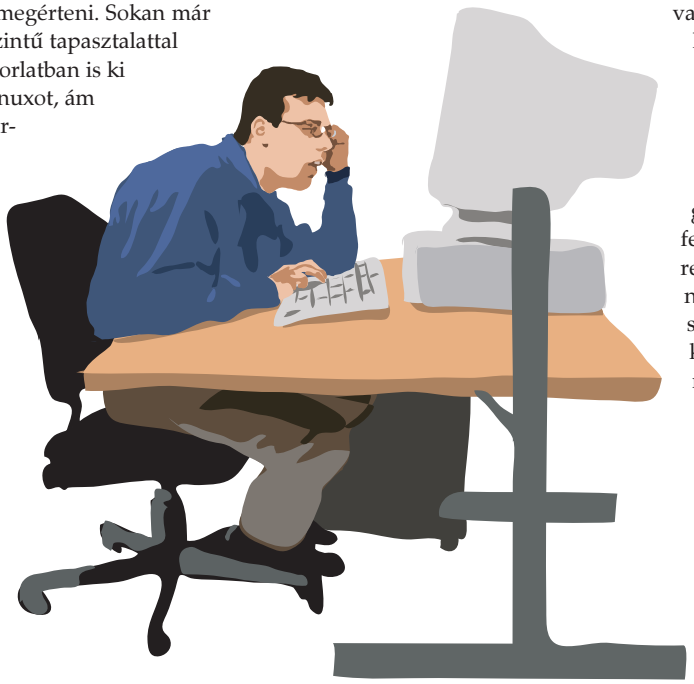
Bárki lehet rendszergazda!

Az SE Linux elkülönített védelmi rétege még a rendszergazdától is megvédi a rendszert. Russell betekintést nyújt a megoldás működésébe, és mindenkit rendszergazdává tesz.

Az NSA Security Enhanced Linuxon 2001 közepe óta dolgoztam, a debianos csomagok elkészítésében és az általános fejlesztésben egyaránt közreműködtem. Amikor Linux-felhasználóknak próbálom ecsetelni a tervezet lényegét, rengeteg félreértéssel szembesülök az SE Linux lényegét illetően. Az SE Linux valódi mibenlétét elég nehéz pusztán a leírásának átolvasásával vagy egy bemutató megtekintésével megérteni. Sokan már rendelkeznek valamilyen szintű tapasztalattal a biztonság terén, és a gyakorlatban is ki szeretnék próbálni az SE Linuxot, ám nincs idejük arra, hogy kísérletezés céljából feltelepítsék. Úgy véltem, a legjobb módszer az SE Linux népszerűsítésére az, ha üzembe helyezek egy bárki által szabadon használható gépet.



Az SE Linuxot normál összeállításon bemutatni nem túl izgalmas dolog, hiszen mindössze a `ps ax` és a `dmesg` az a két parancs, amelynél érzékelhető korlátozás van. Alapesetben a `ps ax` a jogosultság nélküli felhasználónak csak az azonos felhasználói tartományon belül futó folyamatokat mutatja meg, a `dmesg` pedig nem érhető el. Ezeket a korlátozásokat az `OpenWall` lépteti érvénybe, és önmagukban semmi újdonsággal nem szolgálnak. Úgy döntöttem, hogy a világ összes felhasználójának rendszergazdai hozzáférést adok, miközben a védelmet kizárólag az SE Linuxra terhelem, így képességeit a látogatók pontosan felmérhetik.

A 2002. június 6–9. között Karlsruhében, Németországban megrendezett LinuxTagon egy SE Linux bemutatógéppel jelentem meg a Debian pultjánál. Ez volt az első SE Linux „játékgép”. Akkoriban az alapértelmezett házirend még kevésbé volt szigorú. A `setuid` és a `DAC_OVERRIDE` használatát normál felhasználóknak is megengedte (`user_t` tartomány). Átlagos SE Linux-összeállításnál ezzel nincs is gond. Az SE Linux nem használ `uid` hívásokat a jogosultságok odaítélésekor, és ugyan a `DAC_OVERRIDE` lehetővé teszi a Unix hozzáférés-vezérlés felülbírálását, ám az SE vezérlésekét nem. Ez a két lehetőség azért volt elérhető, hogy a `setuidos` programok SE Linux-tartományok nélkül, az `user_t` tartományból is futtathatók legyenek. Az átlagos felhasználónak ez így meg is felelt, de nem akkor, amikor a rendszergazdától kellett megtagadni az azonos tartományban lévő más `uid`-ek elérését. A `user_t`-ből tehát kiszedtem ezeket a lehetőségeket, a rendszergazdai fiókot a `user_r` szerepre korlátoztam, és már kezdődhetett is a mulatság.

Az újabb kiadásokban az alapértelmezett házirend szerint a `setuid` vagy a `DAC_OVERRIDE` használatának lehetőségét a `user_t` nem kapja meg. Egykori játékgépem és egy élő kiszolgáló biztonsági házirendje között tehát az a legfontosabb különbség, hogy a játékgépen a jogosultság nélküli felhasználók elolvashatták a rendszermag üzeneteinek naplóját (`dmesg`) és a biztonsági házirend forrását, megkönnyítve ezzel az SE Linux megismerését.

A próbagépet szándékosan kevésbé biztonságosra állítottam be, mint ahogy egy valódi kiszolgáló esetében tettem volna: jogosultságot adtam a naplófájlok elérésére, a biztonsági házirend olvasására, valamint a jogosultság nélküli felhasználók is rendszergazdaként garázdálkodhattak. Talán ennek is köszönhető, hogy sikerült apróbb rést találni a védelem falán.

A LinuxTag első napján kihasználhatónak tűnő hiányosság jelentkezett a */boot* könyvtár állományaival. Az egyik felhasználó úgy gondolta, a LILO-térképfájlból megtudhatná a LILO jelszavát. Azonnal módosítottam a házirendet, és megtiltottam a */boot* könyvtár elérését, az ilyenfajta gondokat megakadályozandó. Természetesen, ha fizikailag sikerül hozzáférni egy géphez, akkor előbb-utóbb át lehet törni a védelmét, de ezt nyilván a lehető legjobban meg kell nehezíteni.

A rendezvény ideje alatt kezdtem el dolgozni a többféle felhasználói szerep támogatásán. Ennek eredeti oka az volt, hogy munkatársaim egyike komolyabb célokra is használta a játékgépet. Az összes fájlját elvesztette, mert azokat a `root:user_r:user_t` biztonsági környezetben, `uid root`-ként hozta létre, és a többiek is ezt használták a rendszer kipróbálására. Természetesen az `rm -rf /` paranccsal mindenki megpróbálkozott, és ezzel az ő állományai is az enyészeté lettek. Maga a rendszer nem sérült, ugyanis a */bin*, */etc* és egyéb

rendszerkönyvtárak nem választhatók le vagy írhatók a `user_t` által. Miután a barátom a `user1_t` tartományban kapott fiókot, a `user_t` tartomány rendszergazdájaként többé nem lehetett elérni a fájljait.
2002. június 17-én egy Cobalt Qube alapú SE játékgépet az interneten keresztül is – mindenki számára – elérhetővé tettem. Az első gép július 11-ig folyamatosan a hálózaton lógott. Való igaz, ez nem túl nagy eredmény, de a gépet folyamatosan figyelni kellett. Egy ilyen játékszer – ha megtörik, és én nem cselekszem azonnal – bármikor veszélyessé válhat bárki számára, beleértve engem, az internetszolgáltatómat és a használóit is. Vagyis amikor elutazom vagy nagyon leköt a munkám, akkor ki kell kapcsolnom.

A gép beállításai

A gép saját IP Tables-szabályokat kapott, megelőzendő a nemkívánatos, a helyi gépről kifelé irányuló hálózati forgalmat. Egy tűzfal mögé került, ami hasonló megszorításokat alkalmazott az adatátvitelre. Ezzel a módszerrel meg tudtam akadályozni, hogy bárki is belülről kóstolgathassa a tűzfalamat, amíg meg nem töri a játékgépet. Eleinte az SMTP-n kívül szinte minden kimenő kapcsolatot engedélyeztem, de hamarosan úgy döntöttem, csak egy webproxyra engedek adatokat kifelé. Másfajta hálózati hozzáférések céljára SSH-alagutakat lehetett használni. Megtiltottam az X továbbítását is, így ha valamelyik látogató véletlenül engedélyezte volna a saját gépén, a játékgép többi használója nem tudta volna támadni.

Mennyire volt biztonságos?

A gép alig egy napja volt fenn a hálózaton, amikor az egyik felhasználó jelezte, hogy a */etc/shadow* olvasható. A LinuxTag bemutatóján jeleztem, hogy ez a könyvtár szándékosan kívül maradt az érdeklődési körömön, de a hálózatra való csatlakoztatás előtt ki kellett javítani a hibát. A *shadow* fájl tehát `shadow_t` típusú lett, emiatt viszont módosítani kellett a `spasswd` burkoló programot és a hozzá tartozó SE házirendet. A `shadow_t` teljes körű támogatásának megvalósítása nem volt egyszerű, mivel sok esetben a programok a fájlokat újra létrehozva változtatják meg a */etc/passwd* és a */etc/shadow* fájlt, az alapértelmezett `etc_t` környezetbe helyezve őket. Jó megoldás lett volna, ha ezeket a programokat úgy módosítottam volna, hogy az `open_secure(2)` rendszerhívást használják a biztonsági környezetnek a fájl létrehozásakor való megadására. Ezt az ötletet azonban elvetettem, mivel a különféle biztonsági alkalmazásokkal rengeteget kellett volna dolgozni, megkockáztatva, hogy valamilyen hiba miatt megsérül a védelem. Ehelyett inkább írtam egy olyan burkolókódot, amely ezeknek a programoknak a futtatásában segédkezik, és a */etc/passwd* környezetét kilépésük után `etc_t`-re állítja vissza. Ezekre a programokra vonatkozóan a `shadow_t`-t választottam alapértelmezett típusnak arra az esetre, ha fájlt hoznának létre a */etc* alatt. A */etc/shadow* akár `etc_t` típust is kaphatott, a jogosulatlan rendszergazdai felhasználók nem tudtak beleírni. A `user_t` tartomány rendszergazdai felhasználói számára a fájl csak olvasható volt.
A következő napon valaki rájött, hogy a */dev/nvram* nem kapott megfelelő védelmet. Bárki tudta írni, így bármelyik felhasználó össze tudta volna zavarni a BIOS beállításait, lehetetlenné téve a gép elindítását. Akár az is előfordulhatott volna, hogy valaki olyan értékek átadására veszi rá a Qube BIOS-át a rendszermagnak, hogy a következő indításnál már legyengült védelemmel áll volna fel a rendszer. A Cobalt BIOS olyan műveleteket végez el, amelyeket más gépeken a rendszertöltő, például a LILO hajt végre. A házirend módosításával ezt a rést is azonnal betömtem. Fontos megjegyezni, hogy más géptípuson – jelentsen az másfajta processzort vagy kiépítést akár – hasonló apró módosítások végrehajtására lehet szükség a biztonsági házirendben, gondoskodva a */dev* könyvtárban lévő eszközcsomópontok védelméről.
A jelenlegi házirenddel elég kicsi a valószínűsége annak, hogy egy ilyen jellegű hiányosság gondot okozzon, mivel az eszközcsomópontokkal alapértelmezés szerint szinte semmilyen műveletet nem lehet végezni.
Néhányan aggódtak amiatt, hogy valóban jól állítottam-e be a jogosultságokat, és további megerősítést kértek afelől, hogy semmilyen jogszabályba ütköző dolgot nem művelnek, ezért a */etc/motd* tartalmát is módosítottam, biztosítva a látogatókat, hogy a gépet kifejezetten biztonsági próbák céljából csatlakoztattam a hálózatra. Kijelentettem azt is, hogy a védelmi rendszer bármilyen jellegű megtörése megengedett, akár a gép használhatatlanná válásának árán is, amennyiben az alkalmazott módszerről engem tájékoztatnak. Szintén kijelentettem, hogy a gépet nem szabad más gépek ellen irányuló támadások indítására használni, bár ezt a tűzfalszabályok segítségével is igyekeztem megakadályozni. Végül mindenkit megkértem, hogy szolgáltatásmegtagadási (DoS) támadást ne indítson a gép ellen, mert egyrészt semmi érdekes nincs benne, másrészt nem ezek kezelése a kísérlet célja.
Június 20-a után a játékgép működése meglehetősen eseménytelenül folyt. 2003 februárjában az OSDEM rendezvényen ismét megjelentem játékgépemmel a Debian pultjánál, és „szerezd meg a zászlót” versenyt hirdettem. Az érdeklődés elképesztő volt, nemegyszer akár harmincan is figyelték, ahogy valaki a védelmi rendszeren való áttöréssel próbálkozik. Egyikük sikerrel is járt, sikerült elérnie egy fájlt egy megadott nem rendszergazdai fiók alól, miután rendszergazdaként jelentkezett be. Ezt az `EDITOR` környezeti változó értékének módosításával és a `crontab -e` parancs futtatásával érte el. A `crontab` több SE-jogosultsággal futtatta a szerkesztőt, mint az normál esetben történt volna, és így tágabb hozzáférést engedett neki. Igaz, hogy ilyen módszerrel normál kiszolgálón nem lehetne eredményt elérni, hiszen az ismeretlen emberek még SE Linux alatt sem kapnak rendszergazdai hozzáférést, de azért módosítottam a `crontab` házirendjét, megelőzve a további hasonló eseteket. Azt sem szabad elfeledni, hogy a `crontab` alapú támadás egyetlen felhasználói szerepre korlátozódott. A más tartományokban lévő fiókok – például az általam a játékgép üzemeltetésére használt is – érinthetetlenek maradtak.
Az egyetlen folyamatosan fennálló gond az erőforrás-használat volt. Sokan úgy gondolták, hogy a fájlrendszer telítésével vagy az egyéb erőforrások felemésztésével el tudnak érni valamilyen eredményt. Azt hiszem, a DoS-támadásokra vonatkozó kérésem nem volt elég egyértelmű.
Ugyancsak érdekes kérdés volt: hogyan győzhetném meg a felhasználókat arról, hogy valóban rendszergazdaként használják a gépet. A GCC fent volt a gépen, és sokan saját `ps`-változatot vagy egyéb segédprogramokat hoztak, szentül meggyőződve arról, hogy valójában nem rendszergazdák, és csak valami ócska tréfát játszottam velük, módosított segédprogramokkal. Egyikük külön assembly kódot hozott a `getuid()` rendszerhívás használatára, így nyomozva az után, hogy vajon módosítottam-e a *libc6*-ot. Természetesen ő is valóban rendszergazda volt, de azért érdemes eljátszani a gondolattal: vajon hogyan kell módosítani a *libc6*-ot, hogy egy ténylegesen nem rendszergazdaként belépett személy annak is érezze magát?

Minden olvasót arra bíztatok, hogy próbálja ki. Természetesen nem mindenkit volt ilyen nehéz meggyőzni. Volt egy fehérgallérosnak tűnő figura, aki olyan gépek után kutatott, amelyekre egy támadócsomagot (rootkit) tudott felrakni. Nálam is próbálkozott, de rá kellett jönnie, hogy az őt érdeklő könyvtárakat (*/bin*, */sbin* és */etc*) és a bennük lévő fájlokat nem tudja írni. Tőlem kért támogatást a cucc telepítéséhez, de sajnos nem tudtam segíteni neki.

Hogyan végezhetünk saját biztonsági próbát vagy telepíthetünk saját próbagépet?

Ha saját biztonsági próbagépet szeretnénk összeállítani, akkor elsőként megfelelő helyet kell neki keresnünk. Ezt kimondani könnyű, véghezvinni viszont nehéz, ugyanis egy ilyen gép egy csomó hálózati pásztázást és behatolási kísérletet vonz a hálózatra. A legtöbb internetszolgáltató tiltja az ilyesmit, és jó eséllyel le fogja kapcsolni a gépet a hálózatról.

Ha a hálózati csatlakozást lerendeztük, ki kell találnunk valami jó módszert a gép hálózatról való leválasztására arra az esetre, ha valami balul sülne el. A kapcsoló közvetlen elérése például nem tűnik rossznak. A tápellátás vagy a RESET kapcsoló internetes vezérlésének lehetővé tétele is jó megoldás. Ha a távvezérlés nem kivitelezhető, akkor a próbagépet telepítsük egy felügyelhető kapcsoló (switch) külön kapujára, illetve ha ez sem oldható meg, akkor egy Netfiltert futtató linuxos géppel kössük össze keresztcsatolt (laplink) kábellel. Így pillanatok alatt megszüntethetjük a gép hálózati kapcsolatát.

Következő lépésként megfelelő vasat kell szerezni. Egy iPAQ például nem feltétlenül jó választás, mivel programból is teljesen használhatatlanná lehet tenni. Egy átlagos asztali PC nagyjából jónak tűnik. A legrosszabb esetben ki kell benne cserélni az alaplapot, ami kibírható költséget és munkát jelent. Ha esetleg sikerül ingyen szerezni egy gépet, akkor a rendszer teljes halála sem jelent különösebb megrázkódtatást, legalábbis anyagilag. Napjainkban úgyis kiváló eszközök kerülnek a szeméttelepre (az USA-ban – a szerk.).

Ha a gép alapvető beállításainak megadásán túlestünk, akkor megfelelő csomagszűréssel meg kell akadályoznunk, hogy más gépeket támadhassanak róla. Ezeknek a szabályoknak a szigorúsága az internetszolgáltatóval kötött szerződés tartalmának a függvénye. Ha ilyen jellegű hálózathasználatot a szerződés nem engedélyez – mert például otthoni célokra való széles sávú hozzáféréssel bírunk –, akkor szigorú szabályok szükségesek. Ha a szerződés lehetővé teszi kiszolgálók futtatását, akkor több dolgot engedhetünk meg, akár weboldalaknak is helyt adhatunk. Minél nagyobb szabadságot engedünk, annál érdekesebb próbákat lehet végezni. A felhasználók részéről az egyik leggyakoribb panasz az, hogy nincs elég mozgásterük a különféle próbák elvégzéséhez. A következő játékgépemen én már teljes hálózati hozzáférést szeretnék nyújtani, így a felhasználók például levelezhetnek majd a gépen, weboldalakat helyezhetnek el rajta – illetve amit még kérnek.

Tűzfalat a próbagépen és a fizikailag azonos hálózaton lévő összes gépen fel kell húzni. A próbagépen a Netfiltert értelemszerűen a csomagok csendes, naplózás nélküli elvetésére vagy visszautasítására kell beállítani, hacsak a naplók nem tarthatnak számot valakinek a kifejezett érdeklődésére. Az útválasztót úgy kell beállítani, hogy az összes eldobott csomagot naplózza, így hamar értesülhetünk arról, ha a próbagépen valaki sikeresen megkerülte a csomagszűrőt vagy a rendszer védelmét. Ha az internetszolgáltató is tud a próbagép felállításának tervéről, akkor egy egyszerű tűzfal is megteszi. Akadályozzuk meg az SMTP-kapcsolatok létrehozását, a hamisított forrás-IP-címmel küldendő csomagok továbbítását és a webes levelezőrendszerek használatát, például a Hotmailét, ami egyben a webproxyk elérésének megtiltását és a helyi webproxy megfelelő beállítását is maga után vonja.

A helyi hálózaton a próbagépen és az útválasztón kívül más gép lehetőleg ne legyen, ez ugyanis a próbagépről könnyen támadható lenne. Ha több próbagépet is sikerül ugyanarra a fizikai hálózatra kötni, az kellemes mulatság, mivel egymást lehet róluk támadni. Ha csupán egyetlen próbagépre futja, akkor keresztcsatolt ethernetkábellel vagy PPP-t futtató nullmodemmel csatlakoztathatjuk az útválasztóhoz.

A gépek összekötése és a tűzfalak beállítása után kezdődik a munka neheze. Meg kell határozni, milyen hozzáférési kört engedélyezünk, illetve a megfelelő naplózásról is gondoskodni kell. Az SE Linux esetében csak a felhasználói fájlok rendszergazdai bejegyzését kell módosítani, biztosítva a felhasználóknak a rendszergazda szerepet: `{ user_r };`.

Egy másik lehetőség a rendszergazdai bejegyzés teljes eltávolítása az adatbázisból, mivel az alapértelmezett `user_u` személyazonosság csak a `user_r` szerepben engedélyezett, és a jelszóváltoztatások megakadályozásával külön védelem biztosítható. Ha egy jogosultságok nélküli fiók jelszavát akarjuk megváltoztatni, a személyazonosságnak meg kell egyeznie a felhasználónévvel.

A változások a házirend-adatbázis újrafordításával és a rendszermagba való betöltésével léptethetők érvénybe. Ezt követően a rendszergazda már nem rendelkezik jelentősebb hozzáféréssel a rendszerhez, ezért ügyeljünk arra, hogy előbb egy másik fióknak adjunk felügyeleti jogot.

A következő próbagépem üzembe helyezésekor szeretnék jogi jártassággal rendelkező segítőt szerezni, aki átnézi a használati feltételeket, és biztosít arról, hogy a benne foglaltak egyértelmű és jogilag megtámadhatatlan módon írják le a megengedett műveletek körét. A jelszót a használati feltételekkel együtt egy weblapra fogom kihelyezni, és rendszeresen meg fogom változtatni, hogy a látogatók a feltételek elolvasása nélkül ne juthassanak be. Túl sok olyan belépő volt, aki nyilvánvalóan nem olvasta el a feltételeket, különösen a helyi – fork-bombákkal vagy éppen a merevlemez betöltésével végrehajtott – DoS-támadásokra vonatkozó részt.

Ha ilyen játékgépen futtatja valaki az SE Linuxot, akkor tegye meg azt a szívességet, hogy értesít, így írhatok róla a weblapomon.

A játékgépet használók támogatását, illetve az SE Linuxszal kapcsolatos kérdések megválaszolását a ➲ #selinux IRC-csatornán végeztem, az ➲ irc.debian.org kiszolgálón. Ha valaki ilyen biztonsági próbagépet állít össze, akár az SE Linux, akár más rendszer felhasználásával, akkor csatlakozzon a csatornához, ahol kicserélhetjük a tapasztalatainkat.

Köszönetnyilvánítás

A Sun Cobalt részlege egy RaQ kiszolgáló ajándékozásával segítette munkámat. A LinuxTag után az összes SE Linux játékgép Cobalt gépen futott.

Linux Journal 2003. augusztus, 112. szám

Russell Coker

Tíz éve használja a Linuxot. Internetszolgáltatóknál végzett Unix-rendszergazdai munkája során szembesült azzal, hogy a Unix a biztonság területén szorul a legtöbb fejlesztésre.