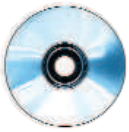

NSA fokozott biztonságú Linux

Rendszered kötelező, finomhangolt hozzáférés-beállításaival azt is korlátozni tudod, hogy a rendszergazda mit tehet.

© Kiskapu Kft. Minden jog fenntartva

Az NSA fokozott biztonságú Linux gyökerei a korábban megjelent, nagy bizalomnak örvendő operációs rendszerig (DTOS) és a Flaskig (Flux advanced security kernel – Flux fokozott biztonsági rendszermag) nyúlnak vissza. A DTOS projekt az amerikai National Security Agency (NSA) és a Secure Computing Corporation (SCC) 1990-es évek elejiközepi együttműködésének az eredménye. A cél egy, a standard biztonsági módszereknél erősebb biztonsági eljárásokkal bíró operációs rendszer készítése volt. A Flask-szerkezet az NSA, az SCC és a University of Utah Flux projektjének a végeredménye, amelyet „kibővítettek, hogy jobban megfeleljen a változó biztonsági előírásoknak" (***Stephen Smalley*** „Flask: Flux Advanced Security Kernel", NAI Labs, 2000. december 26.; ➲ http://www.cs.utah.edu/flux/flask).

Az SE Linux megköveteli a hozzáférés-beállítást (Mandatory Access Control – MAC), míg egy átlagos Unix-rendszer tetszés szerinti hozzáférés-beállítással dolgozik (Discretionary Access Control – DAC). A DAC esetében a felhasználó szabja meg, hogy az egyes elemeknek mekkora hozzáférésük legyen az egyes területeken. Egy Unix-rendszeren például alkalmazhatjuk a `chmod` parancsot a saját könyvtárrendszerünk engedélyeinek megváltoztatáshoz. A MAC esetében a hozzáférés-beállítást egy magasabb szinten lévő felhasználó szabja meg, ő határozza meg a biztonsági beállításokat, amelyek megadják az egyes elemek jogait. Ha létezik egy olyan szabály, amelyik megakadályozza, hogy Bob megnézze Alice otthoni fájljait, és Alice kiadja a `chmod 777` utasítást az otthoni könyvtárában, Bob ennek ellenére sem tudja megnézni őket.

A MAC-et használva a programok minimális jogosultságokkal futnak, egy „megtámadott" folyamat nem tud hozzáférést adni a saját forrásaihoz olyan más folyamatok számára, amelyeknek ez eredetileg nincs engedélyezve. Ez mérsékli a károsodás mértékét abban az esetben, ha egy démon megsérülne. A biztonsági beállítások számos tényezőn múlnak, például a felhasználó jogosultságain, a futtatott program jellegén, azon, hogy mennyire megbízható a program, és milyen a kezelt adatok titkossági szintje vagy épsége.

Mi az SE Linux?

Az SE Linux egy rugalmas, finomhangolt hozzáférés-beállítás a Linux-rendszermagban, amely jelenleg az LSM-vázat alkalmazza (lásd Linux Journal, 2002. november, ***Greg Kroah-Hartman*** „Using the Kernel Security Module Interface"). A jelenlegi felhasználáskor az LSM-csatoló csak a korlátozó hozzáférés-beállítást támogatja, ezért ha a Unix-engedélyek leállítanak egy műveletet, azt az SE Linux sem engedélyezheti. Az SE Linuxot általában további korlátozások életbe léptetésére használják olyan rendszereken, ahol Unix-engedélyek élnek, és ez magától is meglehetősen jól tudja a szükséges hozzáférés-beállításokat. Az SE Linux egy rendszermagfoltból és más segédprogramokhoz tartozó foltok áll, ilyen például a `login` és `cron`. Az NSA felel a hivatalos változatokért. Néhány, nem az NSA-ban dolgozó személy is hozzájárult kódokkal a projekthez. A csomagokat folyamatosan fejlesztik a Debian üzembiztos és fejlesztői változataihoz. Mivel az SE Linux a GPL felhasználási szerződés hatálya alá esik, bárki hozzáírhat és saját módosításokat készíthet. Az SE Linuxot 2.4.19-es és frissebb rendszermagokon lehet használni, és e cikk írásakor – 2003 májusában – a 2.5-ös rendszermagra is fejlesztik.

Miért van szükség a módosított alkalmazásokra?

Mint azt már korábban említettük, az SE Linux egy rendszermagfoltból és módosított segédprogramokból áll. A módosított segédprogramok teszik lehetővé, hogy a rendszer minden fájlja helyes biztonsági tartalommal bírjon. A segédprogramoknak módosított változatai is léteznek, például a `login`, a `cron` és a `logrotate`, valamint olyan programok, mint a `ps` és az `ls`. A `login` esetében például létfontosságú a helyes biztonsági környezet, amikor felhasználó bejelentkezik a rendszerbe. Ha ez nem él, akkor elképzelhető, hogy a felhasználó egyáltalán nem tud belépni. A `login` csomag telepítését a „Hogyan kezdjük?" fejezet írja le az SE Linux HOWTO-jában (lásd a ***Kapcsolódó címeket***), és ez nem is tartozik e cikk keretei közé. Ha azonban az SE Linux telepítésekor elfelejtjük felrakni a `login` csomagot, ez azt eredményezi, hogy nem lesz megfelelő kód kiadva a terminálnak (terminal device), ahol az újraindítást követően bejelentkezne, és ez lehetetlenné teszi a bejelentkezést. A módosítatlan `login` program szintén futtat egy héjprogramot a biztonsági környezetben, amely letiltja a fájlok hozzáférését a felhasználó saját könyvtárában. Például a `login` és `cron` foltjai megmondják a rendszermagnak, hogy melyik biztonsági környezetet alkalmazza. Ezek tényleges végrehajtását a rendszermag végzi. A címkézés létfontosságú, ezért van szükség néhány módosított programra. Mindenki elkészítheti a saját biztonsági irányelveit, amelyek a módosított csomagok telepítése nélkül is alapszintű védettséget biztosítanak, de az alapbeállítás kifinomultabb biztonsági rendszert eredményez.

Gyakran használt kifejezések

Az alábbi kifejezésekkel mindig találkozhatunk, ha az SE Linux leírását vagy a levelezőlisták hozzászólásait olvassuk. Fontos, hogy ezekkel minél jobban megismerkedjünk, még mielőtt telepíteni próbálnánk az SE Linuxot, ezáltal ugyanis a későbbi munkánkat is megkönnyítjük.

- Tartomány
  Egy tartomány részletezi, hogy az egyes folyamatok mit tehetnek meg, és mit nem, illetve azt, hogy egy folyamat milyen műveleteket hajthat végre eltérő típusokon. Ha a `felhasznalo_t` tartományban vagyunk (megkötések nélküli felhasználó tartomány) és a `ps aux` parancsot futtatjuk, akkor csak a `felhasznalo_t` tartományban futó programokat látjuk. Egyéb példák a tartományra: a `sysadm_t`, a rendszergazdai tartomány és az `init_t`, amely az a tartomány, amelyben az `init` fut. A `passwd_t` tartományban a `passwd` programot egy jogosultságokkal nem rendelkező felhasználó futtathatja.

© Kiskapu Kft. Minden jog fenntartva

- Jogosultságok
  A jogosultság határozza meg, hogy mely tartományokat lehet használni. Az irányelveket tartalmazó adatbázis (policy database) határozza meg az egyes felhasználók által hozzáférhető tartományokat. Ha egy adott felhasználónak nincs engedélye (a beállításfájlban) belépni egy tartományba, akkor letiltják. Két példa a jogosultságokra: az átlagos, jogosultságokkal nem rendelkező felhasználói szerep (`felhasznalo_r`) és a rendszergazdai jogosultság (`sysadm_r`).
  Nézzük a következő példát: ahhoz, hogy egy felhasználó a `felhasznalo_t` tartomány alól egy `passwd` parancsot hajtson végre, a `felhasznalo_r` jogosultságú felhasználónak kell begépelni: `passwd_t`; – ez a vonatkozó beállításfájlban van megadva. Ezenkívül más tartományátmeneti szabályokat is fel kell állítani, ezek bővebb ismertetéséről azonban terjedelmi okokból le kell mondanunk. Ez a hozzáadott kód szabja meg, hogy egy felhasználó „felhasználói” (`user_r`) jogosultságokkal elérhesse a `passwd_t` tartományt ahhoz, hogy lefuttassa a `passwd` parancsot. Más kérdés az, hogy vajon a régi tartománynak van-e jogosultsága átlépni az új tartományba.
  Most, hogy meghatároztuk a tartományokat és a szabályokat, összehasonlíthatjuk az SE Linuxot és a Unix `uid`-ot (felhasználói ID). Ha a rendszergazda ural egy programot, amelynek az Unix engedélye 4777 (ez a programot „setuid” rendszergazdává teszi), akkor a rendszer bármelyik felhasználója végre tudja hajtani azt a programot, biztonsági hibát okozva ezzel. Az SE Linuxon azonban ha egy művelet átlép egy védett tartományt, és ha a művelet jogosultsága nem engedélyezett azon a tartományon, akkor a program nem fog futni. Az SE Linux-rendszereken minden művelet egy olyan tartományon fut, amelyik megszabja, hogy az egyes műveletek milyen hozzáférési joggal rendelkeznek.
- Azonosító
  Az SE Linux alatt az azonosító nem ugyanazt jelenti, mint a hagyományos Unix `uid`, amit az olvasók talán ismernek is. Az SE Linux alatt az azonosítók annak a biztonsági környezetnek részei, amelyek azt irányítják, hogy mit tehet meg az ember, és mit nem. Elképzelhető, hogy egy SE Linux azonosítónak és egy hagyományos Unix bejelentkezési névnek ugyanaz a szöveges megjelenítése (és legtöbbször a könnyebbség kedvéért ez igaz is), de fontos megértenünk, hogy két különböző dologról van szó. Alapértelmezés szerint ugyanazok, ha a kérdéses SE Linux azonosító létezik. Ebből kifolyólag, ha `faye` felhasználóként jelentkezem be az SE Linux-rendszerbe, és az irányelv-adatbázisban szerepel `faye`, akkor az én műveleteimhez a `faye` azonosítót rendelik.
  Vegyük például a `su` parancsot a Unix-felhasználó ID-k és az SE Linux azonosítók különbségeit bemutatandó.
  A `su` futtatása nem változtatja meg a felhasználói azonosítót az SE Linux alatt, de megváltoztatja az `uid`-ot, ahogy azt a nem SE Linux-rendszeren is tenné. Ha a `faye` felhasználó az SE Linux-rendszeren gépelné be, hogy `su` ahhoz, hogy rendszergazda lehessen, majd lefuttatná az `id` parancsot, amely kiadja az ő biztonsági környezetét és egyéb adatait, akkor láthatná, hogy az azonosítója továbbra is `faye`, nem pedig root, az `uid` azonban megváltozott.
  Hogy ezt jobban érzékeltessem: ha egy átlagos felhasználó a `faye` loginnal futtatja az `id` parancsot, akkor a következő biztonsági környezetet látja:

```
uid=1000(faye) gid=1000(faye)
groups=1000(faye)
context=faye:felhasználó_r:felhasználó_t
sid=45
```

  Ebben az esetben a biztonsági környezet azonosító része `faye`. Láthatjuk, hogy az `uid` 1000. Tegyük fel, hogy `faye` ezután beüt egy `su` parancsot, és ismét futtatja az `id` parancsot. Most ezt látja:

```
uid=0(root) gid=0(root) groups=0(root)
context=faye:felhasználó_r:felhasználó_t
sid=453
```

  A várttól eltérően az azonosító nem változott rendszergazdára, de az `uid` 0-ra változott. Ha azonban a `faye` felhasználónak engedélye van a rendszergazda jogosultság alkalmazására vagy `sysadm_r`-re, akkor ezt megteheti: a konzolnál történő bejelentkezéskor megadja, hogy ő szuperfelhasználó kíván lenni, vagy a `newrole -r` parancsot adja ki – erről még szólunk a továbbiakban. Ha ezt követően újból futtatja az `id` parancsot, ezt látja majd:

```
context=3Dfaye:sysadm_r:sysadm_t
```

  Így az azonosító ismét ugyanaz marad, de a jogosultság és a tartomány (a második és harmadik mező) megváltozik. Az azonosító ilyen módon történő megtartása ott hasznos, ahol a felhasználó megbízhatósága megkövetelhető. Ez létfontosságú a rendszer biztonsága szempontjából is, mivel a felhasználó azonosítója határozza meg, hogy mely szerepeket és tartományokat lehet használni. Egy átlagos Unixon, ha az embernek van egy `setuid` vagy egy `setgid` programja, amely nem mindenki álltal futtatható, ennek elindíthatóságát nemcsak a bejelentkezett felhasználó engedélyei szabják meg, hanem az is, hogy mely felhasználó felé végeztünk legutoljára egy `su` keresést. Ez a megkötés nem létezik az SE Linux alatt, mivel az az egyén azonosítóját minden művelet végrehajtásakor rögzíti. Ha egy tartományon nincs engedélyünk setuid/setgid program futtatására, akkor nem fogjuk tudni futtatni, mégha egy `su`-val a rendszergazdává váltunk. A jogosultságod határozza meg, hogy mely tartományba van engedélyed belépni, az általad megadható jogosultságokat pedig az azonosítód szabja meg. Ebből kifolyólag az azonosító közvetve irányítja az általad használható tartományok listáját.
- Típus
  Minden tárgy egy típus alá van sorolva, ez a típus határozza meg, hogy mi férhet hozzá az adott tárgyhoz.
  A tárgyak jelen esetben fájlok, könyvtárak, foglalatok (sockets) és más műveletek. Egy típus elgondolásában a tartományhoz hasonlít, a különbség az, hogy a tartomány csak a műveletekre vonatkozik. Másként fogalmazva: a tartomány egy olyan típus, amelyik a műveletekre vonatkozik.
- Átmenet
  Egy lekért művelet során megváltozó biztonsági környezet átmenetére vonatkozik. Az átmenet két csoport egyikébe tartozik. Az első csoport a tartományműveletek átmenete. Amikor egy adott programot futtatunk, az átmenet indulhat a jelenlegi tartományból egy új tartományművelet felé. Ennek bemutatására a `newrole` parancsot használom.

A `newrole` parancs a szereped megváltoztatását szolgálja, mondjuk a `felhasznalo_r`-ről a `sysadm_r`-re – feltéve, hogy `sysadm_r`-ként van hozzáférésed. Ha `felhasznalo_r`-ként indulsz (ez az átlagos, jogosultságokkal nem rendelkező felhasználó), és lefuttatsz egy `newrole -r sysadm_r`-t, hogy megváltoztasd a `sysadm_r`-t, a rendszergazdai szerepet, átmenetet képezel a `felhasznalo_t` tartományból a `newrole_t` felé (ez az a tartomány, ahol az új jogosultság művelet futni fog), és onnan pedig a `sysadm_t` tartomány felé.
A második átmeneti kategória a fájltípusok átmeneti jellege, amikor fájlt nyitunk egy adott könyvtár alatt. Ha egy felhasználó egy fájlt hoz létre a saját könyvtárában, akkor annak a fájlnak a címkéje `user_home_t` lesz. De ha ugyanez a felhasználó egy fájlt a */tmp*-ben hoz létre, akkor az a fájl a `user_tmp_t` címkét kapja. A `user_tmp_t` a */tmp* típusból lett elvonva, amely a `tmp_t`, és abból a tartományból, ahol a fájlt létrehozták, ez a `user_t`. Amikor a felhasználó létrehoz egy fájlt a */tmp* alatt, átmenet zajlik le a `user_tmp_t` típus felé.

- Irányelvek
  Egy irányelv határozza meg, hogy mely típusok mely műveleteket hajthatják végre az egyes tartományokban. Minden démonnak megvan a saját irányelve és a névadási szabály az ***démon-name.te***, ***postfix.te***, ***apache.te*** stb. Egy SE Linuxot futtató gép rendszergazdájaként tetszés szerint szerkesztheted az irányelv fájlokat. Az irányelv-adatbázis az irányelvek forrásfájljainak az összesített listája, amit a rendszermag tölt be indításkor.

SE Linux rendszeren a `spasswd` programot használjuk a jelszó megváltoztatására. A `spasswd` gyakorlatilag a burkolója (wrapper) a Linux-rendszeren használt `passwd`-nek, ez biztosítja, hogy a `passwd` program a helyes tartományon fusson. Ugyancsak ez biztosítja azt is, hogy az SE Linux-azonosítód megegyezzen az egyébkénti Unix-azonosítóddal. Korábban említettem, hogy az átlagos Unix-felhasználó ID-k sokban különböznek az SE Linux-azonosítóktól, de akkor most, amikor a `spasswd`-t futtatom, miért kell egyezniük? A `spasswd` követeli meg, hogy az SE Linux-azonosítóneved megegyezzen a Unix-azonosítóddal. Emlékezz csak, az SE Linux-rendszeren az azonosítód az egyetlen módja annak azonosítására, hogy ki is vagy. Tehát ha legutóbb nem te voltál a megadott Unix-felhasználó, akkor nem tudod megváltoztatni a jelszót.
Ha rendszergazda vagy (`sysadm_r`), egy másik felhasználó jelszavának a megváltoztatásra a `sadminpasswd` programot használhatod. A `sadminpasswd` esetében nem él ez az azonos felhasználónév/azonosító megkötés, mint a `spasswd`-nél, de a `sadminpasswd` csak `sysadm_t`-ként futtatható.

Megengedő és megkötő módok

Az SE Linux kétféle üzemmódban futtatható: megengedőben vagy megkötőben. A megengedő módot hibakeresés céljából használjuk, mivel ott minden naplózódik, de az SE Linux valójában nem köti magát az irányelveidhez. A felhasználó továbbra is tud rendszergazdaként végrehajtani dolgokat, csakúgy, mint egy átlagos Linux-rendszeren. Amíg meggyőződsz róla, hogy minden irányelv kielégítően működik, legjobb a gépet megengedő módban futtatni. Címkéket adunk a rendszerben található tárgyaknak, de semmit sem kényszerítünk ki. A megkötő mód a már megadott irányelveket használja, például a hozzáférés-beállításokat. Csak akkor indítsd a gépedet megkötő módban, ha meggyőződtél róla, hogy az rendesen működik, és már egy ideje rendesen futott megengedő módban. Ne feledd, hogy ha a rendszermagba nincs befordítva a fejlesztői támogatás, akkor itt nem futtahatod megengedő módban a rendszert. Ha a rendszermagban be van kapcsolva fejlesztői támogatás, akkor a gép megengedő üzemmódban indul, és kézzel kell megkötő módra kapcsolnod. Ezt könnyen megteheted egy indító parancsfájl segítségével. A másik megoldás, hogy kiépítesz egy ilyen utat:

```
/etc/rc.boot/avc and /sbin/
avc_toggle
```

Újabb megoldás: írd be a rendszermag parancssorába, hogy `enforcing=1`. Az `avc_toggle` parancsot használhatod a megengedő és a megkötő mód közti változtatásra, és `avc_enforcing` parancsot használhatod annak megállapítására, hogy megkötő módban vagy-e éppen.

Hogyan tovább?

E cikk olvasása után remélhetőleg kedvet kaptál az SE Linux kipróbálásához. Szándékosan hagytam ki a telepítési útmutatót, mivel RPM-ek, az úgynevezett source tarballs vagy Debian-csomagok segítségével is telepíthetsz. Az alapvető lépések leírása egy külön cikket is megtöltene. Egy sereg új dolgot meg lehet tanulni a telepítés előtt, közben és után, és az új felhasználók gyakran összezavarodnak. De ha – még mielőtt bármibe is belefognál – elolvasod a ***Kapcsolódó címek*** részben megnevezett dokumentumokat és megismerkedsz a gyakran használt fogalmakkal, az egészet valamivel könnyebbnek találod majd. Ha elakadsz, izzítsd be kedvenc IRC-ügyfeledet, és menj a #selinux csatornára a ➲ http://irc.debian.org alatt, vagy iratkozz fel az SE Linux-levelezőlistára.

Linux Journal 2003. augusztus, 112. szám

Faye Coker
Jelenleg szabadúszó rendszergazda. Gyakran futtat ISPs-rendszereket és állít át kiszolgálókat Linuxra. Dolgozott Európában és Ausztráliában. A kelleténél többször kérdezik meg tőle Linux-konferenciákon, hogy „eltévedtél?”.

KAPCSOLÓDÓ CÍMEK

Flask (Flux Advanced Security Kernel)
➲ www.cs.utah.edu/flux/flask
Getting Started with SE Linux HOWTO
➲ sourceforge.net/docman/display_doc.php?docid=15285&group_id=21266
NSA Official SE Linux oldal ➲ www.nsa.gov/selinux
NSA SE Linux GYK ➲ www.nsa.gov/selinux/faq.html
NSA SE Linux White Papers
➲ www.nsa.gov/selinux/docs.html
SE Linux levelezőlista ➲ www.nsa.gov/selinux/list.html
SE Linux archívum ➲ marc.theaimsgroup.com/?l=selinux
SourceForge SE Linux Project oldal
➲ http:/sourceforge.net/projects/selinux

© Kiskapu Kft. Minden jog fenntartva