LIDS – A Linux betörésvédelmi rendszere

A fájlok jogosultságainak beállítása néha kevés. A LIDS olyan kernelszintű eszköz, amelynek hatása messze túlmutat a fájlrendszer beállításaival elérhető biztonságon.

Amint egyre többen használnak ***Linuxot***, egyre több biztonsági hibára derül fény. Ez egyrészt természetes, másrészt a fejlesztés nyíltsága miatt a legtöbb ilyen problémát villámgyorsan megoldják a programozók.
Néha azonban az is előfordulhat, hogy nem sikerül valamelyik problémát időben megoldani, így a köztes időszakban a kérdéses szoftvert futtató gépek sebezhetőek lesznek. A rosszindulatú felhasználók a rést kihasználva esetleg root jogosultságot szerezhetnek, és akár nagy pusztítást is végezhetnek. Ez az a pont, ahol nagy hasznát vehetjük a ***LIDS*** rendszernek (***Linux Intrusion Detection System; Linux Behatolásérzékelő Rendszer***).
A ***LIDS*** egy kernelfolt, ami a rendszermag szintjén tesz lehetővé a hozzáférési jogosultságok ellenőrzését, illetve a kapcsolatok kezelését. Két adminisztrációs segédeszköz tartozik hozzá a `lidsconf` és a `lidsadm`, amelyekkel minden szolgáltatása beállítható.
A `lidsadm` tulajdonképpen egy olyan segédprogram, amellyel a ***LIDS***-et egy adott terminálra nézve üzemen kívül helyezhetjük. Így elvégezhetünk olyan beállításokat, amelyeket amúgy maga a ***LIDS*** hivatalból meggátolna. A programmal egyben a jelenleg érvényes beállításokat is megjeleníthetjük.
A `lidsconf` arra szolgál, hogy segítségével egyes fájlok hozzáférési jogosultságait beállíthassuk. Ezek lehetnek bináris fájlok vagy természetesen bármi más. A ***LIDS*** nevezéktanában ezeket a fájlokat objektumoknak (objects) hívják, a velük kapcsolatos engedélyezett vagy tiltott képességek illetve műveletek pedig a szubjektumok (subjects). A ***LIDS*** automatikusan felülbírálja az olyan beállításokat mint például a fájlrendszerben tárolt jogosultságok. Ez azt jelenti, hogy a ***LIDS*** segítségével elérhetetlenné tehetünk gyakorlatilag bármit, legyen az egy fájl, nyers adatokat kezelő eszköz, memóriaterület vagy I/O port. Ráadásul a korlátozások magára a rendszergazdára is érvényesek. Röviden, a ***LIDS*** egy teljes biztonsági modell ***Linuxon*** megvalósítva.

Telepítés

A fejlesztők a telepítéssel kapcsolatos utasításokat szokásos módon a csomagban található ***INSTALL*** nevű fájlban írták le. Ez alapján ugyan mindenki elboldogulhat a rendszerrel, ezzel együtt talán nem árt, ha végigmegyünk a fontosabb lépéseken. A ***LIDS*** stabil változatait mindig az eredeti (vanilla) ***Linux*** rendszermaggal együtt lehet használni. Ennek megfelelően a fejlesztők azt javasolják, hogy a ***LIDS*** foltot mindig az eredeti kernelforrásra, és nem valamilyen terjesztés-specifikus forrásra alkalmazzuk. Utóbbi esetben különböző fordítási hibákba ütközhetünk, mivel a legtöbb terjesztés valamilyen testreszabott kernelt tartalmaz.
Ezen kívül a ***LIDS***-nek közismerten gondjai vannak a nem ***i386*** alapú architektúrákkal.
Nézzük tehát a telepítés lépéseit.
A például a ***lids-2.2.1-2.6.13.tar.gz*** nevű csomagban található változatot a névnek megfelelően a 2.6.13-as kernelforrásra kell alkalmazni.
Az első lépés a foltozás:

```
patch -p1
➥/dir_to_the_patch_file/
➥patch-lids-2.2.1-2.6.13
```

Ha ezzel megvagyunk, a

```
make [x/menu]config
```

paranccsal végezzük el a szükséges beállításokat. Ne felejtsük el a biztonsági szakaszban engedélyezni a ***LIDS*** használatát. Ha ezzel is készen vagyunk, nincs más hátra, mint lefordítani az új kernelt a következő parancsokkal:

```
make
make modules_install
```

A második lépésre természetesen csak akkor van szükség, ha modulba tettünk bizonyos funkciókat.
Másoljuk át a `bzImage` nevű fájlt ***/kernelpath/arch/i386/boot*** könyvtárból saját rendszerünk ***/boot*** könyvtárába, majd konfiguráljuk újra a rendszertöltőt. A következő újraindításnál már a ***LIDS***-szel megerősített kernel fog futni.

A ***LIDS*** állapotáról a

```
lidsadm -v
```

paranccsal kérhetünk jelentést.
Ha itt bármilyen hibaüzenetet kapunk, az azt jelenti, hogy a ***LIDS*** nem épült be a rendszermagba.

Ilyenkor mindenképpen újra kell fordítani a kernelt, persze csak az után, hogy kiküszöböltük a hibát.

A hozzáférési jogosultságok beállítása

Mielőtt hozzálátnánk a különböző szerveralkalmazásokkal kapcsolatos jogosultságok beállításához, nem árt megismerni a `lidsconf` parancs használatának általános szintaxisát:

```
lidsconf -A [-s subject]
➥-o object [-d] [-t from-to]
➥[-i level] -j ACTION
```

A `subject` mezőbe egy program neve kerül, amellyel kapcsolatban valamilyen képességet akarunk beállítani. Az `object` helyén felbukkanhat valamilyen bináris fájl, könyvtár, socket neve, vagy egy képesség.
A `-d` kapcsoló hatására a ***LIDS*** a megadott tartományt végrehajtási tartományként fogja kezelni.
Az idővel kapcsolatos korlátozások beállítására a `-t` kapcsoló szolgál, míg a `-i` az öröklődés szintjét határozza meg.
A `-j` kapcsoló után egy műveletet kell megadni, ami a a következő közül az egyik lehet:

- `DENY`: Megtagadjuk egy adott objektumhoz való hozzáférést.
- `READONLY`: Csak olvashatóvá teszi a kérdéses objektumot.
- `APPEND`: Általában naplózási célra használatos. Azt teszi lehetővé, hogy egy program hozzáfűzzön bejegyzéseket egy bizonyos fájlhoz, de ne törölhesse azt.
- `WRITE`: Bináris állományoknak engedélyezi, hogy írjanak a fájlba.
- `GRANT`: Egy képességgel együtt használatos. Jelentése az adott képesség engedélyezése.
- `IGNORE` és `DISABLE`: Az elsővel egy adott objektumra vonatkozó adott engedélyt lehet visszavonni, a másodikkal egy kiterjesztést lehet hatályon kívül helyezni.

A ***LIDS*** által támogatott képességekről a

```
lidsadm -h | grep CAP
```

paranccsal kérhetünk listát.

A jelenleg támogatott képességek a következők:

- `CAP_CHOWN`: chown/chgrp.
- `CAP_DAC_OVERRIDE`: DAC hozzáférés.
- `CAP_DAC_READ_SEARCH`: DAC olvasás.
- `CAP_FOWNER`: tulajdonos azonosítója, nem azonos a felhasználóval.
- `ID CAP_FSETID`: effektív felhasználói azonosító, nem azonos a tulajdonossal.
- `ID CAP_KILL`: valódi/effektív ID, nem azonos a folyamat-azonosítóval.
- `ID CAP_SETGID`: `set*gid`(2).
- `CAP_SETUID`: `set*uid`(2).
- `CAP_SETPCAP`: átviteli képesség (transfer capability).
- `CAP_LINUX_IMMUTABLE`: változtathatatlan (immutable) és hozzáfűzési fájlattribútum.
- `CAP_NET_BIND_SERVICE`: kapcsolódás 1024-es alatti portokhoz.
- `CAP_NET_BROADCAST`: üzenetszórás (broadcasting) illetve többszörös üzenetszórás (multicast) vétele.
- `CAP_NET_ADMIN`: interfész/tűzfal/útválasztási szabályok változtatása.
- `CAP_NET_RAW`: nyers (raw) socket-ek.
- `CAP_IPC_LOCK`: megosztott memóriaszegmensek zárolása.
- `CAP_IPC_OWNER`: IPC-tulajdonjog ellenőrzése.
- `CAP_SYS_MODULE`: kernelmodulok beillesztése és eltávolítása.
- `CAP_SYS_RAWIO`: `ioperm`(2)/`iopl`(2) hozzáférés.
- `CAP_SYS_CHROOT`: `chroot`(2).
- `CAP_SYS_PTRACE`: `ptrace`(2).
- `CAP_SYS_PACCT`: folyamatok kezelésének beállítása.
- `CAP_SYS_ADMIN`: számos adminisztratív képesség.
- `CAP_SYS_BOOT`: `reboot`(2).
- `CAP_SYS_NICE`: `nice`(2).
- `CAP_SYS_RESOURCE`: erőforrás-határok beállítása.
- `CAP_SYS_TIME`: rendszeridő beállítása.
- `CAP_SYS_TTY_CONFIG`: tty beállítása.
- `CAP_MKNOD`: `mknod` műveletek.
- `CAP_LEASE`: fájlok „bérlése” (lease).
- `CAP_HIDDEN`: rejtett folyamat.
- `CAP_KILL_PROTECTED`: védett programok kilövése.
- `CAP_PROTECTED`: folyamat védelme jelektől.

LIDS segítségével biztosított kiszolgáló felállítása

A továbbiakban feltételezzük, hogy az olvasónak sikerült feltelepítenie mind magát a ***LIDS*** rendszert, mind a vele kapcsolatos adminisztratív eszközöket. Egy viszonylag szoros biztonsági hálóval védett rendszert fogunk felállítani, amelyen egyedül a ***MySQL***, az ***Apache*** és a ***Bind*** futhat.
A beállításoknál feltételezzük, hogy az Apache webkiszolgálót a */usr/local/apache* helyre telepítettük, illetve hogy a naplófájlok a */var/log/httpd* könyvtárban találhatók. Szintén feltesszük, hogy az Apache beállítóállományait tartalmazó könyvtár a */etc/httpd*. Hasonló feltevésekkel élünk a ***MySQL*** rendszerrel kapcsolatban is. A telepítési könyvtárnak a */usr/local/mysql* helyet fogjuk tekinteni. Ha az olvasó rendszerének nem ugyanez a szerkezete, akkor az itt felsorolt parancsokat természetesen megfelelően módosítani kell.
A rendszer teljes biztosításához szükséges dolgok tárgyalása természetesen messze túlmutat e cikk keretein. Ugyanakkor a bemutatott beállítások jó alapot nyújtanak ahhoz, hogy használatban vegyük a ***LIDS*** rendszert.

A rendszer felállítása

Miután elindítottuk a ***LIDS***-szel kiegészített rendszermagot, elkezdhetjük megadni azokat a biztonsági beállításokat, amelyek a rendszer egyes részeihez való hozzáférést szabályozzák. A következő parancsokkal a */sbin*, */bin*, */usr/bin* és a */lib* könyvtárakat tehetjük csak olvashatóvá:

```
lidsconf -A -o /sbin -j
➥READONLY
lidsconf -A -o /bin -j READONLY
lidsconf -A -o /usr/bin -j
➥READONLY
lidsconf -A -o /lib -j READONLY
```

A következő lépésben megadunk néhány hasonló beállítást a */opt*, */etc*

és */usr/local/etc* könyvtárakkal kapcsolatban is, amelyeknek szintén csak olvashatónak kell lenniük. A */etc/shadow* fájlhoz, és a rendszertöltőt tartalmazó fájlokhoz mindennemű hozzáférést megtagadunk:

```
lidsconf -A -o /etc -j READONLY
lidsconf -A -o /usr/local/etc
➥-j READONLY
lidsconf -A -o /etc/shadow -j
➥DENY
lidsconf -A -o /etc/lilo.conf
➥-j DENY
```

Mivel a */etc/shadow*-hoz immár egyetlen folyamat sem férhet hozzá, ez egyben a bejelentkezéseket is megátolja, hiszen a rendszer nem tudja hitelesíteni a felhasználókat. Éppen ezért csak olvasási hozzáférést kell adnunk erre a fájlra a `login` és a `vlock` programoknak. Emellett a `su` parancsnak szintén olvasási joggal kell rendelkeznie a */etc/shadow*-val kapcsolatban, tehát engedélyezzük ezt is:

```
lidsconf -A -s /bin/login -o
➥/etc/shadow -j READONLY
lidsconf -A -s /usr/bin/vlock
➥-o /etc/shadow -j
➥READONLY
lidsconf -A -s /bin/su -o
➥/etc/shadow -j
➥READONLY
```

Ami a `su` használhatóságát illeti, be kell állítanunk még néhány hozzáférési jogosultságot ahhoz, hogy kezelni tudja az *UID* és *GID* értékeket:

```
lidsconf -A -s /bin/su -o
➥CAP_SETUID -j GRANT
lidsconf -A -s /bin/su -o
➥CAP_SETGID -j GRANT
lidsconf -A -s /bin/su -o
➥/etc/shadow -j
➥READONLY
```

A következő lépésben engedélyeznünk kell az `init` és `login` programoknak, valamint az ezekkel kapcsolatos folyamatoknak, hogy írjanak a naplófájlokba:

```
lidsconf -A -o /var/log -j
➥APPEND
lidsconf -A -s /bin/login -o
➥/var/log/wtmp -j WRITE
lidsconf -A -s /bin/login -o
➥/var/log/lastlog -j WRITE
lidsconf -A -s /sbin/init -o
➥/var/log/wtmp -j WRITE
lidsconf -A -s /sbin/init -o
➥/var/log/lastlog -j WRITE
lidsconf -A -s /sbin/halt -o
➥/var/log/wtmp -j WRITE
lidsconf -A -s /sbin/halt -o
➥/var/log/lastlog -j WRITE
lidsconf -A -s /etc/rc.d/
➥rc.sysinit -o /var/log/wtmp
➥-i 1 -j WRITE
lidsconf -A -s
➥/etc/rc.d/rc.sysinit -o
➥/var/log/lastlog -i 1
➥-j WRITE
```

Ismét egy fontos lépés: meg kell adnunk a root felhasználó saját könyvtárával kapcsolatos jogosultságokat. Az egyetlen művelet, amit engedélyezünk az, hogy a *Bash* hozzáfűzhessen bejegyzéseket a `history` fájlhoz:

```
f -A -o /root -j READONLY
lidsconf -A -s /bin/bash -o
➥/root/.bash_history -j APPEND
```

Végezetül engedélyezzük az `init` folyamatnak, hogy leállításkor kilője a folyamatokat:

```
lidsconf -A -s /sbin/init -o
➥CAP_INIT_KILL -j GRANT
lidsconf -A -s /sbin/init -o
➥CAP_KILL -j GRANT
```

Ezen kívül engedélyeznünk kell még az `fstab`-nak és az `init` szkripteknek, hogy fájlrendszereket csatoljanak be, folyamatokat állítsanak le, illetve fájlrendszereket távolítsanak el:

```
lidsconf -A -s/etc/fstab -o
➥CAP_SYS_ADMIN -j 1 -j GRANT
lidsconf -A -s /etc/rc.d/
➥init.d/halt -o CAP_INIT_KILL
➥-i 1 -j GRANT
lidsconf -A -s /etc/rc.d/
➥init.d/halt -o CAP_KILL -i 1
➥-j GRANT
lidsconf -A -s /etc/rc.d/
➥init.d/halt -o CAP_NET_ADMIN
➥-i 1 -j GRANT
lidsconf -A -s /etc/rc.d/
➥init.d/halt -o CAP_SYS_ADMIN
➥-i 1 -j GRANT
```

Az Apache webkiszolgálóval kapcsolatos jogosultságok beállítása

Az *Apache*-nek *setuid* és *setgid* képességekkel kell rendelkeznie ahhoz, hogy működni tudjon. Szintén hozzá kell férni a naplófájlokhoz, más binárisoknak pedig meg kell tiltanunk, hogy a `httpd` bináris állományt módosíthassák:

```
lidsconf -A -s /usr/local/
➥apache/bin/httpd -o
➥CAP_SETUID -j GRANT
lidsconf -A -s /usr/local/
➥apache/bin/httpd -o
➥CAP_SETGID -j GRANT
lidsconf -A -o /etc/httpd -j
➥DENY
lidsconf -A -s /usr/local/
➥apache/bin/httpd -o
➥/etc/httpd -j READONLY
lidsconf -A -o /usr/local/
➥apache -j DENY
lidsconf -A -s /usr/local/
➥apache/bin/httpd -o /usr/
➥local/apache -j READONLY
lidsconf -A -o /var/log/httpd
➥-j DENY
lidsconf -A -s /usr/local/
➥apache/bin/httpd -o
➥/var/log/httpd -j APPEND
lidsconf -A -s /usr/local/
➥apache/bin/httpd -o /usr/
➥local/apache/logs -j WRITE
```

MySQL

Ami a *MySQL*-t illeti, itt is meg kell tiltanunk minden más bináris állománynak, hogy a `mysql` programot módosíthassa. Korlátoznunk kell a *mysql/var* könyvtárhoz való hozzáférést is úgy, hogy ahhoz legfeljebb hozzáfűzési joga legyen a folyamatoknak, magának a `mysqld` démonnak pedig csak olvasási jogot szabad adni a *mysql* könyvtárra:

```
lidsconf -A -o /usr/local/
➥mysql/var -j APPEND
lidsconf -A -o /usr/local/mysql
➥-j DENY
lidsconf -A -s /usr/local/
➥mysql/libexec/mysqld -o
➥/usr/local/mysql -j READONLY
lidsconf -A -s /usr/local/
➥mysql/libexec/mysqld -o
➥/usr/local/mysql/var -j
➥WRITE
```

Bind

A *Bind* rendszernek számos képességgel kell rendelkeznie ahhoz, hogy megfelelően tudjon működni:

```
lidsconf -A -s /usr/sbin/named
➥-o CAP_NET_BIND_SERVICE 53
➥-j GRANT
lidsconf -A -s /usr/sbin/named
➥-o CAP_SETPCAP -j GRANT
lidsconf -A -s /usr/sbin/named
➥-o CAP_SYS_CHROOT -j GRANT
lidsconf -A -s /usr/sbin/named
➥-o CAP_SYS_RESOURCE -j GRANT
lidsconf -A -s /usr/sbin/named
➥-o CAP_SETUID -j GRANT
lidsconf -A -s /usr/sbin/named
➥-o CAP_SETGID -j GRANT
```

Login

A `login` az a program, amely a felhasználók bejelentkezését kezeli egy *GNU/Linux* rendszeren. Ennek megfelelően a következő beállításokat kell vele kapcsolatban megadnunk:

```
lidsconf -A -s /bin/login -o
➥/etc/shadow -j READONLY
lidsconf -A -s /bin/
➥login -o CAP_SETUID -j
➥GRANT
lidsconf -A -s /bin/login -o
➥CAP_SETGID -j GRANT
lidsconf -A -s /bin/login -o
➥CAP_CHOWN -j GRANT
lidsconf -A -s /bin/login -o
➥CAP_FSETID -j GRANT
```

A fenti beállítások elvégzése után le kell zárnunk a kernelt ahhoz, hogy a *LIDS* érvényesülni tudjon. Ehhez adjuk hozzá a következő sort a *rc.local* fájlhoz:

```
lidsadm -I
```

Indítsuk újra a gépet, hogy a beállítások érvénybe lépjenek. Fontos megjegyezni, hogy a bemutatott beállításokkal nem fogjuk tudni futtatni az *X* kiszolgálót sem, mivel az nyers I/O műveleteket használ. Ez persze nem különösebben nagy gond, hiszen a legtöbb kiszolgálónak szánt gépen eleve nem fut az X. ha mégis szükségünk lenne a grafikus felületre, akkor egészítsük ki az eddigi beállításokat a következővel (feltéve, hogy az *X* kiszolgáló bináris állomány a */usr/X11R6/bin/startx*):

```
lidsconf -A -s /usr/X11R6/
➥bin/startx
```

A bemutatott példákból talán érzékelhet, hogy a *LIDS* egy igen hatékony biztonsági kiegészítés, amely akár magától a root felhasználótól is képes megvédeni a rendszert. Mellesleg a használata sem különösebben bonyolult.

Linux Journal 2006. 143. szám

Ifran Habib informatikushallgató a Pakisztáni Nemzeti Tudományos és Műszaki Egyetemen. Középiskolás kora óta erősen érdeklődik a Linux és a nyílt forrású technológiák iránt, a beágyazott rendszerektől egészen a webszolgáltatásokig. Az elmúlt két évben igyekezett népszerűsíteni a Linuxot Pakisztánban, és számos cikket is írt a témáról helyi újságokba és magazinokba.