A fájlrendszer sértetlenségének ellenőrzése CVS-sel

A számítógépre történő behatolás felfedezésének legegyszerűbb módja, ha a fájlrendszer változásait figyeljük.

Erre a feladatra a Tripwire a legjobb ismert megoldás, amely időről időre önműködően átvizsgálja a fájlrendszert, és minden engedély nélküli változást jelent a rendszergazdának. Sokféle változást tud észlelni a Tripwire, többek közt a jogosultságokét, a fájltípusét, a fájlleíró számét (inode), a hivatkozások számáét, a felhasználóazonosítóét (UID) és a csoportazonosítóét (GID), a méretét, a hozzáférés idejéét, a módosítás idejéét, a fájlleíró változásának idejéét és a hamisíthatatlan ujjlenyomat változásáét (amelyet a tíz ujjlenyomat-készítő függvény bármelyikének alkalmazásával létrehozhatunk, például MD5 és Snefru). Ebben a cikkben bemutatjuk, hogy a Concurrent Versions System (CVS – párhuzamos változatkezelő rendszer) Perllel összegyúrva alkalmassá tehető hasonló fájlrendszer-ellenőrző feladatok ellátására, mint amire a Tripwire képes.

Tripwire

A Tripwire telepítésének leegyszerűsített folyamata az alábbi néhány lépésben foglalható össze:

1. telepítsd az operációs rendszert, és azonnal futtasd a Tripwire-t a konzolról, mielőtt még a gépet a hálózatra kötnéd;
2. a fájlrendszer adatait tartalmazó Tripwire adatbázist tartsd írásvédett adathordozón, a rendszertől elkülönítve;
3. rendszeresen futtasd a Tripwire-t, hogy a jelenlegi fájlrendszer adatai és az eredeti adatbázis közötti különbségek kiderüljenek.

CVS

A CVS-t elsősorban programfejlesztők használják, segítségével a forráskód különböző változatai rendszerbe szervezhetők. Az egyik leghasznosabb tulajdonsága, hogy képes nyomon követni az adott forráskóddarab (vagy közönséges szöveg) időbeli változásait attól az eredeti változattól kezdve, amelyet a kezdetekkor raktak a CVS-raktárba. Ha a fejlesztő változtatni akar a forráskódon, kikéri a raktárból, megváltoztatja, és az eredményként előálló kódot visszateszi a raktárba. A CVS önműködően megnöveli a változatszámot, és nyomon követi a változásokat. A módosítások raktárba tétele után a `cvs diff` parancs segítségével pontosan látható, hogy mi változott az új változatban az előző (vagy bármelyik másik) változathoz képest.
Mostanra nagyjából értjük, hogy milyen lépésekből áll a Tripwire alkalmazása. Megmutatjuk, hogy a CVS hasonlóképpen használható, de egy nagy előnye van a Tripwire-rel szemben: a szöveges beállítófájlok különböző változatainak kezelése. Ha gépedet feltörték, és hozzáadtak egy felhasználót a */etc/passwd* fájlhoz, a Tripwire jelenti a tényt, hogy bizonyos számú fájltulajdonság megváltozott, például a módosítási idő vagy az ujjlenyomat, de nem tudja megmondani, hogy melyik felhasználót adták hozzá.
A behatolásérzékelés fő gondja szokott lenni, hogy a vakriasztások száma gyakran nagyon magas, és ez alól az általunk vizsgált megoldások sem kivételek. A rendszerek számától, és így a bonyolultságtól függően a vakriasztások olyan nagy mértékben is jelentkezhetnek, hogy a rendszergazda nem tudja átlátni és kezelni őket. Ezért ha a felhasználó hozzáadásának példáját vesszük, jó lenne, ha a támadásérzékelő program jelentené, hogy kit adtak hozzá a */etc/passwd* fájlhoz, mert ebből könnyebben eldönthetjük, hogy a hozzáadás jogosan történt-e, vagy valami másról van szó. Ez sok óra megtakarított időt jelenthet, mert ha biztossá válik, hogy a rendszert feltörték, az eredeti állapotok visszaállítására az egyetlen biztos megoldás csak az egész operációs rendszer újratelepítése.

Adatgyűjtés

Megtartjuk a Tripwire módszerét, azaz pillanatfelvételt készítünk a kiindulási fájlrendszerről, és a megfigyelendő rendszertől független adathordozón tároljuk. Ehhez valamilyen módon össze kell gyűjtenünk a távoli rendszerek adatait, és egy helyi CVS-raktárban kell tárolnunk. Hozzunk létre egy CVS-nek szentelt gyűjtőgépet a hálózaton! Minden fájlrendszerfigyelő szolgáltatást ezen a gépen egyetlen felhasználó futtat. A szóba jöhető figyelőfeladatok a következők lehetnek: a nyers adatok begyűjtése, az adatok összehangolása a CVS-raktárral és riasztás küldése jogosulatlan adatváltozás esetén. Az SSH-protokollt használjuk a hálózaton keresztül a távoli gépek adatainak átvitelére. Az egyszerűség kedvéért a támadásérzékelő rendszer felhasználójának RSA-kulcsát minden célrendszer rendszergazdájának ***authorized_keys*** fájljába elhelyezzük. Ez lehetővé teszi, hogy ez a felhasználó rendszergazdaként futtathasson parancsokat a célrendszeren, anélkül, hogy az SSH-n keresztül jelszót kellene megadnia. Most, hogy összeállt az általános kép a gyűjtőgépről, megkezdhetjük az adatok gyűjtését.
Kétféle típusú adatot kell a távoli rendszerekről begyűjtenünk: ASCII beállítófájlokat és a parancsok kimeneteit. A parancsok kimeneteinek begyűjtése általában szélesebb kategória a fájlok begyűjtésénél, hiszen a távoli fájlrendszereket valószínűleg nem akarjuk teljes egészében lemásolni. Többek között a következő két parancs szolgáltat fontos adatokat a megfigyelés szempontjából: `md5sum` (MD5-ujjlenyomatot készít a fájlokhoz) és `find / -type f -perm +6000` (azokat a fájlokat keresi meg, amelyeknek az felhasználóazonosító és csoportazonosító jogosultságbitjei be vannak állítva). Az 1. (33. CD Mazazin/CVS könyvtár) és 2. *listán* olyan Perl-kódot mutatunk be, amely adatokat gyűjt a távoli rendszerekről és figyeli az adatok időbeli változásait.
Az 1. lista Perl-kódjában a `Net::SSH::Perl` modult használjuk három adathalmaz begyűjtésére a `192.168.10.5` című gépről, lehívjuk néhány fontos rendszer bináris MD5-ujjlenyomatát, néhány beállítófájl tar-archívumát és az felhasználóazonosító (UID), illetve csoportazonosító (GID) jogosultságbitekkel rendelkező fájlok listáját. A 7. és 8. sorban adjuk meg a célszámítógép IP-címét és a távoli felhasználó nevét, amelyet a `collector.pl` használ majd a bejelentkezésre. Emlékeztetünk, hogy a helyi felhasználó kulcsa már ott van a távoli rendszeren, ezért a bejelentkezéskor nem szükséges a jelszó

2. lista A cvschecker.pl

```
#!/usr/bin/perl

use File::Find;
use strict;
use vars qw(@Files);

my $email = ·root@localhost·;
my $host = "192.168.10.5";
my $cvsroot = "/usr/local/hbids_cvs";
my $collectorCmd =
➥"/usr/local/bin/collector.pl";

system "cvs -d $cvsroot checkout -ko $host";
chdir $host;
system "$collectorCmd"; ### fÆjlok
➥megszerzØse a $host-r l

find(\&findfiles, "/home/mbr/${host}");

### ellenırizz k, hogy megvÆltoztak-e a
➥fÆjlok
for my $file (@Files) {
    if (`cvs -d $cvsroot commit -m . $file`)
{
        my $cvsfile = $file;
        $cvsfile =~ s|^\S+?$host|$host|;
        $cvsfile = $cvsroot ."/" . $cvsfile
        ➥.",v";
        my $prerev = "";

        ### az rlog hasznÆlatÆval kapjuk meg
        ➥az elızı vÆltozatot
        open RLOG, "rlog $cvsfile|";
        my $found = 0;
        while (<RLOG>) {
            if (/^revision\s(\S+)/) {
                $prerev = $1;
                $found++;
            }
            last if ($found == 2);
        }
        close RLOG;

        ### a k l nbsØgek felder tØse a ·cvs
        ➥diff· hasznÆlatÆval
        my $diff = `cvs -d $cvsroot
        ➥diff -r        \

$prerev $file`;
        open TMP, "> /tmp/change";
        print TMP $diff;
        close TMP;
        $file =~ s|^\S+?$host||g;

        ### a vÆltozÆsok elk ldØse levØlben
        `mail -s "MegvÆltozott fÆjl: $host:
        ➥$file"    \
                                $email <
        ➥/tmp/change`;
    }
}
exit 0;

### az  sszes fÆjl feldolgozÆsa a helyi
➥k nyvtÆrszerkezetben,
### kivØve a k nyvtÆrakat Øs a CVS-fÆjlokat
sub findfiles() {
    my $file = $File::Find::name;
    unless (-d $file || $file =~ /CVS/) {
        push @Files, $file;
    }
    return;
}
```

megadása. A 10–13. sorokban a rendszer szempontjából fontos binárisok kis példalistája szerepel, ezekre kell majd az MD5-ujjlenyomatot elkészíteni. Hasonlóképpen a 15–18. sorokban adjuk meg, hogy mely fájlokról készítünk a helyi gépen biztonsági mentést. A 20–24. sorok egy helyi fájlt rendelnek a távoli gépen lefuttatandó parancshoz, minden parancs kimenete ebben a fájlban fog a helyi gépen tárolódni. A 27–33. sorokban rejlik a gyűjtőalkalmazás lényege, a `n_command()` alprogram hívása (36–49. sorok) a `%Cmds` tömbben tárolt minden parancsra. A `run_command()` minden egyes végrehajtásakor új `Net::SSH:Perl` objektumot hoz létre, amelyen keresztül SSH-kapcsolatot nyit a távoli gép felé, bejelentkezik és végrehajtja az alprogramnak átadott parancsot.

A 2. lista Perl-kódja az elektronikus levelek létrehozásáért felelős, amelyek riasztanak, ha a `collector.pl` által begyűjtött fájlok bármelyike megváltozott. Ez úgy történik, hogy kikérjük a pillanatnyi `192.168.10.5` modult a CVS-raktárból (12. sor), lefuttatjuk a `collector-pl`-t (14. sor), hogy friss változataink legyenek a távoli parancsok kimeneteiből és a fájlokból a helyi könyvtárszerkezetben, majd minden egyes (esetlegesen módosult) fájlt visszahelyezünk a raktárba (20. sor). Ellenőrizzük a `cvs commit` parancsának visszatérési értékét (20. sor). Ebből eldönthetjük, hogy a fájl megváltozott-e, mert a CVS önműködően növeli a változatszámot és követi a változásokat. Ha egy bizonyos fájl megváltozott, a `cvschecker.pl` kiszámítja az előző változatszámot (27–36. sorok), végrehajtja a `cvs diff` parancsot a mostani és az ezt megelőző változat között, hogy megkapjuk a különbségeket (39–40. sorok), és a változásokat (47–48. sorok) elküldi arra a levélcímre, amit a 7. sorban adtunk meg.

Támadások érzékelése

Lássuk a *collector.pl*-t és a *cvschecker.pl*-t munka közben pár támadáson keresztül. Tegyük fel, hogy a célrendszer egy Red Hat 6.2-es gép, a fájlrendszer adatait a gép hálózatba kötése előtt elraktároztuk, és a gép IP-címe `192.168.10.5`.

1. példa

Tegyük fel, hogy a `192.168.10.5` gépet feltörték, és rendszergazdaként a következő parancsot adták ki:

```
# cp /bin/sh /dev/... && chmod 4755 /dev/...
```

Ez a */bin/sh* héjat a */dev/* könyvtárba másolja "..." fájlnévvel, és

beállítja az UID bitet. Mivel a fájl tulajdonosa a rendszergazda, és a rendszer bármelyik felhasználója végrehajthatja, a támadónak csak a */dev/...* elérési utat kell ismernie, és bármilyen parancsot futtathat rendszergazdaként. Magától értetődően értesülni szeretnénk róla, hogy ez történt a `192.168.10.5` rendszerrel. A gyűjtőgépen elindítjuk a `cvschecker.pl`-t, ekkor a root@localhost egy olyan levelet kap, amelyből egyértelműen kiderül, hogy a */dev/...* egy új SUID-fájl:

```
From: hbids@localhost
Subject: MegvÆltozott fÆjl: 192.168.10.5:
suidfiles
To: root@localhost
Date: Sat, 10 Nov 2001 17:35:13 -0500 (EST)

Index: /home/mbr/192.168.10.5/suidfiles
===================================================
RCS file:
/usr/local/hbids_cvs/192.168.10.5/suidfiles,v
retrieving revision 1.3
retrieving revision 1.4
diff -r1.3 -r1.4
4a5
> -rwsr-xr-x 1 root root 512668 Nov 10
➥18:40/dev/...
```

2. példa

Tegyük fel, hogy a támadó rendszergazdaként a következő két parancsot adja ki:

```
# echo "eviluser:x:0:0::/:/bin/bash" >>
➥/etc/passwd
# echo "eviluser::11636:0:99999:7:::" >>
➥/etc/shadow
```

Figyeljük meg, hogy az eviluser UID-je és GID-je a */etc/passwd* fájlban 0, és a */etc/shadow*-ban nincs titkosított jelszóbejegyzés. Emiatt a rendszer bármelyik felhasználója az `su - eviluser` begépelésével jelszó nélkül is rendszergazdai jogosultságokat szerezhet. Ahogy az előbb a `cvschecker.pl` futtatása után, most a következő levelet találjuk a rendszergazda postafiókjában:

```
From: hbids@localhost
Subject: MegvÆltozott fÆjl: 192.168.10.5:
➥/etc/passwd
Delivered-To: root@localhost
Date: Sat, 10 Nov 2001 17:43:17 -0500 (EST)
Index: /home/mbr/192.168.10.5/etc/passwd
===================================================
RCS file:
/usr/local/hbids_cvs/192.168.10.5/etc/passwd,v
retrieving revision 1.2
retrieving revision 1.3
diff -r1.2 -r1.3
26a27
> eviluser:x:0:0::/:/bin/bash
```

és

```
From: hbids@localhost
Subject: MegvÆltozott fÆjl: 192.168.10.5:
➥/etc/shadow
Delivered-To: root@localhost
Date: Sat, 10 Nov 2001 17:43:18 -0500 (EST)

Index: /home/mbr/192.168.10.5/etc/shadow
===================================================
RCS file:
/usr/local/hbids_cvs/192.168.10.5/etc/shadow,v
retrieving revision 1.2
retrieving revision 1.3
diff -r1.2 -r1.3
26a27
> eviluser::11636:0:99999:7:::
```

Összegzés

A fájlrendszer változásainak észlelése hatékony módszer a behatolás felfedezésére. Ebben a cikkben bemutattuk, hogy egyszerű Perl-programok segítségével a CVS házi készítésű támadásérzékelő rendszerré fejleszthető. Mostani munkahelyemen, az USinternetworkingnél, amely egy nagy ASP a marylandi Annapolisban, hasonló (bár sokkal kiterjedtebb) egyedi rendszert használunk. Az USiOasis nevű rendszer több száz hálózatba kötött gép fájlrendszerének sértetlenségét ellenőrzi. A gépeken sokféle operációs rendszer, többek közt Linux, HPUX, Solaris és Windows, és sok különböző kiszolgálóoldali alkalmazás fut. A rendszer háttere egy MySQL-adatbázis, egy elég nagy CVS-raktár és egy egyedi web-, illetve CGI-felület, amelyet nagyrészt Perlben írtak. A változások követése a CVS-raktárral még egy fontos előnnyel jár: létezik egy remek megjelenítő eszköz, a Pythonban írott ViewCVS. Az operációs rendszer és az alkalmazások beállítófájljainak a CVS-ben tartása nemcsak a támadásérzékelés szempontjából előnyös, hanem többek között a hálózati és alkalmazáshibák elhárítására, katasztrófa utáni helyreállításra és a rendszer beállításainak követésére is alkalmas.

Linux Journal 2002. február, 94. szám

Michael Rash
(mbr@cipherdyne.com) vezető biztonsági mérnök az ASP-nél a marylandi Annapolisban. Az University of Marylandon szerezte diplomáját alkalmazott matematikus szakon. 1998 óta foglalkozik a Linuxszal. Szabadidejében a Prince George filharmonikus zenekarban hegedül.

Kapcsolódó címek

AIDE ➲ http://www.cs.tut.fi/~rammer/aide.html
FCheck ➲ http://www.geocities.com/fcheck2000
➲ http://portal.acm.org.
Ha egy gépet feltörtek, nagyon is elképzelhető, hogy egy betölthető rendszermagmodullal magát a rendszermagot is megváltoztatták. Ez a parancsvégrehajtásba történő beavatkozást a rendszermag szintjén teszi lehetővé, ezért rendszer szempontjából lényeges binárisokba nem kell a hátsó kapukat építeni. Lásd például:
➲ http://www.uberhaxr.net/kis
A */etc/passwd* és */etc/shadow* fájlok felépítéséről bővebben a megfelelő man(5) súgóoldalak szólnak.
ViewCVS ➲ http://viewcvs.sourceforge.net