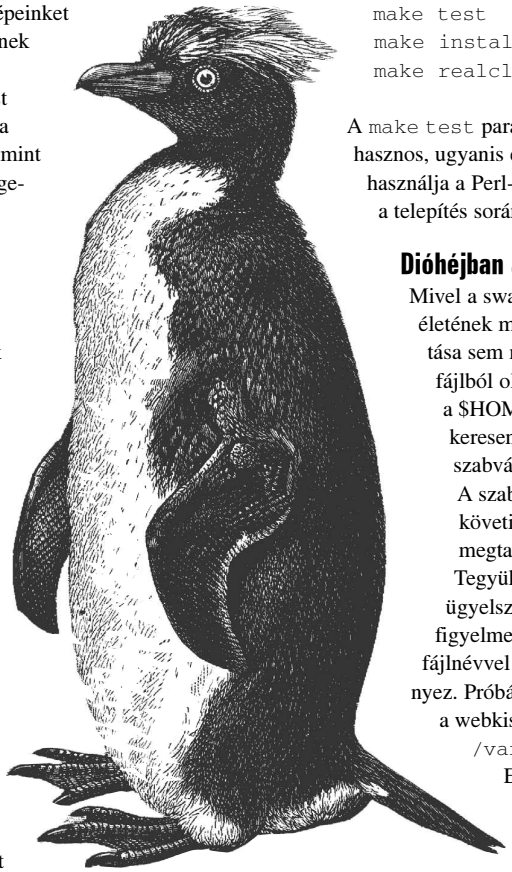
Swatch: önműködő naplófigyelés

A napló tanulmányozásának javasolt módja éber, de lusta emberek számára – így több idejük marad feladataik végrehajtására.

Múltkor a Szaktekintély rovat bemutatott néhány programot, amelyek megvédhetik gépeinket a gonosz szándékú betörőktől. Ezeknek a segédeszközöknek van egy közös feladatuk: a naplózás. Legalább olyan fontos lehet tudomást szerezni a betörési kísérletről, mint távol tartani a betörőket. De kinek akad ideje és türelme – nap mint nap – az általa ellenőrzött összes rendszeren rengeteg nagyméretű és többnyire érdektelen adatot tartalmazó naplófájlt végigolvasni?

A swatchnak (Simple WATCHer) van. A száz százalékig Perlben fejlesztett swatch már az írás pillanatától figyeli a naplófájlokat és rögvest a tettek mezejére lép, amint olyasmit talál, aminek a keresésére megkértük. Ez az egyszerű, rugalmas és hasznos eszköz minden egészséges mértékben óvatos rendszergazda számára kötelezően telepítendő program.



A swatch telepítése

A swatch két módon telepíthető. Az első természetesen az általad kedvelt Linux-terjesztéshez adott bináris csomag, ha van ilyen. A Mandrake jelenlegi változata magában foglalja a swatch RPM-csomagot, de más népszerű terjesztések (például RedHat, SuSE, Slackware vagy Debian) nem tartalmazzák. Ez nem is baj, mert a swatch telepítésének második módja elég érdekes. A swatch forrása, amely a

➲ http://www.stanford.edu/~atkins/swatch

címről tölthető le, a Makefile.PL nevű bonyolult parancsfájlt tartalmazza. Ez a parancsfájl önműködően ellenőrzi a szükséges Perl-modulok meglétét, a Perl 5 CPAN lehetőségének segítségével letölti és telepíti a hiányzó modulokat, majd előállítja a Makefile-t, amellyel a swatch lefordítható.

Miután a szükséges modulokat telepítetted, akár a swatch Makefile.PL parancsfájljával, akár kézzel (és utána futtatod a Makefile.PL-t), a Makefile.PL a következő eredményt adja vissza:

```
[root@barrelofun swatch-3.0.1]# perl Makefile.PL
Checking for Time::HiRes 1.12 ... ok
Checking for Date::Calc ... ok
Checking for Date::Format ... ok
Checking for File::Tail ... ok
Checking if your kit is complete...
Looks good
Writing Makefile for swatch
[root@barrelofun swatch-3.0.1]#
```

Miután a Makefile.PL a swatch Makefile-ját elkészítette, a következő parancsokkal fordítható le és telepíthető:

```
make
make test
make install
make realclean
```

A `make test` parancsot nem kötelező kiadni, de hasznos, ugyanis ez biztosítja, hogy a swatch helyesen használja a Perl-modulokat, így megelőzhetjük a telepítés során esetleg fellépő hibákat.

Dióhéjban a swatch beállításáról

Mivel a swatch programot a rendszergazdák életének megkönnyítésére hozták létre, beállítása sem nehéz. Beállításait a swatch egyetlen fájlból olvassa: alapértelmezés szerint ez a $HOME/.swatchrc. A fájl a swatch által keresendő szövegmintákat tartalmazza szabványos kifejezések formájában.

A szabványos kifejezést azok a műveletek követik, amelyeket a swatchnak a szöveg megtalálásakor végre kell hajtania.

Tegyük fel, hogy egy webkiszolgálót felügyelsz és azt szeretnéd, hogy a rendszer figyelmeztessen, ha valaki egy nagyon hosszú fájlnévvel tártúlcsordulásos támadást kezdeményez. Próbáld ki saját magad, mit ír ilyenkor a webkiszolgáló a naplóba: add ki a `tail /var/apache/error.log` parancsot! Ebből kiderül, hogy a naplóbejegyzés tartalmazza a *File name too long* karakterláncot. Tegyük fel továbbá azt is, hogy levélben szeretnél értesülni az eseményről. Ekkor ezt kell beírnod a .swatchrc fájlodba:

```
watchfor /File name too long/
    mail addresses=mick\@visi.com,
    subject=BufferOverflow_attempt
```

Látható, hogy a bejegyzés a *watchfor* kifejezéssel kezdődik, amelyet egy szabványos kifejezés követ. Ha nem magas szinten ismered a szabványos kifejezéseket (de ugye tervezed a megtanulásukat?), ne aggódj, a legegyszerűbb esetben elegendő – két perjel között – a keresett szöveg egy darabját megadni.

A swatch tetszőleges számú műveletet képes végrehajtani, ha szabványos kifejezésedhez illeszkedést talált. Ebben a példában arra utasítottuk a programot, hogy küldjön levelet a mick@visi.com címre *BufferOverflow_attempt* tárggyal. Vegyük észre a fordított perjelet a @ jel előtt, ha ez nem lenne, a Perl a @-ot különleges karakterként értelmezné. Ha a tárgyban szóközöket szeretnél használni, azokat is meg kell védeni a fordított perjellel például: *subject=Buffer\ Overflow\ attempt*. A levélküldés mellett más műveleteket is használhatunk (lásd még a *táblázatunkban*).

A beállításokról és a műveletekről további részleteket a swatch(1) súgóoldal tartalmaz.
Bővítsük tovább az előbbi példánkat! Tegyük fel, hogy a tártúlcsordulási kísérletek kapcsán küldendő levélen kívül azt is tudni szeretnéd, hogy mikor olvas valaki egy adott weboldalt, de csak akkor, ha éppen be vagy jelentkezve a konzolra. Ugyanebbe a .swatchrc fájlba valami ilyesmit írj be:

```
watchfor /wuzza.html/
       echo=red bell 2
```

Az esemény bekövetkeztekor a gép sípol és a konzolra ír.
Fontos, hogy ezeket az üzeneteket csak akkor fogod látni (és a sípszót hallani), ha ugyanabba a héj-munkafolyamatba jelenkeztél be a konzolra, amelyből a swatch programot is indítottad. Ha kilépsz, elmégy ebédelni és visszatértedkor ismét bejelentkezel, a régi munkafolyamatban elindított swatchfolyamatok üzeneteit nem fogod látni, annak ellenére, hogy ezek a folyamatok még ekkor is futnak.
Ha nem vagy benne biztos, add hozzá a `mail` vagy más, konzoltól független műveletet, például az `exec`-et, amely egy figyelmeztető parancsfájlt indít el. Természetesen csak akkor tedd ezt, ha a kérdéses esemény tényleg annyira fontos.
A figyelmes olvasó már minden bizonnyal észrevette, hogy az előző példa csak olyan Apache webkiszolgálóval működik, amely a hibaüzeneteket és az elérési naplót ugyanabba a fájlba írja. A különböző megfigyelt fájlokhoz nem társítottunk különböző műveleteket, nem is tehettük volna. Mi történik akkor, ha egynél több naplófájlt szeretnél a swatch programmal megfigyeltetni?
Semmi gond: bár minden egyes .swatchrc fájl csak egy megfigyelendő fájlt ír le, nincs akadálya annak, hogy a swatchot több példányban futtasd – mindegyiket a saját .swatchrc fájljával. Másként fogalmazva: a swatch beállításainál a .swatchrc az alapértelmezett neve, ez azonban más is lehet.
Az előző két példát úgy mentheted két fájlba, hogy az előző egyszerű .swatchrc fájl tartalmát például a .swatchrc.hterror fájlba teszed, az előző *watchfor* bejegyzés sorait pedig mondjuk a .swatchrc.htaccess fájlba mented.

A swatch beállításai haladók számára

Eddig olyan műveleteket tárgyaltunk, amelyeket minden egyes alkalommal végre kell hajtani, amikor az esemény bekövetkezik. A swatch viselkedését sokféleképpen, finomabban is szabályozhatjuk.
Az első és legkézenfekvőbb dolog, hogy kihasználjuk a keresési minták szabványos kifejezés voltát. A szabványos kifejezések, amelyek valójában saját szövegformázó nyelvet alkotnak, hihetetlenül hatékonyak, és nagy részben ezek felelősek a Perl, a sed, a vi és sok más unixos segédeszköz varázslatos működéséért.
Illik legalább egy pár *regex*-trükkel tisztában lenni, ezért ismertetek közülük néhányat. Az egyes számú trükk a vagylagosság, ez a szabványos kifejezéshez a „|” jel formájában a logikai *vagy* műveletet adja. Vegyük a következő szabványos kifejezést:

```
/reject|failed/
```

Ez a kifejezés minden olyan sorra illik, amely a `reject` vagy a `failed` szavakat tartalmazza. A vagylagosságot akkor használd, ha azt szeretnéd, hogy a swatch egynél több kifejezésre vonatkozóan hajtsa végre ugyanazt a műveletet.
A második számú trükk a Perlre jellemző szabváynos kifejezésmódosító: a *kis- és nagybetű egyforma*, amely *per-i* néven is ismert, ugyanis mindig a szabváynos kifejezés záró perjele után következik. A `/reject/i` szabváynos kifejezés minden sorra illeszkedik, amely a *reject* szót tartalmazza, legyen az `Reject`, `REJECT`, `rEjEcT` stb. formában írva. Ez legalább olyan hasznos, mint a vagylagosság, a teljesség kedvéért azonban elárulom, hogy a *per-i* az egyik legszámításigényesebb Perl-módosító. Ha a naplónézegetés és az önmegtámadás stb. ellenére sem vagy száz százalékig biztos az aggodalomra okot adó támadás naplófájlbeli kinézetében, a *per-i* segít a találgatásban.

Néhány művelet, amelyet a swatch elvégezhet

Művelet (kulcsszó)	Leírás
echo=normal, underscore, blue, inverse stb.	Az egyező sort kiírja a konzolra – a beállított különleges szövegmódban (az alapértelmezett mód a „normal”).
bell N	A sort kiírja a konzolra és N-szer sípol (alapértelmezés szerint N=1).
exec command	Végrehajt egy parancsot vagy parancsfájlt.
pipe command	A command parancs bemenetére irányítja a kimenetet.
throttle HH:MM:SS	HH:MM:SS ideig vár az egyezés után, és csak ez idő lejárta után hajtja végre a műveletet egy ugyanolyan típusú egyezésre. Így elkerülhető, hogy rövid időn belül bekövetkező nagy mennyiségű swatch-eseménnyel a rendszert akasszák meg.

Amennyiben a szabványos kifejezéseket a lehető legmagasabb szinten szeretnéd kezelni, javaslom *Jeffrey E. F. Friedl* Mastering Regular Expressions című könyvének tanulmányozását.
A swatch szabályozásának másik módja, hogy megadjuk, a nap melyik időpontjában hajtson végre egy adott műveletet. Ezt a művelet után a `when=` kulcsszó beírásával teheted meg. Vegyünk például egy közepesen fontos eseményre vonatkozó .swatchrc bejegyzést, amelyről hétköznap konzolüzenetek formájában szeretnék értesülni, hétvégén azonban levelet óhajtok kapni róla. Ehhez a `when` értékét így kell beállítani:

```
/file system full/
       echo=red
       mail addresses=mick\@visi.com,
       subject=Volume_Full,when=7-1:1-24
```

A `when=` beállítás írásmódja:
`when=napok_tartománya:órák_tartománya`
Láthatjuk, hogy bármikor, amikor *file system full* kerül a naplóba, a swatch a naplóbejegyzést pirossal kiírja a konzolra. Levelet is küld, de csak szombaton („7”) és vasárnap („1”).

A swatch futtatása

A swatch elvárja, hogy aki elindítja, annak a könyvtárában a .swatchrc létezzen. A swatch program ugyanis alapértelmezés szerint ideiglenes fájljait itt is tartja (minden indításkor létrehoz és futtat egy „watcher process” nevű parancsfájlt, amelynek neve egy pontra, valamint a létrehozó swatch-folyamat PID-jére végződik).

Megengedjük-e a Perlnek, hogy a saját moduljait letöltse és telepítse?

A Comprehensive Perl Archive Network (CPAN) Perl programgyűjteményeket összefogó világméretű hálózat. A Perl 5.6.x változata olyan modulokat (többek között a CPAN és a CPAN::FirstTime) tartalmaz, amelyek segítségével Perl-modulokat tölthetünk le, ellenőrizhetünk, sőt akár le is fordíthatunk gcc-vel az Interneten elhelyezkedő CPAN-helyekről. A CPAN és a Perl CPAN lehetőségeinek részletes tárgyalása meghaladja e cikk kereteit, de röviden bemutatom, továbbá egy fontos dologra szintén felhívom a figyelmedet.
Először lássuk a használat módját. Az Example::Module modult (ez nem létező Perl-modul) így telepítheted:

```
perl -MCPAN -e ´install Example::Module"
```

Ha a `-MCPAN` kapcsolót ez alkalommal használtad először, a CPAN::FirstTime modul indul el, és néhány kérdést kell megválaszolnod a modulok letöltésével és telepítésével kapcsolatban. Ezek jól megfogalmazott kérdések és az alapértelmezett válaszok is értelmesek. Mégis fontos odafigyelni a parancs kimenetére, mert a telepítendő modul más moduloktól függhet, esetleg vissza kell menni és végrehajtani például a

```
perl -MCPAN -e ´install Example::PreRequisite"
```

parancsot, mielőtt az első modul telepítését másodszorra is megkísérelnéd.
És most jöjjön a figyelmeztetés: a CPAN biztonsági szempontból se nem sokkal jobb, se nem sokkal rosszabb az egyéb internetes programforrásokhoz képest. Véleményem szerint a CPAN-segédprogramok viszonylag biztonságosak, mivel telepítés előtt minden letöltött modulra kiszámítják az ellenőrzőösszeget, amely erős tikosításnak számító MD5 kódot tartalmaz.
Még ha feltételezzük is, hogy egy adott csomag ellenőrzőösszegét nem cserélik le, amikor a csomagot megpiszkálják (erős feltevés), ez akkor is csak a program engedély nélküli módosítása ellen véd, miután a szerzője feltöltötte a CPAN-ra. Egy gonosz CPAN-fejlesztőt (bárki bejegyeztetheti magát) semmi sem akadályozhat meg abban, hogy kártékony kódot töltsön fel érvényes ellenőrzőösszeggel. Természetesen ugyanez a helyzet a *SourceForge* vagy a *Freshmeat* esetében is.
Remélve, hogy ezt a részt nem hangsúlyozom túl, ha mégis óvakodni szeretnél, akkor a legbiztonságosabban így telepíthetsz egy adott Perl-modult:

1. Keresd meg a modult a ➲ http://search.cpan.org/ címen.
2. Kövesd a hivatkozást a modul CPAN oldalára.
3. Ne a CPAN-ról töltsd le a modult, hanem fejlesztőjének hivatalos weboldaláról, amelyet az *Author Information* alatt adott meg, a 2. pontban már említett weboldalon.
4. Töltsd le a fejlesztő által biztosított ellenőrzőösszeget az imént letöltött tarcsomaghoz (természetesen csak akkor, ha sikerül rábukkanni).
5. Használd a gpg, MD5 stb. programokat a tarcsomag ellenőrzésére (erről akár önálló cikket is lehetne írni a Szaktekintély rovatba). Rengeteg különböző sértetlenség-ellenőrzőt használnak a programterjesztők az MD5-ön kívül, de egyiket sem használja túl sok végfelhasználó.
6. Bontsd ki a tarcsomagot, például: `tar -xzvf groovyperlmod.tar.gz`.
7. Ha az igaz utat követő üldözési mániás kung-fu mester vagy, esetleg azzá szeretnél válni, nézd át a forráskódot hibák után kutatva, jelentsd az eredményt a fejlesztőnek, illetve kürtöld világgá, és gyűjtsd be a nyílt forrás közösségének nagyrabecsülését és hálá-ját. (A nyílt forráskód csak akkor igazán nyílt, ha akad valaki, aki veszi a fáradságot és elolvassa!)
8. Kövesd a modul fordítási és telepítési utasításait, ezeket általában az INSTALL fájl tartalmazza. A teendő többnyire a következő:

```
perl ./Makefile.PL
make
make test
make install
```

Ha a szükséges modulokat a swatch Makefile.PL parancsfájlja hozta a tudomásodra, és az üldözési mániás telepítési módszert szeretnéd használni, akkor le kell írnod a szükséges modulok nevét és „meg kell ölnöd" a parancsfájlt (a jó öreg Ctrl+C-vel) – mielőtt telepíted a modulokat és újra futtatod a swatch Makefile.PL-jét.

swatch-modulok

Perl Module	*RedHat 7 RPM*	*Debian „deb" csomag*
Date::Calc	perl-Date-Calc	libdate-calc-perl
Time::HiRes	perl-Time-HiRes	libtime-hires-perl
Date::Format	perl-TimeDate	libtimedate-perl
File::Tail	perl-File-Tail	libfile-tail-perl

Akad egy harmadik módja is a hiányzó Perl-modulok telepítésének: Linux-változatod FTP-kiszolgálójáról vagy CD-ROM-járól is végrehajthatod. Bár egyik sem éri el a CPAN által nyújtott választékot, a legtöbb Linux-változat tartalmazza a legnépszerűbb Perl-modulok csomagolt változatait. Az alábbi táblázat megadja a swatch-hoz szükséges modulok csomagneveit a RedHat 7 és a Debian 2.2 rendszerekre. A fentiek nem tűnnek a swatch programhoz kapcsolódó ismeretek-nek, nem is azok, de fontosak – egyre több hasznos segédprogram jelenik meg Perl-modulként vagy olyan Perl-parancsfájlként, amely Perl-moduloktól függ, ezért valószínűleg nem a swatch lesz az utolsó alkalmazás, amely a Makefile.PL segítségével telepíthető. Hidd el nekem, megérte ezt a liternyi „tintát" elhasználni arra, hogy a modulok lelkivilágát megismertessem.

A `-c beállítások_útvonala` és a `--script-dir=útvonal` kapcsolókkal a swatch beállításait tartalmazó fájlnak és a parancsfájloknak egyéb helyek is megadhatók. Egyiket se tartsd olyan könyvtárban, amely mindenki által írható! A leghelyesebb, ha ezeket csak a fájlok tulajdonosa tudja olvasni. Ha például azt szeretném, hogy a swatch egyéni beállításaimat a /var/log könyvtárból vegye, és ezt a könyvtárat használja a parancsfájl tárolására is, akkor a következő parancsot kell kiadnom:

```
swatch -c /var/log/.swatchrc.access
➥--script-dir=/var/log &
```

Azt is meg kell mondanom a swatchnak, hogy melyik fájlt figyelje, ehhez pedig a `-t fájlnév` kapcsolót használom. Ha a fenti parancsot a /var/log/apache/access_log figyelésére akarnám használni, az alábbi a parancsot szükséges kiadnom:

```
swatch -c /var/log/.swatchrc.access \
        --script-dir=/var/log \
        -t /var/log/apache/access_log &
```

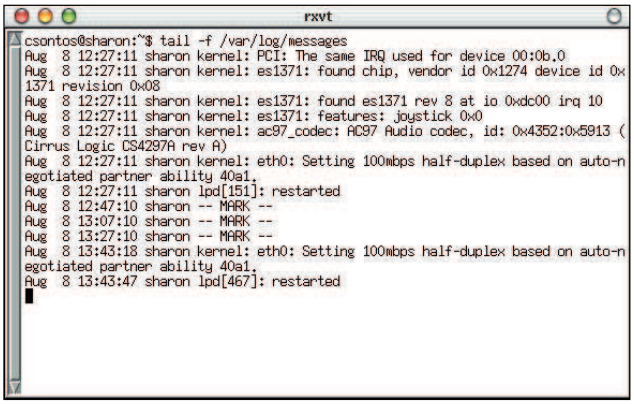
A swatch maga után általában nem takarít ki valami jól, szokása ugyanis, hogy maga mögött hagyja a watcher-process parancsfájlokat. Figyelj erre, és időről időre töröld ezeket a fájlokat a könyvtáradból vagy a `--script-dir` kapcsolóval megadott könyvtárból.
Ha egy időben több fájlt szeretnél megfigyelni, akkor több példányban kell futtatnod a swatchot; legalább a célfájl (a `-t` kapcsoló után) legyen különböző, de ezekre valószínűleg különböző beállításokat is fogsz használni.

A swatch finomhangolása (mindkét irányban)

Miután a swatchot beállítottuk és a program már fut, figyelmünket célszerű az arany középút szabályának betartására fordítanunk. Nem akarjuk, hogy a swatch túl gyakran jelezzen (mindennapos vagy egyszerű műveletekre is riasszon), de azt sem szeretnénk, ha soha semmire nem figyelmeztetne. Vajon mi lehet a helyes megoldás? Annyi különböző válasz létezik, ahányféle dologra a Unixot fel lehet használni.
Akárhogy is, nem kell megmondanom, hogy mi a túlzott részletességű jelentés: meg fogod ismerni, ha találkozol vele. Előfordulhat, hogy egyszer-kétszer megijedsz, amikor olyan események váltanak ki riasztást, amelyek később ártalmatlannak bizonyulnak. Olvasd szorgalmasan a kézikönyveket, állítgasd a .swatchrc-t és haladj tovább!
A másik eshetőséget – amikor túl kevés dolgot figyelsz meg – sokkal nehezebb észrevenni, különösen igaz ez kezdő rendszergazdák esetében. A rendkívüli események, ahogy a nevük is mutatja, ritkán fordulnak elő. Hogyan is vehetnéd észre, miként jelennek meg a naplóban? Első tanácsom, hogy szokd meg a rendszernapló gyakori olvasását, így benyomásod lesz arról, hogyan működik rendszered a mindennapok során.
Még jobb, ha a naplót valós időben olvasod. Ha kiadod a

```
tail -f /var/log/messages
```

parancsot, a rendszernapló utolsó tíz sora kiíródik, és minden ezután következő sor, létrejöttének idejében, egészen addig, amíg a Ctrl+C billentyűkombinációval le nem állítod a `tail` parancs végrehajtását. Ez minden fájl esetén működik, még a gyorsan változó naplófájloknál is.
A másik hasznos dolog a rendszer megtámadása az egyik virtuális konzolon vagy xterm-ben, míg egy másikban a naplófájlokat figyeled. A múlt hónapban és a két hónappal ezelőtt bemutatott Nessus- és Nmap-programok erre a célra tökéletesen megfelelnek. Most azt gondolhatod: „Azt hittem, azért telepítettem a swatchot, hogy ne kelljen a naplófájlokban kotorásznom!” Ez nincs így.
A swatch mérsékli ugyan, de ki nem küszöböli a naplófájlok olvasásának szükségességét.
Miután megkaptad életed első zsebszámológépét, vajon el is felejthetted a négy alapműveletet? Használhatod-e a számológépet, ha nem tudod, hogyan kell összeadni és szorozni? Természetesen nem. Ugyanez vonatkozik a naplófájlok olvasására: nem mondhatod



A tail -f /var/log/messages kimenete

meg a swatchnak, hogy keresse meg azt, amit magad sem tudnál azonosítani, mint ahogyan nem kérdezheted meg azt sem, hogy merre keresd a várost, amelynek elfelejtetted a nevét.

Miért rossz, ha beállítottad a swatchot és elfeledkezel róla?

A fentiek szellemében: ne légy elégedett, ha a swatch csendben marad. Ha a swatch jelzései ritkán jelennek meg, az egyrészt jelentheti azt, hogy a rendszered nem gyakorta van kitéve támadásoknak, másrészt azonban legalább ilyen valószínű, hogy a swatch nem elég nagy területre veti ki a hálóját. Időről-időre pásztázd végig kézzel a naplófájlokat, hogy biztosan ne maradjon ki semmi, és szükség esetén módosítsd a .swatchrc-t.
Végül ne feledd rendszeresen újragondolni a naplóbejegyzéseket létrehozó démonok naplózási beállításait! A swatch nem tud olyan eseményt elkapni, amely nem is kerül be a naplóba. Olvasgasd mind a syslogd(8) súgóoldalt, amely a syslog-démon kezelésének általános leírását tartalmazza, mind a syslogba író programok súgóoldalait, melyekben az ezekre jellemző naplózási beállításokról lelhetsz hasznos tudnivalókat.

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 swatch honlapja
➲ http://www.stanford.edu/~atkins/swatch/
Stephen Hansen és *Todd Atkins* a swatch megalkotói, valamint ők a szerzői a Centralized System Monitoring with Swatch című írásnak, mely a
➲ http://www.stanford.edu/~atkins/swatch/lisa93.html
címen olvasható.
Ajánlott irodalom továbbá *Lance Spitzner* Watching Your Log című írása.
Rövid bevezetés a swatch használatáról, eléhető a
➲ http://www.enteract.com/~lspitz/swatch.html címen.
A cikkben ajánlott könyv: *Jeffrey E. F. Friedl* Mastering Regular Expressions (O'Reilly & Associates, 1998).