A biztonság növelése FTP proxy használatával

Mick azt mutatja be, hogyan lehet biztonsági réteget helyezni a rossz szándékú felhasználók és nyilvános hozzáférésű FTP-kiszolgálónk közé.

Egy nyilvános hozzáférésű FTP-hely üzemeltetése nehéz feladatnak bizonyulhat, de egy tetszőleges FTP-kiszolgálóalkalmazás által nyújtott összes biztonsági szolgáltatás lehetőség szerinti legjobb kihasználása is kemény munkának ígérkezik. Mindennek ellenére a sebezhetőségi pontok előbb vagy utóbb nyilvánvalóvá válnak, és semmissé teszik az addig elvégzett munkát. Vagy tehetünk valami mást is? Fontos módszer, hogy az FTP-proxy alkalmazást a tűzfalunkon futtatjuk. Miközben a szabványos rendszermagba épített Netfilter-kód csak a csomagok vizsgálatára képes, az FTP-proxy a tűzfal számára lehetővé teszi, hogy az minden FTP-műveletben közvetítőként működjön. Ez fokozza a biztonságot az ideiglenes tárterületek (pufferek) túlcsordulásával és sok másfajta FTP-támadással szemben. Ezenkívül megengedi, hogy az FTP-ügyfelek számára megszabjuk, milyen parancsokat hajthatnak végre.
E hónapban azt mondom el, hogy a SuSE ingyenes – és a nem csak SuSE alatt futó – Proxy-Suite (készlet) FTP-proxykiszolgálóját hogyan kell Linux-tűzfalon üzemeltetni, átlátható, de erős védelmet nyújtva minden egyes FTP-művelet számára.

A proxy-suite beszerzése és telepítése

Ha SuSE Linux-változatot üzemeltetsz, telepítheted a proxy-suite csomagot, amely az ftp-proxy bináris példányát telepíti annak beállító állományával és héjprogramjával együtt. Amennyiben az ftp-proxyt áttetsző proxyként kívánod használni, vagy elvárod tőle, hogy LDAP-hitelesítést is végezzen, akkor a legfrissebb változatot kell beszerezned – ez az írás elkészültekor az 1.9-es volt.
A legfrissebb programváltozat működtetéséhez, illetve annak SuSE Linux-változatoktól eltérő rendszerekben való használatához a legjobb megoldás, amelyet választhatunk, ha magunk fordítjuk a forráskódot, amelyet az alábbi címről szerezhetünk be: ➲ http://ftp.suse.com/pub/projects/proxy-suite/src.

Programépítés forráskódból

Az ftp-proxy összeépítését és telepítését érintő összes utasítás megtalálható az INSTALL nevet viselő állományban. Az előzetes beállításoknak megfelelően a beállítási héjprogram ellenőrzi, hogy a libwrap, libldap csomagok telepítve vannak-e, valamint azt, hogy a rendszer támogatja-e szabványos kifejezések (regular expressions) használatát. Red Hat 7.3-as rendszeremen a libwrap és csomag telepítve volt ugyan, de fordításiidő-hibát eredményezett (compile time error), emiatt a libwrapet az alábbi módon kikapcsoltam:

```
# ./configure --without-libwrap
```

és így az ftp-proxy fordítása már megfelelően lezajlott. Mindazonáltal ez a lépés szükségtelen volt, amikor az ftp-proxyt a SuSE 7.1-es rendszeremen fordítottam: magától értetődő, hogy a SuSE- és a Red Hat-változatok libwrap csomagjai eltérőek.
Az ftp-proxy összeépítését és a dokumentáció felmásolását felmásolását is magában foglaló telepítését követően új proxy-kiszolgálónkhoz szükségünk támadhat egy indító héjprogramra. Szerepel az ftp-proxy forráskódjában – az ***ftp-proxy***/ könyvtárban – egy minta héjprogram, az ***rc.script***, amelynek tartalmát a kísérő ***rc.script.txt*** állomány ismerteti.
SuSE változatokban az ***rc.script***-et egyszerűen be kell másolni a ***/etc/init.d*** könyvtárba, majd az egyéni igényeknek megfelelően egy erre mutató közvetett hivatkozást (symbolic link) kell készíteni a ***/usr/sbin*** könyvtárból. Nevezzük át a szóban forgó héjprogramot a ***/etc/init.d/ftp-proxy*** névre, és a közvetett hivatkozást ***/usr/sbin/rcftp-proxy***-ra. Ha valamilyen SuSE 7.x rendszert üzemeltetsz, a ***/etc/rc.config*** állományt a fentieken kívül még a következő sorral is ki kell egészítened:

```
START_FTP_PROXY="yes"
```

A SuSE-től eltérő Linux-változatoknál az ***rc.script*** állomány jelentős mértékben módosításra szorul, mivel ez az állomány nagyon is jellemző a SuSE Linux-változatra. Példák végett vedd szemügyre a Linux-változatod ***init.d*** könyvtárában található egyéb héjprogramokat.
Nos, ha sikerült valami okosat kisütni, javaslom, hogy az átalakított héjprogramot küldd el ***Marius Tomaschewski***-nek (mt@suse.de), aki az ftp-proxy egyik fő fejlesztője, hogy az ő révén mások is hasznosíthassák ragyogó ötleteidet.

Az ftp-proxy beállítása

Amennyiben már akár forráskódból, akár SuSE-csomagból elvégezted az ftp-proxy telepítését, elérkezett az ideje, hogy beállítsuk. A legtöbb beállítható paraméter a ***/etc/proxy-suite/ftp-proxy.conf*** állományban található, illetve a ***/usr/local/etc/proxy-suite/ftp-proxy.conf*** állományban, amennyiben forráskód alapján végeztük a telepítést.
Mielőtt azonban fejest ugranánk az ***ftp-proxy.conf*** állományba, akad néhány apróság, amire figyelmet kell fordítanunk. Először is proxydémonodnak szüksége lesz egy különleges jogok nélküli felhasználói azonosítóra, amelyet használhat. A saját rendszeremen létrehoztam egy ilyen azonosítót:

```
bash-# useradd -u 65500 -g nogroup
➥-d /var/ftp-proxy/rundir -s /bin/false ftpproxy
```

Senki sem jelentkezhet be ezzel a felhasználói névvel. Hogy ezt biztosra vehessük, a ***/etc/shadow*** állományban a proxyfelhasználói bejegyzéshez tartozó sorba tegyünk egy csillagot:

```
ftpproxy:*:12345:0:99999:7:0::
```

Aztán készítened kell egy chroot-dobozt (jail), amelyben a ftp-proxy gyermekfolyamatai majd működni tudnak. A SuSE Linux változat felhasználói számára ez egyszerű feladat – az ftp-proxy indító héjprogramja ezt elvégzi helyettük, ha indítása

1. lista ftp-proxy.conf

```
  ServerType          standalone
  User                ftpproxy
  Group               nogroup
  LogDestination      daemon
# LogLevel            INF
                      # DBG csak hibakeresØs
                      # idejØn hasznÆland !
  PidFile             /var/run/ftp-proxy.pid
# ServerRoot          /var/ftp-proxy/rundir
  AllowMagicUser      no
  AllowTransProxy     yes
  DestinationAddress  192.168.1.2
  ValidCommands       USER, PASS, PWD, CWD,
                      CDUP, PORT, PASV,
                      RETR, TYPE, REST,
                      ABOR, LIST, NLST,
                      TAT, QUIT
```

Tűzfallal kapcsolatos alapismeretek

Ha a tűzfalhasználat számodra még újdonságszámba megy, ez a cikk nagyon keveset fog mondani a számodra. A *Kapcsolódó címekből* kövesd végig *Farrow* és *Power* „A tűzfal-technológiák története" című cikkének gondolatmenetét és a „Building Secure Servers with Linux" című könyvemét is. Mindkettő angol nyelven következő sítlusban tárgyalja a különféle tűzfalelgondolásokat és megvalósításokat.

a `chroot` paranccsal együtt történik:

```
bash-# /etc/init.d/ftp-proxy chroot
```

De még akkor is, ha nem éppen SuSE Linuxot üzemeltetsz, a példaprogramot – a korábban említett ***rc.script*** héjprogramot – meglehetősen egyszerű visszafejteni, hogy kitaláljuk, hogyan is lehetne ezt megoldani. Tulajdonképpen a szükséges ftp-proxy doboz a */var/ftp-proxy/rundir*, ennek tartalmaznia kell azokat a könyvtárakat és állományokat, amelyeket az ftp-proxy használ, valamint a saját */dev/log* különleges naplóállományt, amelyre a helyi `syslog` démon figyelni képes. A `syslog` démonnak a chroot-olt naplózóprogramra való irányításához az indító héjprogramot csupán `-a` kapcsolóval kell kiegészíteni, hogy az elindítsa a rendszernaplózást:

```
syslog -a /var/ftp-proxy/rundir/dev/log
```

A SuSE-rendszerekben ennek a */etc/rc.config* állományban a `SYSLOGD_PARAMS` változón keresztül történő beállítás a megszokott módja. De akár többszörös `-a` utasításokat is kiadhatunk, például akkor, ha a chroot-olt `named`-től fogadunk naplókat.

Az ftp-proxy.conf állomány

Végül eljött az ideje, hogy beállítsuk a proxydémont. Mint azt már korábban említettem, ez az *ftp-proxy.conf* állományban történik meg, amely a */etc/proxy-suite* vagy a */usr/local/etc/proxy-suite* könyvtárak egyikében található. Zavaró vagy bosszantó lehet, hogy a SuSE szóhasználatában a suite elnevezés mindössze egy egyetlen alkalmazásból álló „készletre" utal. Remélhetőleg hamarosan további proxyk fognak elkészülni, és ha annyira hasznosak lesznek, mint az ftp-proxy, én a magam részéről meg fogom bocsátani nekik ezt az aprócska csalást. Ennek az állománynak a leggyorsabb ismertetése a példák felsorolása és részekre szedése (lásd *1. listánkon*). Az első értéket a `ServerType` határozza meg, hogy az ftp-proxyt önálló démonként vagy az `inetd`-ből kell-e futtatni. Jóllehet a programot az imént démonnak neveztem, valójában mindkét üzemmódra képes. Én a nyilvános hozzáférésű kiszolgálógépeken szeretem elkerülni az `inetd` vagy akár `xinetd` futtatását, mivel így az előzetes beállításnak megfelelően elinduló programokat nem kell állandóan kikapcsolgatnom, de a démonként futó programok működéséből származó előnyök miatt is szeretem. Abban az esetben, ha igényeid ettől eltérőek, a `ServerType`-ot `inetd`-re állíthatod, ami szintén működőképes megoldás, ha az `inetd` helyett az `xinetd`-t futtatod.
A felhasználói és a csoportnév magától értetődően elegendő a felhasználói azonosító (UID) és a csoportazonosító (GID) meghatározásához, hogy a kezdőértékek feltöltését követően melyik ftp-proxykiszolgáló alatt fusson. Jó ötlet ezeket különleges jogok nélküli UID-hoz és GID-hez rendelni, hogy csökkentsük a kockázati következményeit annak, hátha egy betörőnek valahogy mégiscsak sikerül eltérítenie egy ftp-proxyfolyamatot. A `LogDestination` kijelöli, hogy az ftp-proxy hová küldje a naplóüzeneteket. Ez lehet egy démon – azaz a helyi rendszernaplózást végző syslog-szolgáltatás – vagy egy állomány, sőt egy csővezeték is. A `LogLevel` a naplózandó adat mennyiségét jelöli: a legtöbb felhasználó számára az `INF` a legjobb választás, azonban a `DBG` – vagyis a lehető legtöbb adat begyűjtése – a hibakeresés során bizonyulhat hasznosnak. A `PidFile` az ftp-proxy számára kijelöli, hol legyen tárolva a mesterfolyamat folyamatazonosítója. Ezt az indító héjprogram használja, amit rendszerleálláskor a `stop` parancs hív meg. Ha viszont az ftp-proxy `inetd` üzemmódban működik, akkor a `PidFile` nincs használatban.
A `ServerRoot` meghatározza a ftp-proxy chroot-dobozának elérési útvonalát. Hagyjuk meg megjegyzéssornak, ha nem szeretnénk, hogy az ftp-proxy a futás során chrootolásra kerüljön (lásd „Chroot kérése az 1.9-es változat alatt" címet viselő keretes írásunkat).

Áttetsző proxyhasználat

Az 1. listában szereplő következő három parancs különösen fontos, ezek határozzák meg, hogy a proxy vajon áttetsző lesz-e. A legtöbb helyzetben az áttetsző proxy a kívánatosabb – a végfelhasználóknak nem kell FTP-ügyfélprogramjuk beállítását elvégezni ahhoz, hogy proxyhasználatukat egyértelműen támogassák. Ehhez csupán az szükséges, hogy az ftp-proxy a rendszermagba fordított Netfilter-kóddal összhangban dolgozzék, amely az FTP-csomagokat a proxydémonunkhoz irányítja, ahelyett, hogy elküldené őket annak a gépnek, amelyiknek tulajdonképpen címezve lettek. Amikor az ftp-proxy ilyen módon átirányított FTP-ügyfélcsomagokat kap, a cél IP-címet annak az új FTP-kapcsolatnak a céljaként használja fel, amelyet a kívánt FTP-kiszolgálóval létesít. A `DestinationAddress` érték az alapértelmezés szerint használandó célt jelöli meg.
Ha felhasználóink számára lehetővé szeretnénk tenni, hogy proxykiszolgálónkat ne áttetsző módon használják, más szóval, hogy FTP-folyamataik kapcsolataikat közvetlenül a proxyki-

Az RFC 959-es szabvány által meghatározott FTP-parancsok

Az RFC 959-es szabvány parancsai	*és azok leírása*
USER	A felhasználó nevének megadására szolgál.
PASS	A jelszó megadására szolgál.
PWD	Az éppen használt munkakönyvtárat írja ki.
CWD	A munkakönyvtár váltására szolgál.
CDUP	A munkakönyvtár szülőkönyvtárára váltó parancs.
TYPE	Adattípus: IMAGE (bináris), ASCII, EBCDIC vagy L (lokális bájtokban kifejezett méret).
MODE	Az adatfolyam típusát állítja be (stream, block, compressed); állítja; a compressed adatfolyam mód visszaélésekre teremt lehetőségeket, ezért nem szabad megengedni, hogy a felhasználók a MODE-ot megváltoztassák.
PASV	Ez a parancs arra utasítja kiszolgálógépet, hogy készüljön fel az adatcsatorna passzív üzemmódú használatára.
PORT	A kiszolgálógépet arra utasítja, hogy az adatcsatornát aktív módban indítsa el (active mode).
RETR	Állomány visszakeresése, tulajdonképpen olvasása.
STOR	Állomány küldése, vagyis felírása a kiszolgálóra.
APPE	Állomány küldése – ha már létezik, akkor az állomány hozzáfűzése a létező állomány végéhez.
REST	Gyors előrecsévélés egy megadott helyzetig. Az utasítást a RETR vagy STOR utasítások egyikének kell követnie.
RNFR	Átnevezés valamiről, amely parancsot ugyanebben a sorban az RNTO parancsnak kell követnie.
RNTO	Átnevezés valamire: ezt a parancsot ugyanebben a sorban az RNFR parancsnak kell megelőznie.
STOU	Állomány felírása, az állomány nevének megváltoztatása, ha az állomány már létezik.
ABOR	Az előzőleg kiadott parancs és annak bármilyen járulékos műveletének törlése (állományátvitel) stb.
DELE	Állomány törlése.
RMD	Könyvtár eltávolítása.
MKD	Könyvtár létrehozása.
LIST	Az állományok nevének és tulajdonságainak felsorolása.
NLST	Csak az állományok nevét mutató rövid lista.
SYST	A kiszolgálógéptől lekérdezi az rajta futó operációs rendszer típusát.
STAT	Egy megadott állomány tulajdonságainak vagy a kiszolgáló gép/kapcsolat állapotának lekérdezése.
QUIT	A kapcsolat leállítása.

Chroot kérése az 1.9-es változat alatt

Talán neked is gondjaid támadnak, amikor az ftp-proxy-t chrootolva futtattad – nekem legalábbis akadtak. Mindjárt az elején ftp-proxy indító héjprogramjában a chrootdoboz használata során furcsa eredményekhez jutottam az ftp-proxy chroot támogatása révén. Az 1.9-es változattal folytatott próbaüzem során az ftp-proxy nem volt hajlandó elindulni, ekkor a *Can't determine Group-ID to use* (Nincs meghatározva, hogy milyen csoportazonosítót kell használni) üzenettel ért véget. Az 1.7-es változattal végzett próbafuttatások során a fentebb említett gond nem jelentkezett, de áttetsző üzemmódban az ftp-proxy nem volt képes megfelelően megoldani a címfeloldást. Lehet, hogy olvasóim jobb eredményekre jutnak, mint jómagam.
Ha kétségek merülnének fel, segítséget kaphatunk egy a SuSE Linux által erre a célra fenntartott levelezőlistáról. A feliratkozáshoz küldj egy elektronikus levelet a proxy-suite-subscribe@suse.com címre.

szolgálóval kezdeményezhessék, akkor az `AllowMagicUser` értékét ***yes***-re állítani. Ezt azonban nem javaslom, ha a proxykiszolgálót külső felhasználók is használni fogják, minthogy pontosan ez az eset áll fenn a nyilvános hozzáférésű FTP-kiszolgálókkal. Az `AllowMagicUser` változó hatására proxykiszolgálónk olyan nyitott proxyként fog működni, amelyet a külső felhasználók alkalomadtán arra is felhasználhatnak, hogy általa más, külső kiszolgálókhoz kapcsolódjanak, esetleg támadási célzattal.
Ha azonban a Netfiltert úgy állítottuk be, hogy a proxy felhasználói kapcsolatokat kizárólag a belső hálózati, „tiszta” oldalról fogadjon el, és az `AllowMagicUser` változót ***yes***-re állítottuk, a felhasználóknak lehetőségük lesz arra, hogy a felhasználói azonosítót és a kukacot (@) követően megadják az elérni kívánt FTP-kiszolgálót, például mick@ftp.wiremonkeys.org. Az `AllowMagicUser` használható, tekintet nélkül arra, hogy az `AllowTransProxy` változó értéke ***yes***-re vagy ***no***-ra lett-e állítva. Fontos megjegyezni, hogyha ez utóbbi változó értéke ***no***-ra lett állítva, és az `AllowMagicUser` változóé szintén, akkor minden FTP-kapcsolat a `DestinationAddress` változót fogja használni. Az egyéb paraméterek között szerepel a `MaxClientString` és a `DestinationTransferMode`. A teljes lista végett olvasd el a ftp-proxy.conf(8) súgóoldalát, ugyanitt a fentebb ismertetett változókról bővebb ismeretek is beszerezhetők.

A Netfilter beállítása transzparens proxyhasználathoz

A működőképes transzparens proxyhasználathoz az IP Tables telepítésére van szükség, ami képes az FTP-csomagokat a helyi proxykiszolgálóhoz irányítani, más szóval a proxykiszolgáló gépen a Netfiltert kell futtatni – a cikkben ezt tételezzük fel. Természetesen ezenkívül szükség lesz az FTP-kapcsolatokat leíró szabályokra is, vagyis arra, hogy a proxykiszolgálóhoz milyen kimenő és bejövő kapcsolatok megengedettek. A továbbítási láncban (FORWARD chain) viszont semmilyen szabályra nem lesz szükség.
Először is az áttetsző proxyhasználat támogatásához a Linux 2.4-es tűzfalra több modult be kell tölteni: az `ipt_conntrack_ftp` és az `ip_nat_ftp` modul szükséges az FTP-kapcsolat nyomon követéséhez; az `ipt_REDIRECT` modul pedig a szabály céljának átirányításához nélkülözhetet-

2. lista Az IP Tables program – a transzparens proxyhasználat parancsai

```
iptables -t nat -A PREROUTING -p tcp -i eth2 \
  --dport 21 -j REDIRECT
iptables -t nat -A PREROUTING -p tcp -i eth0 \
  --dport 21 -j REDIRECT

    # kihagyÆs ...

iptables -A INPUT -p tcp -d $PUBLIC_FTP \
  --dport 21 -m state --state NEW,RELATED \
    -j ACCEPT
iptables -A INPUT -p tcp -s $INTERNAL_HOSTS \
  --dport 21 -m state --state NEW,RELATED \
    -j ACCEPT

    # kihagyÆs ...

iptables -A OUTPUT -p tcp -d $PUBLIC_FTP \
  --dport 21 -m state --state NEW,RELATED \
    -j ACCEPT
iptables -A OUTPUT -p tcp -o eth2 --dport 21 \
  -m state --state NEW,RELATED -j ACCEPT
```

len. A legtöbb Linux-változat valamilyen 2.4-es rendszermagot állít csatasorba, amelyek ezeket a modulokat már tartalmazzák. Amint a modulok betöltése megtörtént, Netfilter indító héjprogramod számára tűzfalszabályokat adhatsz meg, amint az a *2. ábrán* látható. A *2. lista* első két utasítása a tűzfalat arra utasítja, hogy az összes olyan külső és belső csatolófelületén (`eth2` és `eth0`) kapott csomagot irányítsa át, amiknek a TCP 21-es kapuja a célkapuja, vagyis a kiszolgálógépen az FTP számára fenntartott kapu. Fontos megjegyezni, hogy ezek a csomagok semmilyen módon nem lesznek újraírva (mangled), egyszerűen csak át lesznek irányítva a helyi FTP proxydémonhoz.
A 2. lista harmadik és negyedik parancsa arra utasítja a tűzfalat, hogy fogadja el a nyilvános hozzáférésű FTP-kiszolgáló 21-es kapujára küldött összes csomagot – ahol a `PUBLIC_FTP` változó tartalmazza annak IP-címét –, és a belső felhasználók által küldött összes FTP-csomagot, ahol az `INTERNAL_HOSTS` változó tartalmazza a címtartományt a CIDR-jelölésnek megfelelően, vagyis például `192.168.99.0/24`. Az első két sor miatt, ha a harmadik és negyedik sorban lefektetett szabályoknak megfelelő csomag érkezik, az a helyi proxyhoz lesz irányítva.
A 2. lista ötödik és hatodik sora a helyi ftp-proxy démon számára lehetővé teszik, hogy a külső csatolófelületről kapcsolatot létesítsen a megadott nyilvános hozzáférésű FTP-kiszolgálókkal és gépekkel, ez ebben a példában az `eth2` volt.
A 2. listában szereplő sorok nem alkotnak önálló Netfilter-szabálygyűjteményt, csupán azt mutatják, hogyan egészítheted ki a címfordítást (NAT) és egyebeket. A héjprogram a `PUBLIC_FTP` és `INTERNAL_HOSTS` változókat is be kell állítsa. Jó szokás ilyen beszédes neveket használni, ez ugyanis érdekesebbé teszi a kódot.

Az FTP-használatot korlátozó parancsok

Most pedig térjünk vissza az ***ftp-proxy.conf***-hoz (1. lista), pontosabban az ftp-proxy egyik legfontosabb szolgáltatásához: a `ValidCommands`-hoz. Ez nem más, mint a proxy által megengedett, vesszőkkel elválasztott parancsok felsorolása. A sorokat a fordított perjel írásjellel (\) zárjuk le, a lista akár több sorra is kiterjedhet. Az 1. lista alján a `ValidCommands` utasításban az ftp-proxy úgy lett beállítva, hogy az FTP-könyvtárak közötti tájékozódást segítő parancsok megengedettek: `PWD`, `CWD`, `CDUP`; az FTP-olvasó parancsai: `LIST`, `NLST`, `RETR`; ezenkívül néhány felügyeleti parancs: `MODE`, `PORT`, `PASV`.
Itt helyhiány miatt nem áll módomban részletes magyarázatot adni, csupán annyit mondhatok, hogy ezek nem végfelhasználói ügyfélparancsok, hanem olyan, az FTP-protokollnak szóló parancsok, amelyek meghatározását megtalálhatjuk az RFC959-ben (➲ http://ftp.isi.edu/in-notes/rfc959.txt). E parancsokat az FTP ügyfél- és kiszolgálóalkalmazások egymás között használják. A parancsok összesítését *táblázatunk* tartalmazza.
Az ftp-proxy egyik korlátozása az, hogy a külső felhasználók számára nem lehet a belső felhasználókétól eltérő korlátozó parancsokat kialakítani. Emiatt aztán a `ValidCommands`-sal légy óvatos – ha az FTP-kiszolgálóknak a belső felhasználóknak kell állományokat küldenie, akkor nincs lehetőség a `STOR` vagy `STOU` parancsok korlátozására, vagyis ez azt jelenti, hogy ezeket a parancsokat fel kell venni a `ValidCommands` parancsai közé. De ez egyúttal azt a feladatot is magában hordozza, hogy meg kell győződnünk róla, hogy a csak olvasható, nyilvános hozzáférésű FTP-kiszolgáló úgy van-e beállítva, hogy ezeket a parancsokat figyelmen kívül hagyja őket.

Összegzés

Az FTP-proxy fontos biztonsági réteget képez a tisztességtelen emberek és a nyilvános hozzáférésű FTP-kiszolgálógép között. A cikkben ismertettem az áttetsző proxy-használatot a SuSE Linux proxy-suite készletével, ami még sok megismerésre érdemes szolgáltatással dicsekedhet, de ezeket most nem tudtam bemutatni. Bővebb ismeretek végett böngészd át a ***Kapcsolódó címek*** részt. Szerencse fel!

Linux Journal 2002. december, 104. szám

Mick Bauer (mick@visi.com)
Hálózati biztonsági tanácsadó az Upstream Solutions Inc.-nél Minneapolisban (Minnesota). Mick a szerzője a hamarosan megjelenő új O'Reilly könyvnek, amelynek címe „Building Secure With Linux".

KAPCSOLÓDÓ CÍMEK

A hivatalos SuSE Linux Proxy-Suite honlapja
➲ http://www.suse.de/en/whitepapers/proxy_suite
Az FTP – File Transfer Protocol, illetve az RFC, amely meghatározza, hogyan is működik az FTP
➲ http://ftp.isi.edu/in-notes/rfc959.txt
Rik Farrow és *Richard Power* History of Firewall Technologies (A tűzfal-technológiák története)
➲ http://www.spirit.com/Network/net0597.txt
A *Mick Bauer* nevével fémjelzett Building Secure Servers with Linux (Biztonságos kiszolgálógépek építése Linuxszal) című könyv hivatalos honlapja
➲ http://www.oreilly.com/catalog/bssrvrlnx