RSSH

Biztonság korlátok között

Az ssh igen népszerű alkalmazás, amely egy sokféleképpen használható, titkosított csatornát biztosít 2 gép között. Ez a rugalmasság azonban egy többfelhasználós rendszeren biztonsági problémákat is eredményezhet. Az rssh egy olyan segédprogram, amely ennek megoldásában siet a segítségünkre.

Gyakori kívánság, hogy a felhasználók titkosított csatornán mozgathassák állományaikat a gépek között. Noha több *ftp* kiszolgáló is biztosít *SSL* támogatást, ez bizonyos környezetekben problémát okozhat, például ha 2 tűzfal is áll az útban. Szerencsére az *ssh*, az *sftp* illetve *scp* programok segítségével, egyetlen *TCP* kapu (*port*) használatával biztosítja a titkosított átvitelt.

Egy apró probléma azért marad: amíg az *ftp* kiszolgálók esetében a felhasználók beérik egy másra nem használható héjprogrammal (*shell*), például */bin/false*, addig az *ssh* szolgáltatásainak használatához egy tisztességes *shell*, például */bin/bash* is szükséges. Így azonban, aki például *sftp*-vel be tud jelentkezni, az *ssh*-val is be tud, ill. az *sftp* nem tudja bezárni a felhasználót a saját *home* könyvtárába (*chroot*), ami az *ftp* kiszolgálók alapszolgáltatása. Az *rssh* azonban mindkét problémát orvosolni képes.

A telepítéshez töltsük le a legfrissebb verziót a ➲ *http://www.pizzashack.org/rssh/* oldalról, majd csomagoljuk ki, fordítsuk le, és telepítsük a következő módon:

```
tar zxvf rssh-2.3.2.tar.gz
cd rssh-2.3.2
./configure
make
su -c 'make install'
```

Első lépésben állítsuk be a `bela` nevű felhasználó héjprogramját:

```
usermod -s /usr/local/bin/
➥rssh bela
```

Ezután szerkesszük az *rssh* konfigurációs állományát (*/usr/local/etc/rssh.conf*), és adjuk hozzá a következő sort:

```
user = "bela:022:
➥00010:"
```

A felhasználó név után az *umask* paraméter szerepel, majd annak a meghatározása, hogy Béla milyen módon jelentkezhet be a gépünkre. Ahol 0 szerepel, az a protokoll nem engedélyezett, ahol pedig 1, az igen. Az egyes mezők jelentése a következő:

```
rsync:rdist:cvs:sftp:scp
```

Béla tehát kizárólag *sftp*-vel jelentkezhet be. Ha ezután mégis *ssh*-val próbál meg bejelentkezni, akkor egy ehhez hasonló üzenetet kap:

```
$ssh -l bela szerver.ceg.hu
bela@szerver.ceg.hu's password:

This account is restricted by
➥rssh.
Allowed commands: sftp

If you believe this is in
➥error, please contact your
➥system administrator.

Connection to szerver.ceg.hu
➥closed.
```

Béla arról kapott tájékoztatást, hogy az ő felhasználói fiókját az *rssh* korlátozta le úgy, hogy csak az sftp program használata engedélyezett. Jogorvoslatért pedig a rendszergazdához

fordulhat. A naplóban pedig az alábbi üzenet jelent meg:

```
Nov 24 11:31:55 plutonium
➥rssh[2128]: user bela attempted
➥to log in with a shell
```

Lépjünk be most *sftp*-vel:

```
$sftp bela@szerver.ceg.hu
Connecting to szerver.ceg.hu...
bela@szerver.ceg.hu's password:
sftp>
```

A naplóban ezúttal az alábbi listában látható üzenet jelent meg, amely arról tájékoztat, hogy Béla ezúttal *sftp*-vel jelentkezett be:

```
Nov 24 11:32:06 plutonium
➥sshd[2138]: subsystem request
➥for sftp
Nov 24 11:32:06 plutonium
➥rssh[2139]: setting log
➥facility to LOG_USER
Nov 24 11:32:06 plutonium
➥rssh[2139]: setting umask to
➥022
Nov 24 11:32:06 plutonium
➥rssh[2139]: line 52:
➥configuring user bela
Nov 24 11:32:06 plutonium
➥rssh[2139]: setting bela's
➥umask to 022
Nov 24 11:32:06 plutonium
➥rssh[2139]: allowing sftp to
➥user bela
```

A következő listában szereplő konfigurációs részlet segítségével házirendet (*policy*) alakíthatunk ki. Az alapértelmezett házirend szerint semmit sem engedünk meg a felhasználóknak.

```
#allowscp
#allowsftp
#allowcvs
#allowrdist
#allowrsync
```

Ezután az *rssh.conf* állományban definiáljuk, hogy az egyes felhasználók, milyen módon jelentkezhetnek be. Én hasznosnak tartom azt, ha a legtöbb felhasználó által használt módokat engedélyezzük, és csak a megkülönböztetett felhasználókat definiáljuk külön. Néhány példa a következő listában látható:

```
user=bela:011:00100:   # csak
➥cvs
user=geza:011:01000:   # csak
➥rdist
user=joska:011:10000:  # csak
➥rsync
```

Chroot

Az *rssh* arra is lehetőséget ad, hogy a felhasználókat bezárjuk egy adott könyvtárba. Az eljárás nem egyszerű, mert egy komplett futtató környezetet kell felépíteni hozzá. *RedHat* alapú rendszerek esetén használhatjuk ehhez a `mkchroot.sh` héjprogramot, más *Linux* disztribúciók esetén ezt magunknak kell megtennünk.
Az *rssh* ehhez a funkcióhoz az `rssh_chroot_helper` programot használja. Ez úgy működik, hogy ez utóbbi kiolvassa a konfigurációs állományból az adott felhasználóhoz rendelt *chroot* könyvtárt, majd végrehajtja a *chroot()* függvényt, majd az *execv()* rendszerhívással futtatja az adott *ssh* szolgáltatáshoz tartozó démont, például az *sftp-server*-t.
Az *rssh* alkalmas az összes támogatott szolgáltatás (*rdist, rsync, cvs, sftp, scp*) ketrecbe zárására. Adott esetben szükség lehet, hogy az *ssh*-val bejelentkezett felhasználókat is bezárjuk. Ezt azonban nem támogatja az *rssh*. Ez legkönnyebben (?) az *sshd* démonon belül lehetne megvalósítható, esetleg egy másik segédprogram segítségével. Növelhető úgy is a biztonság, ha az *rssh*-val korlátozott felhasználókat egy külön csoportba tesszük (például *rssh*), és a csoportnak csak a feltétlen szükséges binárisokhoz adunk hozzáférést. Ez azonban a kezdeti beállításokon túl utólagos is meglehetősen munkaigényes, például a frissítések után kézzel, esetleg egy erre a célra írt héjprogrammal, kell a jogosultságokat ismét beállítani. A *chroot* környezet kialakításához szükséges részletes tudnivalókat a *CHROOT* állományban találjuk meg. Sok sikert kívánok ehhez, nekem sajnos ez nem sikerült. A feladat nehézségét az is jelzi, hogy a levelező listán a legtöbb kérdés ezzel a funkcióval kapcsolatos.

A jövő

A program írója szerint az *rssh* készen van, nem várható a funkciók további bővülése, legfeljebb a napvilágra került hibákat (*bug*) javítja ki. A biztonságot azzal is növelni lehetne még, ha az *rssh_chroot_helper* program kriptográfiailag leellenőrizné, hogy valóban az *rssh* program hívta meg. A program írója ennek hiányát ilyen irányú tapasztalatainak elégtelen voltával magyarázta meg.
A mai heterogén platformok világában probléma lehet, hogy a **BSD* platformokon hiányzik a *wordexp()* függvény – pedig ez *POSIX.2* része – amely a parancssori argumentumok növelésére használható. A program írója azonban beleunt abba, hogy *BSD* rendszerekre megírja ezt a funkciót. Így aki *Linux* mellett például *FreeBSD* rendszereket is üzemeltet, annak be kell szereznie ezt a függvényt, mondjuk a *glibc2* implementációból. További nehézség lehet az, ha az *sftp* kapcsolatot *Windows* platformról is igénybe akarják venni az ügyfelek, mert nem mindegyik alkalmazást támogatja az *rssh*. A program írója szerint a *WinSCP* javított, és a legutolsó változata már képes vele együttműködni. A szerző egyébként a *FileZilla* ill. a *SecureFX* programokat javasolja.

Sütő János
(jsuto@freemail.hu)
1997 óta használ Slackware Linux-ot. Szabadidejében a postfix clapf nevű vírus- és spamszűrőjét polírozza.