

## AIDE – Ki járt a gépemben?

Ha valaki átrendezné a szobánkat, biztosan észrevennénk. De ha a számítógépünket, vajon azt is? Az AIDE egy olyan alkalmazás, amellyel nyomon követhetjük, mely fájlok, könyvtárak változtak meg a gépünkön.

Sokkal könnyebb egy szabálytalanságot észrevenni egy profi bokszt meccsen, mint azt, ha a támadó egy hátsó ajtót telepített a gépünkre. Az AIDE hasonló alkalmazás, mint a *Linuxvilág 2006. augusztusi* számában ismertetett *Tripwire*, de azzal ellentétben teljesen ingyenes, a GNU GPL licence alatt bocsátották ki.

Az AIDE egy fájl integritás ellenőrző alkalmazás, amely egy adatbázisban tárolja a megadott fájlok, könyvtárak különféle paramétereit, pl. tulajdonos, csoport, méret, időbélyegek és más i-node információk. Ezek mellett még kriptográfiai összegeket (például MD5, SHA1, RMD160) is tárol az egyes bejegyzésekhez. Az adatbázis elkészítése után, amikor ellenőrizzük a gépünket, akkor a megadott fájlok paramétereit összehasonlítja az adatbázisban tároltakkal, és ha eltérést talál, akkor a fájl vagy könyvtár nyilvánvalóan megváltozott, így azt kijelzi nekünk.

Ha egy rosszindulatú felhasználó (*hacker/cracker*) betör a gépünkre, akkor minden bizonnyal telepít egy *root kit*et (jogosulatlan rendszergazda hozzáférést biztosító eszközök), lecseréli a *ps*, *ls*, *netstat*, *who*, stb. programokat, amelyek elárulnák a jelenlétét. Sajnos nem lehetünk akkor sem nyugodtak, ha ezen programok mérete és időbélyegük ismerősek, ezeket ugyanis könnyű manipulálni. Ezekkel szemben sokkal nehezebb hamisítani pl. az MD5 hash értéket. Ma már találtak ütközést (*collision*) nem csak ebben, de az SHA1-ben is. Szerencsére azt azonban rendkívül nehéz megol-

dani, hogy például az *ls* programnak mind az MD5, mind az SHA1, de még az RMD160 összege is változatlan maradjon, és még úgy is működjön, hogy az előbb említett fájl méret és időbélyeg paraméterek is megegyezzenek az eredetiekkel. Ezért az AIDE használatával igen nagy bizonyossággal meg tudjuk mondani egy fájlról, hogy az megváltozott-e vagy sem. Ha paranoiásabbak vagyunk az átlagnál, akkor érdemes, magát az AIDE programot is ellenőrizni. Ebben az esetben másoljuk azt át egy másik (és megbízható) gépre, és ott nézzük meg az MD5, SHA1, stb. ellenőrző összegeit.

Nagyon fontos, hogy az adatbázis megbízható legyen. Miután elkészítettük, tehetjük például írásvédetté tett hajlékonylemeze (*gzip*-vel tömörítve éppen ráfér az én *aide.db* fájlom), kiírhatjuk CD-re, vagy közvetlenül a futás előtt letölthetjük egy biztonságos helyről a gépünkre.

Az AIDE fordításához szükséges az *mhash* csomag, telepítsük, ha nincs meg a gépünkön. Töltsük le az AIDE-t a <http://sourceforge.net/projects/aide/> címről. Csomagoljuk ki, fordítsuk le, és telepítsük a szokásos módon:

```
tar zxvf aide-0.11.tar.gz
cd aide-0.11
./configure --with-zlib
make
su -c 'make install'
```

### Használat

Készítsünk el egy példa konfigurációs fájlt *1.conf* néven, hogy kipróbálhassuk az AIDE képességeit!

Ha elkészültünk, akkor adjuk ki az

```
aide --config-check -c 1.conf
```

parancsot, amely leellenőrzi a konfigurációs fájlt, és kiírja, ha hibát talált. Következő lépésben hozzuk létre az AIDE adatbázisát, amihez adjuk ki az

```
aide -i -c 1.conf
```

parancsot. Ha minden rendben ment, akkor az alábbi üzenetet láthatjuk:

```
AIDE, version 0.11
```

1. Lista Egy példa konfigurációs fájl (1.conf)

```
@@define TOPDIR /

database=file:aide.db
database_out=file:aide.db.new

verbose=5

config_version=0.0.1
report_url=stdout

All=R+a+sha1+rmd160
Norm=s+n+b+md5+sha1+rmd160
MyRule1=R+sha1

# ezt a könyvtárat
# ellenőrizzük
/home/sj/temp/aide-0.11
➔ MyRule1
```

2. Lista Az AIDE detektált egy ismeretlen fájlt

```
AIDE found differences
↳ between database and
↳ filesystem!!
Config version used: 0.0.1
Start timestamp: 2006-08-29
↳ 15:22:07

Summary:
  Total number of files: 162
  Added files:           1
  Removed files:        0
  Changed files:        1
```

-----  
Added files:  
-----

added:/home/sj/temp/  
↳ aide-0.11/teszt

-----  
Changed files:  
-----

changed:/home/sj/temp/  
↳ aide-0.11

-----  
Detailed information about  
↳ changes:  
-----

```
Directory: /home/sj/temp/
↳ aide-0.11
Mtime: 2006-08-29 15:03:02,
↳ 2006-08-29 15:22:05
Ctime: 2006-08-29 15:03:02,
↳ 2006-08-29 15:22:05
```

```
### AIDE database at
/root/aide.db.new initialized.
```

Tegyük fel, hogy másnap reggel szeretnénk látni, hogy vajon nem módosított valaki a gépünkön valamit. Ehhez futtassuk le az ellenőrzést:

```
aide -c 1.conf
```

Egy hibátlan eredmény így néz ki:

```
AIDE, version 0.11
```

3. Lista mindent megtudunk a változásokról

```
Detailed information about
↳ changes:
-----
File: /home/sj/temp/
↳ aide-0.11/config.h
Size: 7270, 7296
Mtime: 2006-04-20 09:22:54,
↳ 2006-08-29 15:28:21
Ctime: 2006-04-20 09:22:54,
↳ 2006-08-29 15:28:21
MD5: EouVGM3qvwxFKmSI8wXmhw==,
f39/zzxsR2UycUd2eoxzbg==
SHA1: yU6M63ipmXZc+56ZrZy+
↳ SH7fG7I=, mHIFiEzYVfAi
↳ 69127rRh6IuzPXU=
```

```
### All files match AIDE
database. Looks okay!
```

Nézzük meg, hogy mi történik, ha egy idegen állomány kerül a megfigyelt könyvtárba! (Ehhez létrehoztam a `/home/sj/temp/aide-0.11/teszt` fájlt). Futtassuk újra az előző parancsot, amely most az alábbi eredményt adja: **2. lista**.

Az eredmény egy összesítéssel kezdődik (hány fájlt nézett meg, hány változott meg, stb.), majd kiírja az idegen fájl nevét, ill. hogy hol történt a változás (ebben az esetben egy új állomány létrejött). Ha törölünk egy fájlt, azt is hasonló módon adja tudunkra az **AIDE**.

Nézzük meg, mi történik, ha módosítunk egy fájlt! A programot ismét futtatva, a kimenet releváns része így néz ki: **3. Lista**.

Láthatjuk, hogy megváltozott a `config.h` mérete, a fájl *i-node*-ban található időbélyegek és az ellenőrző összegek. Bal oldalon az eredeti paramétereket, míg jobb oldalon a futtatáskor számított értékeket találjuk.

Az `aide.db` egy szöveges állomány, ahol az ellenőrzött fájlok, könyvtárak paraméterei találhatóak, soronként egy bejegyzéssel. Minden bejegyzéshez annyi oszlop (paraméter) tartozik, ahányat a konfigurációs fájlban megadtunk, azaz az állományok neve,

4. Lista Az aide.db fájl szerkezete

```
@@db_spec name lname attr
↳ perm uid gid size mtime
↳ ctime inode lcount md5 sha1

/home/sj/temp/aide-0.11/
↳ INSTALL 0 15293 100644 5001
↳ 100 7975 MTA50Tc1Ndc1Mg==
↳ MTE0NTUxNzUzMg== 1147475 1
↳ 6w12fpYI6temRutxxzbGYA==
↳ K7kxSGFRgktBZL18fCfbVogmpoY=
```

különbéle *i-node* adatok (például tulajdonos, csoport, időbélyegek) és kriptográfiai ellenőrző összegek, úgynevezett *hash* értékek (**4. Lista**).

Egy igazi konfigurációs fájl persze bonyolultabb lehet, mint az **1. Listában** szereplő minta. Alább egy összetettebb konfigurációs fájl látható (**5. Lista**).

A konfigurációs fájlban szereplő jelek (például R, L, >) alapértelmezett szabályok, amelyek részletes listája és jelentésük megtalálható az `aide`-vel érkező minta `aide.conf` állományban.

**Tippek**

Az **AIDE**-t a legjobb, ha már rögtön a rendszer telepítése után élesítjük, amikor már minden szoftver telepítve van, de felhasználók még nincsenek a gépen. Célszerű, ha az `aide.db` állományt elkészítése után átmásoljuk egy távoli gépre, és ellenőrzés előtt onnan töltjük le (például `wget` programmal), esetleg összevethetjük a saját és a távoli gépen lévő adatbázisokat. Az is fontos, hogy a rendszeres jelentések eredményét ne tudja a támadó manipulálni, ezért e-mailben is elküldhetjük saját címünkre. A következő verziókban ez valószínűleg beépített funkció lesz. Addig pedig próbáljuk ki, hogy az `aide.conf` fájlban két `report_url` paramétert adunk meg:

```
report_url=stderr
report_url=stdout
```

Majd így futtassuk az `aide` parancsot, akár *cron* feladatként:

```
2 2 * * * /usr/local/bin/aide
↳ -c /usr/local/etc/aide.conf
```

## 5. Lista Egy gyakorlati példa

```

@@define TOPDIR / /boot R
                                     /lib R
# az adatbázis helye /sbin R
database=file:@{TOPDIR}/root/a
ide.db /service R

# a létrehozott adatbázis helye =/tmp L
#database_out=sql:host:port:dat =/tmp/.ICE-unix L
abase:login_name:passwd:table =/tmp/.X11-unix L
database_out=file:/root/aide.db /tmp/.X0-lock L
.new /usr R

# ne tömörítse az aide.db
# fájl /var R
gzip_dbout=no /var/lib R

# mennyi üzenetet írjon ki /var/spool L
# futás közben
verbose=5 =/var/log$ R
/var/log/packages R

# a konfigurációs fájl verziója /var/log/setup R
config_version=0.0.1 /var/log/scripts R

# a jelentést ide írja # a naplófájlok
# lehet még pl. /var/log/*log >
# syslog:LOG_AUTH, stderr /var/log/btmp >
report_url=stdout /var/log/cron >
/var/log/debug >
# néhány saját ellenőrző /var/log/dmesg >
# szabály /var/log/messages >
All=R+a+sha1+rmd160 /var/log/secure >
Norm=s+n+b+md5+sha1+rmd160 /var/log/spooler >
MyRule1=R+sha1 /var/log/wtmp >
/etc R
/bin R
    
```

```

2>/root/jelentes_`date
+%Y%m%d` | /bin/mail -s
"date +%Y%m%d` aide
jelentes" email@cimunk.hu
    
```

A *security-l* listán olvastam egy olyan fájlintegritás ellenőrző megoldásról (amely tetszőleges implementációval megoldható, például *tripwire*, *aide*, *samhain*), ahol egy megbízható gép *ssh*-val bejelentkezik a vizsgálandó gépre, átmásolja a program adatbázisát és magát a program binárisát, lefuttatja az ellenőrzést, annak eredményétől függően küld pl. értesítést, majd letörli az ideiglenes állományait.

Az *AIDE* használata egyéb programokkal is kombinálható.

A *nagios*szal együtt könnyen megoldható az, hogy egyrészt az ellenőrzés automatikus legyen, ill. hogy behatolás detektálása esetén riasztást küldjön.

### Hiányosságok

Más integritás ellenőrző alkalmazásokkal ellentétben az *AIDE* nem tud az adatbázisban csak bizonyos állományokra ellenőrizni. Az összes fájl és könyvtár vizsgálata pedig hosszadalmas lehet. Azonban ez a probléma egyszerűen megoldható, ha több konfigurációs fájl és adatbázist tartunk, és mindig az éppen szükségessé használjuk.

Egy másik apróság, hogy – például a *Tripwire* programmal ellentétben –

az *AIDE* nem tudja súlyosság szerint rangsorolni a változásokat. Ezt némi- leg enyhít(het)ti az, hogy bizonyára az *AIDE* csak egy biztonsági eszköz a sok közül, amivel gépünket egyébként is védjük.

További gond lehet, hogy az *AIDE* adatbázisa nem védett módosítás ellen, sőt nem is képes detektálni, ha az *aide.db* fájl megváltozott. Itt azonban jól jön az adatbázis egyszerűsége: kézzel beleírhatjuk az *aide.db*-re vonatkozó adatokat. A *samhain* például démon módban is képes futni, és mind az elindításakor, mind a leállításakor egy jól beazonosítható jelet hagy, így nem lehet magát az integritás vizsgáló programot lecserélni. Ez a funkció sem található meg az *AIDE*-nél.

Az említett *samhain* érdekessége még, hogy fordításkor egy 64-bites egyedi azonosító képződik minden binárisában. Ha a támadó le is tudja cserélni a programot, a naplófájlokban, email üzenetekben látszani fog, hogy az nem az eredeti program.

Ha sok gépet felügyelünk, akkor bizonyára jól jönne egy központi monitorozó és menedzselési lehetőség, de sajnos ezt az *AIDE*, jelenleg nem támogatja, magunknak kell megfelelő körítéssel ezt a funkciót megvalósítanunk.

### Összefoglalás

Az *AIDE* hatékonyan képes felderíteni, hogy mely fájlok változtak meg a gépünkön. Értékes segítséget ad nem csak a behatolás detektálásához, de az utólagos védekezéshez is, segítségével nem csak azt tudjuk meg, hogy vajon betörték-e a gépünkre, de a telepített trójai és egyéb programokat is azonosíthatjuk. Más eszközökkel kombinálva pedig a hiányosságain is enyhíthetünk.



**Sütő János**

(jsuto@freemail.hu)

1997 óta használ Slackware Linux-ot. Szabadidejében a postfix clap nevű vírus- és spam-szűrőjét polírozza.