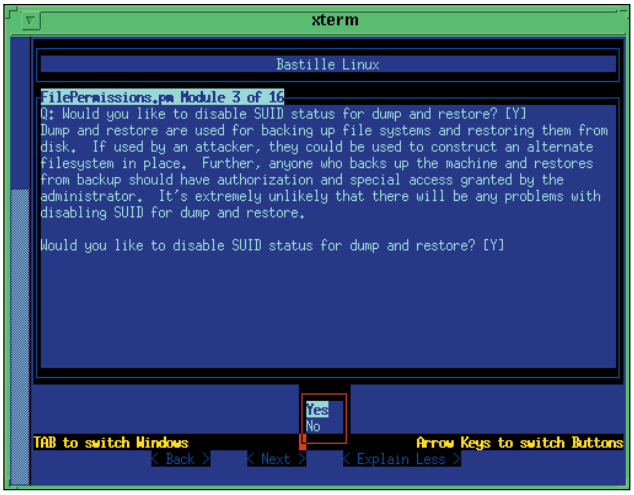
Hogyan erősítsük rendszerünk biztonságát?

A biztonsági rések betömése a Bastille segítségével.

Vajon Ön is azok közé közé tartozik, akiknek, amikor a Linux telepítése után először kiadják a `ps -ef` parancsot, fogalmuk sincs, hogy mit csinál a megjelenő listában felsorolt folyamatok fele? Ne szégyellje mindenkinek el kell kezdenie valahol, rengeteg súgóoldalt át kell olvasni, mire megértjük a Unix-rendszert felépítő rengeteg alkalmazás és démon működését. A súgóoldalak olvasgatását semmivel sem lehet kiváltani, de az is túlságosan nagy könnyelműség lenne, ha biztonsági hézagokban bővelkedő rendszereket üzemeltetnénk mindaddig, amíg el nem érjük a guru állapotot.

A Bastille Linux a rendszer megerősítését szolgáló Perl parancsfájlgyűjtemény, mely biztonságosabbá teszi az operációs rendszert, és ha interaktív üzemmódban használjuk, oktat is. Tiszta és célratörő kérdéseket tesz fel, melyek megválaszolásával egyedi biztonsági szabályozást alakíthatunk ki rendszerünkben. Minden kérdésre részletes választ kapunk, így mire megismerkedünk a Bastille-jal, jó néhány dolgot megtudunk a Linux/Unix biztonságáról is.



InterActiveBastille.pl

Aki pedig már tisztában van a rendszer biztonsági kérdéseivel, és a Bastille használata közben szeretne megtakarítani egy kevés időt, kérheti a Bastille-t, hogy kevesebb magyarázatot jelenítsen meg. Ebben az esetben is ugyanazokat a kérdéseket teszi fel nekünk, viszont nem látjuk a kérdésekhez tartozó hosszadalmas magyarázatokat. Ha „tiszta” Linux telepítésen futtatjuk a Bastille-t, akkor akár ki is hagyhatjuk a kérdéseket és választhatunk egyet az előre összeállított biztonsági sablonok közül. Természetesen, ha igazi nagymenők vagyunk, akkor saját beállításablont írunk majd a Bastille-hoz (vagy ami ennél valószínűbb, a programhoz mellékelt sablonokat fogjuk saját igényeinkhez igazítani).

Miért hasznos a Bastille?

Talán megfordult már a fejünkben az, hogy vajon miért kell olyan sok szolgáltatást engedélyezni a rendszerben alapértelmezés szerint? Nem butaság az, hogy külön parancsfájlokra van szükségünk a felesleges dolgok lefaragásához? Nem lenne-e egyszerűbb, ha ezek már eleve nem lennének ott?

Véleményem szerint a legtöbb Linux-válozat alapértelmezés szerint túlságosan sok dolgot engedélyez. Az azonban tény, hogy a Linux-felhasználók között egyre több a kezdő, és ha a rendszertelepítés után azt tapasztalják, hogy a Linux túlságosan kevés dologra képes (vagy ami még rosszabb, egyáltalán nem is működik), akkor valószínűleg igen kevesen fognak megmaradni mellette, és még kevesebben fognak azzal foglalkozni, hogy rendszerüket megtanulják biztonságosan üzemeltetni.

Más szóval a Linux-változatokat (például a RedHat, a Caldera stb.) összeállítók általában a használhatóságra fektetik a hangsúlyt, nem pedig a biztonságra. Én jobban szeretném, ha a jelenleginél több változat telepítője kínálna a lehető legtöbb szolgáltatást biztosító és „biztonságos” beállítási lehetőségeket is. (A RedHat 7.0 telepítője tartalmaz ugyan ilyen lehetőségeket, de a „biztonságos” lehetőség választása esetén sincs olyan biztonságban a rendszer, mint a Bastille üzembe helyezése után.)

Hogyan született a Bastille?

A Bastille fejlesztőcsapatát *Jon Lasser* és *Jay Beale* vezette. Eredeti céljuk az volt, hogy létrehozzanak egy olyan Linux-változatot, amely a RedHatre épül, de sokkal biztonságosabb annál. Úgy tűnt, hogy céljukat a legkönnyebben úgy érhetik el, ha egy átlagos RedHat rendszerből indulnak ki, és különböző Perl parancsfájlok segítségével fokozatosan biztonságosabbá teszik azt. Ez sokkal rugalmasabb megoldásnak bizonyult, ugyanis olyan parancsfájlokat tudtak létrehozni, amelyek több különböző változat megerősítésére is felhasználhatók.

A Bastille-t felépítő parancsfájlok meglehetősen értelmesek, és sokkal kevesebbet feltételeznek a rendszerünkről, mint korábbi testvéreik (kezdetben kizárólag újonnan telepített RedHat rendszerre tervezték e parancsfájlokat). Egyáltalán nem fontos, hogy a rendszer frissen legyen telepítve, sőt még az sem, hogy RedHat-változat legyen. A Bastille az 1.1.x változat óta számos adatot begyűjt rendszerünkről, mielőtt módosításokat végezne rajta.

A Bastille beszerzése és telepítése

Remélem, mindenki felkészült lelkileg arra, hogy némi oktatás után megerősítse rendszerét. Fontos megjegyezni, hogy az eredményesség változó lehet. Habár a Bastille szinte minden Linux-változattal használható, eredetileg a RedHat segédprogramjának készült, és továbbra is ezeken a rendszereken a leghatékonyabb. A cikk során kitérek majd a nem RedHat-változatokkal kapcsolatos kérdésekre is, az eredményért azonban nem állok jót. Ha bizonytalanok valamiben, akkor látogassanak el a Bastille honlapjára. Ha már itt járunk, meg kell említenünk, hogy ez a hely a Bastille Linux, illetve a hozzá kapcsolódó leírások legbiztosabb forrása. A ➲ http://www.bastille-linux.org/ lap tetején mindig megtaláljuk azt a nagy kövér betűkkel szedett hivatkozást, amely a Bastille Linux legfrissebb változatára mutat. Miután letöltöttük a tar fájlt, helyezzük a /root könyvtárba, és csomagoljuk ki:

```
tar -xzvf ./Bastille-X.Y.Z.tgz
```

Ez az, sikerült a telepítés!

Ne felejtsük el, a Bastille azt feltételezi, hogy a /root könyvtárba telepítettük. Valószínűleg át tudjuk alakítani a Bastille parancsfájlokat úgy, hogy másik könyvtárban is jól érezzék magukat, ezt azonban senkinek sem ajánljuk, mivel a hatás kiszámíthatatlan (ráadásul igen sok parancsfájlról van szó).

A Perl 5 parancsnyelvre is szükségünk lesz ahhoz, hogy a Bastille-t futtatni tudjuk. Ha szeretnénk meggyőződni arról, hogy tartalmazza-e a megfelelő változatot a rendszerünk, akkor egyszerűen adjuk ki a `perl -v` parancsot. Ha a Perl 5.0-nál alacsonyabb változatszámmal válaszol, vagy pedig a *perl: command not found* üzenet jelenik meg a képernyőn, akkor telepítenünk kell rendszerünkre vagy frissíteni kell Perl változatunkat. Egyetlen újabb Linux-változatból sem hiányzik a Perl 5. Nézzünk szét a CD-ROM-on vagy a változathoz tartozó webhelyen.

Az első lépések

A Bastille használata egyszerű. Futtassuk le a /root/Bastille könyvtárban található InteractiveBastille.pl parancsfájlt (lásd *képünkön*). Hosszú kérdéssorozatot kell megválaszolnunk, hogy milyen szolgáltatásokat engedélyezünk. Hogyan kell beállítanunk azokat, hogy megtalálhassuk a szolgáltatások és a biztonság közötti egyensúlyt. Ezek a kérdések különböző témák szerint vannak csoportosítva. A kérdésekre adott feleleteket a *config* nevű fájlban tárolja a rendszer.

Ezután futtassuk a BackEnd.pl parancsfájlt, amely sorra meghívja az InteractiveBastill.pl egyes részeihez kapcsolódó megfelelő parancsfájlokat. A parancsfájlok működését meghatározó változókat a *config* fájl alapján állítja be a rendszer. A válaszoktól függően előfordulhat, hogy egyes parancsfájlokat egyáltalán nem futtat le.

Ha nem szeretnénk ezekkel a kérdésekkel bajlódni, akkor futtassuk az AutomatedBastill.pl parancsfájlt, amely lehetőséget ad arra, hogy különböző alapértelmezett beállítások szerint erősítsük rendszerünk biztonságát. Az AutomatedBastille.pl nagyon egyszerű parancsfájl. Mindössze annyit tesz, hogy meghívja a BackEnd.pl parancsfájlt az előre legyártott beállítási fájllal.

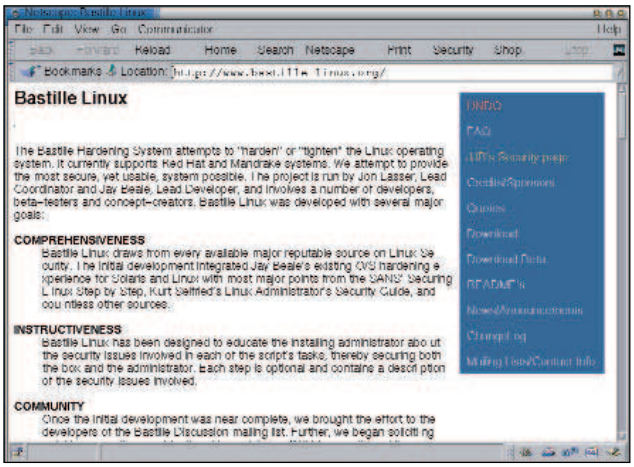
Ezeket az előkészített fájlokat (a Bastille v.1.1.0 változatában Default_Workstation és Default_Workstation_plus_Firewall) könnyedén a saját igényeinkhez igazíthatjuk. Természetesen mi magunk is létrehozhatunk ilyen fájlokat, és az AutomatedBastille.pl megkerülésével közvetlenül a BackEnd.pl parancsfájlt is használhatjuk:

```
./Backend.pl ./beállítási_fájl /root/naplófájl
```

Megjegyzések az InteractiveBastille.pl használatához

Az InteractiveBastille.pl bőséges magyarázatot tartalmaz. Ha elegendő időt szánunk az egyes kérdésekhez tartozó válaszok tanulmányozására, akkor közben igen sokat megtudhatunk a rendszer megerősítésének módszereiről. Ha úgy gondoljuk, hogy már kellőképpen járatosak vagyunk a témában, akkor bármikor átállhatunk a magyarázatokban szegényebb üzemmódra (ez természetesen fordítva is igaz). Az alábbi általános tanácsok hasznosak lehetnek a kezdők számára:

- *Első modul: IPChains.pm* – Az IPChains linuxos tűzfal. Ha gépünkhöz hozzáférhetnek majd az Interneten keresztül más számítógépek, mindenképpen célszerű használnunk. Néhány egyszerű csomagszűrő szabállyal is lényegesen növelhetjük rendszerünk biztonságát.
- *Második modul: PatchDownload.pm* – Ha RedHat rendszert használunk, akkor a Bastille képes arra, hogy RPM-eket töltsön le és telepítsen azokhoz a programokhoz, amelyek telepítésük óta megváltoztak.
- *Harmadik modul: FilePermissions.pm* – Ez a modul korlátozza a hozzáférést bizonyos segédprogramokhoz és fájlokhoz. Ezt elsősorban a SUID jelző kikapcsolásával éri el. A SUID képes arra, hogy úgy indítson el különböző folyamatokat, mintha a rendszergazda hívta volna meg azokat. Így az egyszerű felhasználók is lehetőséget kapnak arra, hogy olyan segédprogramokat futtassanak, amilyen például a mount, a ping vagy a traceroute. Azoknak a felhasználóknak, akik nem rendelkeznek a megfelelő jogosultságokkal, általában nincs is szükségük ezekre a segédprogramokra. A SUID jelző kikapcsolásával ezeket a visszaélésre is alkalmat adó programokat kizárólag a rendszergazda futtathassa.
- *Negyedik modul: AccountSecurity.pm* – Ez a modul lehetőséget ad arra, hogy új rendszerfelügyelői jogosultságokkal felruházott azonosítót hozzunk létre, és általában véve biztonságosabbá tehessük a felhasználói azonosítók kezelését. Ezeket a lépéseket mindenképpen célszerű megtennünk.
- *Ötödik modul: BootSecurity.pm* – Amennyiben rendszerünk elé ismeretlen személyek is leülhetnek, újraindíthatják azt, és az indítási folyamat megváltoztatásával illetéktelen jogokat gyakorolhatnak. E modul segítségével megnehezíthetjük a rendszer ilyen típusú veszélyeztetését.
- *Hatodik modul: SecureInetd.pm* – Ebben a részben internetes szolláltatásainkat tudjuk megerősíteni.
- *Hetedik modul: DisableUserTools.pm* – Tanácsos letiltani a fordító használatát abban az esetben, ha a rendszergazdai jogosultságokkal nem rendelkező felhasználóknak nincs kifejezetten szükségük rá. Amint a legtöbb hasonló esetben, a „letiltás” itt is azt jelenti, hogy az adott szolgáltatást kizárólag a rendszergazda veheti igénybe.
- *Nyolcadik modul: ConfigureMiscPAM.pm* – Itt különféle, a felhasználói azonosítókra vonatkozó korlátozásokat állíthatunk be.
- *Kilencedik modul: Logging.pm* – A legtöbb rendszer alapértelmezés szerint csak igen szűkszavú naplózást engedélyez. Ezzel a modullal kiterjeszthetjük a naplózást és lehetőséget kapunk arra, hogy a naplóadatokat távoli gépekhez továbbítsuk. A folyamatok nyomon követését is bekapcsolhatjuk itt, a legtöbb rendszer esetében azonban erre nincs szükség.
- *Tizedik modul: MiscellaneousDaemons.pm* – Ebben a részben olyan szolgáltatásokat tudunk kikapcsolni, amelyek alapértelmezés szerint bárki számára hozzáférhetők, holott a legtöbb felhasználónak valójában nincs is szüksége rájuk.
- *Tizenegyedik modul: Sendmail.pm* – Magáért beszél.
- *Tizenkettedik modul: RemoteAccess.pm* – Ha az SSH (Secure Shell) még nincs jelen a rendszerünkben, akkor a Bastille képes letölteni és telepíteni azt. Az SSH biztonságos szolgáltatásokat tartalmaz, amelyek képesek helyettesíteni a telnet, az rsh és az



➲ www.bastille-linux.org/

➲ www.enteract.com/~lspitz/

rlogin szolgáltatásokat. Fontos megjegyezni, hogy a Bastille az i386-os gépeken futó RedHat rendszerekhez fordított RPM-eket próbálja meg telepíteni. Ha Linuxunk nem PC-megfelelő környezetben fut, vagy pedig olyan változatot használunk, amely nem képes kezelni a RedHat RMP-eket, akkor ezt a modult nem tudjuk majd használni.

Naplózás és a Bastille futtatása nem RedHat-változatokon

Mi történik azután, hogy az InteractiveBastill.pl létrehozta a config fájlt, a BackEnd.pl pedig megtette a megfelelő lépéseket? Honnan tudhatjuk meg, hogy mi zajlik a rendszerünkben? A Bastille kiváló naplózási lehetőségeinek köszönhetően igen egyszerűen kideríthetjük azt, hogy mely változtatásokat sikerült végrehajtani, és melyeket nem. Ha már itt járunk, célszerű néhány szót ejteni arról is, hogyan használhatjuk a Bastille-t nem RedHat-változatok esetén.
Amint azt korábban már említettük, a Bastille a RedHat-változatra, illetve ennek leszármazottaira (Mandrake, Yellow Dog stb.) lett kialakítva (optimalizálva), mégis elég értelmes ahhoz, hogy más rendszerekkel is elboldoguljon. Tulajdonképpen saját tapasztalataim is ezt az állítást igazolják: lefuttattam a Bastille 1.1.0-t SuSE Linux 6.3 alatt, és örömmel állapítottam meg, hogy a programnak több olyan része volt, amely megfelelően működött, mint ami nem.
A SuSE 6.3 használata során elsőként azt vettem észre, hogy a Bastille helyenként meglehetősen RedHat-központú. A PatchDownload.pm kizárólag RedHat alatt használható, és amint azt korábban már megjegyeztem, a RemoteAccess.pm az SSH 1.2.27 Red Hat i386-os változatát telepíti.
Figyelmeztetéseket kaptam akkor is, amikor a namedet chroot szolgáltatással próbáltam futtatni. Szerencsére a csomagok letöltése és a named futtatása olyan feladatok, amelyek kézzel is könnyen elvégezhetők.
Habár az InteractiveBastill.pl futtatása során kizárólag ez a két nehézség jelentkezett, voltak más olyan részei is, amelyek nem voltak képesek együttműködni a SuSE-változattal. A BackEnd.pl például hiba nélkül lefutott. A futás során hibákat csak akkor vettem észre, amikor átolvastam a Bastille naplóit.
Ezeket a naplókat mindenképpen érdemes átnézni, még akkor is, ha úgy gondoljuk, hogy a Bastille mindent megfelelően végrehajtott. Az értelmes naplózás egyike a Bastille leghasznosabb szolgáltatásainak. Függetlenül attól, hogy kezdők vagyunk vagy Linux-szakértők, nemcsak arról kell tudnunk, hogy mit csinál a Bastille, hanem utána kell néznünk annak is, hogy hogyan csinálja azt.
Ésszerű módon a Bastille a /root/Bastille/log mappába írja a naplóit. A BackEnd.pl két naplót hoz létre: action-log és error-log néven. Az action-log átfogó és részletes leírást tartalmaz a Bastille tevékenységéről, a hibák és a többi váratlan esemény pedig az error-log naplóba kerül.
A SuSE Linux használata során a hibát az okozta, hogy bizonyos fájlok nem voltak ott, ahol a Bastille kereste őket, ugyanis a SuSE egyes fájlokat máshol tárol, mint a RedHat. Igen egyszerűen rá lehet jönni, hogyan lehet kézzel kijavítani a hibát. A Bastille kérdéseinek listája (/root/Bastille/Questions.txt) számtalan ötletet tartalmaz, és ha ismerjük a Perl nyelvet, akkor a parancsfájlokat magunk is módosíthatjuk.
A legegyszerűbb megoldás az, ha módosítjuk a Bastille parancsfájlokban az elérési útvonalakat, majd újra lefuttatjuk a BackEnd.pl-t. A legtöbb elérési út, amelyet a SuSE esetében módosítanom kellett, a /root/Bastille/Bastille/FilePermissions.pl parancsfájlban volt. Minden olyan fájlt, ami a Bastille hibanaplója szerint hiányzott, megkerestem a which segítségével. Ha a keresett fájl létezett, épp csak egyszerűen rossz helyen volt, akkor módosítottam a /root/Bastille/Bastille/FilePermissions.pl megfelelő részét.
Három fájlt gyakran nem talál SuSE alatt a Bastille: a setserial, a chkconfig és az ifdown fájlokat. A chkconfig és az ifdown nem is léteznek a SuSE rendszerekben; ezekkel kizárólag a RedHat illetve az abból származó rendszerekben találkozhatunk. Ezekkel a hibákkal tehát nem kell foglalkoznunk. De mi a helyzet a setseriallal?
A setserial pontos helye a `/sbin/setserial`. Ha kiadjuk a

```
grep setserial /root/Bastille/Bastille/
```

parancsot, akkor a következőt láthatjuk:

```
/root/Bastille/Bastille/FilePermissions.pm:
&B_chmod(0750,"/bin/setserial");
```

Viszonylag kevés nem létező fájl esetén a dolog tehát egyszerű. Sorra vettem az egyes fájlokat, és kiderítettem, hogy léteznek-e, és ha igen, akkor hol vannak. Ezután már csak a megfelelő módosításokat kellett elvégezni a FilePermissions.pm modulban. A hibanaplóban felsorolt fájlok felderítése és a BackEnd.pl újrafuttatása nem tartott tovább húsz percnél.
Volt egy másik gond is, amit meg kellett oldanom. Ezt mutatja be a következő példa:

```
#open /etc/httpd/conf/httpd.conf.bastille
#failed...
#open /etc/httpd/conf/httpd.conf failed.
Couldn´t replace line(s) in
/etc/httpd/conf/httpd.conf because open failed.
#open /etc/httpd/conf/httpd.conf.bastille
#failed...
#open /etc/httpd/conf/httpd.conf failed.
Couldn´t replace line(s) in
/etc/httpd/conf/httpd.conf because open failed.
```

Ezt a hibát a RedHat és a SuSE Apache környezetének az eltérése okozza, amelyről korábban már beszéltünk. Ebben az esetben is a `grep` parancsot használtam:

```
grep httpd.conf /root/Bastille/Bastille/*.pm
```

A parancs kimenete a következő volt:

```
API.pm:
$GLOBAL_FILE{"httpd.conf"}="/etc/httpd/conf/httpd
.conf";
API.pm:
$GLOBAL_FILE{"httpd_access.conf"}="/etc/httpd/con
```

```
f/httpd.conf";
API.orig.pm:
$GLOBAL_FILE{"httpd_access.conf"}="/etc/httpd/con
f/access.conf";
Apache.pm:  my
$httpd_file=$GLOBAL_FILE{"httpd.conf"};
```

Ezekből a sorokból kiderül, hogy az Apache modul egy GLOBAL_FILE nevű változóban tárolja a httpd.conf elérési útját, és ez a változó az API.pm modulban kerül beállításra. Nem kellett mást tennem, mint megváltoztatni ezt az elérési útvonalat az API.pm modulban, majd újra lefuttatni a BackEnd.pl-t (amelyben az Apache.pm hívását tartalmazó soron kívül minden mást megjegyzésbe helyeztem).
Azok a segédprogramok, amelyek biztonságát nem erősítettük meg a Bastille segítségével, könnyen módot adhatnak az egyszerű felhasználóknak arra, hogy rendszergazdai jogokat szerezzenek maguknak a rendszerben. Mindenképpen érdemes tehát rászánnunk az időt ezeknek a hibáknak a felderítésére és kijavítására, főleg abban az esetben, ha egynél több rendszer biztonságról van szó. Nyilvánvaló, hogy minél több azonos típusú rendszer biztonságáról kell gondoskodunk, annál inkább megtérül a munkával töltött idő.

Most már teljes biztonságban vagyunk. Vagy mégsem?

Gondosan átolvastuk és megválaszoltuk az InteractiveBastill.pl által feltett kérdéseket, lefuttattuk a BackEnd.pl-t, átnéztük a Bastille által létrehozott naplófájlokat, és miután kijavítottuk az esetleges hibákat, újra lefuttattuk a BackEnd.pl parancsfájlt. Készen vagyunk?
Nem, és soha nem is leszünk! A biztonsági rendszer karbantartása folyamat, nem pedig egyszeri feladat. A legbiztosabb módja annak, hogy sebezhető rendszert hozzunk létre az, hogy az üzembe helyezés után egyszerűen magára hagyjuk. Ez még akkor is igaz, ha mielőtt valóban magára hagynánk a rendszert, először megerősítjük azt.
Természetesen még a Bastille sem tud előre felkészülni minden olyan programcsomagra, amelyet az adott rendszer alá telepíthetünk. Egyes változatok, például a Debian és a SuSE rengeteg csomagot tartalmaznak, olyan sokat, hogy egy nemrégiben megjelent beszélgetésben Jon Lasser, a Bastille egyik atyja, külön kihangsúlyozta, hogy mekkora gondot okoznak ezek biztonsági szempontból nézve. Hozzátette, hogy ez nem azt jelenti, hogy azokat a rendszereket, amelyek sok különböző programcsomagot tartalmaznak, ne lehetne biztonságossá tenni; mindössze annyit jelent, hogy több munkát igényel.
Van még néhány feladat, amelyeket feltétlenül el kell végeznünk a Bastille futtatása után:

- *A megmaradt szolgáltatások közül tiltsuk le azokat, amelyekre nincs szükségünk.* Nézzünk utána a /etc/rc.d/rc{{n}}.d fájlban (az {{n}} az alapértelmezés szerinti futtatási szintet jelöli, ennek az értékét a `grep initdefault /etc/inittab` parancs kiadásával tudhatjuk meg), hogy melyek azok a szolgáltatások, amelyek továbbra is önműködően elindulnak a rendszer indítása során. A rendszer indításával együtt induló parancsfájlokat a nevük előtt álló nagy S betű jelzi. Ha nem tudjuk, hogy mire jó egy adott démon, akkor adjuk ki a `man` parancsot a démon nevével. Ha nincs szükségünk az adott szolgáltatásra, akkor nevezzük át az `mv` parancs segítségével (ha a név nem S-sel kezdődik, akkor a szolgáltatás nem indul el önműködően).
- *Az alkalmazásaink biztonságossá tételére is szánjunk elegendő időt.* A Bastille segítségével a rendszerünket anélkül is biztonságossá tehetjük, hogy mindent tudnánk a rendszerek biztonságáról. Ez az oka annak, hogy a készítők olyan sok időt szántak az oktatási anyagok beillesztésére. A programfájlgyűjtemény által biztosított alkalmazások azonban nem lehetnek igazán biztonságosak, ha nem értjük a működésüket.

A BIND például, amely a hálózati alkalmazások számára szükséges DNS névfeloldást biztosítja, számos biztonsági szolgáltatást tartalmaz. A Bastille sokat beállít ezek közül, de számos olyan marad, amit nem. A megfelelő man oldalak átolvasása tehát kötelező minden rendszergazda számára.

- *Tiltsuk le a nem használt felhasználói azonosítókat.* A SuSE egyik legbosszantóbb sajátossága az, hogy a /etc/passwd fájl rengeteg, különböző alkalmazásokhoz tartozó jelszóbejegyzést tartalmaz, függetlenül attól, hogy telepítettük-e az alkalmazásokat vagy sem. Ezek között az alkalmazások között sok olyan van, amelyek interaktív bejelentkezésre is lehetőséget adnak (nem a /bin/false héj van megadva mellettük).

Ezzel kizárólag a SuSE-változatban találkozhatunk. Éppen ezért győződjünk meg arról, hogy a /etc/passwd fájlban minden felesleges bejegyzést megjegyzésbe helyeztünk. Ha kétségeink merülnének fel, akkor figyeljünk oda arra, hogy az utolsó mezőben (default shell – alapértelmezés szerinti héj) ne egy valódi héj, hanem a /bin/false szerepeljen – csak a tényleges felhasználói azonosítók esetén van szükség héjra.

- *Folyamatosan frissítsük a rendszert.* Nem lehet eléggé kihangsúlyozni annak fontosságát, hogy mindig naprakészek legyünk. Kövessük nyomon a változatokhoz megjelenő biztonsági javításokat. Ahogy egyre jobban megismerjük a Linux biztonsági rendszerét, azonnal alkalmazzuk az új tudást saját rendszerünkre is. Semmiképpen se hagyjunk magára egy Internetre kapcsolt gépet.
- *Telepítsünk egy betörésészlelő programot.* Az operációs rendszer telepítése után minél előbb telepítsünk betörésészlelő programot is, ilyenek például a tripwire vagy a snort. Ezzel csökkenthetjük annak esélyét, hogy mástól (rendőrség, ideges rendszergazdák stb.) kelljen tudomást szereznünk arról, hogy betörtek a rendszerünkbe, ráadásul más rendszerek megtámadásához is felhasználták.
- *Figyeljük a naplókat!* Nem szabad elfeledkeznünk arról sem, hogy folyamatosan nyomon kövessük a Bastille által készített naplóbejegyzéseket.

A naplófájlok böngészése igen fárasztó is lehet, éppen ezért célszerű olyan parancsfájlokat készíteni, amelyek időnként önműködően átnézik a naplófájlokat és kigyűjtik a gyanús bejegyzéseket, vagy telepíteni olyan eszközt, mint például a swatch, amely képes elvégezni ezt a feladatot.

Mick Bauer
(mick@visi.com) hálózati biztonsággal foglalkozó szaktanácsadó. 1995 óta a Linux elkötelezett híve, 1997 óta pedig OpenBSD prófétaként tevékenykedik. Mick szívesen fogad minden kérdést, és megjegyzést.

Kapcsolódó címek

A Bastille hivatalos honlapja
➲ http://www.bastille-linux.org/

Interjú Jonnal és Jayjel, a Bastille készítőivel
➲ http://slashdot.org/interviews/00/11/08/1616204.shtml

Lance Spitzner Armoring Linux (A Linux felfegyverzése) című cikke ➲ http://www.enteract.com/~lspitz/linux.html

A Linux Documentation Project hivatalos Linux biztonsági lapja ➲ http://www.linuxdoc.org/HOWTO/Security-HOWTO.html