



## Hálózatkezelés a Nagios rendszerrel

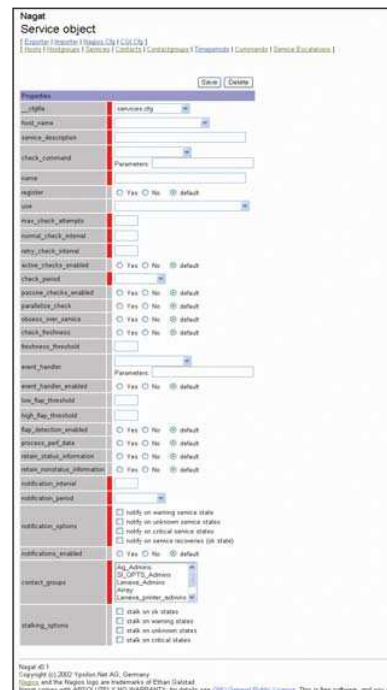
A John Deere különféle alkatrészek és programok keverékére volt kénytelen kiterjeszteni a hálózati kezelőrendszert. A nap hőse a Nagios lett.

**A** mikor elkezdtem dolgozni a John Deere Agricultural Marketing Centerben, 12 állomásunk volt az Egyesült Államok és Kanada különböző pontjain. Ezek a telephelyek eltérő felszereltséggel rendelkeztek: az összes feladatot régi kiszolgálókkal és asztali gépekkel oldották meg – a tartománykezeléstől egészen a nyomtatásig. Ez azonban nem igazán egyezett a központosított IT-szervezetről alkotott elképzelésünkkel. Hogyan kezelhetnénk és figyelhetnénk meg a kiszolgálókat és a WAN csatornákat ennyire változatos helyeken? Körülbelül két évvel ezelőtt úgy döntöttünk, hogy a felhasználói adatokat hálózatunk központjából kivisszük a telephelyekre. A kérdés csak az volt, hogy miképpen tudjuk majd szemmel tartani az összes telephely különböző gyártóktól származó eszközeit? Hogy fogalmat alkothassunk a megfigyelendő eszközökről, bemutatunk közülük néhányat: minden fő helyen volt egy Maxtor 4100 hálózatra kötött tárolóeszköz (NAS) kiszolgáló, illetve egy Compaq 1600, ami a nyomtatási és tartományvezérlői feladatokat látta el. Néhány kisebb telephelyen helyi nyomtatókiszolgálóként Dell GX1 asztali gépeket használtunk. A központi állomáson egy Compaq TaskSmart N2400 volt a legfőbb fájlkiszolgálónk.

A változatos alkatrészek következtében egyetlen gyártó eszközvezérlő-készlete sem felelt meg az igényeinknek, az az ötlet pedig, hogy mindent más-más eszközzel figyeljünk meg, nem igazán tetszett. Így aztán gyártófüggetlen megfigyelőeszközök közül kellett választanunk. Az általunk talált megoldásokat megközelítőleg három nagy csoportba sorolhatjuk. A lista aljára azok a megoldások kerültek, amelyek nem igazán voltak képesek megfigyelni a folyamatosan növekvő számú kiszolgálót és asztali gépet. A legtöbb esetben kénytelenek lettünk volna a programmal érkező eszközöknél maradni. Más programok az úgynevezett élvonalbeli eszközök közé tartoztak, ilyen például Hewlett-Packard OpenView rendszere. Ezt ugyan jól használhattuk volna ott, ahol szeretnénk volna, és képes lett volna megfigyelni is azt, amit akartunk, de az ára messze meghaladta lehetőségeinket.

### A NetSaint projekt

Már éppen kezdtek elveszíteni a reményt, hogy találunk valamit, ami megfelel az igényeinknek, amikor belefutottam a NetSaint projektbe. A NetSaint olyan megfigyelőrendszert ígért, ami megfelelt volna az igényeinknek, és képes lett volna megfigyelni azokat a dolgokat, amiket szeretnénk volna, mégpedig egy nyílt keretrendszer segítségével, ami lehetővé tette, hogy saját bővítményeket (plugin) készítsünk. De a NetSaint segítségével nemcsak egyszerűen megfigyelhetjük a kiszolgálókat és a szolgáltatásokat, de azt is lehetővé teszi, hogy a bemutatott irányvonalat követve munkáinkat előretekintőbb módon készítsük el. Minden nagyszerűen ment: a NetSaint megbízhatóan végezte munkáját a John Deere egyik kisebb irodájában. Ahogy telt-múlt az idő, a mezőgazdasági részleg belekezdett az IT-részleg korszerűsítési programjába, azzal a célkitűzéssel, hogy a szakértői területek színvonalát a teljes részlegben megemeljék. Ebben a projektben ismét felmerültek a megfigyelés régi bonyo-



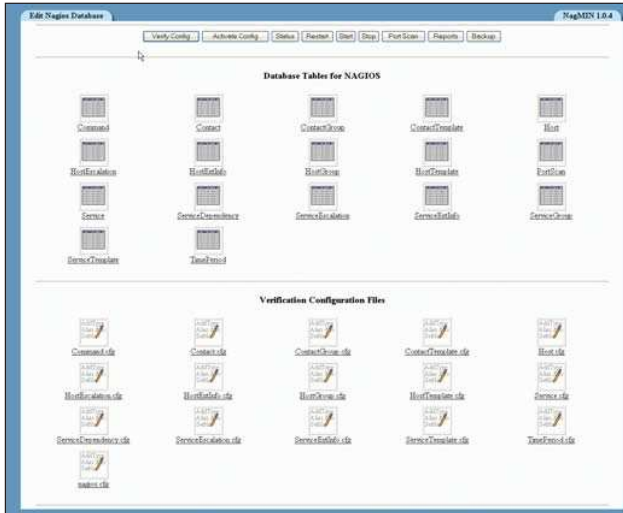
1. kép A Nagios-szolgáltatások lapja

egységeknek lehetővé teszi, hogy a kialakult irányzatok adatait grafikonon ábrázolják. Ugyancsak nehézséget jelentett az eddigieknél is szórta megfigyelendő alkatrészkészlet, a Network Appliance-től kezdve a Sunon át a Dell-gépekig. Továbbra is voltak pénzügyi korlátaink, ez azt jelentette, hogy az OpenView-osztályú termékek kiestek a lehetőségek közül. Mivel termelési környezetünkben a NetSaintet futtattuk, úgy döntöttünk, nekilátunk és megpróbálunk NetSaint alapú megoldást keresni a mezőgazdasági részleg számára. Az első dolog, amit megtanultam, hogyha egy ideig nem követem a nyílt forrás közösségének a történéseit, akkor amikor visszatérek, igen sok változást tapasztalok. Esetünkben a NetSaint nem létezett többé; helyét átadta utódjának, a Nagiosnak (<http://www.nagios.org>).

### Nagios – az utód

A Nagios több szempontból a NetSaint evolúciójának a következő lépcsőfokát jelenti. Némi szomorúsággal szemlélttem, hogy a pingvin eltűnt a földalról, de lassan hozzászoktam a hiányához. A rendszer tanulmányozása után úgy döntöttünk, hogy a többszintű megoldást választjuk, s a megfigyelési projektünkben szóba kerülő minden egységnél külön Nagios-kiszolgálót helyezünk el. Elsősorban ezért döntöttünk így, hogy egyetlen kiszolgáló se terhelődjön túl, és a szükséges megfigyelési szintet folyamatosan fenntarthatjuk. Ezt szem előtt tartva Moline-ban, Illinois államban telepítettük a fő megfigyelőgépet,

dalmi. Mivel egy központosítottabb IT-szerkezet felé próbáltunk meg elmozdulni, a többi megfigyelőprogram valamennyi képességével fel kellett vernetni magunkat. Továbbá minden egység saját maga hozhatta meg az igényeinek a legjobb megfelelő IT-döntéseket. A kérdés tehát az volt: miként fogjuk össze a sokféle megfigyelőrendszert, és hogyan mozdulunk el egyetlen egységes megoldás irányába? Az új megfigyelőnek képesnek kell lennie kiszolgálók százait és szolgáltatások ezreit figyelemmel kísérni, miközben az egyes



2. kép A NagMIN főlapja

majd az első gyermek kiszolgálót a kansasi Lenexában, a mezőgazdasági részleg marketingosztályán helyeztük el.

## Telepítés

A szülő- és a gyermek kiszolgálón ugyanaz a Nagios telepítés futott. Mindkét gépre Red Hat 8-at telepítettünk, majd felpakoltuk a Nagios rendszert. A rendszerrel érkező leírás csaknem teljes értékű, és szépen végigvezet bennünket a telepítés lépésein. A Nagios mellett minden kiszolgálóra egy Nagat nevű programot is feltelepítettünk (1. kép).

## A Nagat beállítóprogram

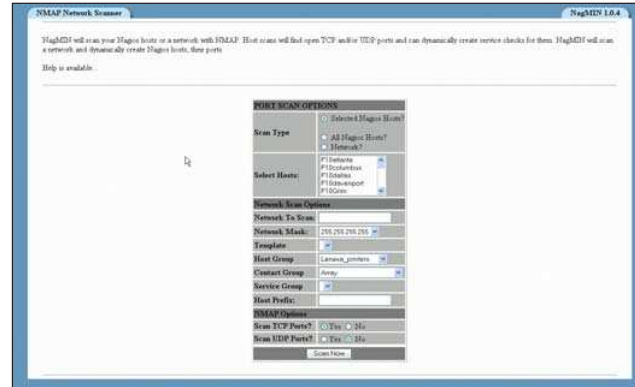
A Nagat a Nagioshoz szánt webalapú beállítóprogram. Segítségével mindössze egy weboldalon kell néhány mezőt kitöltenünk, és a válaszaink alapján elkészül a beállításfájl. Így egyetlen terminálablak megnyitása nélkül a kevésbé tapasztaltak is be tudják állítani a Nagios-t, illetve annak szolgáltatásait és gépeit. A Nagat használata során Red Hat 8-as telepítésünk alatt belefutottunk néhány nehézségbe. Megfelelőségi gondjaink elhárítására a PHP és az Apache egyik korábbi változatát kellett feltelepítenünk (PHP 4.1.2, Apache 1.3). Miután ezt megoldottuk, hamar be tudtuk indítani a rendszert. A Nagat ugyan néhány hibát is magában rejtett, de könnyen kiküszöböltük őket. Az egyik hiba a *service edit* oldalon bukkan fel: a rendszer nem menti a kapcsolatsoportot (contact). A javításhoz mindössze annyit kell tennünk, hogy az alábbi sorokat a 39. sor case-szerkezetéhez illesztjük:

```
$saveobject[ 'contact_groups' ] =
    @implode(" ", $saveobject[ 'contact_groups' ] );
```

Ezek után kapcsolatsoportunkba már gond nélkül kimenthetjük a változásokat, a szolgáltatás frissítése közben.

## A kiszolgálók beállítása

Miután mindent feltelepítettünk, elérkeztünk a projekt – véleményem szerint – legnehezebb részéhez: a kiszolgálók beállításához. Végül maga a kiszolgálóbeállítás közel sem volt olyan nehéz feladat, mint kitalálni, hogy mit is várunk el rendszereinktől, majd végigjárni az odáig vezető utat. Az első döntés, amit meg kellett hoznunk – és amit éppen most vonunk vissza –, a Nagios forditása beépített adatbázis-kezelés nélkül.



3. kép A NagMIN kapupasztázó felkutatja a megfigyelendő szolgáltatásokat

Egy teljes évig meg is felelt az igényeinknek, de ahogy újabb egységeket építettünk be Nagios megfigyelő rendszerükbe, az általa nyújtott adatok elérése egyre fontosabbá vált. Többé már nem voltak elégségesek a Nagios saját kis grafikonjai. Magunk szeretnénk volna az adatokat megkapni és megjeleníteni, emberek és egységek szerint, ezt pedig nem éppen egyszerű megtenni az eredeti Nagios-telepítésünk által adott egyszerű szöveges állományokkal.

## A Nagios beállítása

A következő lépést a főrendszer beállítása jelentette, ami a legtöbb szempontból magától értetődő volt. Akad néhány dolog, amit érdemes egy kicsit jobban megnézegetni, ha a legtöbbet szeretnénk Nagios rendszerünkben kihozni. Az egyik ilyen képesség az intelligens ellenőrzés (smart check) módszere. Ahelyett, hogy rendszerünkön az összes ellenőrzést egyszerre futtatná (ez igen nagy processzorfelhasználással járna), a Nagios szétosztja az ellenőrzést, mondjuk ötperces időközökre, jelentősen mérsékelve ezáltal a CPU-felhasználást. A másik dolog, amit érdemes megnézni: a párhuzamosított szolgáltatás-ellenőrzés; ezáltal egy időben több ellenőrzést is futtathatunk, ami sokat segít, ha több processzorral rendelkezünk.

A szülő- és a gyermek kiszolgáló beállítása szinte teljesen megegyezik, akad azonban két lényeges különbség is. A gyermek kiszolgálón alapértelmezés szerint a figyelmeztetéseket ki kell kapcsolni, illetve be kell állítani az aktív ellenőrzést (active check), míg a szülő kiszolgálón ennek éppen az ellenkezőjét kell végrehajtanunk. Mindezt azért tesszük, hogy csökkentsük a terhelést a főkiszolgálón és az összes figyelmeztetést egyetlen gépen jelenítsük meg. (A kétféle kiszolgálóhoz használható beállításfájlok az 51. CD Magazin/NAGIOS könyvtárban találhatóak.) A következő lépés a Nagios által végrehajtható parancsok megadása lesz. Kísérletet tettem arra, hogy a rendszert használó emberek eszébe véssem, hogy a Nagios: keretrendszer. Ez alatt azt értem, hogy ő maga semmilyen ellenőrzést nem hajt végre, ez a munka a Nagios által meghívható bővítményekre marad. És ez nagyon jó így, hiszen ezáltal könnyedén készíthetünk saját bővítményeket, mindaddig, amíg ragaszkodunk a Nagios nyújtotta keretrendszerhez. Az általunk írt bővítmények segítségével olyan rendszereket is képesek leszünk megfigyelni, amiket az eredeti Nagios bővítményeivel nem tudtunk volna. Jelenleg képesek vagyunk megfigyelni például a NetApp fájlkezelőt (filer), és kiolvasni a lapszámlálót a HP nyomtatókból, valamint a Compaq Insight Manager adatait be tudjuk építeni a Nagios rendszerbe. A Microsoft Windows-kiszolgálók megfigyelésének talán

4. kép Nagios-szolgáltatásrészletek

legegyszerűbb (de nem az egyetlen) módja, ha kihasználjuk az NSClient képességeit. Az NSClient programot windowsos gépeinken szolgáltatásként futtassuk. Futtatásához nyissunk meg egy kaput a Windows-kiszolgálón – biztonsági okokból érdemes jelszóval védeni a kapuelérést. Írásom születésekor még nem lehetett titkosítást használni a program és a Nagios kiszolgáló közötti kapcsolatban. A bővítmény lehetővé teszi, hogy elérjük a kiszolgáló memória- és lemezfoglaltságát, a processzorteljesítményt, illetve egyéb olyan adatokat, amiket Windows alatt egyébként a teljesítménykezelő (performance manager) eszköz segítségével érhetnénk el. Linux- és Unix-rendszereinkhez némileg eltérő módszert használunk. A Nagios Service Check Acceptor (NSCA) nevű programot démonként futtatjuk az `inetd` alól. A program használatához a Nagios egy `check_nsc` nevű bővítményt indít el, ami titkosított csatornát hoz létre a két számítógép között, majd a bővítmény segítségével lefuttatja a számítógép ellenőrzéséhez szükséges kódokat. A rendszer beállítása magától értetődő: válasszuk ki a két gép között használható titkosítási módszert, majd az ügyfelén adjuk meg a `plugin` parancsot. Ezt követően az adatokat máris átszippkázhatjuk Nagios rendszerünkre. Mint azt korábban említettem, jelenleg éppen azzal foglalkozunk, hogy a Nagios által alapértelmezés szerint használt fájlokat háttéradatbázisba töltjük át. Ezt a módszert arra szeretnénk használni, hogy jobb adatábrázolást érjünk el olyan eszközök segítségével, mint például a Crystal Reports. A Nagios önmagában is támogat két adatbázis-kezelőt: a MySQL-t és PostgreSQL-t. A Nagios rendszerrel együtt néhány parancsfájlunk kapunk, amelyek előállítják az adatbázist, és elkészítik Nagios rendszer működéséhez szükséges táblákat és mezőket. Mi a PostgreSQL mellett döntöttünk. Az adatbázis felállítása után a `--with-mysql-xdata` vagy a `--with-pgsql-xdata` kapcsolók valamelyikével újra futtatnunk kell a beállítási parancsfájlokat. Az `xdata` úgy állítja be a rendszert, hogy mindenhez adatbázisokat használjon. Jelenleg az egyetlen dolog, ami nem támogatja az adatbázisokat, maga a beállításfájl. Végül egy érdekességre szeretném felhívni a figyelmet: idén márciusban jött ki egy kiegészítés, mégpedig Nagios rendszer NagMIN elnevezésű Webmin bővítménye (2. kép). Ez a rendszer nemcsak a Nagios webes beállítását teszi lehetővé (mint a Nagat), hanem néhány olyan dolgot is felkínál, amit eddig egyetlen eszköz sem. Először is lehetővé teszi a Nagios-beállításfájlok adatbázis-támogatását. Másodszor, ha a Nagios-telepítésünket akarjuk beállítani, igencsak jól jön kapuvizsgálat (port scan) képessége (3. kép). Még nem próbáltam ki e képesség minden lehetőségét, de állítása szerint hálózati felderítést végez, és a fellelt adatokat beilleszti a Nagios beállításfájlaiba,

megkímélve bennünket attól, hogy a rengeteg adatot mind nekünk kelljen felvinnünk. A NagMIN a legtöbb rendszeren jól működik, de a különleges rendszereket valószínűleg mindenképpen kézzel kell felvinnünk a rendszerbe. Amennyiben további tájékoztatásra lenne szükségünk a bővítménnyel kapcsolatban, látogassunk el a SourceForge.net honlapjára. Nagios kiszolgálónkat a szülő-gyermek viszony alapján állítottuk be. Ehhez némileg eltérő beállítások szükségesek, mintha egyszerű önálló rendszert készítenénk. Egyrésztől a webfelületet csak a szülő-kiszolgálóra telepítettük fel (4. kép). A gyermek-kiszolgálók mindössze annyit tesznek, hogy ellenőrzési adataikat passzív ellenőrzési módszerrel egyszerűen áttöltik a szülő-kiszolgálóra. Ennek kivitelezéséhez egy OCSP parancsot adtunk ki, ami lefuttatja az adatokat a Nagios szülő-kiszolgálóra közvetítő parancsfájl. Azáltal, hogy a parancsfájl futtatására az OCSP parancsot használjuk, minden egyes Nagios-ellenőrzési folyamat után le fog futni. Éppen ezért tulajdonképpen csak a gyermek-kiszolgálónak kell aktív ellenőrzéseket futtatniuk. Így a Nagios-szülő-kiszolgálónak csak a webfelületet kell futtatnia, illetve hiba esetén elküldenie a jelentést.

Most, hogy körülbelüli képet kaphattunk arról, hogyan állítottuk be a Nagiosot, engedjék meg, hogy elmeséljem, milyen előnyökkel járt a használata. Ennek az évnek az elején vásároltunk egy TempTrax nevű digitális hőmérőt, ami együttműködött a Nagiosszal. A hőmérőt fő számítógépteremünk hőmérsékletének a megfigyelésére használtuk. Március eleje környékén aztán kiderült, milyen fontos is számunkra a Nagios. Egyik pénteken körülbelül déltájt üzenetet kaptam, miszerint a számítógépterem hőmérséklete emelkedik. Miközben öltözködtem, a rendszer válságos helyzetről szóló üzeneteket kezdett nekem küldözgetni a számítógépszoba hőmérsékletével kapcsolatban. Mire beértem a munkahelyemre, a hőmérséklet 80 fok körül volt, és egyre emelkedett. A Nagiosnak hála elég időm maradt, és fel tudtam hívni a légkondicionálásért felelős szakembereket, akik megjavították a berendezést, még mielőtt a számítógépeket komoly kár érte volna. Később kiderült, hogy a számítógépszoba A/C egységét vezérlő másik két rendszer képtelen volt kitárcsázni és riadóztatni a rendszert felügyelő szakembereket. Ha a Nagios nem figyelmeztetett volna bennünket a számítógépszobában előállt hibára, másnap reggel sokkal nagyobb bajjal találtam volna szemben magam. Egészében véve azt kell mondanom, hogy a Nagios kiváló hálózati megfigyelőeszköz. Keretrendszerrel van szó, ami önmagában nem túl sok mindenre képes, de pontosan ez az, ami olyannyira egyszerű eszközzé teszi. Mivel keretrendszer, csak azt végzi el, amire szükségünk van. Lévény nyílt, a bővítményeket tervezni hozzá éppolyan egyszerű, mint ellenőrzési adatainknak Nagios által is érthető formátummá alakítása. Így a bővítményünk mindent felhasználhat, amit csak a Nagios nyújtani tud, és ez nem kevés. Végül a Nagios olyasvalami, amire mindenképpen vetnünk kell legalább egy pillantást – helyi telepítésünkön akár ki is próbálhatjuk. Kötve hiszem, hogy kiábrándulunk belőle.

Linux Journal 2003. július, 111. szám



Richard C. Harlan

(harlanrichardc@johndeere.com)

Hálózati mérnök a lenexai John Deere Agricultural Marketing Centerben, Kansasban.