Fájlrendszer-címkézés a SELinuxban

A SELinux sokkal kifinomultabban vezéreli a fájlokhoz való hozzáférést. Lássuk!

Most, hogy az ***NSA Security Enhanced Linux*** (***Fokozott Biztonságú Linux***) része lett a 2.6-os rendszermagnak, és így eljut a különböző ***Linux*** terjesztésekbe, valószínűleg egyre többen telepítik majd a ***SELinuxot***, és kezdenek kísérletezni vele. Ezeknek a jövőbeni felhasználóknak kíván segítséget nyújtani ez a cikk, közelebbről is megvizsgálja a SELinux alatti fájlrendszer-címkézést. Bár ez az írás alapvetően a középhaladóknak szól, a következő bekezdésben rövid áttekintést adok a SELinux néhány alapelvéről, a hálózaton található forrásokban pedig szerepel néhány hivatkozás a részletesebb információk forrásaira. ***Faye Coker*** bemutató cikke (***Linux Journal, 2003. augusztus***) különösen ajánlott azoknak, akik csak most kezdenek ismerkedni ezzel a különleges területtel.

SELinux áttekintés: címkézés és hozzáférés-szabályozás

A ***SELinuxban*** valamennyi, biztonsági szempontból fontos objektumhoz, például a feladatokhoz, a fájlleírókhoz és magukhoz a fájlokhoz egy-egy biztonsági környezet tartozik, vagyis egy-egy olyan címke, amely magában foglalja az adott objektumhoz kapcsolódó biztonsági jellemzőket. A szabványos ***SELinuxban*** ez egy kettősponttal elválasztott karakterlánc, amely azonosító, szerep és típus értékekből áll. Ezeket a címkéket egy ***biztonsági kiszolgáló*** nevű rendszermag-összetevő rendeli hozzá a megfelelő objektumokhoz, mégpedig a biztonsági házirend adatbázisában megadott szabályok alapján. A szerző munkaállomásán található */etc/shadow* fájl biztonsági környezeti címkéje például a következő:

```
system_u:object_r:shadow_t
```

A `system_u` és az `object_r` a fájlokban használt általános azonosító és szerep értékek, míg a `shadow_t` a fájl típusa. Ez utóbbi egy olyan jellemző, amely meghatározza, hogyan lehet elérni az adott fájlt. Egy folyamatot a következőképpen lehet címkézni:

```
root:staff_r:staff_t
```

A ***SELinux*** azonosító itt `root`, amelyet a ***SELinux*** a szabványos ***UNIX*** azonosítónál tartósabb azonosítófajtaként rendel hozzá. A szerep itt `staff_r`, ami azt jelöli, hogy a folyamat az ehhez a szerephez rendelt összes jogosultsággal rendelkezik. A folyamathoz kapcsolt típus a `staff_t`. A folyamatoknál a típus jellemző meghatározza, hogy azok hogyan férhetnek hozzá az objektumokhoz, és hogyan léphetnek kapcsolatba más objektumokkal. A folyamatok típus jellemzőjét gyakran tartományként is emlegetik.

Hozzáférés-szabályozási döntések

A ***SELinux*** a biztonsági környezeti címkék segítségével a folyamatok és az objektumok közötti kapcsolatrendszert határozza meg. A rendszer működése közben folyamatosan döntéseket hoz azzal kapcsolatban, hogy melyik folyamat melyik objektumhoz férhet hozzá, és pontosan hogyan. De hogyan történik a gyakorlatban mindez? A ***SELinux*** „kapaszkodókat" (hooks) helyez el az alapvető rendszermagkód stratégiai pontjain, például ott, ahol a felhasználó éppen olvasni kezdi a fájlt. Ezek a kapaszkodók lehetővé teszik a ***SELinux*** számára, hogy felfüggessze a rendszermag rendes adatforgalmát, és kifinomult döntéseket kényszeríthessen ki a hozzáférések szabályozása terén a megfelelő kérésekkel. A hozzáférés-szabályozási döntések általában folyamatok (például `cat`) és objektumok (például */etc/shadow*) között történnek, meghatározott jogosultságok megszerzése végett. (Példánknál maradva a `cat`-nek nyilván olvasási jogosultságot kellene szereznie az adott fájlra.). A döntéskérések a hozzáférés vektortárba (***Access Vector Cache*** – ***AVC***) kerülnek, amely átadja a kéréseket a biztonsági kiszolgálónak, kiértékelésre. A biztonsági kiszolgáló a kapott adatokat egyezteti a biztonsági házirend adatbázis tartalmával, majd meghatároz egy eredményt, amely az ***AVC***-ben tárolódik, és visszatér a megfelelő ***SELinux*** kapaszkodóhoz. A ***SELinux*** kapaszkodó ezután vagy engedélyezi a forgalom folytatását, vagy egy `EACCES` értéket ad vissza, a döntés eredményétől függően. A folyamatokhoz és az objektumokhoz rendelt biztonsági környezeti címkék ezeknek a döntéseknek a meghozatalát támogatják. Az *1. ábrán* ennek a folyamatnak egy egyszerűsített változatát láthatjuk.
A következő kód azt szemlélteti, hogy a biztonsági környezeti címkék hogyan működnek egy valódi rendszeren, illetve mit lát maga a felhasználó az egész folyamatból:

```
$ id -Z
root:staff_r:staff_t

$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

```
Az ellenőrzési napló a következőt rögzíti:
avc:  denied  { read } for  pid=13653 exe=/bin/cat
name=shadow dev=hda6 ino=1361441
scontext=root:staff_r:staff_t
tcontext=system_u:object_r:shadow_t tclass=file
```

A `cat` program, amelynek a biztonsági környezeti címkéje esetünkben `root:staff_r:staff_t`, nem kapta meg az engedélyt egy `system_u:object_r:shadow_t` címkéjű fájl olvasására.

A ***SELinuxnak*** természetesen halvány fogalma sincs arról, hogy mi az a `cat` vagy mit tartalmaz a ***/etc/shadow*** fájl, őt egyszerűen csak a megfelelő biztonsági környezeti címkék, a célobjektum (ebben az esetben a fájl) osztálya és az éppen kérelmezett engedély típusa érdeklik. Az ezekkel megadott információ számára tökéletesen elegendő a döntés meghozatalához.

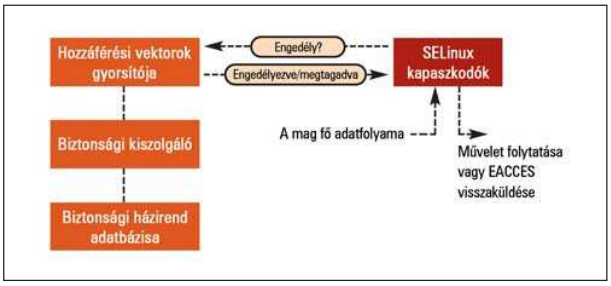
A ***SELinux*** tervezésében fontos szempont, hogy a címkék az objektumok összes biztonsági jellemzőit magukban foglalják. A rendszermagban a biztonsági kiszolgáló, a felhasználói térben pedig a ***libselinux*** értékeli ki azokat a megfelelő pillanatokban. Maga a rendszermagkód és a felhasználói tér többi része semmi egyebet nem tesz, csupán számára ismeretlen adatként továbbadja a címkéket. Ennek a rendszernek az egyik óriási előnye az, hogy a címkéket anélkül lehet újabb biztonsági jellemzőkkel kiegészíteni, hogy ehhez újra kellene fordítani az alkalmazásokat vagy újratervezni az alapvető ***SELinux*** kódot.

Kiterjesztett jellemzők

Egy jellegzetes lemez alapú ***Linux*** fájlrendszerben minden fájlt egyedileg azonosít egy olyan fájlleíró, amely a fájl kulcsfontosságú adatait tartalmazza, beleértve a ***UNIX*** tulajdonjogot és a hozzáférési adatokat. Amikor a rendszermag egy fájlra hivatkozik, akkor a lemezről a memóriába olvassa ezt a fájlleírót. A ***UNIX*** szabványos jogosultságellenőrzése egyszerűen a fájlleíróban szereplő adatokat használja. A ***SELinux*** tulajdonképpen ezt a szabványos ***UNIX*** biztonsági protokollt terjeszti ki, és biztonsági környezeti címkéket használ a finomított hozzáférés-szabályozási döntések meghozatalára.

Maga a ***Linux*** megvalósítja a kiterjesztett jellemzőket (***Extended Attributes*** – ***EAs***), más néven ***EAs*** vagy ***xattrs***. Ezek a név/érték párok fájlokhoz rendeltek, akár a kiterjesztések a hagyományos fájlleíró alapú jellemzőkhöz. A kiterjesztett jellemzők segítségével úgy lehet szabványosított módon szolgáltatásokat adni a fájlrendszerekhez, hogy a jellemzők felületei fájlrendszerektől függetlenek legyenek. A kiterjesztett jellemzőkkel kapcsolatos szolgáltatások közé tartoznak például a hozzáférés-szabályozási listák (***Access Control Lists*** – ***ACL***), a karakterkészletleíró-adatok tárolása a fájladatok mellett és a ***SELinux*** biztonsági környezet címkézés.

A kiterjesztett jellemzők a névtereken belül tárolódnak, lehetővé téve ezzel a különböző osztályokba tartozó kiterjesztett jellemzők elkülönített kezelését. A hozzáférésszabályozási listák a `system.posix_acl_access` és a `system.posix_acl_default` névterekben kapnak helyet. A ***SELinux*** biztonsági környezeti címkéi a `security.selinux namespace` névtérben találhatóak.



1. ábra Az SELinux hozzáférés-szabályozási döntései a rendszermagban

Akit részletesebben is érdekelnek a ***Linux*** kiterjesztett biztonsági jellemzői az olvassa el az `attr(5)` súgóoldalt.

Kiterjesztett jellemzők biztonsági címkézése

A kiterjesztett jellemzők természetesen „kézzel” is módosíthatók a `getfattr(1)` és a `setfattr(1)` segédprogramokkal. Ha például látni akarjuk egy fájl ***SELinux*** biztonsági környezeti címkéjét, akkor a következőt kell tennünk:

```
$ getfattr -n security.selinux /tmp/foo
getfattr: Removing leading ´/´ from absolute
➥path names
# file: tmp/foo
security.selinux="root:object_r:sysadm_tmp_t\000"
```

Figyeljük meg a kiterjesztett jellemző biztonsági névterének leírását! A rendszer használatát megkönnyítendő a ***SELinux***-hoz kapunk egy `getfilecon(1)` nevű burkolóprogramot is, ami bizonyos helyzetekben nagyon megkönnyítheti az életet. Segítségével ugyanis nem nekünk kell meghatározni a kiterjesztett jellemző névteret, és tisztább eredményt is ad.

A szöveg alapú címkék használata biztosítja, hogy értelmes, olvasható biztonsági jellemzők tárolódjanak a fájladatokkal. Ezek a címkék változatlanok maradhatnak vagy lefordíthatók, ha a fájlrendszer egy másik, valószínűleg más biztonsági rendet alkalmazó rendszerre fűződik be. Ennek ellenpéldája, ahogy a fájl tulajdonosa felhasználói számazonosítóként tárolódik a fájl leírójában. A felhasználói azonosító jellemzően egy értelmes értékhez van hozzárendelve az ***/etc/passwd*** fájlon keresztül – egy másik rendszerben más jelentése lehet.

Ahhoz, hogy egy fájlrendszer támogassa a ***SELinux*** biztonsági környezet címkéket, kiterjesztett jellemző támogatásra és kiterjesztett jellemző biztonsági névtér kezelőre van szüksége. Jelenleg az ilyen fájlrendszerek közé tartozik az ***ext3***, az ***ext2***, az ***XFS*** és a ***ReiserFS*** – ez utóbbi külső foltot alkalmaz. Ráadásul a ***devpts*** fájlrendszer hamis biztonsági kezelővel rendelkezik, amely megengedi a kiterjesztett jellemző alapú hozzáférést a ***pty***-k rendszermagban található címkéihez.

Tehát, mikor kerülnek címkézésre a fájlok? A ***SELinux*** rendszer telepítése során, a `setfiles(8)` segédprogram jellemzően arra szolgál, hogy a fájlrendszer összes olyan fájlját megcímkézze, amely támogatja a kiterjesztett jellemző biztonsági címkézést. Az olyan csomagkezelő eszközök, mint az ***RPM***, szintén megcímkézhetik a fájlokat a telepítés so-

A SELinux fájlrendszer-címkézés fejlődése

A ***SELinux*** első változata 2000-ben más elvet használt a fájlrendszerek címkézésére, mint amit az ebben a cikkben tárgyalt kiterjesztett jellemzők megközelítés használ. Az állandó biztonsági azonosítók (***Persistent Security IDs*** – ***PSIDs***) a biztonsági környezet címkék egész megvalósításai az ext2 fájlleíró használaton kívüli mezőjében tárolódtak. A fájlok leképezését mindkét fájlrendszerben arra használta a ***SELinux***, hogy megkeresse a fájlok ***PSID***-jét a fájlleíró alapján, és leképezze a biztonsági környezet címkére. Ennek a megközelítésnek az volt a hátránya, hogy minden fájlrendszert külön módosítani kellett, hogy támogassa a ***PSID***-ket. Ezért a kiterjesztett biztonságra nem volt jó általános megoldás az árral szemben haladó rendszermagban. Az ***LSM*** projekttel egy általánosított hozzáférés-szabályozási váz valósult meg a ***Linux*** rendszermagban. Mivel az ***LSM***-ben nincsenek fájlrendszerre jellemző kapaszkodók, a ***SELinux*** eltávolodott a módosított fájlrendszer megközelítéstől, és a ***PSID***-ket hagyományos fájlban tárolta, a leképező fájlok mellett. Ez lehetővé tette a ***SELinux*** számára, hogy tisztán ***LSM*** alkalmazásként legyen használható, rendszermag-foltozás nélkül. Ahhoz is hozzájárult, hogy a címkézés több fájlrendszeren működjön, de teljesítmény és következetesség szempontjából nem volt optimális. A fájlok rendszermagból való elérése továbbra is gondot okoz. A ***SELinux*** fősodorbeli rendszermagba olvadásának folyamataként, a közösségtől kapott több visszajelzés alapján a ***SELinux*** a jelenlegi kiterjesztett jellemzőkre épülő fájlrendszer-címkéző modellre váltott. A kiterjesztett jellemzők szabványos alkalmazásprogramozói felülettel rendelkező alkalmazásokat biztosítanak, a fájlrendszereket pedig hasonlóképpen más biztonsági modellek is használhatják, még ugyanazon a fájlrendszeren belül is, a kiterjesztett jellemző névterek által biztosított elkülönítés alkalmazásával.

rán, a rendszergazdáknak viszont gyakran kézzel kell beállítaniuk a biztonsági környezetet, a `chcon(1)` vagy a `setfilecon(1)` segédprogrammal.

Fájlok létrehozása

Amikor létrejön egy fájl, a biztonsági rend egy megfelelő szabálya általában leírja, hogyan kell címkét hozzárendelni a szülőkönyvtár és az aktuális feladat biztonsági környezete alapján. Íme egy példa:

```
$ id -Z
root:staff_r:staff_t
$ ls -dZ /tmp
drwxrwxrwt+ root  root system_u:object_r:tmp_t
/tmp
$ touch /tmp/hello
$ getfilecon /tmp/hello
/tmp/hello       root:object_r:staff_tmp_t
```

Ebben az esetben a biztonsági rend egy olyan szabályt tartalmaz, amely megszabja, hogy a `staff_t` által a `tmp_t` könyvtárban létrehozott fájlokat a `staff_tmp_t` típus címkével kell ellátni. Ha nincs egyértelmű szabály, akkor a fájlokat a szülőkönyvtár környezetével kell megcímkézni.
A kitüntetett alkalmazások felülbírálhatják az imént említett szabályt, úgy, hogy biztonsági környezetet írnak a */proc/self/attr/fscreate* fájlhoz. Ezután ez a biztonsági környezet szolgál az összes újonnan létrehozott fájl megcímkézésére. A `setfscreation(3)` könyvtárfüggvény magában foglalja ezt a lehetőséget. Ahhoz, hogy kézzel visszaállítsunk egy biztonsági környezetet, használjuk a `restorecon(8)`-t.
Az olyan helyzetek kezelésére, amikor címke nélküli fájlok keletkeznek, fejlesztés alatt áll egy `fsck`-szerű segédprogram. Ez az indításkor futtatandó alkalmazás biztosítja majd, hogy minden fájl helyesen címkézett, mielőtt több felhasználós üzemmódba lép.

Viselkedésmódok címkézése

Az előző részben az olyan fájlrendszerekben való fájlcímkézésről volt szó, amelyek támogatják a lemezen található kiterjesztett jellemzőket, és rendelkeznek kiterjesztett jellemző biztonsági névtér kezelővel. Amikor egy ilyen fájlrendszer helyesen van befűzve, akkor úgy mondják, hogy a `xattr` címkéző viselkedésmódot alkalmazza.
Amikor a ***SELinux*** kezdeti értéket ad egy fájlrendszernek, például a befűzés során, a következő naplóüzenet keletkezik:

```
SELinux: initialized (dev hda6, type ext3), uses
xattr
```

A `uses xattr` záradék azt jelenti, hogy a fájlrendszer a fent leírt `xattr` címkéző viselkedésmódot alkalmazza. Sok fájlrendszer nem támogatja a kiterjesztett jellemzőket, és azok közül, amelyek támogatják, sem mind rendelkezik biztonsági névtér kezelőkkel. A lemezes fájlrendszereknél lehet, hogy még senki sem végezte el a kódolási munkát, vagy egyszerűen a kiterjesztett jellemzőknek nincs értelme a *vfat*-hez hasonló öröklött fájlrendszereken.
A pszeudo-fájlrendszerek burjánzása a Linux alatt kezdődött. A fájlrendszerek egyre kedveltebb felhasználó-rendszermag alkalmazásprogramozói felületté válnak. Ezek közül a ***procfs*** a legszembetűnőbb, amely a felhasználói tér és különböző rendszermag-összetevők közötti felület. A ***procfs*** hosszú történetének köszönhetően rengeteg hulladékot halmozott fel, és az új felhasználó-rendszermag alkalmazásprogramozói felületeket arra ösztönzi, hogy különböző fájlrendszerekként valósuljanak meg. Ezek a fájlrendszerek a rendszermagra épülnek, és nincs valódi kiterjesztett jellemző támogatás. Ilyen például az ***usbfs***, a ***sysfs*** és a ***selinuxfs***.
Az ilyen kiterjesztett jellemzőket nem támogató rendszerek különböző címkézési viselkedésmódokat használnak, az egyes fájlrendszer-fajták biztonsági rendjéhez igazodva.
Az átmeneti ***SIDs*** címkézési viselkedésmód a ***devpts***, ***tmpfs*** és ***shmfs*** fájlrendszereknél használatos. Az ezekben a fájlrendszerekben található fájlok igény szerint a rendszermagban kerülnek címkézésre, az adott feladat és az érvényben lévő fájlrendszer biztonsági környezete alapján.
A ***devpts*** egy különleges átmeneti ***SIDs*** fájlrendszer. Kiterjesztett jellemző alkalmazásprogramozói felület elérést biz-

tosít a *pty*-khez egy álkiterjesztett jellemző biztonsági kezelőn keresztül. A kitüntetett alkalmazások, mint amilyen az `sshd` ezt a szolgáltatást a *pty*-k újracímkézésére használja, az átmeneti *SID* címkék felülbírálásával.
A *SIDs* címkézési viselkedésmód feladat egyszerűen megcímkézi az aktuális feladattal azonos biztonsági környezetben lévő fájlt. A *pipefs* és a *sockfs* fájlrendszerben létrehozott csővezetékek illetve csatlakozópontok esetében használatos.
A `genf_contexts` címkézési viselkedési módot olyan fájlrendszereknél használják, amelyek alkalmatlanok az *xattr*, átmeneti *SIDs* és feladat *SIDs* címkézésre. A biztonsági rendben a biztonsági környezet címkék fájlrendszer/elérési út párokhoz vannak rendelve. Az elérési út összetevő célja, hogy lehetővé tegye a fájlrendszer finomabb léptékű címkézését. Ez a szolgáltatás különösen a *procfs*-nél fontos, amely olvasható és írható rendszermag-adatok összevisszasága, beleértve a *systcl* felületet.
A legtöbb nem kiterjesztett jellemzős fájlrendszer a `genf_contexts` címkézést használja, általában úgy, hogy az egész fájlrendszer egyetlen biztonsági környezetre van állítva. Ennek gyakori példája a *sysfs*, a *vfat*, az *nfs* és az *usbdevfs*.

Befűzési pont címkézése

A 2.6.3-mas rendszermagba újonnan került szolgáltatás a befűzési pont címkézése, vagy másként a környezetbefűzés. Ennek az a fő célja, hogy lehetővé tegye, hogy az egész fájlrendszer biztonsági környezetét meg lehessen határozni egy befűzési beállítással. A befűzési pont címkézése bármilyen fájlrendszerre megvalósítható, és felülbírálja a rendes címkézési viselkedésmódot.
A befűzési pont címkézésének egy meghatározott felhasználási módja annak lehetővé tétele, hogy a különböző *NFS* befűzések külön-külön lehessen címkézni, befűzési időben. Olyan fájlrendszerek általános esetenkénti címkézéséhez is hasznos, amelyek nem támogatják a kiterjesztett jellemző biztonsági címkézést, valamint máshol címkézett kiterjesztett jellemző címkézésű fájlrendszerek címkézésére. Ez utóbbi például a törvényszéki munkában lehet hasznos.
A címke nélküli öröklött fájlrendszereket is lehet, hogy *SELinuxra* alkalmas operációs rendszereken kell befűzni. Ugyan a fájlrendszertípus támogatja a kiterjesztett jellemző biztonsági címkézést, esetleg nem akarunk maradandó biztonsági környezet címkéket adni a fájlrendszerekhez. A befűzési pont címkézése segítségével rendszermag alapú címkéket rendelhetünk hozzá, amelyek nem íródnak a lemezre.
Mivel a befűzési pont címkézése új szolgáltatás, és nem elég alaposan dokumentált, vegyük egy kicsit részletesebben. Amikor a *SELinux* engedélyezett a rendszermagban, három új lehetőség válik elérhetővé a befűzési pont címkézése számára:

- `context`: A fájlrendszer összes fájlját és magát a fájlrendszert a meghatározott biztonsági környezet szerint címkézi. A */proc/self/attr/fscreate* alkalmazásprogramozói felületet, amelyet fent taglaltunk, a fájlrendszer figyelmen kívül hagyja. Ez felülbírálja a meglévő címkézési viselkedésmódot, és a befűzési pont címkézésére változtatja. Ezzel a lehetőséggel a fájlrendszer címkéi a felhasználó számára kizárólag olvashatóak, annak ellenére, hogy a házirend által meghatározott átmenetek még működnek azokon a fájlrendszereken, amelyek támogatják a kiterjesztett jellemző biztonsági címkézést.
- `fscontext`: A teljes fájlrendszer címkéjét (azaz a fájlrendszert magát) beállítja a meghatározott biztonsági környezetre állítja. Ez lehetővé teszi a fájlrendszerek finomabb léptékű vezérlését, azáltal, hogy hozzájárul, hogy a címkéket a befűzés alapján lehessen beállítani, a házirendben meghatározott fájlrendszer alapú beállítás helyett. Mivel a `context` lehetőség is ezt a működést valósítja meg, nem lehet együtt használni a két lehetőséget. Ez a beállítás csak olyan fájlrendszereken működik, amelyek támogatják a kiterjesztett jellemző biztonsági címkézést. A teljes fájlrendszer biztonsági környezetei egy adott fájlrendszeren belül, a fájlok létrehozása során hozott hozzáférés-szabályozási döntéseknél, fájlrendszerek befűzésekor és leválasztásakor, fájlrendszer-jellemzőkhöz való hozzáféréskor és maga a fájlrendszer újracímkézésekor használatosak.
- `defcontext`: Beállítja a címkézetlen fájlok alapértelmezett biztonsági környezetét a házirendben meghatározott érték helyett. Ahogy az `fscontext` lehetőségnél is,

Biztonsági mentés és visszaállítás

A *SELinuxot* használó rendszergazdák számára számos változó feladat egyike a biztonsági mentés és visszaállítás. Amikor archívumot hozunk létre, hogyan őrződnek meg a biztonsági környezet címkék az archívumon belül? A válasz az igen rugalmas `star(1)` segédprogram használata, amely kiterjesztett jellemző támogatással rendelkezik.
Az archívumok biztonsági környezet címkékkel való befolyásolásához használjuk az `xattr` parancsot. Archívumok létrehozásakor meg kell határozni az `exustar` formátumot. Például:

```
$ star -xattr -H=exustar -c -f cups-log.star
➥/var/log/cups
```

létrehozza a */var/log/cups* könyvtár archívumát, a fájlokban megőrizve a biztonsági környezet címkéket.
A kicsomagoláshoz egyszerűen használjuk a `xattr` kapcsolót:

```
$ star -xattr -x -f cups-log.star
$ ls -Z var/log/cups/
-rw-r--r--+ root     sys
➥system_u:object_r:cupsd_log_t error_log
-rw-r--r--+ root     sys
➥system_u:object_r:cupsd_log_t error_log.1
```

Mint látható, a biztonsági környezet címkék megmaradtak.

csak olyan fájlrendszerekkel működik, amelyek támogatják a kiterjesztett jellemző címkézést, és érvénytelen, ha a `context` lehetőség van meghatározva, mivel az is ezt a működést valósítja meg.

A rendszermagban a ***SELinux*** a `mount(2)` alatt elemzi és kiiktatja a biztonsági befűzési beállításokat, rendes beállításokat átadva a fájlrendszerfüggő kódon keresztül. A hagyományos fájlrendszereknél nem kell ügyelni a biztonsági beállításokra, így nincs szükség a módosításokra. Ez azért lehetséges, mert a legtöbb fájlrendszer általában név/érték párokat használ befűzési lehetőségként, amelyet a ***SELinux*** könnyen befolyásolhat.
A bináris befűzési beállítás adatokkal rendelkező fájlrendszerek esetében, amilyen a ***NFS***, az ***SMBFS***, az ***AFS*** és a ***Coda***, különleges módon kell eljárni. Ezek közül csak az ***NFSv3*** támogatott a ***SELinux*** fejlesztésének mostani szakaszában.
Íme egy példa, a `context` lehetőség működésére, mivel valószínű, hogy a három befűzési lehetőség közül ez a leggyakrabban használt. Egy naplófájlokat tartalmazó hajlékony lemez érkezett az asztalunkra, és szeretnénk befűzni a ***SELinux*** gépre, néhány elemző programot futtatni rajta. Amiatt, ahogy a házirend meg van határozva, a fájlokat a `system_u:object_r:var_log_t` címkével kell ellátni, hogy a naplóelemző program megfelelően működjön. Ha ezzel a módszerrel fűzünk be, elkülöníthetjük az adatot a lemezen, lehetővé téve a ***SELinuxnak***, hogy megvédje egymástól az operációs rendszert és a lemez tartalmát.
Fűzzük be a lemezt:

```
$ mount -v -t vfat
➥-o context=system_u:object_r:var_log_t
➥/dev/fd0 /mnt/floppy
/dev/fd0 on /mnt/floppy type vfat
(context=system_u:object_r:var_log_t)
Mit mond az ellenőrzési napló?
SELinux: initialized (dev fd0, type vfat), uses
mountpoint labeling
```

Ez az üzenet ígéretesnek tűnik. Következőnek ellenőrizzük, hogy a lemez fájljai úgy vannak címkézve, ahogy gondoltuk. Rendszerint a `getfilecon(1)` hívást alkalmaznánk, de a `getfattr(1)` egyértelműbb hibaüzeneteket ad:

```
$ getfattr -n security.selinux
/mnt/floppy/access_log
/mnt/floppy/access_log: security.selinux:
➥Operation not supported
```

Mi történik? Egy `ls -Z` parancs is megmutatja, hogy a fájl üres biztonsági környezettel rendelkezik:

```
$ ls -Z /mnt/floppy/access_log
-rwxr-xr-x+  root  root  (null)
➥/mnt/floppy/access_log
```

A hajlékonylemezen található ***vfat*** fájlrendszer nem rendelkezik kiterjesztett jellemző támogatással, és a biztonsági környezet címkézése tisztán a rendszermagon belül történik. Kiderül, hogy a rendszermagon belüli címkézés jól működik, de a felhasználói tér eszközei nem képesek megjeleníteni a kiterjesztett jellemző alkalmazásprogramozói felületen található címkéket. Ez az aktuális kiterjesztett jellemző megvalósítás korlátja, amelyet még elegánsan meg kell oldani.
Van viszont egy trükkös módja, hogy megnézzük, mik a fájlok címkéi – az ellenőrzési napló alkalmazásával, amely hozzáférési üzenetek naplózásakor mindig rögzíti a célobjektum biztonsági környezetét.
A `getfattr(1)` használata a következő ellenőrzési bejegyzést idézte elő:

```
avc:  denied  { getattr } for  pid=12354
exe=/usr/bin/getfattr
path=/mnt/floppy/access_log dev=fd0 ino=132
scontext=root:staff_r:staff_t
tcontext=system_u:object_r:var_log_t tclass=file
```

Tehát a fájl címkézése helyes (`system_u:object_r:var_log_t`), amely a `context` befűzési lehetőségen keresztül került a befűzés parancshoz.

Jövőbeni munka

Most is lehet ugyan biztonsági környezeti címkéket rendelni az ***NFS***-sel befűzött fájlrendszerekhez, de azok csak helyben működnek, a rendszermagon belüli hozzáférés-szabályozási döntéseknél. A címkék nem kerülnek átvitelre a hálózaton a fájlokkal együtt. Ezen a területen fejlődés tapasztalható az ***NFSv2/v3*** protokollokon és kódon történt ***SELinuxra*** szabott módosításoknak köszönhetően. Hosszabb távon az ***NFSv4***-be várhatóan bekerül a hálózati címkézés nevesített jellemzőkön keresztül, amelyek részei a kiterjedtebb ***NFSv4*** leírásnak. Ez lehetővé tenni, hogy az ***NFS*** ügyfél is és a kiszolgáló is megvalósítsa a ***SELinux*** biztonságot a hálózati fájlokon. Más hálózati fájlrendszerek számára szintén hasznos lenne a támogatás, ahogy a ***Trusted BSD SELinux*** kapujával való együttműködés is.

Linux Journal 2004. október, 126. szám

James Morris (jmorris@redhat.com) egy Sydney-i ausztrál rendszermag fejlesztő, aki jelenleg a Red Hat-nek dolgozik, Bostonban. Rendszermag-karbantartó a SELinuxban, a hálózati és titkosító API-ban, valamint LSM fejlesztő és az Emeritus Netfilter Core csapat tagja.