Ellenség a (tűz)falakon belül

A biztonsággal nem lehet elég sokat foglalkozni. Ez természetesen nem azt jelenti, hogy tűzfalunkat a nap 24 órájában kellene még tovább finomítani, mert tökéletes valószínűleg akkor sem lesz. Előbb-utóbb valakinek úgyis sikerül „fogást találni" rajta , viszont ha már megtörtént a baj, akkor célszerű lenne azt a lehető legrövidebb idő alatt felismerni.

Pontosan ezt a feladatot vállalja magára a ***Tripwire***, mely tulajdonképpen nem más, mint egy fájlváltozás figyelő alkalmazás. Egy átlagos ***Linux*** rendszerben több ezer, vagy akár több tízezer fájl is lehet, ezért nem várható el, hogy ezeket állandóan szemmel tartsuk. Általában pedig már csak akkor kezdünk el hibát keresni, ha valami nem működik megfelelően, hiszen kedvenc operációs rendszerünk stabilitása miatt szerencsére nem túl gyakran kell ezt megtennünk. Pedig az ártó szándékkal gépünket uralni vágyó betörők nagy részének nem az érdeke, hogy rögtön hatalmas károkat okozzanak nekünk.
Jól bevált gyakorlat, hogy egy sikeres betörés után a feltört rendszerben egy ún. „backdoor" (hátsó kapu, hátsó ajtó) programot helyeznek el, mely segítségével később már nem kell újra feltörni a rendszert, hanem a hátrahagyott „bejáraton" keresztül a betörő észrevétlenül, akár hónapokon keresztül is visszalátogathat hozzánk. Mindezt anélkül is megteheti, hogy ez a tudomásunkra jutna.
Akármilyen furcsa is a nálunk ki-be járkáló betörő sokkal kártékonyabb lehet, mintha a támadást rögtön felfedeztük volna. Arról nem is beszélve, hogy először lehet, hogy csak a tűzfalunk esik áldozatul, és csak idővel utána a többi számítógép.

A Tripwire bemutatása

Sok behatolásfigyelő alkalmazás közül most behatóbban a ***Tripwire*** használatával ismerkedünk meg, mely alapvetően a rendszer fájljaiban történt gyanús változásokat figyeli. A ***Tripwire*** az eltárolt szabályok és fájladatbázis alapján átfésüli a fájlrendszerünket, eltárolja és megjeleníti a detektált változásokat. A szabályok segítségével könnyen a testreszabhatjuk azt, hogy mely fájlokon milyen típusú ellenőrzést végezzen az alkalmazás. Ajánlatos a fájlokról az adatbázist még azelőtt elkészíteni, mielőtt az adott számítógépet a hálózatra kötnénk.
Ha végképp szeretnénk biztonságban tudni az adatbázist és az alkalmazást, akkor írjuk ki egy csak olvasható médiára, például egy ***CD***-re vagy pedig tároljuk egy megbízhatónak ítélt távoli számítógépen. Ellenkező esetben a támadónak lehetősége van módosítani a ***Tripwire*** adatbázist, illetve magát az alkalmazást is. Szerencsére ezt azért nem annyira egyszerű megtenni, mert a ***Tripwire*** saját jelszavas védelemmel rendelkezik, illetve az általa generált kimeneti fájlokat digitálisan

1. Lista A Tripwire beüzemelése

```
DIR=/etc/tripwire
SITE_KEY=$DIR/site.key
LOCAL_KEY=$DIR/$(HOSTNAME)-local.key

#központi és helyi kulcs generálása
twadmin -- generate-keys -- site-keyfile $SITE_KEY
twadmin -- generate-keys -- local-keyfile $LOCAL_KEY

#a konfigurációs fájl és szabálygyűjteményt aláírása
twadmin -- create-cfgfile -- cfgfile $DIR/tw.cfg -- site-
➥keyfile $SITE_KEY $DIR/twcfg.txt
twadmin -- create-polfile -- cfgfile $DIR/tw.cfg -- site-
➥keyfile $SITE_KEY $DIR/twpol.txt

#jogosultságok beállítása
cd $DIR
chown root:root $SITE_KEY $LOCAL_KEY tw.cfg tw.pol
chmod 640 $SITE_KEY $LOCAL_KEY tw.cfg tw.pol
```

2. Lista Példa a szabályok kialakítására

```
(
        rulename="Elso kritikus szabalyom",
        severity=100
)
{
        /etc/sudoers            ->$(SEC_CRIT) ;
        /etc/shadow             ->$(SEC_CRIT) ;
        /etc/passwd             ->$(SEC_CRIT) ;
}
```

aláírja. Ezért a fenti megoldások megvalósítását most nem részletezem. Periodikusan futtatva a ***Tripwire-t*** állandó képet kapunk arról, hogy rendszerünkben milyen fájlok változtak meg, természetesen minden nagyobb általunk generált fájl változás (például biztonsági frissítés) után célszerű a kezdeti adatbázist újra frissíteni.

A Tripwire telepítése

Ajánlatos a ***Tripwire*** honlapjáról beszereznünk a legfrissebb stabil verziót (ez a cikk írásának pillanatában a 2.4.0.1). Majd pedig ezt telepíteni forrásból, itt meg kell jegyeznem, hogy a ***Tripwire-nek*** van egy szabad és egy kereskedelmi verziója, ebben a cikkben csak a szabad forráskódú verziót ismertetem. A szokásos módon telepítsük az alkalmazást, majd pedig futtassuk le az ***install.sh*** szkriptet, amennyiben ez nem állna rendelkezésünkre, akkor a következő parancsok segítségével ugyanazt a hatást tudjuk elérni. Először létre kell hoznunk a központi és helyi kulcsot, majd a kulcsok segítségével alá kell írnunk a konfigurációs fájlt és a szabálygyűjteményt. A nem kódolt formátumú konfigurációs állományokat pedig távolítsuk el. Futtassuk le az *1. Listában* látható kódot.
Most már kiadhatjuk a

```
tripwire -init
```

parancsot, melynek hatására elkészül a ***Tripwire*** adatbázis. Ezt az alkalmazás a helyi kulccsal írja alá. Ne felejtsük el a ***twcfg.txt*** és a ***twpol.txt*** állományokat eltávolítani.

A

```
tripwire -check
```

paranccsal pedig bármikor ellenőrizhetjük a fájlok integritását.

A szabályok testreszabása

A ***tw.cfg*** fájl tartalmazza azokat a szabályokat amely alapján az alkalmazás eldönti, hogy az adott fájl a fájladatbázis részét képezze-e vagy sem. Az alapbeállításként használt konfiguráció általában jó, de természetesen sokkal jobb, ha testreszabjuk, már csak azért is, mert megítélésem szerint túl sok fájlt figyel egyszerre a program az alapbeállítással.
Mivel a ***twcfg.txt*** és a ***twpol.txt*** fájlokat előzőleg töröltük le, ezért vissza kell nyerni őket először ahhoz, hogy módosítani tudjuk tartalmukat. A

```
twadmin -- print-cfgfile >
➥/etc/tripwire/twcfg.txt
```

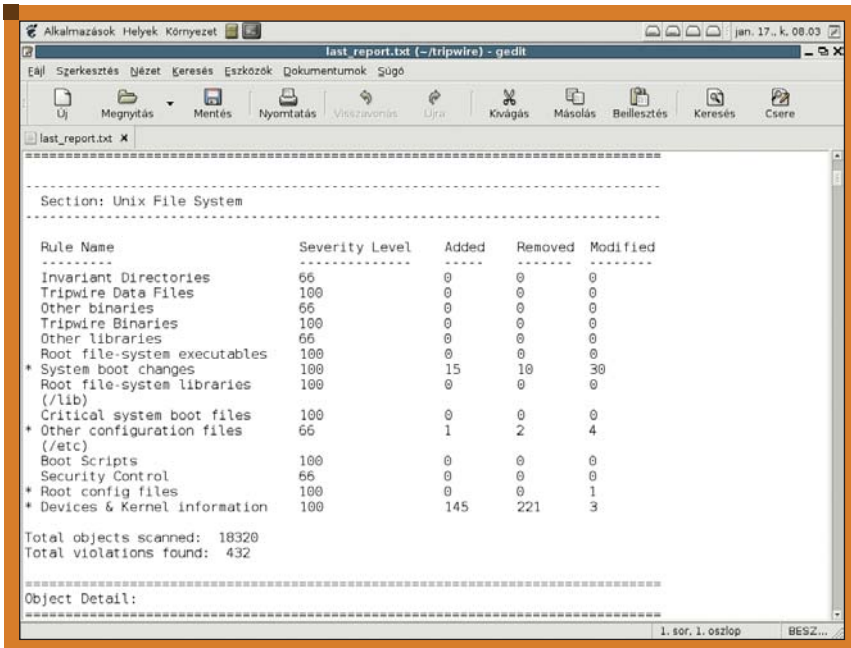
és

```
twadmin -- print-polfile >
➥/etc/tripwire/twpol.txt
```

parancsokkal tudjuk újra számunkra is olvasható formátumra hozni. Beletekintve a ***twcfg.txt*** fájlba láthatjuk, hogy a ***Tripwire*** a fájlokat különféle kategóriák szerint vizsgálja. Gyorsan fussunk végig az egyes kategóriákon.

- `SEC_CRIT`: Olyan fájlok, melyek a rendszer szempontjából kritikusak, és nem változhatnak meg, ilyen például maga a ***Tripwire.***
- `SEC_BIN`: Olyan bináris fájlok, melyeknek nem lenne szabad változni.
- `SEC_CONFIG`: Olyan konfigurációs fájlok, melyek ritkán változnak, de ugyanakkor gyakran használjuk őket.
- `SEC_LOG`: Ide tartoznak azok a fájlok, melyeknek mérete általában folyamatosan nő, de hozzáférési jogaik nem változhatnak.
- `SEC_INVARIANT`: ezek azok a könyvtáraink, melyek hozzáférési jogai nem változhatnak

Természetesen magunk is hozhatunk létre újabb kategóriákat. Az egyes kategóriákat még súlyosságuk szerint is



■ ***1. ábra*** A Tripwire jelentés megtekintése

besoroljuk az alábbi három már előre definiált súlyossági szinttel:

- `SIG_LOW`: Nem kritikus fájlok, melyek esetében általában kicsi a behatolás esélye.
- `SIG_MED`: Nem kritikus fájlok, melyek esetében számottevő a behatolás esélye.
- `SIG_HI`: Kritikus fájlok, melyek a behatolás elsődleges célpontjai lehetnek.

Természetesen újabb súlyossági szinteket is tudunk definiálni. A súlyossági szintek később megkönnyítik a kritikus fájlok gyakoribb ellenőrzését.
A *2. Lista* egy olyan szabályrendszert mutat be, melyhez hasonlóakat mi is könnyen létrehozhatunk.
Az egyes szabályokat a nevük alapján is módunkban áll leellenőriztetni, a fenti példát követve `tripwire -- check -- rule-name "Elso kritikus szabalyom"`. Amennyiben az összes egy bizonyos szintű vagy annál magasabb súlyosságú szabályt akarjuk leellenőrizni, akkor azt a

```
tripwire -- check -- severity
➥40
```

paranccsal tehetjük meg.
Mivel a szabályok bizonyos esetekben fedhetik egymást, ezért az egyes szabályokon belül lehetőségünk van arra, hogy megtiltsunk bizonyos fájlok vagy állományok vizsgálatát. Például az `"Elso kritikus szabalyom"` már leellenőriz 3 állományt a */etc* könyvtárban, ezért célszerű megkeresni azt a szabályt, amely átvizsgálja a teljes */etc* könyvtárat és úgy módosítani, hogy ezeket a fájlokat már később ne ellenőrizze le újra, tehát a kérdéses szabályhoz írjuk hozzá a következőt:

```
!/etc/sudoers
!/etc/passwd
!/etc/shadow
```

A változások ellenőrzése

A *Tripwire* jelentések alapértelmezésben a */var/lib/tripwire/report* könyvtárban találhatóak. A jelentés fájlneve tartalmazza a kiszolgáló nevét és a jelentés elkészítésének időpontját. Mivel az alkalmazás kódoltan és aláírva tárolja el a jelentést, ezért először számunkra is olvasható formátumra kell alakítani. Ezt a

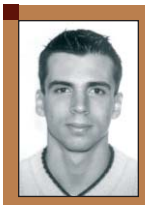
```
twprint -- print-report --
➥twrfile "kiszolgalo-legutolso
➥jelentes.twr"
```

paranccsal tehetjük, mivel a program az alapértelmezett kimenetre dolgozik, ezért inkább irányítsuk át egy fájlba, majd egy megfelelő editor program segítségével tekintsük meg a jelentést.

Az *1. ábrához* hasonló kimenetet fogunk a fájlban találni.
Sajnálatos módon a kimenet elég hosszú, de az elején egy táblázatban vannak összefoglalva a bekövetkezett változások. Természetesen a *Tripwire* beállítható úgy is, hogy minden egyes jelentés alkalmával e-mail formájában értesítsen minket.

Mindennapi használat

Érdekes kérdés ugyanakkor, hogy milyen gyakran futtassuk le az ellenőrzést, hiszen egy-egy terheltebb órában leterhelni a kiszolgálót nem túl szerencsés, ugyanakkor kívánatos volna minél gyakrabban, és nem csak naponta képet kapni a rendszerről.
Mivel a *Tripwire* lehetőséget biztosít arra, hogy ne csak az egész adatbázist, hanem annak bizonyos darabjait ellenőriztessük le egyszerre a rendszerrel. Így például azt a néhány tényleg kritikus fájlt (***index.html***, ***passwd***, ***sudoers***) akár 10 percenként is ellenőrizhetjük, míg a további fájlokat egy kevésbé terhelt napszakban naponta egyszer. Használjuk okosan a lehetőséget a szabály neve vagy az azonos súlyossággal megjelölt szabályok alapján való ellenőrzésre.
Természetesen 100%-os biztonságot még a *Tripwire* használatával sem érünk el, hiszen adódhatnak olyan helyzetek, amikor például frissítjük a rendszert, majd egy rosszindulatú behatoló elvégzi a módosításait bizonyos fájlokon. Utána mi elkészítjük az általunk jónak hitt fájladatbázist, a változásokat természetesen a saját beavatkozásunknak tudjuk be, ugyanakkor azt egy rosszindulatú személy követte el. Ekkor hiába végzünk állandóan ellenőrzést a hibát a *Tripwire* nem fogja jelezni, hiszen egy már eleve hibás fájladatbázissal történik az összehasonlítás.



Horváth Ernő
ernohorvath@gmail.com
24 éves, műszaki informatikus. Három évvel ezelőtt ismerkedett meg komolyabban a Linux rendszerekkel és emellett érdeklődik még a robotika és a biztonságtechnika iránt is. Ha lenne szabadideje sokat kirándulna, biciklizne és filmeket nézne.