Betörésérzékelés mindenkinek

Telepítsd a Tripwire-t, hogy elkaphasd a betörőket!

Bár reménykedünk benne, hogy megerősített rendszerünk védelmi szempontból áthatolhatatlan, százszázalékos biztonság nem létezik. A réseket már a rendszer védelmi vonalán fel kell ismerni. A Tripwire Open Source szabadon felhasználható és nyílt forráskódú programcsomag, amely a lehetséges réseket meglehetősen jó eséllyel már megnyitásuk pillanatában felfedezi.
A Tripwire-hoz hasonló sértetlenség-ellenőrzők a titkosításban használt módszerekkel „ujjlenyomatot” vesznek a rendszer futtatható bináris állományairól, beállításokat tartalmazó fájljairól és minden másról, amelyek a behatolás során vagy annak következtében megváltozhatnak. Ezután ezeket a fájlokat időről-időre összevetik a tárolt ujjlenyomatokkal, és a különbségeket levélben elküldik a rendszer gazdájának.
A Tripwire a legjobban ismert és a legérettebb sértetlenség-ellenőrző rendszer, írásunkban ezt tárgyaljuk részletesen. Érdekes lehet még az Advanced Intrusion Detection Environment (AIDE) is, amely több felületen fut, mint a Tripwire Open Source, és az FCheck, és amelyik teljesen Perlben íródott, ezért még az AIDE-nél és a Tripwire-nél is kevésbé felületfüggő (ráadásul Windows alatt is fut). A cikk végén a *Kapcsolódó címek* részben találhatók a hivatkozások az AIDE és az FCheck weboldalaira.

Mi a sértetlenség-ellenőrzés?

A sértetlenség-ellenőrzés olyasmi, mint a biztonsági mentés: remélhetőleg sosincs rá szükség, de az ég óvjon attól, hogy éppen akkor ne álljon a rendelkezésünkre, amikor szükségünk lenne rá. Akárcsak a biztonsági mentés, a sértetlenség-ellenőrzés is egy nagyobb terv része. Ha rendszerünket megerősítettük, megfoltoztuk, és a legmagasabb szinten karbantartjuk (vagy legalább a józan ész elvárta szinten gondját viseljük), akkor a sértetlenség-ellenőrző biztosítja a végső védőhálót, amely felfogja a betörő támadását, bármilyen okos legyen is.
A sértetlenség-ellenőrző működési elve egyszerű: ha egy fájl váratlanul megváltozott, jókora az esélye, hogy egy behatoló változtatta meg. Az első dolgok egyike például, amelyet a betörők a rendszergazda jogainak megszerzése után meg szoktak tenni, az, hogy a gyakran használt rendszerprogramokat (például ls, ps és netstat) olyanokra cserélik le, amelyek látszólag a szokásos módon működnek, de elrejtik azokat a fájlokat, folyamatokat vagy kapcsolatokat, amelyek lebuktathatnák őket.
A sértetlenség-ellenőrzők segítségével létrehozhatunk egy ellenőrzőösszeg-adatbázist a fontos rendszerprogramokról, beállítási fájlokról vagy bármi másról, amitől nem várjuk vagy nem akarjuk, hogy megváltozzon. Ha rendszeresen összevetjük ezeket a fájlokat a sértetlenség-ellenőrző adatbázisával, rendszerünk feltörésének esélyét csökkenthetjük. Minél hamarabb szerez tudomást a rendszergazda a betörésről, annál nagyobb az esélye, hogy elkapja vagy legalább elkergeti a behatolókat.
Figyelem, a legjobb sértetlenség-ellenőrző sem ér semmit, ha nem megbízható az adatbázisa! Nagyon fontos, hogy a lehető leghamarabb létrehozzuk ezt az adatbázist az operációs rendszer biztonságos forrásból történő telepítése után. Megismétlem, a sértetlenség-ellenőrző telepítése, beállítása és fenntartása nem éri meg a fáradtságot, ha nem tiszta rendszeren hozták létre az adatbázisát.

Tripwire – az első és mindmáig a legjobb sértetlenség-ellenőrző

Számos más ünnepelt és hasznos dologgal együtt a Purdue COAST Project (➲ http://www.cerias.purdue.edu/coast/) „ajándékozta” a világnak a *Dr. Eugene Spaffort* és *Gene Kim* által írt Tripwire-t is. Eredetileg nyílt forrású, szabad program volt, majd 1997-ben kereskedelmi termék lett, és a díjmentes használat az akadémiai, illetve egyéb nem kereskedelmi célú felhasználásra korlátozódott.
Szerencsére tavaly októberben a Tripwire Inc. megjelentette a Tripwire Open Source Linux Edition névre hallgató terméket. A Tripwire kereskedelmi változatai ezelőtt olyan lehetőségeket tartalmaztak, amelyek az Academic Source Release-ből hiányoztak. Ezzel szemben a Tripwire Open Source a kereskedelmi termék többé-kevésbé friss változatának felel meg. Az olyan vállalati környezetben kihasználható lehetőségeket leszámítva, mint a sok gépből álló rendszer központi karbantartása, sokban hasonló a Tripwire for Servers termékhez.
A Tripwire Open Source csak a nem kereskedelmi Unixokra ingyenes, például Linuxra és Free/Net/OpenBSD-re. Tulajdonképpen csak a RedHat Linux és a FreeBSD támogatott, de semmi akadálya, hogy más Linux- és BSD-változatokon is ugyanolyan jól lefordítható és futtatható legyen. Csak a régebbi Academic Source Release használható szabadon a kereskedelmi Unixokon, például Solarison vagy IBM AIX-en, egyébként meg kell vásárolnunk a kereskedelmi változatot. Írásunk további részében a Tripwire Open Source Linux Edition változatával fogok foglalkozni.

A Tripwire beszerzése, lefordítása, telepítése

A cikk írásának pillanatában a Tripwire Open Source legújabb változata: a 2.3.1-2. A forráskód `tar`-csomagként letölthető a ➲http://sourceforge.net/projects/tripwire/ weboldalról. Javaslom, hogy ezt a csomagot töltsd le, fordítsd le és telepítsd. A Tripwire története során csupán egyetlen jelentős biztonsági hibája akadt (az is csak egy szolgáltatás-megtagadási kockázat volt). Ezt a programot azért használjuk, mert üldözési mániánk van. Az üldözési mániások számára csak a legújabb (megbízható) változat a megfelelő. Az eddig elmondottak figyelembevételével megemlítem, hogy a RedHat 7.0-ban található bináris változat szintén meglehetősen új. Amennyire meg tudom állapítani, a RedHat v2.3-55 RPM-je és a hivatalos forrás v2.3.1-2-változata között csak két, a biztonságot nem érintő hibajavítás történt, tehát valószínűleg nem vállalsz óriási kockázatot, ha a RedHat 7.0 RPM-jét telepíted. De ne mondd, hogy én nem szóltam!
A Tripwire Open Source lefordításához helyezd a forráscsomagot a /usr/src könyvtárba, és csomagold ki:

```
tar xzvf ./tripwire-2.3.1-2.tar.gz
```

Ezután ellenőrizd, hogy megvan-e a rendszereden a közvetett hivatkozás a `/usr/bin/gmake`-ről a `/usr/bin/make`-re (GNU-make ugyanis nem minden Unixnak része a, ezért a Tripwire kifejezetten a `gmake`-et keresi – természetesen a legtöbb Linux-rendszeren ez a `make`). Ha nincs meg, a következő paranccsal létrehozható: `ln -s /usr/bin/make /usr/bin/gmake`.
Ezenkívül ellenőrizni kell az alkönyvtárak teljes rendszerét a

`/usr/share/man` alatt, ugyanis a Tripwire a `man4`, a `man5` és a `man8` alá akar majd súgóoldalakat telepíteni. Az én Debian-rendszeremen hiányzott a `/usr/share/man/man4`, ezért a telepítő létrehozott egy /usr/share/man/man4 nevű fájlt, amely a súgóoldalt tartalmazta ahelyett, hogy az ilyen nevű könyvtárba másolta volna. Végül el kell olvasni a forrás README és INSTALL fájljait, belépni a forrásfa src könyvtárába (például /usr/src/tripwire-2.3.1-2/src), és a változók értékeit az src/Makefile-ban szükség szerint módosítanunk kell. Győződj meg róla, hogy a megfelelő SYSPRE sor elől törlöd-e le a megjegyzésjelet (SYSPRE = i386-pc-linux, SYSPRE = sparc-linux stb.)! Felkészültünk a fordításra, tehát írd be a `make release` parancsot. Mivel ez jó darabig eltart, közben akár egy szendvicset is bekaphatsz. Amikor a fordítás elkészült, a könyvtárszerkezetben lépj eggyel feljebb, például /usr/src/tripwire-2.3.1-2, továbbá add ki a következő két parancsot:

```
cp ./install/install.cfg .
cp ./install/install.sh .
```

Kedvenc szövegszerkesztődben nyisd meg az install.cfg-t, és ha az alapértelmezett útvonalakat rendben találod, vizsgáld meg a *Mail Options* részt. A Tripwire-nak itt adhatod meg, hova küldje a naplóbejegyzéseit. Ha a TWMAILMETHOD=SENDMAIL beállítást választod, a Tripwire a megadott helyi levelezőt (alapértelmezés szerint a `sendmail`-t) használja arra, hogy jelentéseit elküldje egy helyi felhasználónak vagy csoportnak.
A TWMAILMETHOD=SMTP beállítást választva, és a TWSMTPHOST és TWSMTPPORT értékeit megadva a Tripwire a külső levélcímre küldi el a jelentéseit a megadott SMTP-kiszolgálón és -kapun keresztül. Ha később meggondolnád magad, a Mail Options beállításait bármikor megváltoztathatod.
Ha a rendszert – amelyre a Tripwire-t telepíted – több felhasználó is használja, továbbá valaki rendszeresen belép és olvassa a rendszergazda leveleit, akkor a SENDMAIL-módszer a megfelelőbb. Ha a rendszer karbantartása egy másik gépen keresztül távolról történik, az SMTP-módszer a jobb.
Miután kedved szerint kitöltötted az install.cfg-t, ideje telepíteni a Tripwire-t. Írd be: `sh ./install.sh`. Két jelszót fog kérni a program, a *site* jelszó a Tripwire beállítási fájljait védi, a *local* pedig a Tripwire adatbázisát és jelentéseit. Így lehetséges egységes házirendet alkalmazni több gépre, míg az adatbázis karbantartásának és a jelentések létrehozásának felelősségét el lehet osztani.

Megjegyzés az RPM-ekről

Az RPM-ek telepítése egyszerűbb és gyorsabb (de jegyezzük meg ismét, hogy a legfrissebb Tripwire-változat nem feltétlenül érhető el ebben a formában). Miután felraktad az RPM-et, az egyetlen dolog, amit meg kell jegyezned, az, hogy le kell futtatnod a `/etc/tripwire/twinstall.sh` parancsot, amellyel létrehozhatod a `site` és `local` jelszót. A parancsfájl a forrás `install.sh` parancsfájljához hasonlóan működik, lásd az előző bekezdést.

A Tripwire használata

Amennyire hasznos a Tripwire, olyannyira elterjedt róla az a hír, hogy nehéz beállítani (ez természetesen minden sokoldalú bonyolult eszközre igaz). A helyzet valójában nem olyan rossz; ha követed azokat az egyszerű lépéseket, amelyeket itt megadok, akkor hatékonyan használhatod a Tripwire-t a rosszfiúk elfogására. Lépj be hát a felhasználók előkelő csoportjába, akik nemcsak telepítették, hanem használják is a Tripwire-t!

A beállítófájl kezelése

Első feladatunk a beállítófájl, a tw.cfg ellenőrzése.
Ezt a fájlt a telepítő ugyan titkosította, de kényelmünk kedvéért twcfg.txt néven egy szöveges változat is bekerült a /etc/tripwire könyvtárba. Itt kell megváltoztatnod azokat a beállításokat, amelyekkel kapcsolatban a telepítés óta meggondoltad magad.
Ha valamit megváltoztattál, a beállítófájlt a következő paranccsal titkosítsd újra:

```
twadmin --create-cfgfile --site-
➥keyfile/site.key twcfg.txt
```

Ebben az esetben a site.key a telepítés idején létrehozott kulcs, és a twcfg.txt az a szöveges beállítófájl, amit az imént szerkesztettél és titkosítani szeretnél. Ez magától értetődőnek tűnhet, hiszen ezek a fájlok alapértelmezett nevei, de bármi másnak is elnevezheted őket. Ne mulaszd el megadni a kulcsfájlt, különben a `twadmin` hibát jelez (és ne feledd, a gyakorlat célja a beállítófájl titkosítása).
Figyelem, soha ne hagyd meg a Tripwire beállító- (tw.cfg) és házirend- (tw.pol) fájljainak szöveges változatát a merevlemezeden. A szerkesztés és a titkosítás után töröld a szöveges változatokat, később bármikor helyreállíthatod őket a következő paranccsal:

```
twadmin --print-cfgfile > mycfg.txt
```

ahol természetesen a mycfg.txt-t bármivel helyettesítheted. Bár nem ismertettem még a Tripwire binárisait (jobb őket a megfelelő környezetben tárgyalni), gondolom már kitaláltad, hogy a `twadmin` segítségével kezelheted a Tripwire beállításait, kulcsait és (legalábbis kezdetben) a házirendfájlokat.

A házirendfájlok kezelése

A házirendeket a Tripwire beállítófájljaihoz hasonlóan szövegfájlként szerkeszthetjük, telepítés előtt azonban alá kell írni és titkosítani kell őket. A beállítófájllal ellentétben a `twadmin` programot egy adott rendszeren csakis az első telepítéshez használjuk, a továbbiakban a Tripwire programot alkalmazzuk a házirendfrissítési módban.
A Tripwire telepítése után első ízben a következő paranccsal hozhatjuk létre a házirendet:

```
twadmin --create-polfile twpol.txt
```

ahol a twpol.txt annak a szöveges házirendfájlnak a neve, amelyet telepíteni szeretnénk.
Ahogy már a beállítófájloknál is említettük, szöveges házirendfájlt

sehol ne hagyj a rendszereden. Ha később olvasni vagy módosítani akarod a házirendet, visszaállíthatod a következő paranccsal:

```
twadmin --print-polfile > mypol.txt
```

A mypol.txt bármi lehet, a házirend szöveges változatának elnevezése a te dolgod.

Házirend létrehozása vagy szerkesztése

Kezdődjék a komoly varázslat! A Tripwire számára a házirendfájl olyan, mint az ember számára az agy: ez mondja meg, mit keressen és mit tegyen vele. Kissé barátságtalan a felhasználókhoz, talán nem annyira, mint a sendmail.cf, de azért készülj fel néhány rövidítés bemagolására.

A Tripwire Open Source tartalmaz egy alapértelmezett házirendfájlt, természetesen ezt is használhatod a sajátodként. Mivel azonban az alapértelmezett házirendet olyan RedHat-rendszerre dolgozták ki, amelyre szinte minden telepítve van, használatbavétel előtt a házirendfájlt alaposan át kell szerkesztened.

Ejtsünk egy-két szót a finomhangolásról. Ha a házirend nem ellenőriz elég fájlt, vagy nem nézi meg eléggé az ellenőrzött fájlokat, akkor a Tripwire nem teljesítheti a feladatát és a gondok észrevétlenül megmaradhatnak. A másik véglet sem szerencsés: ha a Tripwire nagyon odafigyel azokra a fájlokra, amelyek megváltozhatnak, „hamis találatokat” fog jelezni, melyek elterelhetik a figyelmedet a valódi gondokról.

Nem valószínű, hogy mindent elsőre sikerül értelmesen beállítani. Szinte biztos, hogy időről-időre igazítanod kell a házirenden, különösen az első sértetlenség-ellenőrzés futtatása után. Ezért, még ha pontosan ugyanazt a RedHat-rendszert használod is, amelyre a Tripwire Open Source házirendjét tervezték, meg kell tanulnod a Tripwire-házirend formai követelményeit. Munkára fel!

A házirendfájl szerkezetét úgy fogom elmagyarázni, hogy egy működő házirendfájlt kezelhető darabokra osztok. Az első darab a mintafájl legelejéről származik és néhány változónak ad értéket:

```
WEBROOT=/home/mick/www;
CGIBINS=/home/mick/www/cgi-bin;
TWPOL="/etc/tripwire";
TWDB="/var/lib/tripwire";
```

Kevesebbet kell gépelnünk, ha ezeket a változókat használjuk. A következő darabban néhány érdekesebb változónak adunk értéket:

```
BINS        = $(ReadOnly) ;  # Binárisok, amelyek
                             # nem változhatnak
SEC_INVARIANT = +tpug ;      # Könyvtárak, amelyek
                             # tulajdonosa/engedélyei
                             # nem változhatnak
SIG_MED        = 66 ;        # Fontos, de nem
                             # rendszerkritikus fájlok
```

Az első változócsoporttal ellentétben – amelyek egyszerű útvonal-rövidítések voltak – ezek kicsit érdekesebbek. Az első sorból kiderül, hogyan lehet egy változónak egy másik változó értékét megadni, hasonlóan a `bash`-héj formai követelményeihez, de itt zárójelbe kell tenni a második változó nevét.

A második sor adja meg a „tulajdonságmaszkot”. A tulajdonságmaszkok azoknak a fájltulajdonságoknak a rövidítései, amelyeket a Tripwire megvizsgál. Mivel a tulajdonságmaszkok sokszor érthetetlen és kezelhetetlen alakúak, a legtöbb ember változónevekkel hivatkozik rájuk. A Tripwire számos előre megadott változóval rendelkezik, amelyek a gyakran előforduló tulajdonságmaszkoknak felelnek meg.

1. lista Példa házirendfájl

```
WEBROOT=/home/mick/www;
CGIBINS=/home/mick/www/cgi-bin;
TWPOL="/etc/tripwire";
TWDB="/var/lib/tripwire";
BINS= $(ReadOnly) ;    # Binárisok, amelyek
                       # nem változhatnak
SEC_INVARIANT = +tpug ;# Könyvtárak, amelyek
                       # tulajdonosa/engedélyei
                       # nem változhat
SIG_MED= 66 ;          # Fontos, de nem
                       # rendszerkritikus fájlok
# Mick weboldala
(
  rulename = "MickWeb",
  severity = $(SIG_MED),
  emailto = mick@uselesswebjunk.com
)
{
  $(TWPOL)             -> $(Readonly) ;
  $(WEBROOT)           -> $(ReadOnly) (recurse=1) ;
  !$(WEBROOT)/guestbook.html ;
  $(CGIBINS)           -> $(BINS)      ;
  /var/log/httpd       -> $(Growing)   ;
}
```

Az első sor ezek egyikét tartalmazza, a *ReadOnly* tulajdonságmaszk azokra a fájlokra illik rá, amelyek semmilyen módon nem változhatnak, mint például a futtatható fájlok. Ha eljön az ideje, a tulajdonságmaszkokat részletekbe menően tárgyalni fogjuk.

A harmadik sor nevet ad a szigorúsági szintnek. Fontosságuk szerint a szabályok között szigorúsági szintekkel tehetünk különbséget. Amikor a `tripwire` parancsot a `--severity N` kapcsolóval adjuk ki, csak az `N`-nél nagyobb vagy egyenlő szigorúságú szabályok kerülnek értelmezésre. Ha nem használjuk ezt a kapcsolót, az összes szabály érvényes lesz. Amennyiben egy szabálynak a szigorúsági szintje nincs megadva, alapértelmezés szerint nulla lesz. Az ilyen szabály csak akkor kerül értelmezésre, ha a `--severity` kapcsolót nem használjuk. Most már tudunk egyet s mást a házirend változóiról, valamint használatukról, lássuk ezután a szabályokat!

```
# Mick weboldala
(
  rulename = "MickWeb",
  severity = $(SIG_MED),
  emailto  = mick@uselesswebjunk.com
)
{
 $(WEBROOT)    -> $(ReadOnly)(recurse=1) ;
  !$(WEBROOT)/guestbook.html ;
  $(CGIBINS)   -> $(BINS)      ;
  /var/log/httpd     -> $(Growing)   ;
}
```

Ez meglehetősen nagy falat, ezért kezdjük a szabályszerkezettel! A szabályok állhatnak egyedül vagy a közös jellemzők által meghatározott csoportban. Ez a lista több közös tulajdonsággal (zárójelben) rendelkező csoportot mutat (kapcsos zárójelek között). A szabálycsoport neve „MickWeb”, a csoport szigorúsági szintje 66, és a csoport-

1. táblázat **Tulajdonságmaszk-értékek**

Tulajdonság	*Leírás*
-	A következő tulajdonságok figyelmen kívül hagyása
+	A következő tulajdonságok feljegyzése és ellenőrzése
a	Hozzáférés ideje
b	Kiosztott blokkok száma
c	Csomópont időbélyege (létrehozás/változtatás)
d	A fájlleírónak helyet adó eszköz azonosítója
g	Fájltulajdonos csoportazonosítója
i	Fájlleíró (i-node)
l	A fájl mérete nő („növekvő fájl")
m	Módosítás ideje
n	Hivatkozások száma (fájlleírók hivatkozásszámlálója)
p	Engedélyek és módosító bitek
r	A fájlleírók által hivatkozott eszköz azonosítója (csak eszközfájlokra érvényes)
s	Fájlméret
t	Fájltípus
u	Fájltulajdonos felhasználóazonosítója
C	CRC-32 ellenőrzőösszeg (nem biztonságos, de gyors)
H	Haval-ellenőrzőösszeg (biztonságos, de lassú)
M	MD5-ellenőrzőösszeg (biztonságos, de lassú)
S	SHA-ellenőrzőösszeg (biztonságos, de lassú)

2. táblázat **Előre megadott Tripwire tulajdonságmaszk-változók**

ReadOnly	Széles körben elérhető, de írásvédett fájlok. Érték: +pinugtsdbmCM-rlacSH
Dynamic	A felhasználók saját könyvtárának és a gyakran változó fájlok megfigyelésére használható. Érték: +pinugtd-srlbamcCMSH
Growing	Olyan fájlokhoz való, amelyek csak nőhetnek. Érték: +pinugtdl-srbamcCMSH
Device	Jó az eszközökhöz és azokhoz a fájlokhoz, amelyeknek csak a tulajdonságait kell ellenőrizni, a tartalmát nem. Érték: +pugsdr-intlbamcCMSH
IgnoreAll	Ellenőrzi a fájl meglétét vagy hiányát, de semmi mást. Érték: pinugtsdrlbamcCMSH
IgnoreNone	Mindent ellenőriz. Egyéni maszkok létrehozására használható, például mymask = $(IgnoreNone) -ar; Érték: +pinugtsdrbamcCMSH-l

tal kapcsolatos jelentések a mick@uselessjunk.com címre érkeznek. A tulajdonságokat vesszővel választjuk el, továbbá az összes szabály pontosvesszővel ér véget.

Tulajdonságokat az egyes szabályokhoz külön is hozzárendelhetünk. A csoport első szabályánál a `recurse 1`-re van állítva, ami azt jelenti, hogy a /home/mick/www könyvtárat egy szintmélységig kell ellenőrizni (azaz a könyvtárat magát és közvetlenül alatta mindent, de semmi többet). Megjegyzendő, hogy a könyvtárakat a program alapértelmezés szerint teljes mélységükig ellenőrzi, ha a `recurse` tulajdonság alapértéke *True*.

A szabálykifejezésben megadott tulajdonságok általában felülbírálják szabálycsoport felett zárójelben megadottakat. Kivétel az `emailto` tulajdonság, amely összeadódó vonással bír: ha a csoportnak létezik közös levélcíme és a csoport egyik szabályának másik levélcíme, akkor az ahhoz a szabályhoz tartozó jelentéseket mindkét címre elküldi. Mindössze néhány tulajdonság létezik: `rulename`, `severity`, `emailto` és `recurse`. A tulajdonságokról további ismereteket a *Kapcsolódó címek* részből kiindulva szerezhetsz.

A MickWeb-csoport tulajdonságai után néhány szabály következik. Figyeld meg, miként használjuk a változókat az objektumok (a fájlok és a könyvtárak a Tripwire szóhasználatában) és a tulajdonságmaszkok megadására. Valójában egyik szabály sem használja a tulajdonságmaszkok „kézzel írott" formáját. Ez a bevett gyakorlat, amely elfogadható.

Közvetlenül az első szabály alatt, amely a Tripwire-rel közli, hogy WWW könyvtáram első szintjét írásvédettként kezelje, egy felkiáltójellel kezdődő kifejezés található. Ezt a kifejezést megállási pontnak hívják, feladata a szabály alól való kivételek megadása. Ebben az esetben a megállási pont arra utasítja a Tripwire-t, hogy ne vegye figyelembe a /home/mick/www/guestbook.html változásait. A tulajdonságok nem vonatkoznak a megállási pontokra (és nem is szükséges megadni őket).

Íme, itt egy teljes házirendfájl (legalábbis műszaki értelemben, mivel egyáltalán nem ellenőrzi a rendszer binárisait – az igazi házirendek sokkal hosszabbak). Az *1. listán* a szétvágatlan változatot olvashatjuk.

Talán észrevetted, hogy az egész fájl csak egy közvetlen hivatkozást tartalmaz egy tulajdonságmaszkra: a SEC_INVARIANT változó értéke `+tpug`. Mit is értsünk ez alatt?

A tulajdonságmaszk fájl- vagy könyvtártulajdonság, amelyet az adott objektumra nézve ellenőrizni kell vagy figyelmen kívül kell hagyni. A pluszjel után következő tulajdonságokat kell ellenőrizni, a mínusz utániakat pedig figyelmen kívül kell hagyni. A tulajdonságok rövidítései az *1. táblázatban* olvashatók.

A Tripwire leírása részletesen tárgyalja ezeket a tulajdonságokat. Ha nem ismered valamelyik titokzatos fájltulajdonságot (például: fájlleíró hivatkozások számlálója), akkor olvasd el Card, T'so és Tweedie „Design and Implementation of the Second Extended Filesystem" című cikkét (lásd még *Kapcsolódó címek*). Az ellenőrzőösszegekről csak annyit, hogy egy szabályban általában nem érdemes egy vagy két biztonságos ellenőrzőösszegnél többet használni, mert meglehetősen számításigényesek. Másfelől ne hagyatkozz teljesen a CRC-32 ellenőrzőösszegre, mert gyors ugyan, de sokkal könnyebb becsapni.

Ahogy már korábban említettem, a Tripwire rendelkezik néhány előre megadott (bedrótozott) változóval, amelyek tulajdonságmaszkokat írnak le. Ezek láthatók a *2. táblázatban*.

A legtöbb esetben egyszerűbb ezeket az előre megadott értékeket használni, mint saját magadnak kirakosgatni egyet. A változókat további tulajdonságokkal egyesítheted, például:

```
/dev/console   -> $(Dynamic) -u ;
# Dynamic, de az UID változhat
# Ez ugyanaz, mint a következő
/dev/console   -> +pingutd-srlbamcCMSH-u
```

Amit Mick kihagyott

A cikkben végig a Tripwire parancsainak „hosszú formáját” használtam. Minden két mínuszjellel (--) kezdődő kapcsolónak létezik „rövid” megfelelője, például a következő két parancs egyenértékű:

```
twadmin --print-cfgfile
twadmin -m f
```

Ha már belejöttél a Tripwire használatába, javaslom, hogy sajátítsd el a rövid formák alkalmazását. Ahogy *Neal Stephenson* rámutat az „In the Beginning Was the Command Line” (Kezdetben volt a parancssor) című esszéjében, az ínhüvelygyulladás a hozzánk hasonló kockafejűek számára olyan, mint a bányászoknál a fekete tüdő. Nem szeretném, ha bárki azt gondolná, hogy a kedves olvasóknál bármelyik kialakulásáért is felelős vagyok.
Kezdetben valószínűleg jobban használhatók a Tripwire angol szavakból összerakott hosszú parancsai. A Tripwire Open Source Reference Card (lásd *Kapcsolódó címek*) megfelelő táblázatot tartalmaz a Tripwire-programcsomag hosszú és rövid kapcsolóiról.

Miután létrehoztál valamit, ami értelmes házirendnek látszik, telepítened kell. A rendszer első Tripwire-házirendjét a következő paranccsal telepítheted:

```
twadmin --create-polfile policyfile.txt
```

A Tripwire beállításának utolsó lépése az adatbázis létrehozása. Fontos tudnunk, hogy semmi értelme olyan rendszeren létrehozni a Tripwire-adatbázist, amely működik egy ideje, mert lehet, hogy már betörtek rá! A Tripwire telepítése, beállítása és elindítása minél közelebb legyen az operációs rendszer telepítéséhez.
Az adatbázis létrehozásához használjuk a `tripwire` parancsot: `tripwire --init`. Ennél egyszerűbb már nem is lehetne, igaz? A `--init` kapcsolót csak új adatbázis létrehozására használd.
Ha a Tripwire-házirendet a későbbiek folyamán meg szeretnéd változtatni, célszerűbb a következő parancsokat használni:

```
twadmin --print-polfile > mypolicy.txt
  # kiírja a pillanatnyi házirendet
vi mypolicy.txt
  # változtasd meg a házirendet
tripwire --update-policy mypolicy.txt
  # telepítsd az új házirendet --
  # NE HASZN`LD ERRE A TWADMINT!
```

Tripwire-ellenőrzések futtatása

Az adatbázis telepítése után rendszeresen futtathatjuk az ellenőrzéseket, amelynek legegyszerűbb módja a `tripwire --check` parancs kiadása. Ez az összes védett fájlt összehasonlítja az adatbázissal, és a jelentést kiírja a képernyőre, valamint egy bináris fájlba. A jelentés a következő paranccsal tekinthető meg:

```
twprint --print-report --report-level N
        ➥--twrfile /path/file
```

ahol `N` egy szám 0-tól 4-ig, 0 az egysoros összefoglaló, 4 az összes részletet tartalmazó jelentés. A /path/file a legfrissebb jelentés teljes elérési útja és neve (a jelentés alapesetben a /var/lib/tripwire/report könyvtárba kerül).

Ha azt szeretnéd, hogy a Tripwire levélben értesítse a házirendben megadott személyeket, az ellenőrzést a következő módon futtasd:

```
tripwire --check --email-report
```

A jelentés ez esetben is megjelenik a szabványos kimeneten és a /var/lib/tripwire/report könyvtárban. Ha a Tripwire RPM-et RedHat 7-esre telepítetted, a rendszer már be van állítva a Tripwire elmaradhatatlan ellenőrző futtatására. Az RPM tartalmazza a `/etc/cron.daily/tripwire-check` parancsfájlt. Ha az emailto tulajdonságot a Tripwire-házirendedben használtad, akkor az utolsó előtti sort a parancsfájlban szerkeszd át, hogy így nézzen ki:

```
test -f /etc/tripwire/tw.cfg &&/usr/sbin/tripwire
➥--check --email-report
```

(A sorban a `--email-report` kapcsoló eredetileg nem szerepelt.)
A Tripwire nem sokat árul el, ha nem rendszeresen futtatod, akár kézzel, akár a cronból/anacronból, vagy a kettő valamilyen együttes alkalmazásával.

Megszegték a szabályokat, mi a teendő?

Mi történik, ha a Tripwire a szabályok megsértését jelenti? Ez rajtad múlik. A szabályok megszegése gyakran a túlságosan megszorító házirend miatt történik, és nem tényleges támadás áll a háttérben. Neked kell eldöntened, melyik veszélyes, és mit teszel ellene.
Ha csak vaklárma volt, a szabályszegés után valószínűleg frissíteni akarod a Tripwire-adatbázist, hogy a megfigyelt fájlok és könyvtárak törvényes változásainak megfeleljen. Két módszer létezik erre.
Az egyik a `tripwire` parancs futtatása frissítési módban:

```
tripwire --update
➥--twrfile/var/lib/tripwire/report/myhost-date.twr
```

Az utolsó érték az a jelentés, amelyet a frissítés alapjául akarsz felhasználni. A parancs megnyitja a jelentést a tw.cfg-ben megadott szövegszerkesztővel, ahol megadhatod, hogy a Tripwire az adatbázisában melyik megváltozott fájlt vagy könyvtárt ne frissítse. Más szavakkal, amikor kilépsz a szövegszerkesztőből, a Tripwire csak azoknak a bejegyzéseknek az ellenőrzőösszegét változtatja meg az adatbázisában, amelyek mellett „x” található. Kezdetben mindegyik mellett ott az „x”.
Íme, egy kivonat egy Tripwire-frissítés munkafolyamatából:

```
Remove the "x" from the adjacent box to prevent
updating the database with the new values for
this object.

Modified:
[x] "/home/mick/www"
```

Ha a bejegyzés mellől letörlöm az „x”-et, kilépek a szövegszerkesztőből és futtatom az ellenőrzést, a /home/mick/www változása újra a jelentésbe kerül, mert az adatbázis nem frissült. Ha a változás megfelel a házirendnek, hagyd ott az „x”-et, ha nem megfelelő, esetleg nem vagy benne biztos, töröld.
A Tripwire-adatbázis frissítésének másik módja, hogy az ellenőrzést „interaktív” módon végezzük, amely után azonnal frissítés következik. Ezért a

```
tripwire --check -interactive
```

ugyanaz, mint a

```
tripwire --check
tripwire
➥--twrfile/var/lib/tripwire/report/reportname.twr
```

de az első azzal az előnnyel bír, hogy nem kell megkeresni a jelentés fájlnevét (ez nehezen található ki, mert az időbélyeget is tartalmazza).
Ha hamis találatokat kapsz, fontold meg házirended finomhangolását. A finomhangolást a *Házirend létrehozása* vagy a *Házirend szerkesztése* fejezetben megismertek szellemében végezd.
Lépj tovább, és nehezítsd meg a betörők dolgát!
Mielőtt írásomat befejezném, szeretnék két ragyogó ötletet adni, amelyeket *Ron Forrestertől,* a Tripwire Open Source projektvezetőjétől hallottam:

1. A tw.cfg-ben mindig legyen MAILNOVIOLATIONS=TRUE, így meghallhatod a Tripwire szívverését, vagyis ha a Tripwire a cronból óránként fut, és több mint egy óráig nem kapsz jelentést, akkor tisztában lehetsz vele, hogy történt valami.
2. Mindig hagyj meg egy-két szabályszegést (mondjuk a /etc/sendmail.st-t), így a behatolónak nehezebb jelentést hamisítania. Igencsak egyszerű olyan jelentést létrehozni, amelyben nincs szabályszegés, de ha akad benne egy-két ismert szabályszegés, a hamisítás már sokkal nehezebb.

Remélem, kezdetnek elég lesz ennyi, hiszen sok mindenről még szó sem esett, mást is csak érintettünk. Higgy nekem, a munka, amit az eszköz kiismerésébe és kezelésének elsajátításába fektetsz, megéri a fáradságot. A Tripwire Open Source Manual (lásd *Kapcsolódó címek*) részletesen és jól tárgyal mindent. Sok szerencsét!

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

Kapcsolódó címek

A Tripwire Open Source projekt lapjai ➲ http://sourceforge.net/projects/tripwire. Itt található a legfrissebb Tripwire Open Source forráskód és leírás.

Tripwire Open Source Manual és Tripwire Open Source Reference Card ➲ http://prdownloads.sourceforge.net/tripwire/tripwire-2.3.0-docs-pdf.tar.gz. Ez a lap kötelező olvasmánynak tekintendő (pdf formátum). Ha ez a hivatkozás nem működik, próbáld ezt: ➲ http://sourceforge.net/project/showfiles.php?group_id=3130.

A Linux-binárisok letöltési helye a Tripwire Open Source ➲ http://www.tripwire.org/.

A Tripwire Academic Source Release letöltési helye ➲ http://www.tripwire.com/downloads/tripwire_asr/index.cfml?

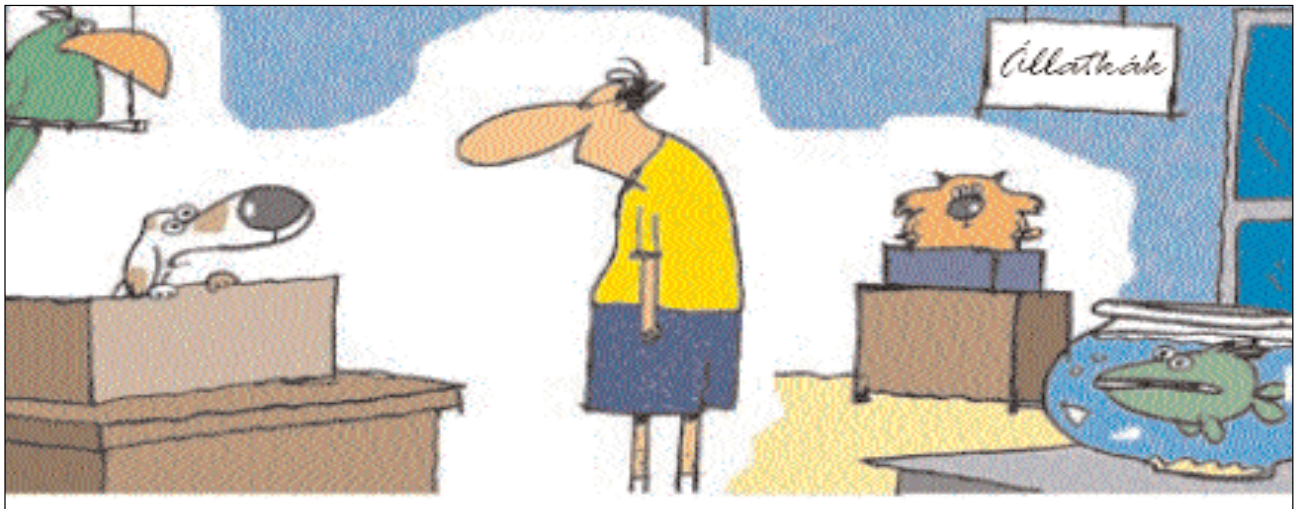
Advanced Intrusion Detection Environment (AIDE) ➲ http://www.cs.tut.fi/~rammer/aide.html.

FCheck ➲ http://www.geocities.com/fcheck2000/.

Perlben írt és könnyedén hordozható sértetlenség-ellenőrző: „Tripwire – The Only Way to Really Know" ➲ http://securityportal.com/topnews/tripwire20000711.html.

Cikk *Jay Beale* (Bastille-Linux fejlesztője) tollából a Tripwire Academic Source Release használatáról „Design and Implementation of the Second Extended Filesystem" címmel ➲ http://web.mit.edu/tytso/

Rémy Card, Theodore T'so és Stephen Tweedie kiváló írása a Linux ext2 fájlrendszeréről. A "Basic File System Concepts" fejezet különösen érdekes lehet a Tripwire felhasználói számára ➲ www/linux/ext2intro.html.



„Vigyél magaddal! Szobatiszta vagyok, csendes és vettem pár számítógépes leckét. Ha segítségre van szükséged a programoddal kapcsolatban mindig számíthatsz rám!"