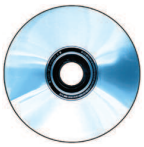
# Rejtett szaglászás-, betörésérzékelés és naplózás

A támadók képtelenek átírni a naplóállományokat, ha kapcsolódni sem tudnak a naplózó géphez. Ismerkedjünk meg a rejtőzködés módozataival!



A Linuxvilág 2001. decemberi számában, a `syslog` rendszernapló beállításáról szóló írásomban már megemlítettem a lopakodó naplózást, vagyis az IP-címmel nem rendelkező központi naplózó kiszolgálógépet, amelyet remekül el lehet rejteni a betolakodók szeme elől.
Ráadásul a naplózókiszolgálók csoportja nem is az egyetlen, amelyik előnyt képes kovácsolni egy kis rejtőzködésből. A betörésérzékelő rendszerek (Network Intrusion Detection System – NIDS) szondái és a szaglászók IP-cím nélkül is tökéletesen működnek, úgyhogy az általuk védett hálózatnál is kevésbé sérülékenyek. E havi írásomban egyetlen IP-cím nélküli hálózati kártyán fogom a sokrétű és nagy teljesítményű Snort háromféle használatát bemutatni: rejtett szaglászóprogramként, rejtett NIDS-szondaként és rejtett naplózóként.
Amennyiben már jól ismered a Snortot, remélem, most megtapasztalod, milyen könnyű is rejtett üzemmódban használni. Ha pedig most készülsz vele megismerkedni, ez a cikk életmentő tanfolyam lesz a számodra. A cikkben szereplő valamennyi parancs és beállítás mind az IP-címmel rendelkező, mind az IP-cím nélküli hálózati kártyákon egyformán jól működik.

## Vajon mire is jó a rejtőzködés?

Az Internethez kapcsolódó számítógépek működtetése kockázatos. Minden esetben, amikor valamilyen szolgáltatást kínálunk, fennáll a lehetősége, hogy egy rosszindulatú felhasználó valamelyik alkalmazás biztonsági résén keresztül kibillenti a rendszert rendes működéséből, vagy egyszerűen olyan mennyiségű szolgáltatás-megtagadással (Denial-of-Service) járó támadással halmozza el, hogy azt már nem lesz többé képes elviselni. A web- és FTP-kiszolgálók, valamint a más végfelhasználói közreműködést feltételező kiszolgálók esetében ezek a kockázati tényezők sohasem küszöbölhetők ki, csupán a lehető legkisebb mértékűre csökkenthetők vagy feltartóztathatók.
A hálózati szondák és naplózók azonban abból a szempontból egyediek, hogy természetükből fakadóan befogadó szerepet játszanak: adatokat csak fogadnak, viszont a maguk részéről semmit sem kell küldeniük. Ezért ha befogadó jellegükből előnyt kovácsolunk, az általuk védett hálózatból elérhetetlenné válnak, s ez jó ötletnek bizonyulhat.
Eredményül egy olyan rendszert kapunk, amelyet csak konzolról lehet irányítani, vagy külön hálózati kártyát kell beleépíteni, amely IP-címmel rendelkezik. Abban az esetben, ha a rendszerbe két hálózati kártya van építve, két fontos tanácsot érdemes megfogadni: először is az IP-továbbítást ki kell kapcsolni. A második, hogy az IP-címmel ellátott kártyát a szaglászó-naplózó hálózattól eltérő hálózatra kell kapcsolni. A példánál maradva ez egy külön hálózat lehet az NIDS-szondák, rendszerfelügyeleti és naplózó munkaállomások részére.

## Fizikai és rendszerszintű felépítés

A hálózati kártya (Network Interface Card – NIC) telepítése egyszerű feladat. Feltéve, hogy a hálózati kártyát a rendszermagod támogatja, a Linux önműködően felismeri azt, a kezeléséhez csupán be kell tölteni a megfelelő modul(oka)t.
Mindamellett az egyes Linux-változatok nagyon is eltérő módon végzik el a hálózati kártyák kezdeti jellemzőinek beállítását. Red Hat-változatra épülő rendszeremben a második kártya telepítéséhez létre kellett hoznom egy új állományt, a */etc/sysconfig/network-scripts/ifcfg-eth1*-et, az alábbi tartalommal:

```
DEVICE=eth1
USERCTL=no
ONBOOT=yes
BOOTPROTO=
BROADCAST=
NETWORK=
NETMASK=
IPADDR=
```

Annak ellenére, hogy a Red Hat Kudzu eszköze az új csatolókártyát önműködően érzékelte, a hálózati beállító héjprogramja hibajelzéssel ért véget, amikor az IP-címet nem adtam meg. Saját */etc/sysconfig/network-scripts/ifcfg-eth1* állományom létrehozását követően a Red Hattel sikerült úgy működésbe hoznom a kártyát, hogy nem kellett neki IP-címet adnom. Lehetséges, hogy a különböző, a Red Hattől eltérő Linux-változatokban ugyanezt az eredményt más és más módon lehet elérni.

## Rejtett szaglászás

Ha már telepítetted és üzembe helyezted rejtett hálózati kártyádat, és a megfigyelni kívánt hálózathoz is csatlakoztattad, eljött a rejtett szaglászás kipróbálásának az ideje. A cikk hátralevő részében feltételezni fogom, hogy gépedre már telepítve van a Snort program. A legtöbb Linux-változat saját Snort-csomaggal rendelkezik, a legfrissebbet pedig a ➲ http://www.snort.org (a 41. CD Magazin/Snort könyvtárában is megtalálható) címről szerezheted be. Ha Snortot NIDS-ként kívánod használni, különösen fontos a Snort legfrissebb változatának beszerzése.
A szaglászó üzemmódú Snort használatához nem kell mást tenni, csak kiadni az alábbi parancsot:

```
snort -dvi eth1
```

A `-d` kapcsoló a Snortot az alkalmazás adatainak visszafejtésére utasítja, a `-v` pedig arra, hogy a csomagok tartalmát írja ki a konzolra, a `-i` után adhatjuk meg a megfigyelni kívánt kártyán. A `-C` kapcsolóval a programot a hexadecimális adatok átlépésére lehet utasítani, így csak az ASCII-karaktereket fogja megjeleníteni (1. lista, lásd 41. CD Magazin/Snort könyvtár).
A Snort a szaglászást az IP-címmel nem rendelkező hálózati kártyán is hibátlanul végzi.

## Rejtett betörésérzékelés

A betörésérzékelés önmagában is hatalmas terület, a Snort betörésérzékelő képességei pedig sokfélék és erőteljesek. Mielőtt alaposabban elmélyednénk a témában, úgy érzem, fel kell hívnom a figyelmet, hogy ennek a témának épp csak felszínét érintettem: a Snort alapértelmezéshez közeli beállításokkal való működtetése távolról sem a leghatékonyabb módja a Snort NIDS-ként való használatának.

A Snort NIDS üzemmódban való indításához mindössze a */etc/snort/ snort.conf* állomány szerkesztését kell elvégeznünk, és a Snortot démonmódban elindítanunk. Ezután már csak a *snort.conf*-ban meghatározott szabályokat kell időnként frissíteni, amint újabb támadási aláírások válnak hozzáférhetővé. Tekintsük át az egyes lépéseket!

Annak ellenére, hogy a `-c` lehetőség révén tetszőleges beállítóállomány kijelölhető, a legtöbb ember mégis a */etc/snort/snort.conf* állomány használata mellett szokott dönteni. E cikk hátralevő részében azzal a feltételezéssel élek, hogy választásod szintén erre a lehetőségre esett. A *2. lista* csonka, de jelentésében teljes Snort-beállítóállományt mutat be. Mint az világosan látható, a Snort-beállításban találhatók meg a teljes körű lehetőségek, a változómegadások, az „előfordítói" és kimenetszabályozó utasítások (direktívák) és a Snort-szabályok. A teljes körű lehetőségek (vagyis `config`-utasítások) a legtöbb lehetőség beállításához kellemesen használható közvetlen kapcsolók, amelyek a Snortnak indítózászlókként adhatók át, és gépelést takarítanak meg.

A Snort-szabályok által használt változók a betörésérzékelést pontosabbá teszik. Ha például a `DNS_SERVERS` változóban megadjuk névkiszolgálóink IP-címét, akkor a Snort figyelmen kívül fog hagyni bizonyos, a DNS-kiszolgálónk által küldött csomagokat, amelyek egyébként támadási kísérleteknek tűnhettek volna.

Az előfordítói utasítások az előfordítói modulok beállítására használhatók, amelyek tulajdonképpen olyan csomagmódosító Snort-elemek, amelyek a csomagokat még a szabályokkal való ütköztetés előtt módosítják.

A `frag2` előfordítói zászló például újra összeállítja a feldarabolt csomagokat, de egyúttal figyel az IP-cím töredékalapú és töredékjelleghez kapcsolódó rendellenességekre is.

A kimenő utasítások olyan feldolgozás utáni beállítómodulok, amelyek a Snort-riasztások vagy más módon megfigyelt csomagok naplózását és tárolását teszik lehetővé. Későbbi összehasonlítás és elemzés végett a csomagokat MySQL-adatbázisba lehet rögzíteni és a későbbiek folyamán olyan utólagos adatbázis-feldolgozó eszközzel tanulmányozni, mint a ➲ http://www.andrew.cmu.edu/snortacid.html címről letölthető ACID-program.

Végül maguk a szabályok közvetlenül kilistázhatók, mint az a 2. listán bemutatott „Vegyes üzemű Cisco Catalysthez távoli hozzáférés" riasztásnál történt; vagy szövegállományba fűzhetők, mint az a 2. lista hátralevő részében látható. Az utóbbi módszerrel könnyedén lehet használni a szabályokat tatalmazó állományt, amelyet a ➲ http://www.snort.org/dl/signatures/snortrules.tar.gz (41. CD Magazin/Snort könyvtár) címen a Snort fejlesztőcsapat félóránként frissít. A Snort NIDS üzemmódú, a beállítóállományt felhasználó indításához a következő parancsot használjuk:

```
snort -c /etc/snort/snort.conf -D -i eth1
```

Nem szabad elfelednünk, hogy korábban bemutatott példáinkban rendszerünk rejtett fogadófelületének az `eth1` csatolókártyát állítottuk be.

*2. lista* A snort.conf minta beállítóállomány

```
# 0. lØpØs: Set global options:

config logdir: /var/log/snort

# 1. lØpØs: hÆl zati jellemzık
# testreszabÆsa:
var HOME_NET 192.168.1.0/24
var EXTERNAL_NET any
var SMTP $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var DNS_SERVERS 192.168.1.250/32
var RULE_PATH ./

# 2. lØpØs: az elıford t k beÆll tÆsa

preprocessor frag2
preprocessor stream4: detect_scans
preprocessor stream4_reassemble
preprocessor portscan: $HOME_NET 4 3
                       ➥portscan.log

# 3. lØpØs: a kimenetet kezelı beØp lı
# modulok beÆll tÆsa
output database: log, mysql, user=root
dbname=snort host=localhost

# 4. lØpØs: a szabÆlykØszletek testreszabÆsa

alert tcp $HOME_NET 7161 -> $EXTERNAL_NET any
(msg:"Vegyes  zemß Cisco Catalysthez tÆvoli
➥hozzÆfØrØs";
flags:SA; reference:arachnids,129;
reference:cve,CVE-1999-0430;
classtype:bad-unknown; sid:513; rev:1;)

#   A sz vegÆllomÆnyok elØrØsi œtvonala,
#   amelyekben tovÆbbi kiegØsz tı szabÆlyok
#   adhat k meg:

include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules

# (stb. ...)
```

Az alapértelmezésnek megfelelően a Snort a riasztásokat a */var/log/snort/alert* naplóállományba rögzíti, míg a kapupásztázó tevékenységet a */var/log/snort/portscan.log* állományba. Ahogyan a *3. listán* látható, az egyes riasztásokban megjelölt csomagok naplózása a */var/log/snort* könyvtár alkönyvtáraiban fog megtörténni.

A kijelölt NIDS-szondának már a rendszerbetöltés során el kell indítania a Snortot. A Snort hivatalos RPM-csomagja telepíti a */etc/init.d/snortd* indító héjprogramot. Amint elvé-

gezted a Snort igényeid szerinti beállítását, ezt a héjprogramot a `chkconfig` paranccsal tedd futtathatóvá a kívánt futási szinteken. Ha a Snortot forrásprogramból telepítetted, szükség lehet egy saját indító héjprogram készítésére.
A Snort NIDS üzemmódú működtetése maga is megérdemelne egy cikket, még inkább cikksorozatot, de az eddig elmondottak elegendőek ahhoz, hogy bemutassuk: a Snortot tényleg lehet IP-cím nélküli hálózati kártyával használni, továbbá azt is felvázolhassuk, hogyan kell munkára bírni a NIDS üzemmódú programot.

## A rejtett üzemmódú naplózás

Elérkezett az ideje, hogy a korábban említett két módszert egy harmadikká egyesítsük, vagyis rejtett üzemmódú naplózásssá. A szokásoknak megfelelően a központi naplózó kiszolgálógép a `syslog`-ot vagy vagy a `syslog-ng`-t futtatja. És nem csalás, nem ámítás: a naplócsomagok a Snort programmal IP-címmel nem rendelkező hálózati kártyán keresztüli befogása valóban lehetséges, ezt aztán tovább lehet adni a syslognak vagy a syslog-ngnek. De ha már egyszer adott a lehetőség, miért is ne lehetne a Snort számára egyszerűen lehetővé tenni, hogy a naplóállományt közvetlenül maga rögzítse?
Az általam most bemutatásra kerülő módszer a Snort, a `tail` és az `awk` eszközt használja a központi gépen működtetett naplózóalkalmazás helyett, ami azt jelenti, hogy a naplóbejegyzéseket küldő gépeken a `syslog` vagy a `syslog-ng` beállításait az alábbiakban ismeretett leírásnak megfelelően a továbbiakban is el kell végeznünk.
Tegyük fel, hogy egy hálózati szakaszra felfűzött kiszolgálógépek naplóállományait egyetlen naplózó kiszolgálógépen szeretnénk gyűjteni. Tételezzük fel ezen kívül azt is, hogy most a naplóállományok titkosságát kevésbé tartjuk fontosnak, mint az épségüket. Senkinek semmiféle hallgatózó szöszmötölésével nem törődünk, viszont nem szeretnénk, ha bárki is matatna a központi gép által egyszer már elfogadott naplóbejegyzések között. A fenti kívánalmakat figyelembe véve tehát a naplózókiszolgálót a helyi hálózatra IP-cím nélküli hálózati kártyával csatlakoztatjuk, és a helyi hálózat hamis IP-címére küldött naplócsomagokat górcső alá vesszük.

## A kiszolgálók beállítása a rejtett naplózó használatára

Minden kiszolgálón, ahonnan a naplózógépnek naplócsomagokat szeretnénk küldeni, két teendőt el kell végeznünk.
Az egyik, hogy minden egyes küldő rendszert beállítsunk, hogy az üzeneteit milyen ál IP-címre küldje. Az ***ál***, illetve ***hamis*** alatt azt értem, hogy ezt az IP-címet semelyik gépnek nem szabad kiosztani, de az adott hálózatban ténylegesen érvényes címnek kell lennie. Tételezzük még fel azt is, hogy helyi hálózatunk címe `192.168.1.0/24`-es, és a naplózáshoz használt „hamis” cím `192.168.1.111`. Minden hálózatra kötött, szabványos `syslogot` használó kiszolgáló */etc/syslog.conf* beállítóállományába be kell szúrnunk a

```
*.info                  @192.168.1.111
```

bejegyzést.
Ezzel szemben azokon a gépeken, amelyeken a `syslog-ng` naplózást használjuk, a következő néhány sort kell beillesztenünk a */etc/syslog-ng/syslog-ng.conf* állományba:

```
destination d_loghost { udp(ip(192.168.1.111)
                        ➥port(514)); };
filter f_info { level(info); };
```

*3. lista* A rejtett naplózáshoz szükséges /etc/snort/snort.conf beállítófájl

```
var EXTERNAL_NET any

config dump_payload
config dump_chars_only
config logdir: /var/log/snort
preprocessor frag2
log udp 192.168.1.20/32 any ->
➥192.168.1.111/32 514
(logto:"logged-packets";)
```

```
log { filter(f_info); destination(d_loghost); };
```

Mindkét esetben szükség lehet – mint az a fenti két példából kitűnt – a túlságosan általános „info” vagy „magasabb” („higher”) fontossági jellemzők helyett egyéni csomagszűrő szempontok megadására.
Az egyes gépeken futó naplózó alkalmazások beállítóállományaiba a megfelelő sorok beillesztésén kívül minden naplóbejegyzést küldő gépnek szüksége lesz a hamis IP-címre vonatkozó ARP-bejegyzésre, hogy a gép képes legyen elvégezni a címfeloldást. Amennyiben helyi hálózatodban elosztó működik, már maga az ARP-cím is lehet nem létező, de valósághű IP-cím.
Ha azonban a hálózatodban kapcsoló üzemel, ehelyett a naplózó kiszolgálógép hálózati kártyájának MAC-címét, azaz fizikai címét kell megadnod.
A naplóbejegyzéseket küldő gépen vagy -gépeken statikus, azaz állandó ARP-bejegyzés az alábbi paranccsal hozható létre:

```
arp -s 192.168.1.111 00:04:C2:56:54:58
```

ahol a `192.168.1.111` a naplózókiszolgálónk ál-IP-címe, a `00:04:C2:56:54:58` számsorozat pedig ugyanezen gép hálózati kártyájának valódi vagy hamis-MAC-címe.
Mostanra a kapcsolóval szerelt helyi hálózatunk valamenynyi küldő gépnek beállításával végeztünk, úgyhogy a naplóbejegyzéseket mindegyikük a `192.168.1.111`-es címre küldi; az elosztóval szerelt hálózat esetén pedig azok a bejegyzések a rejtett naplózókiszolgáló alhálózatára lesznek irányítva. Nem marad más feladatunk, mint magának a rejtett naplókiszolgálónak a beállítása.

## A Snort beállítása a rejtett naplózókiszolgálón

A betörésérzékelési üzemmódnál bemutatottakhoz hasonlóan a Snortot rejtett naplózóként ebben az esetben is mindössze egyetlen állománnyal, a */etc/snort/snort.conf* szerkesztésével állíthatjuk be. A *3. listán* a Red Hat változatra épülő rejtett naplózógép ***snort.conf*** állományába nyerhetünk betekintést. Lássuk, hogyan is épül fel!
Először is egy változónak történő értékadást látunk: `EXTERNAL_NET any`. A Snort NIDS-üzemmódú működéséhez semelyik másik változó itteni használatára nincs szükség. Tekintsünk át néhány beállítóutasítást:
a `dump_payload` a `-d` parancssori kapcsolónak felel meg, a `dump_chars` pedig a `-C` kapcsolónak, míg a `logdir` parancs a Snort naplóállományai számára a saját könyvtárat jelöli ki,

utóbbival egyenértékű a `-l` (nem egyes, hanem kis l betű!) lehetőség használata.
A 3. listán végighaladva felfedezhetünk egy előfordítói utasítást: a `frag2` előfordítói utasítást az előre beállított értékekkel hívjuk meg. Lehetséges, hogy a nagyobb méretű naplóállományok feldarabolásra kerülnek, de mégha ilyen állapotban vannak is, ez a lehetőség újraegyesíti őket számunkra.
Végül itt következik munkánk értelme: a felhasználói igényeket tükröző Snort-szabály. A Snort-szabályok megalkotása semmivel sem bonyolultabb feladat, mint mondjuk IP Tables- vagy IP Chains-szabályok létrehozása – csupán a TCP/IP-protokollok működési elvét és az alkalmazások viselkedésének ismeretét tételezi fel. A „Snort Felhasználói Kézikönyv" (elérhető a ➲ http://www.snort.org/docs/writing_rules címen) mindezt világosan és mindenre kiterjedően elmagyarázza. Haladjunk végig lépésről lépésre a 3. listában bemutatott Snort-szabályon:

```
log udp 192.168.1.20/32 any ->
➥192.168.1.111/32 514 (logto: "logged-packets";)
```

A szabály a tevékenység naplózását érintő szabállyal kezdődik. Ebben az esetben a Snortot csomagnaplózóként használjuk. Így a */var/log/snort/alert* állomány állandó írogatása helyett azt szeretnénk, ha a Snort minden, a szabálynak eleget tevő csomagot rögzítene a naplóban, bármiféle riasztás küldése nélkül.
Most következik a szabály ellenőrzőprotokollja, az UDP. A `syslog` üzeneteket általában UDP-protokollon keresztül küldik. A szabály protokollját a forrásgép IP-címe követi, a CIDR-jelölésnek megfelelően. A naplóállományokat küldő gép IP-címe `192.168.1.20`, és pontosan az e gép küldte csomagokat szeretnénk alávetni a szabályoknak. A `/32` a teljes 32 címbit vizsgálatát előíró CIDR-rövidítés, ami azt jelöli, hogy ez inkább egy közönséges gépcím, semmint egy címtartomány. A példahálózatunk gépeiről érkező csomagok társításához a `192.168.1.0/24` címet jelöltük ki.
Az IP-címet a forráskapu követi, vagyis ebben az esetben az „any" (bármelyik). Az „any" a Snort-szabályokban gyakori megjelölés, mert néhány kivételtől eltekintve a TCP/IP-alkalmazások önkényesen kijelölt, magas azonosítószámmal ellátott kapukról küldenek csomagokat.
A szabály közepén található az irányjelző műveleti jel (direction operator) `->`, amely azt jelzi, hogy a jel bal oldalán álló IP-cím és kapu a csomag forráshelyéhez tartozik, míg a jobb oldalon álló hasonló adatok a célhelyet jelölik.
A másik irányjelző műveleti jel (a „< >") azt fejezi ki, hogy a forrás és cél IP-címek és a hozzájuk tartozó kapuk kölcsönösen felcserélhetők. Más szóval ez annyit tesz, hogy a Snortnak a csomagokat a megadott szabályoknak alá kell vetnie, haladjanak azok bármely IP-cím felé a meghatározott kapukon keresztül.
Az irányjelző műveleti jeltől értelemszerűen jobbra található a célhelyet meghatározó IP-cím és kapumegjelölés, a `192.168.1.111/32` és `514`.
A `192.168.1.111` az a cím, ahová kiszolgálógépeink az UDP-protokoll 514-es kapuján keresztül a naplóbejegyzéseiket küldik.
Végül itt találhatjuk a szabály kiadása során érvényesíthető, zárójelek között megadott választható lehetőségeket. Ha több ilyen lehetőséget szeretnénk egyidejűleg használni, ;-vel (pontosvessző) kell őket elválasztani egymástól. Ezúttal csupán egyetlen lehetőség szerepel: `logto:"logged-packets"`.
A `logto`: lehetőség révén megadhatunk egy állományt, ahová a szabálynak eleget tevő csomagokat lehet rögzíteni – ez a */var/log/snort/logged-packets* állomány volt. Az állománynév elől azért hagyhattuk el az útvonal nevét, mert a `logdir` lehetőséggel a Snort által használandó könyvtárat már korábban kijelöltük: */var/log/snort*.
Ha a `logto:` lehetőséget nem vesszük igénybe, a Snort a naplókönyvtárának új alkönytár kezdi el a naplóbejegyzéseket gyűjteni, minden egyes, a szabálynak eleget tevő IP-cím számára egy-egy külön alkönyvtárat hozva létre.
A rejtett üzemmódú naplózás céljainak azonban jobban megfelel, ha a `logto:` lehetőséggel az egyes szabályoknak eleget tevő csomagok naplózására egyetlen állományt jelölünk ki. E módszerrel a csomagokat inkább a szabályok szerint rendezhetjük, semmint az elvégzett műveletek alapján.
Hűha, ezt hosszabb volt elmagyarázni, mint a rejtett naplózást beállító ***snort.conf*** fájl hátralevő részét!
Ám a legfontosabb rész mégiscsak maga a szabály. Amennyiben több kiszolgálógép adatait szeretnéd gyűjteni, helyes, ha mindegyik naplóbejegyzéseit külön-külön állományba gyűjtöd.

## Összefoglalás

A Snort a szaglászás-, betörésérzékelés és naplózás sokrétű és nagyteljesítményű eszköze. Rejtett üzemmódú szaglászóként vagy NIDS-egységként való beállítása könnyű, és rejtett üzemmódú naplózómegoldásba történő beépítése is hasonlóképpen kivitelezhető. Minden jót a naplózási és NIDS-kísérletekhez, legyenek akár rejtettek, akár nem!

***A cikkhez tartozó listák megtalálhatóak a 41. CD Magazin/Snort könyvtárában.***

*Linux Journal 2002. október, 102. szám*

*Mick Bauer*
(mick@visi.com) hálózati biztonsági tanácsadó az Upstream Solutions Inc.-nél Minneapolisban (Minnesota). Mick a szerzője a hamarosan megjelenő új O'Reilly-könyvnek, amelynek címe „Building Secure Servers With Linux", de ő írta a „Network Engineering Polka" című művet is. Büszke apja gyermekeinek.

### Az elosztók és kapcsolók hatása az ismertetett módszerre

Az írásunkban ismertetett módszerek csak akkor működőképesek, ha a Snortot futtató gép és a naplózni kívánt kiszolgálógépek a helyi hálózatnak azonos szakaszára csatlakoznak. Osztott, vagyis jelelosztóval (hub) összekötött hálózatoknál ez azt jelenti, hogy a megfigyelni kívánt gépek és a naplózást végző gép között csak elosztók lehetnek.
Kükönböző hálózati szakaszokon lévő gépekkel a leírt módszer nem működik, csak az átjáró(k) külön beállítása után.