Az rsync (2. rész)

Az rsync-modulok fájlrendszerszintű beállításai és kapcsolatok létrehozása.

Az előző alkalommal névtelen felhasználók számára helyeztünk üzembe egy rsync-kiszolgálót. Az első kódrészlet – a múlt hónapban is mintaként használt ***rsyncd.con***f állomány – néhány biztonsági szempontból hasznos beállítást szemléltet. Példánkhoz visszatérve most ejtsünk néhány szót az rsync-modulok (könyvtárak) fájlrendszerszintű beállításairól. A fő elvek ebben az esetben is megegyeznek a névtelen FTP `chroot`-környezetekben használtakkal. A különbség mindössze annyi, hogy a `chroot` használatához nem kell futtatható vagy beállítófájlokat bemásolni, mint azt néhány FTP-kiszolgálónál látni.

Az rsync beállítóállományában csak apróbb módosításokat kell végrehajtani az elérési útvonalak és engedélyezett állomások között, és máris a névtelen felhasználók rendelkezésére állhatunk. Ez még elég sovány dolog. Hogyan tudjuk fogadni a névtelen feltöltéseket, illetve miként hozhatunk létre külön modult a hitelesített felhasználók számára? A második kódrészlet mindkét kérdésre megadja a választ.

Először szükségünk lesz egy bejövő (a továbbiakban `bejovo`) nevű modulra, aminek az elérési útvonala */**home**/**bejovo***. Most is a nyilvánosan írható könyvtárak esetében (lásd „Tips for Securing Anonymous FTP” a „Building Secure Servers with Linux” című műben) alkalmazott elvek jutnak érvényre, egyetlen fontos különbséggel: a névtelen rsync szolgáltatás esetében a kérdéses könyvtárra mindenkinek futtatási és írási jogot kell adni, vagyis 0733-as módot kell beállítani. Ha ezt elmulasztjuk, a feltöltések anélkül lesznek sikertelenek, hogy a felhasználó erről bármiféle hibaüzenetet kapna, vagy a kiszolgáló naplójába bejegyzés kerülne.

Ebben az esetben is érdemes megfogadni néhány, az FTP-kiszolgálókra is érvényes tanácsot: a könyvtár tartalmának változását mindig kövessük figyelemmel, nehogy visszaéljenek a szolgáltatásunkkal, és a tartalma soha ne legyen bárki által olvasható. A feltöltött fájlokat a lehető leghamarabb – célszerűen a `cron`-ból – mozgassuk át egy nyilvánosan el nem érhető könyvtárba.

A `[bejovo]` csoportban az egyetlen új beállítás az átvitelek naplózása. Ezzel arra utasítható az rsync, hogy egy kicsit részletesebb naplót készítsen a fájlátviteli próbálkozásokról. Alapértelmezett esetben a beállítás értéke: `no` (nem). Emellett a már ismert `read-only` beállítás is `no` értéket kapott, felülbírálva ezzel az átfogó beállítások között szereplő `yes`-t (igen). Nincs hasonló eszköz arra, hogy az rsync tudomására hozzuk a könyvtár írható voltát. A lehetőségeket a könyvtárra megadott engedélyek szabják meg.

A példa második része egy korlátozott elérhetőségű, `Zenebolondok` nevű modult ad meg. Ebben három új beállítással ismerkedhetünk meg. Az első a `list`, ez adja meg, hogy a modul megjelenjen-e, amikor a távoli felhasználók a kiszolgálón elérhető modulok listáját kérdezik le. Alapértelmezett értéke `yes`.

A másik két beállítás, az `auth users` és a `secrets file` szabja meg a csatlakozó ügyfelek hitelesítésének módját. Az rsync hitelesítési összetevője, ami csak démonmódban érhető el, egy meglehetősen komoly, 128-bites, MD5 alapú kihívás-válaszadás jellegű eljárást használ. Ez két okból is jobb a normál FTP-hitelesítésnél. Egyrészt a rendszer nem továbbítja a jelszavakat a hálózaton keresztül, így nem lehet őket lehallgatni. Ettől még – elméletileg – nyers erővel (brute force), kivonatokkal kivitelezett támadást lehet indítani a kiszolgáló ellen!

Másrészt az rsync nem használja a rendszer hitelesítő adatait, a felhasználónév–jelszó párosokat saját állományban tárolja. Ezt az állományt kizárólag az rsync használja, semmilyen kapcsolatban nincs a */**etc**/**passwd*** vagy a */**etc**/**shadow*** állománnyal. Ha tehát valaki eredményesen támadja az rsyncet, a felhasználók rendszerazonosítói nem kerülnek közvetlen veszélybe, hacsak valaki nem végzetesen rosszul állította be az rsync könyvtárhozzáféréseit vagy az adott könyvtárakra vonatkozó jogosultságokat.

Maguk az adatátvitelek az FTP-hez hasonlóan titkosítás nélkül folynak – tehát igaz, hogy az rsync ellenőrzi a felhasználók adatait, ám az adatok épségét nem biztosítja, illetve bizalmasságukat sem védi a hallgatózóktól. Ilyen elvárások esetén SSH vagy Stunnel felett kell futtatni.

A `secrets file` beállítás adja meg az rsync felhasználónév-jelszó párosokat tároló állományának nevét. Ez hagyományosan a */**etc**/**rsyncd.secrets***, de gyakorlatilag bárhol, tetszőleges névvel elhelyezhető, és a ***.secrets*** végződés sem kötelező. Ennek a beállításnak nincs alapértelmezett értéke. Ha az `auth users` beállítást engedélyezed, akkor a `secrets file` értékét is meg kell adnod. Az alábbi részlet egy `secrets file` tartalmát szemlélteti:

```
watt:shyneePAT3
bell:d1ngplunkB00M!
```

A /etc/rsyncd.secrets mintaállomány tartalma

A második kódrészlet `auth users` beállítása adja meg, hogy a `secrets file`-ban szereplő felhasználók közül kik jogosultak a modul elérésére. Minden olyan ügyfélnek, aki ehhez a modulhoz próbál csatlakozni – feltéve, hogy átjutott a `hosts allow` és a `hosts deny` hozzáférés-vezérlési listák révén emelt akadályokon –, meg kell adnia nevét és jelszavát. Ne feledd pontosan megadni az érintett fájlokra és könyvtárakra vonatkozó jogosultságokat, ugyanis ezek szabják meg, hogy csatlakozás után a hitelesített felhasználók mit művelhetnek. Ha az `auth users` beállításnak nem adsz értéket, akkor a rendszer nem követeli meg a felhasználók hitelesítését, és a modul névtelen rsync-hozzáféréssel is elérhető lesz. Démonmódban ez az rsync alapértelmezett viselkedése.

Nagyjából ez az, amit egy névtelen és hitelesített felhasználók által egyaránt elérhető rsync-szolgáltatás beindításához tudnod kell. A parancssori és a beállítófájlban található további beállításokról – köztük az itt nem tárgyalt, a naplóüzenetek testreszabására használhatókról – az rsync(8) és az rsyncd.conf(5) súgóoldalak szolgáltatnak bővebb tájékoztatást.

1. lista Egy rsyncd.conf mintafájl

```
# kizÆr lag Ætfog  hat k rrel megadhat
# beÆll tÆsok
syslog facility = local5

# Ætfog  jellegß, de a modulokn Æl is
# megadhat  beÆll tÆsok
use chroot = yes
uid = nobody
gid = nobody
max connections = 20
timeout = 600
read only = yes

# pØldamodul:
[public]:
   path = /home/public_rsync
   comment = NyilvÆnos fÆjlok
   hosts allow = helyi.valami.org,
                 ➥10.18.3.12
   ignore nonreadable = yes
   refuse options = checksum
   dont compress = *
```

2. lista További rsyncd.conf modulok

```
[bejovo]
path = /home/bejovo
comment = Ide  rhatsz, de nem olvashatod
read only = no
ignore nonreadable = yes
transfer logging = yes

[Zenebolondok]
path = /home/cvs
comment = Zenebolondok CVS-tÆr
list = no
auth users = watt,bell
secrets file = /etc/rsyncd.secrets
```

Az rsync használata távoli rsync-kiszolgáló elérésére

Nehogy elfeledjem: még nem mondtam el, hogy rsync-kiszolgálóhoz hogyan kell ügyfélként kapcsolódni. Az írásmód roppant egyszerű, a távoli állomás nevének beírásakor egy helyett két kettőspontot kell használni, és a kívánt modulhoz abszolút helyett relatív elérési útvonalat kell megadni. Példaként vegyük elő az előző hónap felállását, amiben a helyi gépet helyinek, a távolit pedig távolinak hívtuk, és tegyük fel, hogy az ***ujcuccok.tgz*** fájlt szeretnénk letölteni, illetve a távoli gépen démonmódban fut az rsync. Emellett azt is feltesszük, hogy nem emlékszünk már rá, hogy a távoli gépen milyen nevű modul alatt találhatók az új állományok. Először tehát le kell kérdezni az elérhető modulok listáját:

```
[root@helyi darthelm ]# rsync tavoli::
public      NyilvÆnos fÆjlok
bejovo      Ide  rhatsz, de nem olvashatod
```

(Korántsem véletlen, hogy a modulok neve így alakul, a példákban is ezeket állítottuk be. Természetesen a Zenebolondok modul neve sem véletlenül nem jelenik meg a listában.)
A keresett könyvtár neve ***public***. Ezek után az alábbi parancscsal tudod az aktuális munkakönyvtáradba másolni az *ujcuccok.tg* fájlt:

```
[yodeldiva@helyi ~]# rsync
➥tavoli::public/ujcuccok.tgz
```

A kétszeres kettőspont és az elérési útvonal formátuma is eltér az SSH-módnál megismerttől. Míg az SSH abszolút elérési útvonalat vár a kettőspont mögött, az rsync démon egy modulnevet keres, ami a fájl elérési útjának gyökereként szolgál. Csak a példa kedvéért lássuk ugyanezt a parancsot SSH-módban:

```
[yodeldiva@helyi ~]# rsync -e ssh
➥tavoli:/home/public_rsync/ujcuccok.tgz
```

A két parancs természetesen nem teljesen egyenértékű, hiszen a távoli gép rsync démonfolyamatát a könyvtár tartalmának névtelen – vagyis hitelesítést nem végző – felhasználók számára való elérhetővé tételére állítottuk be, az SSH viszont minden alkalommal megköveteli a hitelesítést (igaz, ezt önműködővé is lehet tenni, ha nulla hosszúságú RSA vagy DSA kulcsot alkalmazunk; lásd a „Building Secure Servers with Linux” negyedik fejezetét). Nem is ez volt a lényeg, hanem az, hogy megmutassam az elérési útvonalak kezelésében jelentkező különbséget.

Az rsync bújtatása Stunnel segítségével

Az utolsó rsync használati mód, amiről szót ejtek, a démonmódban futó rsync és az Stunnel párosítása. Az Stunnel olyan általános célú TLS- vagy SSL-burkoló, ami bármelyik egyszerű TCP alapú átvitel titkosítással – és választható módon X.509 tanúsítvánnyal – védett beágyazására használható. Igaz, hogy SSH-módban futtatva az rsync titkosítással ruházható fel, ám elveszti démonmódban elérhető szolgáltatásait, amik közül a névtelen hozzáférés lehetőségét emelném ki. Az Stunnel segítségével legalább olyan jó titkosítás érhető el, mint SSH-val, de megtartható a névtelen kapcsolatok támogatása is.

Mi a helyzet az önhívással (recursio)?

Azzal kezdtem, hogy az rsync mennyire hasznos, ha nagyobb adatmennyiségeket – programarchívumokat, CVS-fákat – kell másolni, de összes példámban csak egy-egy állomány másolásáról volt szó. Ennek oka az, hogy elsősorban az rsync biztonságos használatát akartam bemutatni.
A rengeteg ügyféloldali (parancssori) rsync-kapcsoló felfedezésének örömét meghagytam neked, kedves olvasó. Mindenre kiterjedő leírást az rsync(8) súgóoldalon találsz. Különösen figyelemre méltó a `-a` (vagy `--archive`) kapcsoló, ami a `-rlptgoD` rövidítése, és a legtöbb fájltípusra – eszközökre és közvetett hivatkozásokra is – vonatkozóan önhívó működést ír elő; illetve a `-C` (vagy `--cvs-exclude`), ami az rsyncet CVS stílusú fájlkihagyási szabályok használatára utasítja a másolandó fájlok kiválasztásakor.

Az Stunnelről részletesen is olvashatsz a „Building Secure Servers with Linux” 5. fejezetében, ahol sok példában az rsyncet is viszontláthatod. Akadhatnak olyanok is, akik számára újdonság következik:

Ebben az esetben az alábbi lépéseket kell a kiszolgálóoldalon végrehajtani

1. Az ***rsyncd.conf*** beállításait a megszokott módon kell megadni.
2. Az rsyncet a `--port` kapcsolóval kell meghívni, és 873-tól eltérő kapuszámot kell megadni, például:
   `rsync -daemon --port=8730`
3. Egy Stunnel-figyelőt (listener) kell életre hívni a 873-as TCP-kapun, ami minden, ezen a kapun érkező bejövő kapcsolatot az előző lépésben megadott kapura fog irányítani.
4. Ha nem akarod, hogy bárki kedvére kapcsolódhasson a géphez, a ***hosts.allow*** állományban tiltsd le a 2. lépésben megadott kapura vonatkozó nem helyi kapcsolatokat. Emellett (vagy ehelyett) az IP Tables beállításait is hasonlóan módosíthatod.

Az ügyféloldalon az alábbiak szerint alakulnak a dolgok

1. Lépj be rendszergazdaként, állíts be egy Stunnel-figyelőt a 873-mas TCP-kapura (feltéve, hogy nincs rsync-kiszolgáló a helyi gépen, ami már használja ezt a kaput), ez az összes erre a kapura érkező bejövő kapcsolatot a távoli kiszolgáló 873-mas TCP-kapujára fogja irányítani.
2. Amikor kapcsolódni akarsz a távoli kiszolgálóhoz, a távoli kiszolgáló neveként `localhost`-ot kell megadnod. A helyi Stunnel-folyamat csatlakozik a kiszolgálóhoz, és rsync-csomagjaidat a távoli Stunnel-folyamat felé továbbítja. A távoli Stunnel-folyamat visszafejti az általad küldött rsync-csomagokat, és átadja őket a távoli rsync-démonnak. A válaszcsomagok természetesen ugyanezen a titkosított kapcsolaton keresztül érkeznek.

Mint látható, maga az rsync most is nagyon hasonló beállításokat használ, mint a névtelen csatlakozások esetében – a munka túlnyomó része az Stunnel-továbbítók beállításával kapcsolatos.

Linux Journal 2003. április, 108. 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

- http://www.rsync.org/
- http://samba.anu.edu.au/rsync
- http://sunsite.dk/info/guides/rsync/rsync-mirroring.html
- http://freshmeat.net/projects/rsync/