fejlécét, hogy megfelel-e az előírásoknak: a címzett megtalálható-e a rendszeren, kell-e valamilyen műveletet végrehajtani a levél fejlécén – például a címzett nevének átírását –, esetleg meg kell-e hívnia a `trivial-rewrite` programot. Ha végzett, akkor „lerakja” – fájlba menti – az üzenetet az `incoming` sorba (queue). Ebbe a sorba kerülnek tehát azok a levelek, melyeket a kiszolgáló elfogadott.
- `trivial-rewrite`
  Szükség esetén átírja a levél fejlécét, és visszaadja azt a `cleanup` programnak. A programot a `qmgr`, azaz a „sorkezelő” is meg tudja hívni.
- `qmgr`
  Az `incoming` sorból áthelyezi az üzeneteket az `activ` sorba, valamint felügyeli a levél sorsát. Az `activ` sorból kerülnek elküldésre a levelek helyileg vagy távolra. A sor mérete korlátozott, nehogy túlterheljék a kiszolgálót, olyan módon például, hogy túl sok levelet próbálnak vele elküldetni, és így nem marad elég memória az egyéb tevékenységekre, vagy éppen csak a levelek fogadására. A `qmgr` arra is felügyel, hogy azok a levelek, melyeket először próbál elküldeni – akár helyileg, akár távolra –, elsőbbséget élvezzenek azokkal szemben, melyeket nem sikerült először elküldeni, és ezért későbbi továbbításra várnak. A levelek innen ahhoz a programhoz kerülnek, amit a továbbítás megkövetel. Ezek a `local`, `smtp` és a `pipe` programok. A Postfix pillanatfelvételeiben elérhető még a `virtual` is, erre azonban a cikksorozat későbbi részében térek ki.
- `local`
  A helyi felhasználók számára fogadja a leveleket, ellenőrzi a `.forward` fájlok meglétét, valamint az `alias` adatbázist. A `.forward` fájlokat használjuk arra, ha egy programnak akarjuk elküldeni a levelet, vagy másik címre próbáljuk azt továbbítani. A rendszer az alábbi üzenettároló formátumokat ismeri: mbox és Maildir. A kettő közötti különbségről egy következő alkalommal írok.

A `.forward` fájlon keresztül a vezérlést más programoknak tudjuk adni, például a `procmail`-nek, bár a `local` önmagában is képes erre. Az `alias` adatbázist arra használjuk, amire a neve is utal: neveket leképezünk más nevekre. Jó példa erre a Kis.Ferenc kisf névre alakítása, amit már akármelyik unixos rendszer elfogad – gyanúm szerint a windowsos rendszerek is hasonló névleképezéssel oldják meg ezt a helyzetet. A cikksorozat következő részében ezt is részletesebben ismertetem majd.

- `smtp`
  A nem helyi felhasználók számára továbbítja az üzeneteket. Tehát ha én az ago@lsc.hu címről akarok levelet küldeni az info@linuxvilag.hu címre, akkor az én kiszolgálóm ezzel kapcsolódik a Linuxvilág kiszolgálójához. A távoli rendszerekkel csak smtp protokollon keresztül tartja a kapcsolatot (figyelem: nem az egész rendszer, csak ez a részprogram!).
- `pipe`
  Más vagy eltérő típusú rendszernek továbbítja az üzeneteket, de még a helyi rendszeren. Tehát én például egy adatbázisba továbbítom a leveleket, melynek szerver része fogadja a leveleket, letárolja azt, majd később elküldi azokat. Leggyakoribb alkalmazása, ha UUCP-n keresztül továbbítjuk a leveleket, használhatjuk azonban ezen keresztül az `lmtp` protokollt ismerő programokat is. Az `lmtp` protokollt használja például a `Cyrus-IMAP` rendszere is.



Wietse Venema, a Programozó

Ezek után a levél remélhetőleg kézbesítésre került. Ha ez mégsem történt volna meg, akkor visszapattan. Ennek kétféle oka lehet: elsőként a levél nem felelt meg valamilyen feltételnek és Postfixünk soha többé nem fog próbálkozni az elküldésével. Ezt okozhatja a levél tartalma, a küldő személye vagy más is, például a címzett már nem található meg a fogadó kiszolgálón. A második szintén nem túl szerencsés eset, ha a levelet csak átmenetileg nem tudjuk elküldeni, például a fogadó levelezőkiszolgáló időleges hibája miatt. Ekkor a levél `deferred` jelzőt kap, és a rendszer adott időközönként, illetve időtartamig megpróbálja elküldeni. Azokról a levelekről, amelyeket visszautasítottak, egy bejegyzést készít a `bounce` program a bounce nevű sorba, majd a levél eredetijét vagy annak egy részét visszaküldi a feladónak, kiegészítve azt a hiba okával. A bounce sor nem tárolja el a leveleket! Ha a levelet csak időlegesen utasította vissza a fogadó fél, akkor egy bejegyzés kerül a defer sorba, a levelet magát pedig a deferred sorban helyezi el. Ezt a helyzetet szintén a `bounce` program kezeli le.

Van egy program, amihez hasonlót eddig egyetlen más levelezőrendszernél sem láttam, a `master` démon, amely mindig ott fut a háttérben. Ez felügyeli az összes többi démont, a rendszer állapotát, és nem engedi, hogy a rendszert alkotó programok „elvaduljanak” és megállásra vagy szolgáltatásmegtagadásra kényszerítsék a rendszert. Ha túl messzire merészkednének, akkor visszafogja őket. Mivel ez vezérli az összes többit, és a Postfix modularitása megengedi, könnyű olyan kiszolgálót készíteni, amelyik csak küldeni tud.

A következő részben olyan rendszert állítunk be, melynek teljes a szolgáltatáskínálata, visszautasítja a levélszemetelők küldeményeit, és nem engedi, hogy kiszolgálónkat nyitott továbbítóként használják.

Deim Ágoston (ago@lsc.hu)
Kedveli a sört, szereti a futást és imádja Szabó Lőrinc verseit. Nem hisz vakon egyik rendszerben sem. Vonzódik a BSD-hez is. Tagja az LME-nek és a MBE-nek. Mottója: a gép nem lehet fontosabb az embernél.