

1.344 FT

II.
ÉVFOLYAM
8. SZÁM



tech.net

A MICROSOFT MAGYARORSZÁG SZAKMAI MAGAZINJA

Small Business Server (II. rész)
Fun – A fejlődés megállíthatatlan



RRAS

Routing & Remote Access

Még egy kis

Microsoft



Barcelona

A „tech.net magazin Brainstorm” a Dupla KV rovathoz hasonló, ám a személyes kérdésfelvetést és vitát is lehetővé tevő rendezvény, melynek célja:

- az elsőre talán ismeretlen technológiák élő bemutatása
- a cikkekhez kapcsolódó kódok megírása/kipróbálása
- a terjedelmi okokból kimaradt információk átadása

E magazinnal együtt a rendezvényre érvényes belépőjegyet minden előfizetőnkhez eljuttattuk. További információk a belépőjegyen olvashatók.

Várjuk Önöket a NetAcademia Mesterkurzusokon



A legjobbakat tanítjuk.

(Bár belépőjegyet adtunk, ez a tény önmagában nem biztosítja helyét a rendezvényen. A jegy célja előfizetőink elsődlegességének biztosítása, de a korlátozott résztvevői létszám (100 fő) miatt a regisztráció kötelező. Jelentkezzen, amíg nem késő!)

<http://technet.netacademia.net/brainstorm>



KAPCSOLJON!

HunNet
RL512W



512 kbit/sec-os
szimmetrikus
forgalomfüggetlen
csatlakozás
az internetre
havi 35.000 Ft-ért

Tel: 06-40-hunnet (486-638)



Miről nem írtunk? Mi az a NATE?

◀ KÖSZÖNTŐ



Szó érte a ház elejét. Kritikát kaptunk. Olvasóink kérdezik, mikor láthatnánk már végre egy jó kis cikket például a WINS-ről (és sok egyéb ősi technológiáról)? Ezt könnyen megmondhatom! Ha minden a terveink szerint halad, bizony soha! Én a magam részéről már a WINS felhására várok! Legyünk túl rajtuk, amilyen hamar csak lehet, hogy továbbra is az érdekes dolgokkal foglalkozhassunk lapunk hátsó részén. Óriási örömről számolunk, hogy alkalomról alkalomra egyre többen vesznek részt (ingyenes) Mesterkurzusainkon. Mint azt néhány hónappal korábban beígértem, összeállítottunk egy tematikus képzéseket is. De nem azért, mert valamilyen élni is kell (ezek az alkalmazások ugyanis fizetősek lesznek), hanem mert úgy érezzük, elérkezett az alkalom a hazai informatikusokkal ilyen formában történő „kapcsolattartásra”. Szorgoskodásunknak több oka van. Egyfelől tartanunk kell a bevált tájékoztatási formát, az állandóan fejlődő technológia ismertetését a tech.net magazinban, másfelől véget szeretnénk vetni egy igen kellemetlen kényszerfeladatunknak. Lássuk sorjában.

1. Kényszerszaktanácsadás

Cégünk tevékenységei között elég jelentős szerep jut a szaktanácsadásnak. Mind levlistáinkon, mind kiszállásos formában műveljük a gyógyítást. Nincs is semmi baj a bevételekkel, de nagy baj van az időbeosztással. Sok esetben túlzott munkát végzünk, szétkonfigurált, agyonkattintott, beduglott vizsgálóokra és fűtőkre kell alkalmaznunk a ráolvasásos hibaelhárítást. A ráolvasásnak pedig tudvalegve sokkal nagyobb a hatása, ha éjfél táján végzik (az sem hátrány, ha éppen holdtölte van :-). Szép pénzeket lehet kérni az éjszakai vajakölésért - csak egy a bökkenő: mindannyian imádunk aludni. Minden pénzélni többet ér (ne), ha az éjszakáinkat néha ágyban, párnák közt tölthetnénk ahelyett, hogy guru-széken, pizzába könyökölve menne a hókuszpókusz. Néha elképesztően egyszerű „hibák” miatt áll egy-egy cég (például betelik az SQL Server tranzakciónaplója), mely elő sem fordult volna, hogo ha én azt tudtam volna...

2. A tájékoztatás és a technológiai verseny

Éppen hogy behoztuk cikkeinkkel azt a lemaradást, amit általában halmozunk fel, hogy tavaly szeptember előtt nem is léteztünk, máris itt az új kihívás: a forradalmi .NET. (A .NET forradalomról lásd e havi Tech.Ed beszámoló cikkeinket!) Kétoldalról nehezedik ránk nyomás: egyfelől azok részéről, akik az időnságokat a lehető legkevesebb késéssel követve már

a bétatermékekkel kísérleteznek (hajrá dotnet!), másfelől azok részéről, akiknél ilyen-olyan (akár cégpolitikai) okokból kifolyólag még mindig a Windows NT 4.0 jut osztályrészül. Az első csoport természetesen a még-még-még újabb dolgok megismerésében érdekel (s bevallom, minket is az ilyen cikkek írása izgat), míg a második csoport szívesen olvasna ismertető cikkeket például a DHCP Serverről, az NTFS-ről, DNS-ről stb. A csapat tehát széthűz. Bevallom, nem szívesen írnék cikket sem a DHCP-ről, sem a WINS-ről, sem a többi olyan technológiáról, melyek éveket óta velünk vannak, s nem is igazán változnak. Sokkal jobb lenne, ha továbbképzés keretében, interaktív módon foglalkoznánk velük, mellyel két legyet ütnénk egy csapásra: nem kellene leírunk :-), és átkerülne sok-sok olvasó a második csoportból az elsőbe: „Tudjuk, unjuk a DHCP-t, az Active Directoryt!”

3. Az állóvíz felkavarása

Piaci felmérések szerint Magyarországon évente közel háromszor informatikus vesz részt valamelyik MCP tanfolyamon. Ez szép szám. De sajnos ugyanaz a háromszor ember bukkan fel évről évre. Ez viszont elképesztő. Ez a magyar valóság: körbe-körbe járunk! Miközben az informatikai felsooktatásról évről évre újabb és újabb tízezer fiatal kerül a munkaerőpiacra, ezek elhelyezkedés után gyakorlatilag azonnal a felejtés útjára lépnek; mintha bizony megállna az idő azok számára, akik sikeresen elhelyezkedtek – pedig nekik is ketyeg! További elképesztő statisztikáim is vannak. A legnagyobb informatikai csapattal rendelkező cégek alkalmazottainak 4-5%-a jut el valamilyen komoly továbbképzésre. És a többiek? További 15-16% mindenféle külön képzés nélkül tartja a fejlődési iramot. Maradt 80%. Ők mit tesznek? Ha új feladatot kapnak, leülnek egy vizsgálóló elé anélkül, hogy a megfelelő ismeretekkel rendelkezzenek a tranzakciókezelésről, a biztonsági részekről, a scriptelési lehetőségekről stb. Főnökük milliós routereket vásárolgat, pedig ha tudnák, mi mindent tud a Windows RRAS!

Felölősségünk

Az informatikusok felelőssége (beruházási döntéshozók vagyunk!), hogy szakmai érvek alapján segítsek szegény főnökeiket abban, hogy a lehető legjobb döntéseket hozzák meg. Ehhez sajnos mi magunknak is a lehető legjobbaknak kell lennünk, de az a nyereség csak a miénk, hogy pont azal az eszközzel dolgozhatunk, ami a feladatra alkalmas!

Költségparadoxon

A világ boldogabbik felén már elég rég ismerik e törvényt. Van is piaca és kultúrája a posztgraduális képzésnek! (Közben egy svájci oktatóközpont prospektusa: száz (100) tanterem tizenhárom városban. Kis magyar valóság: a legnagyobb CTEC oktatóközpont talán ha hat tanteremmel bír, azok kapacitáskihasználtsága is alig 60%. Vidéki telephely? Ugyan!)

NATE (NetAcademia Továbbképzési Estek)

A NATE tanfolyamok célja, hogy kitörjünk a háromezres körből, s lehetőleg minél több informatikusnak tegyük elérhetővé az ismereteket. A nemtanulásnak sokszor anyagi okai is vannak. Sírdogálunk itt Budapesten, hogy milyen alacsonyak a tanfolyamárak, hisz alig negyede a moszkvai áraknak, de pont a négyszerese annak, amit egy vidéki árákhoz szokott informatikus hajlandó lenne kifizetni. (A Szovjetunióban mindig minden jobban megy...) Hát még ha nem is egy hallgatójelölt akadna cégenként! Azután ott van a munkából való kiesés kérdése. Magyarországon írd és mondd háromezer kiváltságos van, aki egy-egy hétre a tanulás hőbortjának áldozhat. A többiek? Dógozzanak! Ezen is, s sok minden máson is megpróbálunk változtatni, segíteni. Önös érdeklünk, és a közérdek megegyezik: a száguldó informatikai vonatra igenis fel kell szállni! A NATE jelentheti a repülőrajtot, a robogó vonat elcsípését. Mégpedig minimális munkakieséssel. Hogyan? Talán csak nem e-learning? Nem.

Az e-learning halott?

Igen. De nem sávszélességek miatt. Már tudjuk: az e-kereskedelem nem helyettesíti a hagyományos vásárlást; a CD nem váltja fel a könyveket; az e-learning nem hozza a hagyományos oktatás hatékonyságát. Ha valaki munkaidő után nem hajlandó szakkönyvet kezébe venni, akkor nem hajlandó weben bogarászni sem. Ha valakinek nehezen megy az új koncepciók önálló feldolgozása (vagyunk egy páran, mintegy 87%-unk ilyen!), annak az e-learning sem adja meg a kezdőlökést. A megoldás a személyes kontaktuson, tapasztalatátadáson, interakción alapuló oktatásban rejlik: ez az úgynevezett tanfolyam. Hetente egyszer, délutánonként, harcedzett előadó tolmácsolásában kell megrágni azt a negyvenórás tananyagot, melynek feldolgozása után akár még az MCP vizsga is lehetető.

A legelső NATE tanfolyam témakörei

A fenti monológot legjobban talán úgy támaszthatnám alá, hogy felsorolom azokat a témaköröket, amelyek az ősi NATE tanfolyam alapját képezik (egyébként a hivatalos 1560-as tananyag). Az árákról egyelőre annyit, hogy sokat fog érni az újságelőfizetés a díjak kalkulációjánál!

- ☞ A Windows 2000 telepítése: kézzel, automatikusan (Setup Manager Wizard), RIS-sel

- ☞ Áttérés Windows 2000-re: tartományok frissítése, Mixed és Native mód, munkaállomások átállítása
- ☞ DNS: DDNS, a névtér tervezése, delegáció, forwarderek, Internet-kapcsolat stb.
- ☞ Active Directory: telepítés, tartománya, erdő, FSMO szerepek, az Active Directory logikái és fizikai felépítése, Telephelyek (site) megtervezése, replikáció
- ☞ Rendszerfelügyelet: Active Directory Users and Computers, szervezeti egységek (Organizational Units), felhasználók, csoportok, jogosultságok, delegáció
- ☞ Csoportos házirend (Group Policy), registry-módosítók, felhasználói felület beállítás, szoftvertelepítés, biztonság, scriptek (Startup, Shutdown, Logon, Logoff), mappa-átírányítás, GPO öröklődés
- ☞ Terminal Services
- ☞ Fájlok és nyomtatók: elosztott fájlrendszer (DFS), NTFS és megosztási jogok, Encrypting File System, kvótázás
- ☞ Routing és RAS: protokollok tömege, telefonos hálózat, virtuális magánhálózatok (PPTP, L2TP), IPSec, NAT, Út-választási protokollok (RIP, OSPF)
- ☞ DHCP és WINS: DHCP és DDNS, DNS és Active Directory
- ☞ Lemezkezelés: dinamikus lemezek (dynamic disks), RAID 0, 1, 5
- ☞ Katasztrófák ellen: mentés, helyreállítás, Safe módok, Active Directory Authoritative Restore, Recovery Console

Plusz, plusz, plusz!

Ami a hivatalos MOC könyvből kimaradt, nálunk elérhető: tartománykonzolidáció, a Kerberos hitelesítési protokoll, LDAP, LDIF, saját házirend faragása, scripting (WMI, ADSI stb.), nyílt kulcsú titkosítás, PKI, Smart Cards bejelentkezés.

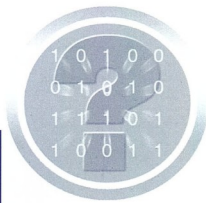
<http://nate.netacademia.net>

Izgalmas idők elé nézünk! Akit nem sikerült meggyőzőnöm, jöjjön el szeptember 11-én délután bemutatkozó délutánunkra! Most pedig (virtuálisan) jöjjenek velünk Barcelonába, a Tech.Ed konferenciára! (A színes oldalakért elnézést kérünk, többé nem fordul elő :-)) Lássuk a jövőt!

Fóti Marcell
főszerkesztő



2001 Augusztus



tech.net¹

A Microsoft Magyarország Szakmai Magazinja

Szerkesztőség

Főszerkesztő: **Fóti Marcell**

marcellf@netacademia.net

Főszerkesztő-helyettes: **Fülöp Miklós**

mick@netacademia.net

Szerkesztőség címe:

1105 Budapest, Ihász utca 13.

Tel.: 263-2732

technet@netacademia.net

Nyilvános levelezési lista:

tech.net@lyris.netacademia.net

Kiadja és terjeszti

a **NetAcademia Kft.**

Terjesztési, előfizetési információ:

Tel.: 263-2732

terjesztes@netacademia.net

Megjelenik havonta, ára 1.344 Ft

Példányszám: 3.000

Minden jog fenntartva, beleértve (a részleteket illetően is) a sokszorosítás, a nyilvános előadás, fordítás jogát. A magazinban közötti cikkeket, képeket és illusztrációkat a kiadó engedélye nélkül közölni, reprodukálni tilos.

Előfizethető megrendelőlevélben a szerkesztőségéknél:

1105 Budapest, Ihász utca 13.

Fax: 261-7145

<http://technet.netacademia.net/subs>

Hirdetésfelvétel:

Bársonykalapács Marketing

Felelős: **Udvarev Rita**

Tel./Fax: 214-0923

info@velvethammer.hu

1027 Budapest, Fő utca 67. V. 1.

Grafikai tervezés, kivitelezés,

nyomdai előkészítés:

Bársonykalapács Marketing

Művészeti vezető: **Balogh Zoltán**

Bársonykalapács © Copyright 2001

Nyomda:

Cerberus Kft.

1066 Budapest, Lovag u. 14.

Felelős vezető: **Schmidt Gábor**



Small Business Server

Small Business Server (II. rész) – Az első lépések **5. old.**



Windows 2000

Routing and Remote Access Services (I. rész) – DUN **10. old.**

NetMon (VIII. rész)..... **14. old.**



Tech.Ed

Tech.Ed 2001, Barcelona..... **17. old.**

Tech.Ed – napról napra..... **27. old.**



Jogi esetek

Az ASP-ről jogászszemmel..... **24. old.**



Szerszámosláda

SysInternals külföldkiadás..... **31. old.**



Business Internet

ASP sulí (VII. rész)..... **36. old.**



Fun

A fejlődés megállíthatatlan..... **41. old.**

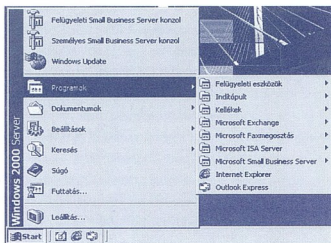
Small Business Server (II. rész)

– Az első lépések

Miután sikeresen telepítettük a Small Business Server-t, a teendők listája jelenik meg a képernyőn. Érdekes a lista elemein sorban végigmenni, mert így eljutunk a kiszolgálónk alapvető beállításáig. Ha a Teendőlista ablakot indulás után bezártuk volna, később is bármikor hozzáférhetünk, ha valamelyik rendszerfelügyeleti konzolból elindítjuk.

A rendszerfelügyeleti konzolok

A Small Business Server 2000 két különböző felhasználói szintnek szánt felügyeleti eszközt tartalmaz: egyet a rendszergazdának, egy másik, kisebb tudású, de egyszerűbben kezelhető példányt pedig a cég valamelyest hozzáértő végfelhasználója számára.



☞ A Start Menü a Small Business Server 2000 telepítése után

Amint látható, az egyes rendszerkomponensek (Exchange, ISA Server, az ábrán nem látható SQL Server) saját menüi is megmaradtak, de a Start menü tetején megtaláljuk a Felügyeleti illetve Személyes Small Business Server konzol ikonokat is. Bármelyiket kipróbálhatjuk, de mi használjuk egyelőre a Felügyeleti konzolt (hacsak külön nem hivatkozunk másra, az SBS konzol néven e cikk keretein belül is mindig erre gondolunk).

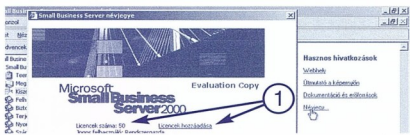


☞ A Felügyeleti Small Business Server konzol, benne a teendők listája

Miután a konzolt megnyitottuk, keressük meg benne a teendőlista sort, és vegyük sorra annak tartalmát! Az üdvözlés után az „Ügyfélcímek hozzáadása” sor következik. Ha ide kattintunk, az SBS súdo megfelelő oldala jelenik meg a képernyőn, amely leírja, hogyan lehet licenc-csomagokat telepíteni a rendszerbe.

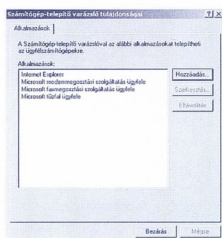
Mint azt már az előző részben írtuk, a rendszer a tartományban létrehozható felhasználók számát nem, az SBS-ügyfélként konfigurálható felhasználókat (és munkaadókat) viszont korlátozza.

Ha a licenccsomag már a kezünkben van, azt egy egyszerű telepítéssel adhatjuk hozzá a rendszerhez (futtassuk a lemezről a setup.exe programot). Ha viszont azt szeretnénk tudni, hogy a kiszolgáló pillanatnyilag hány licenccel rendelkezik, ne is próbálkozzunk a Vezérlőpultban keresséni: ez az információ is a felügyeleti konzolból érhető el.



☞ A licencinformációkat a felügyeleti konzol „Kiszolgáló állapot (BackOffice kezdőlap)” oldalán, a Névjegyz felíratra kattintva megjelenő dialógusablakban találjuk (1).

Ha ezzel készen vagyunk, térjünk vissza a teendőlistához. A következő lépésben gondoljunk az ügyfélszámítógépek telepítésére! Az ügyfélszámítógép-telepítő lemez (amit később majd minden felhasználó részére létrehozunk) lehetővé teszi, hogy a számítógépre automatikusan ügyfélalkalmazásokat telepítsünk. Ha a teendőlista „Ügyfélszámítógépek megadása” sorára kattintunk, egy dialógusablak jelenik meg, ahol ellenőrizhetjük az ügyfélszámítógépek telepítéseihez azokra kerülő alkalmazások listáját (sőt, a listához akár újabbakat is adhatunk).



☞ Az ügyfélszámítógépekre telepítendő szoftverek bővített listája

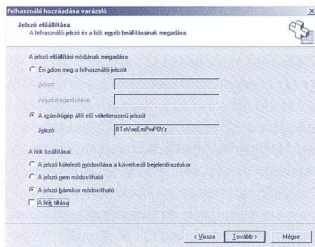


A teendőtlista ezen sora tehát félrevezető, mert egyelőre nem az ügyfélszámítógépek telepítéséről van szó (*szerecsére a kattintásra megjelenő magyarázat már helyes*). Az ügyfélszámítógépekre Outlook, Internet Explorer, megosztott modem-, fax-, és tűzfalkliens kerül, de a telepítendő szoftverek listája bővíthető, ha a Hozzáadás gombra kattintunk, elindul az „Ügyféltelepítés bővítése” varázsló, amelynek megadhatjuk a telepítendő program Windows 9x, Windows NT, Windows 2000 verziójának elérési útját. Ugyanez a dialógusablak egyébként a későbbiekben bármikor elérhető közvetlenül is, a konzol számítógépek oldalának „Ügyfélalkalmazások definiálása” ikonjára kattintva.

Felhasználók létrehozása

De haladjunk tovább! A teendőtlista következő eleme, a „Felhasználók hozzáadása” már komoly varázslót indít. A varázsló segítségével egy-egy felhasználó komplett virtuális munkahelyét alakítjuk ki:

- ☞ Az első oldalon a felhasználó teljes nevét, telefonszámát, irodájának leírását, valamint felhasználói nevét kell megadnunk.
- ☞ Ezután a felhasználó jelszavának létrehozása következik. Megadhatjuk mi a felhasználó jelszavát, vagy kérhetjük a varázslót, hogy generáljon egyet (*ebben az esetben a kitalált jelszót ne felejtjük el felírni!*).

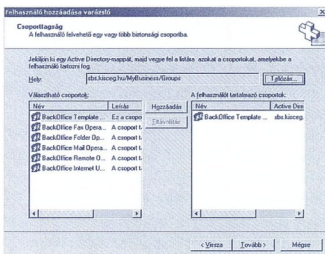


☞ **A jelszó megadására szolgáló panel. Ha a varázslót a felhasználói konzolból futtatjuk, bizonyos funkciók letiltódnak**

- ☞ Ugyanezen az oldalon adhatjuk meg, hogy a felhasználónak mikor kell megváltoztatnia a jelszavát, illetve, hogy a felhasználói fiók tiltott.
- ☞ Ha telepítettük az Exchange kiszolgálót, ezután a felhasználó elektronikus postafiókjának létrehozása következik. Az „E-mail aliasnév” mezőben megadott név lesz a felhasználó e-mail címének @ előtti része. Megadhatjuk még, hogy a varázsló melyik Exchange kiszolgáló melyik postafiók-tárolójába hozza létre a postafiókot, bár egy újonnan telepített SBS esetén nincs sok választásunk.
- ☞ Ezután a felhasználói sablon kiválasztása / beállítása következik. A felhasználói sablon nem más, mint egy-egy „szabványosított” beállítás-csoport, ami csoporttagságot, jogosultságokat, egyéb beállításokat tartalmaz. A sablonok segítségével könnyebb a felhasználók adatainak beállítása. Az SBS már eleve tartalmaz három beépített sablont, a felhasználók, kiemelt felhasználók illetve a rendszergazdák részére. Ha új sablont szeretnénk létrehozni, válasszuk az „Egyedi beállítások...” le-

hetőséget, és ne felejtjük el beállítani az „Új sablon létrehozása ezekkel a beállításokkal” opciót sem. Ekkor a felhasználó egyedi beállításaiból egy új sablon készül, amit a következő felhasználók létrehozásakor már elég kiválasztani. Legalább egyszer válasszuk az „Egyedi beállítások” mezőt, hogy megismerjük, mit rejt egy-egy sablon! Példánk is ezen az útvonalon halad tovább.

- ☞ Az első lépés a csoporttagság beállítása: jelöljük ki azokat a csoportokat, amelyekbe a felhasználó tartozni fog. Az egyes csoportok tagjai speciális jogosultságokhoz jutnak: külön csoport van például a levelezőkiszolgáló, a fax kezelői részére. A BackOffice Internet Users csoporttagság például az Internet-hozzáférést biztosítja a felhasználó számára.



☞ **A felhasználó csoporttagsága. Jól figyeljék meg, hogy melyik csoporttagság mire jogosít fel!**

- ☞ Ezután a terjesztési (levelezési) lista-tagság beállítása következik. Alapértelmezésben egyetlen levelezési lista létezik, a cég közös, saját listája, aminek általában minden felhasználó tagja. Ha újabb terjesztési listákat veszünk majd fel, szükség lehet a sablonok módosítására.
- ☞ A következő ablakban különféle címadoatok megadására lesz szükség: utca, város, irányítószám, ország, stábti.
- ☞ A következő oldal a csatlakozási engedélyeket tartalmazza. Az SBS ugyanis távolról is elérhető, akár RAS, akár virtuális magánhálózat, illetve terminálkliens segítségével is. Ha a terminálként való bejelentkezést is engedélyezzük, megadhatjuk, hogy a felhasználó bejelentkezés után teljesértékű munkaszállt, a felhasználói konzolt, vagy csak egyetlen alkalmazást (*pl. sol.exe :-)*) használhasson.
- ☞ Ezután választhatunk: elindítjuk a számítógép telepítése varázslót (*ami létrehozza a számítógép fiókját az Active Directory-ban és elkészíti az ügyfélszámítógép-telepítőtem*), vagy későbbre halasztjuk. Válasszuk a varázsló indítását („*számítógép telepítése most!*”).
- ☞ A varázsló első kérdése a számítógép nevére vonatkozik. Adjuk meg itt az ügyfél számítógépének leendő nevét (*a régóta ismert szabályokat szem előtt tartva: legfeljebb 15 betű, semmi speciális karakter, a név nem egyezhet meg egyetlen felhasználó, sem tartomány nevével*).
- ☞ A következő oldalon az ügyfélalkalmazások telepítését kérhetjük, illetve tilthatjuk le. A beállítások alapja az általában előzőleg ellenőrzött lista, de ez számítógépenként megváltoztatható (*azt is kérhetjük, hogy ne települjön automatikusan alkalmazás*).



☞ Ezután a varázsló felajánlja, hogy létrehozza az ügyfélszámítógép-telepítőlemez.

☞ A következő kattintásra végetér a varázsló, és az utolsó oldalon áttekinthetjük az összes tervezett változtatást. A tulajdonképpeni munka a „Befejezés” gombra kattintva indul el.

A varázsló létrehozza a felhasználói fiókot, a felhasználó saját munkakönyvtárát, beállítja a jogosultságokat, létrehozza a számítógépfiókot, bejelentkezési parancsfájl generál és még számos fontos feladatot elvégz a háttérben. A telepítés befejezése után a felhasználó már bejelentkezhet a tartományba, és hozzáfér a számára engedélyezett erőforrásokhoz is.

Nyomatató hozzáadása

A teendőlista következő eleme a „Nyomatató hozzáadása”. Ez a parancs a Windows 2000 saját nyomatótelepítő varázslóját indítja el, így ennek használatában nem lesz semmi különleges. A varázsló indítása előtt győződjünk meg, hogy a nyomtatókat nem ismerte-e már fel a Windows 2000 automatikusan, hiszen akkor nincs szükség újabb telepítésre. Ellenőrizzük, hogy a nyomtató tulajdonságai között, a „Megosztás” oldalon a „Listázva itt:Cím tár” (sic! – [MICK]) opció be van-e kapcsolva (mármint, hogy a nyomtatót bejegyezte-e a rendszer a címtárba). Mert ha igen, a Windows 2000-felhasználók élhetnek a címtár olneyelivel: kereshetnek nyomtatót a címtárban a nyomtató helye, az általa nyújtott szolgáltatások, papírméret, felbontás, egyéb paraméterek alapján. (Na jó, a Small Business Server felhasználói valószínűleg nem üzemeltetnek olyan sok nyomtatót, hogy így kelljen keresni, de a lehetőség megvan rá – és hát, ki tudja? :-)).

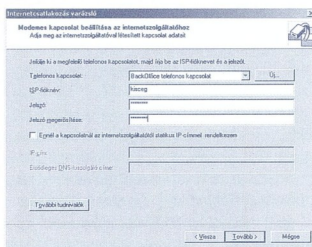
Az Internet Information Services konfigurálása

Mármint, a teendőlista fenti nevű menüpontja lesz a következő bekezdés témája. Ha az ISA Server-t telepítettük a kiszolgálóra, akkor néhány beállítást az IIS-en is el kell végeznünk. Kattintsunk a teendőlistán a címben említett feladatra! Az elinduló „varázsló” (inkább csak tanonc) egyetlen ablakból áll, és a kiszolgáló belső IP címét kérdezi. Ha megadjuk (a belső hálózati kártya IP címét megnézhetjük az ipconfig paranccsal, az alapértelmezés 192.168.16.2), a varázsló leválasztja az IIS-t az összes többi IP-címről. A többi majd az ISA Server feladata lesz.

Az Internetsatlakozás varázsló

A következő feladat a kiszolgáló Internet-kapcsolatának beállítása, amely az eddigieknél jóval szerzeágzóbb feladat lehet. Cikkünkben egy olyan kis cég kiszolgálóját konfiguráljuk – lásd a leírást az előző számban –, akik modulus kapcsolattal csatlakoznak a saját Internet-szolgáltatójukhoz, ők tartják fenn a cég néhány postaldátáját is. A modemet az SBS telepítésekor már csatlakoztatták a kiszolgálóhoz. A feladat most egy modulus kimenő kapcsolat és (egyelőre) egy POP3-alapú levelezés beállítása lesz. Kattintsunk a teendőlistán az „Internetsatlakozás varázsló” sorra. A bekészítőképernyő után következő első oldalon máris kiválaszthatjuk, milyen jellegű hálózati kapcsolatot akarunk kiépíteni. Válasszuk a „Modem vagy terminálkártya” opciót és kattintsunk a Tovább gombra. A megjelenő ablakban az internetszolgáltató eléréséhez szükséges telefonos hálózati kapcsolat beállításait kell megadnunk.

Az Új... gombra kattintva hozunk létre egy új telefonos kapcsolatot (ha még nem volt), vagy válasszuk ki egy meglévőt.



☞ Megadjuk az Internetszolgáltatóhoz történő bejelentkezéshez szükséges adatokat

A következő oldalon az Exchange kiszolgáló levelezéssel kapcsolatos beállításai következnek: engedélyezhetjük, illetve tilthatjuk az Exchange kiszolgálón működő SMTP (levélküldés) illetve POP3 (olvasás) protokollokat. Egyelőre engedélyezzük mindkettőt, majd lépünk tovább.

A következő ablakban fontos adatot kell megadnunk: a varázsló az Internetes tartománynev után érdekli (lásd a tartománynev-probléma elemzést az előző részben). Most a levelezés tartománynevet kell megadnunk (esetünkben: kiscseg.hu), ami nem feltétlenül egyezik meg a kiszolgáló tartománynevével.

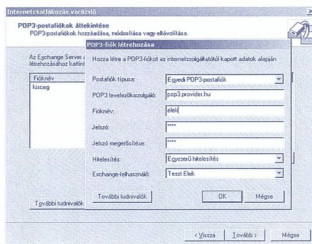
Ezután az SMTP levélküldési beállítások következnek. A levélküldése kétféle módon történhet:

- ☞ Az Exchange kiszolgálón minden kimenő levelet egy külső kiszolgálóhoz (célszerűen az internetszolgáltatóéhoz, ha engedi) továbbít, küzdjön az a konkrét továbbítással
- ☞ Az Exchange kiszolgálón ön maga kézbesíti a leveleket a DNS információk alapján. Ehhez nem kell külső segítség, viszont terhelheti a céges vonalakat.

Ha a szolgáltató lehetővé teszi az első opciót, válasszuk azt, az a nekünk mindenképpen az olcsóbb megoldás (ezesetben meg kell adnunk a célkiszolgáló nevét vagy IP címét).

A következő oldalon az Exchange levelezés fogadásának beállításai láthatók. Az ETRN levelezést egy későbbi alkalommal elemezzük majd részletesen, a POP3 postafiók-letöltésekhez erre nincs szükség (válasszuk a „Nincs jelzéküldés” lehetőséget).

A POP3 postafiók-beállítások következnek. Minden külső postafiókhoz hozzunk létre egy beállítást:



☞ A külső POP3 postafiók megadása



Kétféle POP3 postafiók készíthető: globális és egyedi. Az egyedi postafióknak megvan a helyi gazdája, az adott postafiókba érkező levelek egy-egy helyi felhasználó postaládájában landolnak. A globális postafiókokból letöltött levelek a céges terjesztési listára kerülnek.

Ezután a külső postafiókok ellenőrzésének gyakoriságát állíthatjuk be (*modemes kapcsolatnál, mint esetünkben is, ez érzékeny szempont*) – az alapértelmezés egy óra. A kimenő levelek a bejövő levelek lehetőségek kerülnek elküldésre. Az ISA Server (*a tűzfal*) konfigurációja következők: be kell állítanunk, milyen szolgáltatásokat engedünk be az Internet-felől. A varázsló a következő lehetőségeket ajánlja fel:

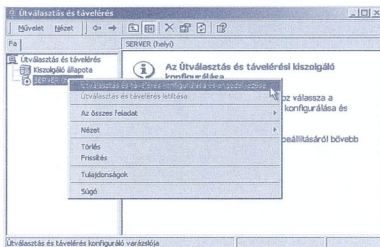
- ☑ Levelezőkiszolgáló – kétirányú SMTP és bejövő POP3 forgalom engedélyezése
- ☑ Webkiszolgáló – az Internet Information Server elérése HTTP és HTTPS protokollok segítségével
- ☑ Webes levelezőkiszolgáló – az Exchange Server webes felületének (Outlook Web Access) engedélyezése
- ☑ Virtuális magánhálózat – a kiszolgáló elérése PPTP virtuális magánhálózat segítségével
- ☑ POP3 – az Exchange postafiókok elérése POP3 protokoll segítségével
- ☑ FTP – az IIS FTP szolgáltatásának engedélyezése
- ☑ Terminálkiszolgáló – a kiszolgáló elérése terminálkliens segítségével.

A beállítások jóváhagyása után a varázsló nagy munkába kezd, aminek a végén cégünknel létrejön a beállításoknak megfelelő Internet-hozzáférés.

A távelérés konfigurálása

A teendőlista további elemei már mind-mind csak a sugóra mutogatnak, konkrét varázslókat közvetlenül már nem indíthatunk nekünk. A következő lépés a távelérés konfigurálása, ami a RAS, az esetleges virtuális magánhálózatok, illetve a terminál-hozzáférés beállítását jelenti. A virtuális magánhálózat létrehozásától egyelőre tekintsünk el (*egyrészt az elég komoly tűzfal-téma, másrészt pedig általában csak állandó hálózati kapcsolat esetén van értelme*), de a betárcsázásos, illetve terminálon keresztüli hozzáférések van létjogosultsága (*persze csak ha nem foglalt a modem – legjobb, ha erre külön módon van fenntartva*).

A távelérést (*kapcsolt vonali telefonos hálózati kapcsolatot*) a Windows 2000 „Útválasztás és távelérés” (*ismerősebb nevén Routing and Remote Access, RRAS*) szolgáltatása biztosítja, így annak beállításához is külön eszközt kell használnunk. (Érdemes megjegyezni, hogy a most következő információ érnél fogva nem csak a Small Business Server 2000-re, de minden Windows 2000 Server-re is érvényes, legyen az SBS-tag vagy sem.) Miután megkerestük és elindítottuk, kattintsunk jobb gombbal a fában a kiszolgálónk nevére, és az „Útválasztás és távelérés konfigurálása és engedélyezése” menüpontot választva indítsuk el az RRAS varázslót.



☑ Az RRAS szolgáltatás beállítása és engedélyezése

A megjelenő varázsló üdvözlését követő oldalon válasszuk a „Távelérési kiszolgáló” opciót. A következő oldalon ellenőrizzük, hogy a TCP/IP szerepel-e a felsorolt protokollok között, majd lépünk tovább. Most azt kell eldöntönnünk, hogy a betárcsázó, és így a belső hálózat részévé való számítógép milyen IP címet kapjon. Az egyértelmű, hogy egy belső IP címre van szüksége, a kérdés már csak az, hogy ezt az IP címet honnan szerzi.

Ha a hálózatban van DHCP kiszolgáló (*vagy éppen maga az SBS kiszolgáló az*), választhatjuk az automatikus címkiosztást; a kliens ilyenkor a DHCP-től kap IP címet, mintha csak a helyi hálózatra csatlakoztatott volna.

Ha nincs DHCP kiszolgáló, kézzel kell megadnunk egy legalább két címből álló tartományt, lehetőleg ugyanabban a belső címtartományban, ahol a többi ügyfélszámítógép is tartózkodik (*például 192.168.2.100-192.168.2.107*). A kiszolgáló aktuális IP címét az ipconfig parancssori eszközzel ellenőrizhetjük. A varázsló RADIUS használatát firtató kérdésére adjunk nemleges választ. A varázsló ezzel befejezte az adatok gyűjtését; némi telepítgetés után elindul a távelérés szolgáltatás, amit az MMC ablakban a kiszolgáló neve mellett megjelenő zöld ikon (és a konfigurációs fa lényeges megváltozása) is jelez.

A távelérés engedélyezése

Maradjunk még ebben az ablakban! Keressük meg a fában a „Távelérés házirendjei” csomópontot, és kattintsunk rá. Egyetlen házirendet találunk a listában, aminek a neve mindent elárul: „Hozzáférés engedélyezése, ha van betárcsázási engedély”. Ha kettőt kattintunk erre a sorra, megjelenik a házirend tulajdonságlapja. Láthatjuk, hogy a hozzáférés feltétele az, hogy vasárnap 00:00 és szombat 24:00 között érkezzen be a hívás – ennek a feltételnek valószínűleg mindenki meg fog majd felelni :-).

De figyeljük csak meg az oldal közepén található beállítást: „Ha a felhasználó teljesti a fenti feltételeket ... Ne kapjon távelérési engedélyt”. Most akkor jöhet vagy nem jöhet? Bizony jöhet, mert ugyan ez a házirend korlátozhatja ugyan a felhasználók hozzáféréseit, de a felhasználói objektum adott beállítása felülbíráhatja ezt.

Magyarul, akinek a felhasználói tulajdonságlapján a hozzáférés engedélyezve van, az jöhet. Ellenőrizzük ezt a beállítást az előző lépésben létrehozott felhasználónál (*az „Actív Directory – felhasználók és számítógépek” eszközt az SBS konzolból is elérjük*), a tulajdonságlap „Behívás” oldalán.



Teszt: Elők tulajdonságai

Tárvédelem	Terminálszolgáltatások profila				
Exchange General	Email Adókönyv				
Albáncok	Cím	Fők	Profil	Telefonoszektorok	Szervezet
A következő csoportok tagja					
Tárvédelmi engedély (behívás vagy VPN)					
<input checked="" type="radio"/> Elérés engedélyezése					
<input type="radio"/> Elérés tilbása					
<input type="radio"/> Elérés vezérlése a RAS-házirend alapján					
<input type="checkbox"/> Hízőazonosító ellenőrzése					
Visszahívási beállítások					
<input type="checkbox"/> Nincs visszahívás					
<input type="checkbox"/> Hívó által megadott számon (csak útválasztás és RAS)					
<input type="checkbox"/> Mindig ezen a számon					

• A felhasználó betárcsázási beállításai

Itt adhatjuk meg, hogy a felhasználó használhatja-e a szolgáltatást, vagy sem (a harmadik opció – mármint hogy a RAS házirend döntse el – azért nem választható, mert az SBS tartományunk NT4-kompatibilis, ún. mixed módban üzemel, és ezt a beállítást az NT4 tartomány nem képes megemészteni). Ugyanitt található a visszahívásra vonatkozó beállítások is. Térjünk vissza most a házirendhez! Ha a házirend idő „korlátozása” non-stop, a házirend beállítását pedig a felhasználói beállítás ügyis felülbírálja, akkor miért van szükség házirendre? Megkapjuk a választ, ha a „Profil szerkesztése” gombra kattintunk – valamilyen profil ugyanis érvényesül mindenkire, aki bejelentkezik a kiszolgálóra. Esetünkben ez az egyetlen profil, tehát ez hullik minden felhasználó fejére (hacsak nem csinálunk másmilyen profilt mondjuk a hétvégi időszakra – no pont ez a RAS házirend lényege).

Beállítások profil módosítása

Helyezés	Titkosítás	Speciális
Behívási megközelítés	IP	Működés
<input type="checkbox"/> Vignál beírása ennyi idő úresjáról után:	<input type="text" value=""/>	perc
<input type="checkbox"/> Egy munkanemert hosszának korlátozása:	<input type="text" value=""/>	perc
<input type="checkbox"/> Elérés korlátozása ezekre a napokra, órákra		
<input type="checkbox"/> Beállítások csak erre a számon		
Behívási módoszer korlátozása		
<input type="checkbox"/> ADSL-DNT - aszimmetrikus DSL (inkább költség)		
<input type="checkbox"/> ADSL-CAP - aszimmetrikus DSL-vidéki applikáció fáda...		
<input type="checkbox"/> Ethernet		
<input type="checkbox"/> ISDN - ISDN digitális aláló vonal		
<input type="checkbox"/> XDSL - szimmetrikus DSL		

OK Mígga Alkalmaz

• RAS profil

Ha végignézegetjük a profil oldalait, láthatjuk, hogy mely korlátozásokat hozhatunk a felhasználóval szemben:

- Korlátozhatjuk a behívás időtartamát, a maximális üresjáratit időt
- A behíváshoz használt telefonszámot
- A behívási „módszer” – nevezük inkább behíváshoz használt médianak, hiszen ma már nem csak telefonon keresztül lehet RAS-olni, hanem sok más módon is, nézzük csak meg a listát (némi rövidítés fordítása kifejezetten szárazkötő :-)
- IP és intelligens multilinck-beállítások, csomagszűrő (!)
- Hitelesítési és titkosítási beállítások

Aki nem felel meg a követelményeknek, nem érvényesül rá a házirend – akire pedig egy házirend sem érvényesül, az nem használhatja a távoli elérést! Ezért az alapértelmezett, mindenkire érvényes házirend alapértelmezésben a rendszerben.

A terminálszolgáltatás elérésének beállítása

Ahhoz, hogy a rendszergazdákon kívül más felhasználók is igénybe vehessék a Windows 2000 terminálszolgáltatását, további beállításokra van szükség. Mindenekelőtt, magára a terminálszolgáltatásra kell jogot adnunk: indítsuk el a Start / Programok / Felügyeleti eszközök / Terminálszolgáltatások konfigurálása eszközt, abban pedig válasszuk ki a „Connections” (sic! – [mICK]) csoportot. Kattintsunk jobb gombbal a jobb oldalon megjelenő RDP-Tcp sorra és válasszuk a Tulajdonságok parancsot. A megjelenő tulajdonságlap „Engedélyek” oldalán vegyük fel új felhasználóként a már létező „BackOffice Remote Operators” csoportot, és adjunk részükre „Felhasználói hozzáférés” engedélyt. (Ha egy felhasználót a „Kiemelt felhasználók” sablon segítségével hozunk létre, tagja lesz a fenti csoportnak – persze a csoporttal kézzel is bárkit hozzáadhatunk).

Még egy fontos beállítás van hátra: az alapértelmezett tartományi házirend nem teszi lehetővé, hogy halandók belépjenek a tartományvezérlő szentélyébe. A szentségtöréshez nyissuk meg a Start / Programok / Felügyeleti Eszközök / Tartományvezérlő biztonsági házirendje eszközt, és ott keressük meg a Biztonsági beállítások / Helyi házirend / Felhasználói jogok kiosztása csomópontot. A jobb panelen keressük meg a „Helyi bejelentkezés” sort, kattintsunk rá kétfőt, majd a megjelenő dialógusablakba vegyük fel a fent említett csoportot is. A beállítások néhány perc múlva érvényesülnek.

Legközelebb folytatjuk a Small Business Server 2000 alapbeállításainak áttekintését, addig is jó kísérletezést kívánok mindenkinek!

Külp Miklós
mick@netacademia.net



Routing and Remote Access Services (I. rész) – DUN



A Windows 2000 rengeteg beépített komponense közül talán a hálózati komponensek szolgáltatásválasztéka a leggazdagabb. Ez egyáltalán nem meglepő, ha tudjuk, hogy a Windows 2000 minden változata (*Professional, Server, Advanced Server*) „Internet ready”. Más részről mára a kiszolgálószolgáltatások olyan széles választéka jelent meg, amiről öt évvel ezelőtt csak álmodozni tudtunk. Emlékszem, 1995-ben, a Windows NT 3.1 megjelenésekor szinte egyetlen „értelmes” kiszolgálótermék sem létezett. Hol volt még az Exchange? Volt helyette Microsoft Mail (*jó kis shared folder megoldás...*)! Hol volt az SQL Server? IIS? Proxy? SMS? Nehéz is volt elképzelni, vajon mi a csudának kellett egy Windows NT, amikor a Novell Network fileakkora vason nyújtotta pont ugyanazt. Nagyon változott a világ: már senkinek sem elég, amit anno a csúcsok csúcsának hittünk: fájl és nyomtatószolgáltatás mindenkinek. És hol volt az Internet? Sehol! Ma már elképzelhetetlen, hogy egy informatikusnak ne legyen email címe; akkoriban Microsoft megoldázzállítónál dolgozva sem jutott, csak egy céges postafiók a Rolitronnál (@zabene.hu).

S nem fejleszthetne boldog-boldogtalan hálózatban futó szolgáltatást, ha az operációs rendszer helyett a programozóknak kellene megvalósítani a magaszintű hálózati szolgáltatásokat. De antitrószert per ide, tözsekrach oda, a Microsoft telemtönte operációs rendszerét hasznosabbnál hasznosabb hálózati szolgáltatással. Már-már annyi van, hogy képtelenség kiigazodni közöttük: IPsec, Qos, IGMP, OSPF, BAP, VPN, EAP, Radius, L2TP stb. Mi a csudák ezek? Ha egyelőre talán ismeretlenek is, annyit elárulhatunk: szabványos technológiák. Már megint RFC olvasásra kell buzdítanom kedves olvasóinkat. Minden vállalatnál folyamatosan fejlődik a hálózat. Nemrég beszerzett, de máris elavult útválasztókat dobunk ki laza mozdulattal, a hubot switchre cseréljük, a Cisco routert pedig RRAS-ra (...). Hálózattervezési szempontból is érdemes ismerni a fenti rövidítések mögött meghúzódó technológiákat. Cikksorozatomban a Routing and RAS szolgáltatásválaszték szeretném konkrét, szemléletes példákkal ismertetni, hogy – ha már egyszer megvettük – használjuk ki, ami a kezünkben van. Természetesen ezer részen keresztül is lehetne elemezni a sok-sok hálózati szolgáltatást, a RRAS-ra koncentrálni mégis kézen tartható az információ mennyiség. Természetesen ez azzal jár, hogy bizonyos ősi hálózati szolgáltatások kimaradnak e sorozatból. Így siránkozva bár, de lemondtam a DHCP, DNS, WINS és társaik ismertetéséről.

Ügyfelek és kiszolgálók

Ügyfél-kiszolgáló korszakban élünk, minden hálózati szolgáltatás felbontható kiszolgáló-, és munkaállomásokra komponensekre. Nincs ez másként az RRAS-nál sem: RAS kiszolgálókról és DUN (*Dial Up Networking*) ügyfelekről beszélhetünk. A RRAS is kétféle bontható: Routing (1) és Remote Access (2) komponensekre. Ide-oda csapongva maj a DUN, hol az RRAS telepítését, majd funkcióit elemzem majd.

Ami a W2000 Serverben RRAS - az a W2000 Professionalban DUN névre hallgat.

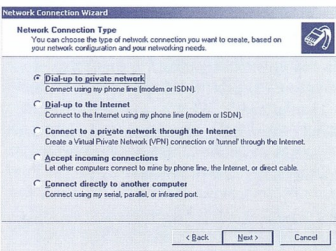
Eközben természetesen némi kavarodásban is osztályrészünk lesz, hisz rengeteg kiszolgálófunkció megtalálható akár a Windows 2000 Professionalban is, gondoljunk csak a Connection Sharingnek nevezett NAT-re, vagy egyszerű TCP/IP routálásra.

Ami az RRAS-ban NAT - az a DUN-ban Connection Sharing!

Az azonban általánosan elmondható, hogy a RRAS kiszolgáló mindent jobban tud. Jobb a NAT-ja, okosabb az útválasztója, izmosabb a VPN-je, mint az egyszerűbb DUN-nak, továbbá csinos kis házirendjei vannak, melyekkel igazán finoman hangolható a működése. Egy kevés IAS-sal szerezve a házirendek központosíthatók is (*íme egy policy, mely nem Active Directory integrált!*).

DUN szolgáltatások

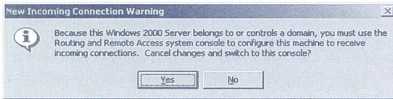
A DUN alapvető szerepe, hogy a központi RRAS kiszolgáló által felkínált szolgáltatáshalmazt kiaknázza kapcsolatot építson fel. Hol tudnánk utána nézni, hogy mire képes még? Nagyon logikus helyen van: Start menü -> Settings -> Network and Dial up connections. A Windows 2000-ben végre egy helyre került (*szinte*) az összes hálózati szolgáltatás, legyen az lassú, vagy gyors, modemes vagy Ethernet-alapú. Az ablakban kettőt kattintva a Make New Connection ikonon elindul a DUN bővítés, és megmutatja a DUN képességeit.



☛ A Network Connection Wizard

- ☛ Dial-up to private network: ez a lehetőség igazi informatikusoknak való, a kapcsolátpézt minden részlete a kezünkben van. Mivel feltételezem, hogy minden olvasóm legalább hatvszoros futott végig a varázsló ezen ágán, nem részletezném.
- ☛ Dial-up to the Internet: ez a pont igazi jüzcseremege: elindul az Internet Connection Wizard, amely a bővös 4298300-os számot felhívva megpróbálja letölteni a magyarországi Internetszolgáltatók aktuális listáját.

- ☞ Connect to a private network: virtuális magánhálózatokhoz történő csatlakozás esetén használandó. Érdekesége, hogy – virtuális hálózat lévén – egy másik, „nem-virtuális” kapcsolat használatára szorul, arra ráépül. Ha a két kommunikáló fél között nincs élő hálózati kapcsolat, akkor a virtuális sem épülhet fel!
- ☞ Accept incoming connections: ez a lehetőség mini RAS kiszolgálóvá változtatja munkaállomásunkat, egyetlen kapcsolat fogadására felkészítve a másinát. Tartományi tag gépek a varázsló a következő üzenettel hajt el Windows 2000 Serveren (szépen mutatva a DUN és a RRAS közös gyökerét):



☞ **Tartománytag gépeken bejövő kapcsolatok kezelésére DUN helyett RRAS használandó**

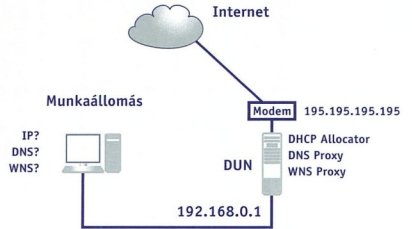
- ☞ Connect Directly to another computer: közvetlen kábelkapcsolat (soros vagy párhuzamos) két gép között. SP1 előtt bizonyos géptípusok esetén nem működött. Most vizsgáljuk meg, mi van a varázslat mögött, hisz Next-et nyomogatni mindenki tud, ebből nem lesz hálózattervezés. Ami mindegyik kimenő kapcsolattípusban közös, az a megoszthatóság, és a kívánságra történő tárcsázás.

Kapcsolatmegosztás (Connection Sharing = NAT = RFC 3022)

A hálózati kapcsolat ugyanolyan közös erőforrássá vált, mint egy nyomtató, vagy egy könyvtár? Végül is igen. Odaklatty és már meg is van osztva. Bámulatosan leegyszerűsítették a redmondí fiúk ezt a funkciót, pedig a háttérben égszakadás-földindulás történik! Ellentétben ugyanis a fájl- és nyomtatómegosztással, ahol a munkaállomás is tisztában van az igénybe vett szolgáltatás jellegével (sőt, együttműködik a kiszolgálóval a feladat végrehajtásában és például dinamikusan letölti a legfrissebb nyomtatómeghajtót), a Connection Sharing (a továbbiakban: CS) ügyfelei bambán küldözgetik csomagjaitka a „kiszolgálónak”, akiről egyszerűen azt hiszik, hogy egy Default Gateway. A kapcsolatot megosztó gép feladata a megosztásra csatlakozók teljeskörű vizsgálása; öntudatlan ügyfelekkel ez nem is történhet másként.

A 3022-es RFC megfogalmazása szerint: a NAT feladatra IP címek áttálcázása egyik IP tartományról egy másikra anélkül, hogy a végfelhasználók ezt észlelnék.

A kapcsolatmegosztás nemcsak egyszerű, hanem egyéb előnyei is vannak: segítségével egy teljesen hétköznapi, potom árú, sőt sok esetben ingyenes Internetkapcsolatot teszünk elérhetővé egy kis hálózat néhány gépén, anélkül, hogy további modemekre, vagy a tárcsázások megkapott dinamikus IP címen kívül további címekre lenne szükségünk. Az alábbi ábrán egy CS kiszolgáló üzembehelyezése előtti pillanatot láthatjuk, amikor az égszakadás-földindulás beindul.

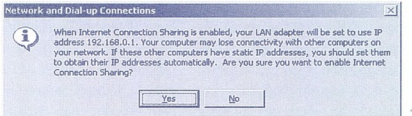


☞ **Hálózatunk a Connection Sharing bekapcsolása előtti másodpercen**

Vajon hogyan kellene beállítani a munkaállomás IP címét, hogy kilásson a netre? Belső címet kell használnunk, mivel egyetlenegy valós IP címünk van, amit tárcsázáskor kapunk a szolgáltatótól - további valós IP címek nem állnak rendelkezésre. Természetesen a DUN gép legyen a Default Gateway. Ha a munkaállomások száma több, mint egy, már érdemes dinamikus IP cím kiosztáson gondolkoznunk. A CS is erre gondol, és gyorsan telepít egy ál-DHCP Servert, egy szköp nélkül, elengedett kézzel üzemelő IP címosztogatót (DHCP Allocator). A teljesen automatikus DHCP működésnek ára van! A DHCP Allocator gépen:

- ☞ a munkaállomásoknak kiosztott IP címek kötelezően a 192.168.0.0 tartományból származnak
- ☞ az Allocator önmagát tünteti fel DNS Serverként a leosztott IP címekben
- ☞ az Allocator önmagát tünteti fel WINS Serverként a leosztott IP címekben

Ennek megfelelően a DUN hálókártyája a 192.168.0.1 címet veszi fel, hogy azonos hálózatra kerüljön saját kinevelse!



☞ **A Connection Sharing ragaszkodik a 192.168.0.1 címhez – a NAT nem**

Mindezeket túl a munkaállomásokat dinamikus IP cím felvetelre kell állítani. Ez van inges. És működik. Sőt, ha begondolunk, a CS igen okos! Mit kell tenni a DUN géppel? Jobbkattaj, Properties, Sharing és kész. Mit kell tenni a munkaállomásokkal? Semmit, hisz automatikus IP cím beállításal települnek. Semmit sem lehet beállítani, hisz semmit nem is kell beállítani. Vagy mégis? Mi van, ha nem vezethető be pl. a 192.168.0.0 IP cím tartomány? Ha az ímént felsorolt szigorú szabályok valamelyike nem engedhető meg hálózatunkon, a CS helyett az RRAS-féle NAT-ot kell választanunk, mert az sokkal rugalmasabb. Lebeszélhető akár a DHCP Allocator telepítéséről, a DNS Proxyról, a 192.168.0.0 IP tartományról – egyszóval minden paramétere állítható.

Címfordítás

Következő vizsgálatunk tárgya a CS működése. CS=NAT, azaz Network Address Translator, címfordító kisiparos, aki a

belső IP címekből külsőt csinál, és vice versa. Vegyük például a webböngészést.

Egyszerű eset

A munkaállomás a böngésző elindul (*mondjuk*) a 192.168.0.11 belső IP címen, és a 2345 porton. A CS átalakítja ezt a belső címet az egyetlen létező valós címre (195.195.195.195), a portot pedig érintetlenül hagy(*hat*)ja. A távoli webböngésző a 195.195.195.195 címről, és 2345 portról érkező kérészt észlel, s a választ is oda küldi. A CS „emlékszik”, hogy melyik belső gép nevében járt el, és oda továbbítja a csomagot.

A Connection Sharing címfordítási táblázata

LAN-ról	Internet-re
192.168.0.11	195.195.195.195

Byonyolult eset

Legyen két munkaállomásunk (192.168.0.11 és 192.168.0.22). Mindkettő elindul a böngésző. Hogy-hogynem, a véletlen fura játéka folytán mindkét gépen mindkét böngésző a 2345-ös porton indul. Míg az első gép böngészőjét a CS simán kiengedi az előző módon, addig a második gépet már nem ereszteti az előző módon, addig a második gépet már nem ereszteti ki változatlan portszámmal, mert a 2345 már FÖGLALT. Akármekkora varázsló is a CS azért egyetlen kimenő IP címről két azonos számú TCP portot nem képes nyitni a külvilág felé. Így kénytelen-kelletlen másik portot ad neki, például a 2345 utáni első szabad portot, a 2346-ot.

A Connection Sharing címfordítási táblázata

LAN-ról	Internet-re
192.168.0.11:2345	195.195.195.195:2345
192.168.0.22:2346	195.195.195.195:2346

Belátható, hogy minél több belső gépet kell „kifordítani” a valós címre, annál nagyobb eséllyel kell az IP címen kívül a portszámot is babrálni. Ez olyan esetekben okozhat gondot, ha a portszám nemcsak a TCP/UDP fejlécben szerepel, hanem a csatornában utazó alkalmazásprotokollban is. Tipikus példája ennek az FTP (*az aktív és passzív FTP összehasonlításáról a júniusi tech.net magazinban olvashatnak*), melynél élénk portszámátvitel zajlik a PORT parancs segítségével. Ha a CS nem módosítaná megfelelően az FTP parancsot, meghiusulna a kapcsolat. De módosítja. Mert tud FTP-ül. Az alkalmazásszintű portbuherátor-kiegészítőket NAT editoroknak hívják, ezek segítségével válik lehetővé a következő protokollok sikeres címfordítása:

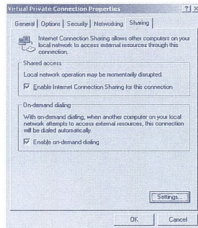
• FTP, ICMP, PPTP, NetBIOS, H.323, Direct Play, LDAP alapú ILS regisztráció, RPC

További NAT editorokat a Windows 2000 nem tartalmaz. Ha valami véletlenül mégis működik (*pl. HTTP*), az azért lehetséges, mert az adott protokoll külön editor nélkül is elviseli a jobbra-balra címfordítást.

Alkalmazáspublikálás

A CS fő funkciója természetesen a dolgozók böngészéseinek kijuttatása az Internetre. Néhány helyen ugyanakkor felmerülhet annak igénye, hogy egy belső gépen futó szolgáltatást elérhetővé tegyünk az Internet felől, beengedjünk bizonyos forgalmakat (*ez természetesen fix IP cím*

igényel – mondjuk, hogy van, és helyesen be is állítottuk). Ilyenkor meg kell fordítani a CS-t, és a kintről, mindenféle címről érkező hívásokat egyetlen belső géphez kell hajjítani. Ha például egy webkiszolgálót publikálunk, akkor a CS-t rá kell venni, hogy a hozzá kintről befutó, 80-as porti kéréseket belső webmasinánkhöz irányítsa:

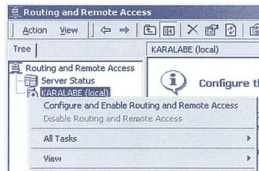


Webkiszolgáló láttatása a kinti világ felé

Már csak azt kell biztosítanunk, hogy a CS kapcsolat kívánságunkra magától felépüljön. Ezt megint a kapcsolat tulajdonságlapján, a Sharing fülön találjuk. A demand dialing segítségével a CS mindig felkappa a telefont, valahányszor kinti címet próbálunk elérni.

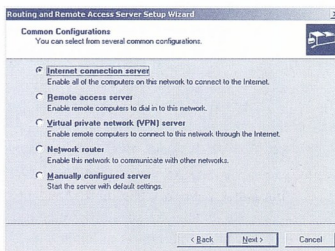
A Connection Sharing nagytestvére: a NAT

Ha a fenti szolgáltatást tartományi tag Windows 2000 Serveren keressük, akkor ne a DUN-ba kattintsunk be, hanem az RRAS-t. Hogyan kell telepíteni Serverre az RRAS-t? Sehogya. Gyárilag rajta van, csak le van tiltva, engedélyezniük kell:



Az RRAS engedélyezése

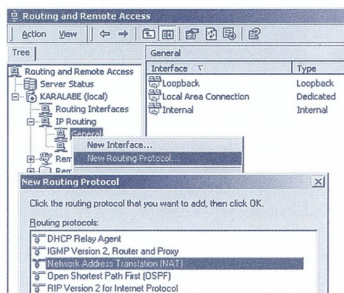
Az engedélyezési parancsra elindul egy varázsló, melyről csak tüzetezesebb vizsgálat derítheti ki, hogy nem ugyanaz, mint ami a legelső ábrán szerepelt (*és amelyről részletesebben e havi SBS cikkünkben olvashatnak. Munkamegosztás!*)



Az RRAS telepítővarázsló

- ☞ Internet Connection Server: NAT-ot varázsol a gépre
- ☞ Remote Access Server: betárcsázó ügyfelek fogadására
- ☞ VPN Server: virtuális magánhálózat végpontja
- ☞ Network Router: RIP vagy OSPF protokollal, esetleg kézi-
leg beállított LAN-LAN, vagy WAN-LAN router
- ☞ Manually configured: csak profioknak. Én – szerényen - a
manuálisan megoldást szoktam használni.

További különbség az előző DUN varázslóhoz képest, hogy egyen RRAS a fenti funkciók mindegyikét akár egyszerre, egyidőben el tudja látni, az összes funkciót fel lehet rá utólag aggatni. Kézi esetben így lehet a NAT-ot utólag hozzáadni:



☞ Új szolgáltatás telepítése utólag

A NAT telepítése után a következő beállítási lehetőségeink lesznek, melyek nem léteztek a CS-féle kistestvér megoldásban:

Funkció	Connection Sharing	NAT
Hálózati kártyák szerepe beállítható	Nem	Igen
Egynél több valós IP cím használata	Nem lehet	Lehet
Egyedi alkalmazások portösszerendelése	Nincs	Van
Címfordítási tábla TTL-je	Nem módosítható	Módosítható
DHCP Allocator	Fix	Beállítható
DNS és WINS proxy	Fix	Átállítható

A CS esetében a nem-NAT kártya automatikusan belsőnek számít. Egynél több kártyát nem is képes kezelni, mert a belső hálózaton a 192.168.0.0 cím is megváltoztathatatlan. Ha több (fix) IP címet kaptunk a szolgáltatóunktól, NAT használata esetén lehetővé válik ezek egyidejű használata, ami nagy segítség, ha rengeteg kapcsolatot kell kiszolgálni, mert később fognak el a szabad portok (bár 65535-nél több kapcsolatot egyetlen NAT kiszolgálóra bízni nem szokás).

A NAT címfordítási táblázatának TTL-jét is átállíthatjuk (mennyi idő után fejejtse el egy összerendelés. TCP-nél alapértelmezésben 1440 perc, azaz 1 nap). A kistestvér CS-nél ez a paraméter módosíthatatlan. A NAT további előnye, hogy beállítható, mi legyen a DHCP Allocator címtartománya (ha egyáltalán bekapcsoljuk). Ez adja az RRAS-féle megoldás igazi erejét: nem kötelező átlnni a belső hálózaton a 192.168.0.0 tartomány használatára!

A DHCP Allocator és DNS Proxy nemcsak ki/be kapcsolható, hanem csodálatos statisztikákat is kérhetünk ténykedésükről.

A CS/NAT és a Proxy/ISA összehasonlítása

Az eddigiekből bizonyára úgy tűnik, hogy a CS/NAT páros pontosan ugyanazt tudja, mint a Proxy/ISA versenyzők. Címet fordít mindegyik. Ez igaz, azonban a Proxy/ISA duó ezeken felül tud még egyet-s-mást! Az alábbi összehasonlító táblázatból kiolvasható, érdemes-e ISA-ra költönni? (Igen.)

Funkció	CS/NAT	Proxy/ISA
Autentikáció	Nincs	Tetszőleges
Jogosultságkezelés	Nincs	Felhasználónként
Gyorsítótár	Nincs	Van
Csomagszűrés	Nincs	Van
Tűzfalfunkciók	Nincs	Van
Költség	Ingyenes	Fizetős

A CS/NAT párosnak akkor, és csak akkor nincs költségvonzata, ha az oprendszert megvettük (ugyebár)!

Fóti Marcell
marcellf@netacademia.net

A cikkben szereplő URL-ek:

[1] RFC 1631 és 3022: The IP Network Address Translator (NAT) <http://www.ietf.org/rfc/rfc3022.txt>

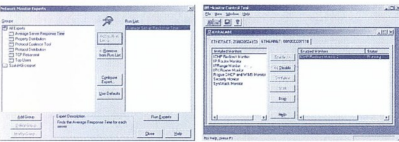
NetMon (VIII.rész)



Eddigi NetMon témaköreink esetében a Network Monitor offline elemzőképességét használtuk ki, azaz elkaptunk egy bizonyos hálózati forgalmat, s utólag, a kandalló melegénél elemeztük. Nem vettünk igénybe semmilyen segédeszközt az elemzés elvégzéséhez a saját józan paraszti eszünkön kívül, pedig a NetMon kétféle lehetőséget is tartalmaz! Az egyik elemzési eszköz az Experts névre hallgató jószág, melyet az elkaptott hálózati forgalomra eszrevet csodálatos statisztikákat kaphatunk. Sok esetben azonban az utólagos okoskodás késő bánat, mert a baj már megtörtént (például Distributed DOS támadás áldozatul esett kiszolgálónk, vagy a hálózat munkadőlomásai egytől egyig elveszítették kapcsolatukat a külvilággal).

A másik elemzési eszköz a Monitor Control Tool, mely segítséget nyújt ismert hálózati problémák azonnali, real-time érzékelésében is. Futtatásával folyamatos hálózatmonitorozásra nyílik lehetőség; kihasználja a promiszkus üzemmódban rejlő elemzési lehetőségeket, közismert hálózati problémák elég jelentős halmozát érzékeli, és jelenti számunkra. További érdekes lehetőség, hogy akár távoli hálózatok figyelése is megvalósítható, ha minden alhálózaton beizzítunk egy-egy Monitor Control Toolt, s ezeket központi helyszínről felügyeljük. A cikk hátralévő részében szándékosan felváltva írok mindkét lehetőségéről, ezzel is hangsúlyozva, hogy KETTŐ van belőlük, melyek felhasználói felülete eléggé hasonló – de ezzel a hasonlóságok listájának végére is értünk. A kavarodás csökkentése érdekében mindkét eszköz mellé odateszem a megfelelő ikont, így első pillantásra látszani fog, éppen melyikről van szó.

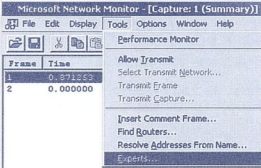
Eszköz	Ikon	Üzem mód
Monitor Control Tool		Online
Expert		Offline



◀ Baloldalon az Expertek, jobboldalon a Monitor Control Tool felhasználói felülete

Végül egy mondat erejéig megemlítem az általam ki NEM próbált, de olvasott lehetőséget: a NetMon WMI Provider segítségével programozási eszközökkel hozzáférhetünk a Monitor Control Tool riasztásaihoz, s tetszőleges módon reagálhatunk rájuk: SQL Serverbe naplózhatjuk, emailt írhatunk stb. Hol találjuk az elemzési eszközöket?

 **A NetMon Experts fellelési helye**
NetMon Tools->Experts.



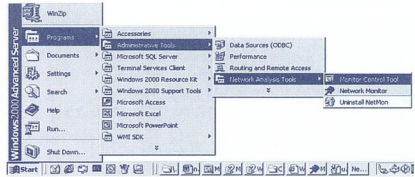
◀ A szakértők offline elemzésre használhatók

Vigyázzunk a NetMon ravasz menürendszerével, ez a menüpont csak akkor jelenik meg, ha egy elkaptott hálózati forgalom részletező ablakában állunk!

A felhasználás módja inkább memória-, mint processzor- vagy hálózatintenzív, hisz egy elmentett trace fájlom végzünk statisztikázást. Tipikus esetben néhány másodperc alatt lefut, így – ellentétben a másik eszközzel – különböző tervezést nem igényel a használata.

 **A Monitor Control Tool elérése**

Külön programként található a Start menüben. Nem (csak) ebből következik, hogy ezt az eszközt nem érdemes egyidőben futtatni a NetMonnal: real-time elemzésével jelentősen leterheli azt a gépet, amelyiken futtatjuk.



◀ A Monitor Control Tool nem azonos a NetMonnal!

Hálózat-, és processzorintenzív működése miatt általában dedikált gépen futtatjuk – ott viszont éjjel-nappal, mert a hálózati problémák (különösen a hackertámadások) nem csak nappal léphetnek fel.

 **Mit tudnak a szakértők (experts)?**

Utólagos okoskodásra, jobbára statisztikakészítésre szolgálnak. Az alábbi táblázat röviden összefoglalja az expertek szerepét:

Expert	Mire való?
Average Server Response Time	Általunk kiválasztott (pl. FTP, SMTP stb.) kiszolgálóalkalmazások válaszidejének kiszámítása

Property Distribution	Egyes protokollok mezőinek megoszlása (pl.: TCP – Destination Port)
Protocol Coalesce Tool	Protokoll-összevonó. Töredezett csomagok összevonása egyetlen nagyobbba
Protocol Distribution	Protokollelosztás. Mennyi egyedi protokoll van az elkapott forgalomban?
TCP Retransmit	Újraüldött TCP csomagok száma
Top Users	A legtöbbet forgalmazó felhasználók kilistázása

Mit tud a Monitor Control Tool?

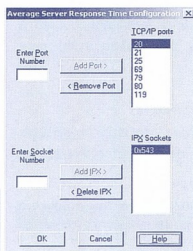
A real-time elemzés azon a felismerésen alapul, hogy a tipikus hálózati problémák tipikus hálózati forgalmat generálnak – magyarul a forgalom „mintázatának” (pattern) figyelésével lehetőség nyílik sok-sok esemény felismerésére. Az alábbi lista a beépített mintafelismerőket tartalmazza, melyeket a cikk hátralevő részében részletes elemzésnek vetünk alá, hisz első ránézésre – talán – nem egyértelmű, hogy milyen baj származhat ICMP Redirect üzenetek váratlan megjelenéséből. Tehát a lista:

Monitor	Mire való?
ICMP Redirect	Hibás routerüzenetek észlelésére
IP Router	A „hálózati routerek” életjeleinek figyelése
IPRange	A hálózaton kiosztott IP címektől eltérő címek felbukkanása ellen
IPX Router	Ugyanaz, mint az IP Router Monitor, azzal a különbséggel, hogy nem használjuk, mert úgyszincs IPX a hálózatunkon :-)
Rouge DHCP and WINS	Hamis DHCP és/vagy WINS észlelésére
Security	NetMon kukkolók kitiltása
SYN Attack	SYN támadás felismerésére

Most pedig térjünk rá az egyes Expertek és Monitorok gyakorlati hasznára. Ha a főnök görcsöl a számítógépével, mert dög lassú, és rád fogja, hogy azért nem boldogul, mert lassú a hálózat, akkor a válasz...

Average Server Response Time Expert

Ezzel az egyszerű eszközzel az elkapott hálózati forgalom alapján meg tudjuk állapítani az egyes kiszolgálók általunk kiválasztott portjain a válaszidőket. Gyárilag a következő portok szerepelnek a statisztikában: 20, 21, 25, 69, 79, 80, 119. Azaz FTP, SMTP, TFTP, Finger, HTTP és NNTP. Ez természetesen szabadon változtatható.

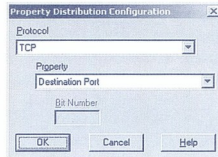


► Mely portok válaszidejét mérjük?

Ha például egy webkiszolgáló válaszidejére vagyunk kíváncsiak, egyedül a 80-ast hagyjuk meg. Ha pedig az előbbi listában fel nem sorolt SQL Server reakcióidejére, vegyük fel a 1433-as portot. Futtassuk le ezt a szakértőt mondjuk az áprilisi számhoz mellékelte `http.cap` fájlon, mely a szokásos helyről, az [1] címről tölthető le. Ebben a `www.netacademia.net` webkiszolgáló böngészésének hálózati forgalma szerepel, 5 hop távolságból. Az eredménynek 0,389009 másodpercnec kell lennie a fájlban lévő 78 HTTP kérés-válasz alapján. Bizonyító erejű információ került a kezünkbe ezáltal, hogy nem a hálózat a lassú, hanem a főnök XT-je :-). Most legyen az a feladat, hogy megállítsuk: Nincs-e adott gépnek mely portjain kopogtatnak a felhasználók? Gyocsd-támadás-előkészítés? Port scanning?

Property Distribution Expert

Ez a szakértő az elkapott hálózati forgalomból tetszőleges protokoll tetszőleges mezőjére képes statisztikát készíteni, jelen esetben a májusi `w2kboot.cap` fájl lesz kísérleti anyagunk az [1] címről. Állítsuk be az expertet, hogy a TCP protokoll Destination Portja alapján statisztikázzon nekünk egyet!



► A TCP protokoll Destination Portjait számláljuk

Az eredmények megnyugtatóak. Összesen négy kiszolgálóport dolgozott (135, 139, 389, 445), hackernek nyoma sincs. Ezek összesen a kapcsolatok $7+3+10+7=27\%$ -át adják.

Protocol	Property	# of Frames	% of Frames	Value	Label	% of Src
TCP	Destination Port	599	74			
TCP	Destination Port	42	5	135	Location Service	7
TCP	Destination Port	19	2	139	NETBIOS Session Service	3
TCP	Destination Port	60	7	389		10
TCP	Destination Port	43	5	445		7
TCP	Destination Port	75	9	1028		12
TCP	Destination Port	32	3	1029		5
TCP	Destination Port	10	1	1031		1
TCP	Destination Port	8	1	1032		1
TCP	Destination Port	16	1	1033		2
TCP	Destination Port	10	1	1037		1
TCP	Destination Port	20	2	1036		3
TCP	Destination Port	14	1	1042		2
TCP	Destination Port	12	1	1044		2
TCP	Destination Port	10	1	1046		1
TCP	Destination Port	8	1	1049		1
TCP	Destination Port	10	1	1051		1
TCP	Destination Port	20	2	1052		3

► A TCP protokoll Destination Portjait számláljuk

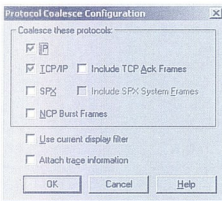
A 135-ös és 139-es porton futó szolgáltatásoknak a névt is visszakaptuk. A 389-es port az LDAP szolgáltatása, a 445-ös pedig a NetBIOS nélküli (letiltott NetBIOS-sal zajló) fájl-és nyomtatási szolgáltatás kapuja. Az összes többi (kliens)port használata elenyésző.

Következő feladatunk különleges: egy fájl kell darabjaiból összeállítanunk. Az Ethernet szabvány korlátai miatt a maximális keretméret nem haladhatja meg az 1514 bajtot. Minden adat, mely ennél nagyobb (lenne), feldarabolva ke-

rül a hálózatra, s a fogadó gép feladata a darabok összeillesztése. Akkor mi miért vacakoljunk vele? Mert nem MI vagyunk a fogadóállomás! Nálunk a NetMonban „csak úgy” a maga nyers, töredezett formájában megjelent, s ebből kelne visszaállítani az eredeti adatot...

Protocol Coalesce Tool

Végül is a probléma nem megoldhatatlan, hisz az eredeti fájl darabkái egyazon TCP csatornán haladnak kijelölt céljuk felé. Amit a célállomás meg tud tenni, azt mi is! Összegyűjtjük egy adott TCP csatorna összes csomagját, és abból kihalásszuk az adatot. Ismét dolgozzunk a http.cap fájlj! Eredetileg 181 frame van benne. Ha erre ráuszítjuk a Protocol Coalesce Toolt, akkor összesen tíz marad, de azok jó nagyok ám! Az összevont adatok külön fájlba kerülnek, valami hasonló néven: http(Coalesced)01.cap. Nézzünk bele! Az ötödik csomag, mely egy HTTP Response, összesen 58524 bájtos lett; egy csomplett GIF fájlba találunk benne! A teljesség kedvéért megmutatom, hogy ez az eszköz is finomhangolható:



► A Protocol Coalesce Tool tuningolása

Használtam már a Protocol Coalesce-t XLS, illetve TXT fájlok összeszedésére, bár az előbbi némi utófeldolgozást igényelt, mielőtt az Excel hajlandó lett volna megnyitni. Újabb kihívás: a főnök robbantott tortadiagramban szeretné látni a hálózatunkban használt egyes protokollok arányait...

Protocol Distribution Tool

A feladat nem igazán életszerű, mert a főnökök jobbra nem tudják, mi az, hogy protokoll, de jobb nem jutott eszembe. Ki mást érdekelne a protokollok egymáshoz viszonyított aránya?

Ismét a wk2boot.cap fájl kózzuk, s a szakértő futása után az eredményeket egy hirtelen mozdulattal a vágólappra helyezzük, onnan pedig Excelbe. Mostantól a Kedves Olvasó fantáziájára bízom, hogy vajon hogyan állt elő e csodálatos eredmény (eredetileg színes volt!):



<input type="checkbox"/>	LLC
<input checked="" type="checkbox"/>	SNAP
<input checked="" type="checkbox"/>	SMB
<input type="checkbox"/>	IP
<input checked="" type="checkbox"/>	ARP_RARP
<input checked="" type="checkbox"/>	ICMP
<input type="checkbox"/>	UDP
<input type="checkbox"/>	NBT
<input checked="" type="checkbox"/>	TCP
<input checked="" type="checkbox"/>	DNS
<input type="checkbox"/>	MSRPC
<input type="checkbox"/>	Browser
<input type="checkbox"/>	TRAIL
<input type="checkbox"/>	STATS
<input checked="" type="checkbox"/>	R_LOGON
<input type="checkbox"/>	LDAP

► Hálózatunk protokolljainak megoslása

E megalzó feladat után következzen egy szakmai kihívás: megfigyelhető-e a hálózatban olyan hiba, mely miatti egyes TCP csomagokat újra kell küldenie a feladónak?

TCP Retransmit

Az ismételten elküldött TCP csomagok sok minderről, többek közt bizonytalan hardverekről, illetve túlterhelés miatti csomag elvesztésről árulkoznak, mégis, sokszor rejtve maradnak, mert hiába gúvadunk a Network Monitor felett, az ember összpontosítási képessége véges: ezernyi csomag között garantáltan nem találjuk meg a tökéletesen azonosakat. Márpedig az újraküldés pontosan éről ismerhető fel: egy csomagot bitről bitre, pontosan megismétel a feladó. A feladat gépesítését kiált!

Mi okozhat TCP újraküldést? TCP csomagvesztés! Ha akár egy adatscsmag, akár egy ACK nem érkezik meg a Retransmit időintervallumon belül, akkor újraküldésre kerül. Mitől vezsethet el csomag? Természetesen ennek ezernyi oka lehet, beleértve a múlt rosszullétet is, hisz egyetlen bit hibája miatt a csomag sorsa megpecsételődik: ha elromlik az ellenőrzőösszeg, az útválasztók elhajítják. Ennél kevésbé misztikus a torlóadás esete. Ha van egy 128k-s bérlet vonalunk, s azt a belső 100 Mbit-es hálózatról rendesen meghajítjuk, bizony eldugul. Az útválasztók ilyenkor duguláselhárításképpen eldobják azt a felesleget, amit semmiképpen sem tudnak kituszkolni a lassú vonalon. Továbbá ilyenkor visszaszólnak az eredeti feladónak, hogy lassítson (ICMP Source Quench), más kérdés, hogy ezt figyelembe veszik-e a gépek, vagy sem (általában igen). Szerény véleményem szerint ez az expert éri a legtöbbet a készletből, mert közvetlenül felhasználható hálózati hibák keresésére. Rendszeres futtatásával elejét vehetjük valami nagyobb katasztrófa, tökéletes dugulás kialakulásának. Utolsó expertünk éppen kapóra jön a főnök legújabb agy-amenésének kiszolgálásában: állapítsuk meg, mely felhasználók viszik el a sávszélességet...

Top Users Expert

Ennek a szakértőnek összesen annyit kell mondanunk, hogy az IP cím, vagy a MAC Address használja fel a forgalom szortírozásához, illetve, hogy mekkora top-listát készítsen (az alapértelmezés tíz „fő”). Ezt lefut-tatva valami ilyen eredmény kapunk:

Folytatás a 18. oldalon ►►►



Mint azt már előző számunkban beharangoztuk, a lehetőségekhez képest szeretnénk a magyar informatikusokat „teljes körűen” tájékoztatni a Tech.Ed konferencián elhangzottakról. Ez azonban enyhén reménytelen, tekintettel az ott elhangzott összesen 206 féle, 13 előadótéremben, tíz témakörben párhuzamosan folyó szekcióelőadásra, melyen természetesen egyetlen ember sem képes egyszerre résztvenni – kivéve természetesen Hermionét (*Harry Potter 3. kötet*).



◉ Welcome to Tech.Ed!

Ízelítőként beszámolunk azokról az előadásokról, ahol magunk is résztvettünk, de bizony mi is türelmetlenül várjuk a Post Conference DVD-t, amelyen megtekinthetjük mindazt, amiről lemaradtunk.

Barcelona tökéletesen felkészült több ezres tömegek befogadására; az 1992-es olimpia városa látott már milliós turistaáradatot is. Azt hihetnénk, Spanyolországban vagyunk, de nem: minden helybéli rögtön és erélyesen helyreigazít. Barcelona Katalóniában van (*ami melleleg Spanyolország egyik tartománya, de ezt ne mondd vissza, mert elvernek*). Nem spanyolul beszélnek, hanem katalánul (*ami csak a paritásbitben különbözik a spanyol nyelvtől, de ezt sem szabad nyilvános helyen fejtegetni*). A város fejlettsége egyébként fényévekre áll Budapestétől: légkondicionált tömegközlekedési eszközök (*a buszok is!*), mágneskártyás jegykezelés, flat monitoros kivetítők a metróban, zenélő szökőkutak, pálmákkal szegélyezett ötször öt sávós sugárutak.



◉ A híres zenélő szökőkút a Montjuïc hegy lábánál

E mellett ebben a városban élt és alkotott Antoni Gaudí építész, akinek jellegzetes épületei igencsak rányomják bélyegüket Barcelona látképére. Szubjektív, fényképekkel illusztrált Barcelona-beszámoló az [1] címen található.

A Tech.Ed egyébként nemcsak konferencia, hanem kiállítás, tanfolyam, könyvesbolt, és persze kikapcsolódás a megfáradt informatikusnak. Talán ennek, és a korábbi események pozitív visszhangjának köszönhető, hogy Magyarországról összesen 162 szakember érkezett. Ez olyan hatalmas szám, hogy kénytelen voltam lemondani egy összesített csoportkép készítéséről. Ugyanakkor mi, 162-en egyszerűen elvesztünk az összesen kilencezres tömegben, mely tömeg szinte elveszett a Montjuïc 2 olimpiai csarnokban.



◉ Tech.Ed party az egyik szabadtéri olimpiai létesítményben

A rendezvény hatalmas méretei talán megbocsáthatóvá teszik a legelső híradások pontatlanságait, sőt tévedéseit. A MOM nemhogy hangszúlyos szerepet nem kapott, hanem épenséggel a legkevesebb előadást érdemelte ki (*az i.....hu-n megjelent cikkben a szerző nemcsak a konferencia fő témakörét nem találta el, de még a közlekedésről sem volt helyes fogalma*). Hiába no, a MOM egyáltalán nem nevezhető forradalmi terméknek, míg a .NET igen!



◉ Montjuïc

Folytatás a 19. oldalon ►►►

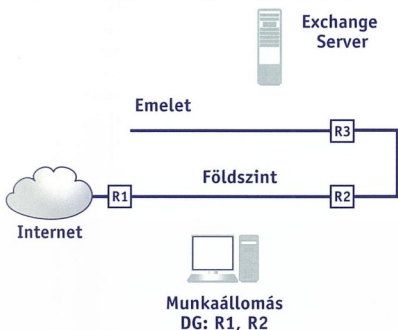
Description	Name	Address	Frames	% Frames	Bytes	% Bytes
Top Packet Destination	Unknown	129.9.64.26	3	0	483	0
Top Packet Destination	Unknown	212.97.8.36	3	0	612	0
Top Packet Destination	Microsoft	FFFFFFFFFFFF	4	0	240	0
Top Packet Destination	Unknown	198.32.1.116	5	0	395	0
Top Source Packet	Unknown	002019A1B9502	2	0	94	0
Top Source Packet	Unknown	00495301180	4	0	240	0
Top Packet Destination	Unknown	172.16.255.255	44	5	6142	2
Top Packet Destination	Unknown	172.16.0.1	288	35	67617	39
Top Source Packet	Unknown	172.16.0.17	343	42	93899	41
Top Packet Destination	Unknown	172.16.0.17	456	56	131338	59
Top Source Packet	Unknown	172.16.0.1	456	56	131338	59

☞ Toplista (vagy feketelista?)

A 172.16.0.17-es gépnél ül kolléga jöhet a munkakönyvéért! :) És most térjünk át a Monitor Control Toolal végezhető realtime elemzésekre.

ICMP Redirect Monitor

Az ICMP Redirect üzenetek alkalmasak arra, hogy munkaállomásokon és más gépeken automatikusan módosuljanak az útvonalatábla-bejegyzések, ha egy új útválasztó bukkan fel a hálózaton, illetve ha egy munkaállomás nem a megfelelő útvonalat választva próbál forgalmazni. Az alábbi ábra egy tipikus paradox helyzetet mutat, ahol egy munkaállomás egyszerre két irányba is szeretne forgalmazni, amihez két Default Gateway kellene beállítani.



☞ Gyakori, ám megoldhatatlan(nak tűnő) útválasztási feladat

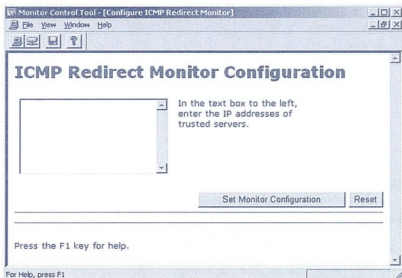
Igen ám, de hiába gépelünk be két DG címet (*R1 és R2*), a Windows csak a legelsőt hajlandó használni mindaddig, amíg az él (úgynevezett *dead gateway detection algorithm*ot használ), és csak akkor áll rá a másodikra, ha az első kimúlt. Ennek következtében két rossz megoldás közül választhatunk. Az első rossz megoldásban az R1, a második rossz megoldásban az R2 a legelső Default Gateway. Hogy ez miért rossz? Mert bárhová is kellene küldeni egy IP csomagot (az emeletre vagy az Internetre), munkaállomásunk mindig a DG-nek küldi, rábízva a továbbítást a routerre. Könnyen belátható, hogy ha egy csomag az emeletre való, de gépünk R1-nek küldi, akkor mandínerből jut el végleges rendeltetési helyére: egyszer megpattan R1-en, mely R2-n és R3-on keresztül ügyesen feldobja az emeletre. Eközben élénk sávzélességparazítás zajlik, ugyanis a csomag kétszer megy át a földszinti

alhlón! Mit lehetne ez ellen tenni? Valamilyen módon megcáskál fel kellene okosítani a munkaállomásokat, hogy kapásból jó irányba induljon el az IP csomag!

☞ Kézzel úgy okosíthatnánk fel, hogy felvennénk a munkaállomás Route táblájába egy bejegyzést arról, hogy merre van az emelet. Ezt kissé macerás megtenni, ha a gépek száma nagy.

☞ DHCP Server esetén számításba jöhetne a 033-as számot viselő Static Route opció, melynek pontosan az a feladata, hogy a sok-sok IP paraméter átadása mellett lehetőség legyen a kliensek Route táblájának automatikus módosítására is. Milyen kár, hogy nem működik! (Not implemented yet.)

☞ R1 pontosan tudja, hogy pazarló módon kerül fel a csomag az emeletre, hisz ő maga pumpálja vissza ugyanabba a subnetbe. R1 minden ilyen esetben ICMP Redirect üzenetet küld az eredeti feladónak, hogy nem jó gatewayt használ. Az okos munkaállomások pedig hálásan köszönik, és ez alapján módosítják Route táblájukat. A következő alkalommal már jó helyre, R2-nek megy az emeletre szánt csomag. A buták nem veszik figyelembe az ICMP Redirectet, s továbbra is „mandínerből” kommunikálnak az emeleti gépekkel. Az NT4 SP2 óta okos, előtte buta volt.

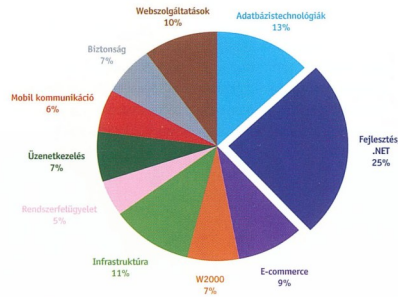


☞ Az ICMP Redirect Monitor beállítása igen egyszerű: fel kell sorolnunk a valódi routereket, minden más címre érkező ilyen üzenet hamis!

Ez a monitor tehát egyfelől arra szolgál, hogy kiderüljön: valamely gépek sávzélességtétközlő módon kommunikálnak a hálón, másfelől azért kell állandóan futnia, mert hibás/hamis routerek üzenetei tökéletesen rombadonthetik egy adott hálózati kommunikációját. Még az is előfordulhat, hogy két alhálózat összes keresztforgalmát magairányítja valaki, például Claudia Sniffer (Így hívjuk egymás közt azt az információbűvárt, aki snifferrel dolgozik) lerakja laptopját a sarokba, és vár. Kis idő elteltével az összes inter-subnet forgalom átfolyik a laptopon, és kiválóan lehet NetMonozni, össze lehet szedni a jelszavakat. Bizony, bizony! A switch nem akadály! (Elnézésért, de ezúttal URL-t nem adnék a jutilithoz, pedig nélkülözhetetlen NetMon segédeszköz ... :-)

Folytatás a 20. oldalon >>>

Ezen a konferencián bizony a fejlesztők és jövőlátók találhatták meg leginkább a számításukat, mert a .NET tisztán elvitte az előadások több, mint egynegyedét, s visszaköszött más (például mobil kommunikáció) témakörökben is. Itt végre megtudhattuk, hogy a .NET nem egyenlő az SQL Server 2000 XML támogatásával. Ez csak a jéghegy csúcsának a csúcsa. A .NET forradalmi változás! Az alábbi torta-diagram a Tech.Ed Conference Guide előadászámai alapján az egyes témakörök súlyát hivatott szemléltetni.



☞ **Arccal a .NET felé!**

Témakörök

Az alábbiakban a tíz témakört sorolom fel, mindegyikhez röviden odaírom az elhangzott előadások főbb mondanóit:

1. Adatbáziskezelés

OLAP, adatbányászat, DTS fejlesztés, adatok elérése XML formában, az Office XP használata OLAP kockák forgatására, replikáció, mentési stratégiák. Őriási tejesítmény elérése: lekérdezések tuningolása, indexelés, futtatás clusteren és Datacenter Serveren. Full text indexelés.

2. Fejlesztés, .NET

.NET framework, ASP.NET, a C# programozási nyelv, alkalmazásfejlesztés mobil eszközökre, a .NET és a COM+ együttműködése, a DLL Hell megszüntetése, verziókövetés, áttérés COM-ról a Common Language Runtime használatára, Windows Forms, ADO.NET, Smart Tague, SOAP, a .NET és a Java összehasonlítása, Design Patterns, VisBas.NET, Garbage Collection (aki sok ismeretlen, zavaros fogalmat talál felsorolásban, ne aggódjon: kitisztítjuk!)



☞ **Diadalív**

3. E-commerce

Varázslatos fejlesztések BizTalk Serverrel, Commerce Serverrel, Content Management Serverrel, IIS5 és IIS6 használata.

4. Windows 2000, áttérés, bevezetés

IIS6 a Whistlerben (*erről már írtunk!*), az Active Directory fejlesztései (*erről is!*), AD tervezés, együttműködés a meglévő DNS infrastruktúrával, Metadirectory, több erőt magában foglaló hálózatok tervezése, az Encrypting File System változásai a Windows XP-ben, Voice Over IP, hálózattervezés Visioval.



5. Infrastruktúra

Sysinternals eszközkészlet, kékhálalelemzés, Internet Explorer 6, többfelhasználós alkalmazásfuttatás Windows XP-n, GDI+ a felhasználói felületen, alkalmazáskompatibilitás XP-n, a 64 bites Windows bemutatása, NLBS, a Windows XP memóriakezelése és feladatütemezője.

6. Rendszerfelügyelet

A MOM használata, scriptelés, MOM Management Packs, AppCenter Server, SMS upgrade és Site Recovery.

7. Üzenetkezelés és csoportmunka

Exchange Server különböző pózokban: Internetszolgáltatónál, clusteren, workflow közepén, BizTalkkal integrálva. Digital Dashboard, Outlook COM Add Ins, Sharepoint Portal Server, Sharepoint Team Services, Exchange Store programozása.



8. Mobil kommunikáció

Mobile Information Server, Windows XP Embedded, drótnélküli hálózatok használata, .NET alkalmazások elvárszolt hardvereken, Microsoft Mobile Explorer böngésző futtatása telefonokon, Compaq iPAQ dicséret, SQL Server for Windows CE, multimédia használata drótnélküli vackokon.

Folytatás a 21. oldalon >>>

Most pedig engedtessek meg nekem, hogy a fent vázolt hackermegoldás alapján levezzem a lokális hálózatok teljeskörű lehallgatásának stratégiáját.

Azért osztom meg e bűnös gondolatokat Kedves Olvasóinkkal, mert nekem már akadt dolgom ilyen méretű összeesküvéssel. Ahol milliárdok utaznak a belső hálón, ott előbb-utóbb megjelennek a tolvajok! Azt mondhatják egyesek, hogy felesleges a Monitor Control Toolal hackerek nyomába eredni, mert a támadásokat a tűzfalak ugyanis megfogják. A külső támadásokat valóban. De az NSA (*National Security Agency*) felmérése szerint az ártó szándékú számítógépes bűnelkövetések elsőprő többsége nem külső, hanem belső támadás!

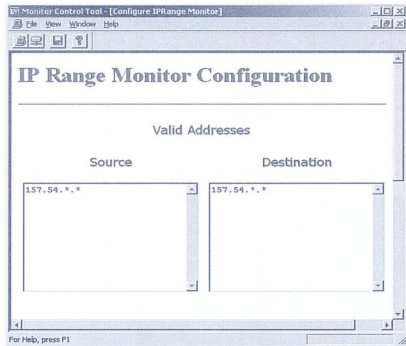
A fentebb emlegetett „segédeszköz” óriási hátránya, hogy egy subneten belül nem jár túl a switch eszén. Miért akadályoz minket egy switch a NetMonozásban? Azért, mert hiába kapcsol a hálókártyánk promiszkuszu üzemmódba (*mely lehetővé teszi idegen csomagok elkapsát is, lásd NetMon szozrat 1. rész*), a várva várt csomagok nem jutnak el hallgatózó gépünkig. Ennek oka az, hogy a switch az eredetileg üzenetszórásos Ethernet csatornát pont-pont kapcsolatokkal váltja ki. Üzenetszórásos esetben (*például vékony koaxra felűzve, vagy UTP-t Hubbal használva*) az összes hálózati forgalom eljut mindenhova, maximum a gépek nem veszik figyelembe, ami nem nekik szól. Ennek a megoldásnak tagadhatatlan előnye az olcsóság, ám minél több gép van a csatornában, annál kevesebb hasznos adat vihető át az ütközések (*collision*) miatt. Ezen segít a switch. Ha két masina egymással kommunikál, akkor a switch egy szempillantás alatt közvetlenül összeköti őket – majd bont. Ennek a megoldásnak óriási előnye, hogy – például – egy tíz megabites hálózat nem fog az üzenetszórás miatt telítődni (*collision*), hanem hébe-hóba akár sokszor tíz megabites (*virtuális*) teljesítményt is produkálhat. Igen ám, de ha nincs üzenetszórás, nem sokra megyünk a jó drága, SMS-béli full-featured Network Monitorral! Ugyanolyan keveset lát, mint ingyenes párja, hisz nem jut el hozzá minden.

Claudia Sniffer azonban nem esik kétségbe. Elgondolkodik: vajon le lehet-e bútitani egy switchet, hogy hűként működjön? Nos, igen. Gondoljunk csak a broadcastokra. Azokat a switch is teríti, nem is tehet mást. Hub módra viselkedik akkor is, ha nem ismeri fel a címzett MAC Addressét. Ez tipikusan a legelső néhány csomagot jelenti egy switch-port életében. Ha tehát valahogy meg tudjuk akadályozni a switchet a MAC Addresssek betanulásában, akkor szórní (*szakszerűbben: floodolni*) fog. Az elgondolás nem teljesen hülyeség, az NLBS például előszeretettel rejtegeti Multicast MAC Addressét a switchek árgus tekintete elől, de ez a megoldás Claudináknak nem fog bejönni: nincs lehetősége az összes oprendszer kódjának módosítására. A switch tehát elvileg lebutítható, de a gyakorlatban ez kivitelezhetetlen. Vagy mi lenne, ha az előző, ICMP Redirecthez hasonló trükkkel egy adott gépen folytatnánk át az összes forgalmat? Ez már menni fog. Csak azt kell biztosítanunk, hogy az ÖSSZES gép külön subneten érezze magát, ugyanígy ebben az esetben egy ROUTEREN keresztül kommunikálnának egymással. Ki kell használnunk, hogy a tipikus vállalati hálózati számítógépek dinamikus IP cím felvételre (*DHCP*) vannak állítva. Ha sikerülne egy olyan DHCP Servert üzembe állítani, mely minden kérelmezőnek más-más IP subnetről ad címet, s ráadásul a Default Gatewayt is ránk állítja, nos ak-

kor már majdnem célbaértünk. Az összes gép a mi „route-rünkön” keresztül forgalmaz, így a switch tehet egy szíveseget :-). Elárulom: van ilyen DHCP Server a „píacon”. Hogy lehet védekezni ennyi gonoszág ellen? Hát Monitor Control Toolal!

IP Range Monitor

A hálózatot kiosztott IP címektől eltérő címek felbukkanása ellen használható oly módon, hogy megadjuk az érvényes, illetve ha van tiltólistánk, az érvénytelen címeket, így a külön subnetre eső címek kiosztását azonnal felülelhetjük.



► **Az IP Range Monitor: érvényes és érvénytelen feladó és célcímek beállításá**

Rogue DHCP and WINS

Hamis DHCP és/vagy WINS észlelésére! Eredetileg olyan problémák észlelésére találták ki, amiket a rendszergazda maga okoz, amikor vizsgálra felkészülés közben DHCP és WINS Servereket telepít fel-le. Ha megjelenik egy idegen címetek osztó DHCP Server a hálózatban, akkor a munkaállomások sem az eredetitől, sem a próbakiszolgálótól nem fog címet feladni, mert egymásnak ellentmondó két IP cím ajánlatból nem tud választani. Emellett sikeresen felhasználható e monitor a bütykölt DHCP Server ellen is. Fel kell venni a megbízható DHCP és WINS Serverek címeit, s a többiit elintézi a Monitor Control Tool.

Security Monitor

Emlékszünk még a Bone protokollra? Ezzel jelzi minden NetMon, hogy éppen futtatják. A Security Monitor Control Toolal megakadályozható, hogy idegen gépről futtasson a NetMon. Ezt megint Claudia Sniffer ellen vethetjük be, de ne feledjük: nem csak Microsoft Network Monitor létezik a világon! A többi snifferprogramra ez a monitor - érthető okokból - hatástalan! NetMon ellen azonban kiváló. Nemcsak hogy észleli az idegen kukkolókat, de bejelük is fojtja a szót (*leállítja a NetMont*), és ha akarjuk barátságatlan üzenetet küld Claudináknak.

Remélem a figyelmes olvasónak feltűnt, hogy a Security, a Rogue DHCP és az IP Range monitorok csak üzenetszórásos csatornán észlelik, amivel megbiztuk őket; switches kör-

Folytatás a 22. oldalon ►►►

9. Biztonság

ISA, PKI, CAPICOM, SSL/TLS, vírusmentesítés Exchange alá, kriptográfiai homályosítás, Secure RAS, az Active Directory PKI integrációja, IPsec, EFS.

10. Webszolgáltatások

UDDI, XML mindenhol, SOAP, SAX, Passport, ASP.NET, WSDL – no és természetesen ez az egész témakör egy az egyben rácsatlakozik a .NET előadások anyagára.

Hát, még így utólag végigfutva is lenyűgöző mennyiség! És ezek csak a témák, némelyikről akár öt-tíz előadás is szót. Nem állíthatjuk, hogy a Microsoftnak ne lenne mondanivalója *(nekünk pedig ne lenne tanulnivalónk!)* Már csak egy dolog hiányzik a boldogságomhoz: egy valóban fejlődő Magyarország, ahol a fejlődés ütemét a vállalatok is követik, és nem a Windows for Workgroupssal bírkóznak.

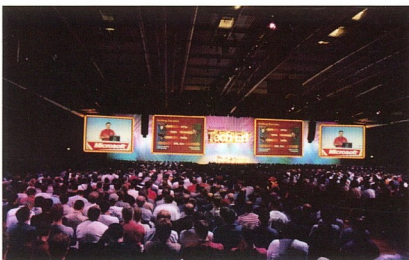
UDDI ide? Ebbe az országba? Minek? Egy kezemen meg tudom számlálni a potenciális webservice fejlesztőket! Fényévekre vagyunk akárcsak a spanyoloktól *(bocsánat, katalánoktól)* is. Hajrá magyarok!



☞ Antoni Gaudí lakóháza a Güell parkban

Előadók

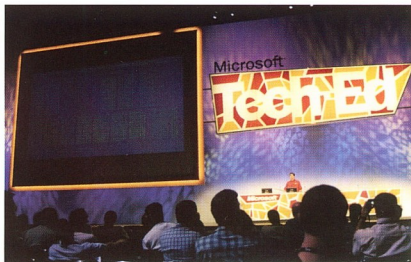
Természetesen a legizgalmasabb témakörökből is lehet poscsék előadást „alkotni”, ha mindenféle jöttment alakokra bizzák a feladatot. Noha valóban előfordulhatott volna poscsék előadás *(206 között már csak a nagy számok törvénye miatt is kellett lennie 1-1 poscséknak)*, ilyenekkel nem találkoztam, sőt, az előadások elsőprő többségét az adott témakör specialistái, és közismert, kiváló előadók tolmácsolásában hallhattuk. Kiemelnék egy-két úberelhetetlen személyiséget:



☞ Anders Hejlsberg vezetett előadása a .NET-ről

Anders Hejlsberg, a C# nyelv atyja. Jól mutatja a konferecia súlypontját, hogy a teljes tömeg, 9000 ember előtt elhangzott vezető előadást egy programozóagyú szakemberre bízta *(figyu, i...hu! Súlypontok!)* A C#-ot megelőzően 1996-tól a Visual J++ és a Windows Foundation Classes projekteken dolgozott. 'S '96 előtt? Nos, alapítotagként Borlandnál tevékenykedett... Az ő nevéhez fűződik a Turbo Pascal, később a Delphi megalkotása.

Mark Russinovich, a Sysinternals eszközkészlet alkotója, az Inside Windows 2000 MS Press könyv társszerzője. Mark nem tett mást, mint a megfelelő időben, a megfelelő helyen, a megfelelő oldalra állítva hackertudását beledebuggolt a Windows NT kernelébe, s ravasz kernelhívásokra, valamint röptében telepített eszközmeghajtókra alapozva létrehozta a FileMon, RegMon és sok más ismert, nélkülözhetetlen monitoreszközt.



☞ Mark Russinovich kékhálál demója

Az egykori harcos eszközkészletét a Microsoft is megvásárolta, és például az Office XP fejlesztésénél, a kompatibilitási teszteknél használták *(mit ír az XP és hova?)* Cspikelődő humorával, összekacintásaival rendkívül élvezetessé tette előadásait!

Előadásai: A Tour of the Sysinternal Tools és Introduction to Windows 2000 Crashdump Analysis



☞ Sagrada Família

Folytatás a 23. oldalon ►►►



nyezetben NEM! Akkor talán mi magunk kényszerülünk a switch becsapására? Szerencsére nem. Okosabb switcheken van egy úgynevezett monitoring port, ahová minden beérkezett csomagot ki lehet küldetni: ide kell feldugni a Monitor Control Toolt futtató gépet. A portok működése továbbra is kapcsolt marad, ám az általunk kiválasztott portok forgalma megjelenik a monitoring porton, a NetMonozók össze ösziinte öröme.

SYN Attack Monitor

A legbonyolultabb, de talán legkevésbé használható eszköz a SYN támadás felismerésére való monitor. Nem az a baj, hogy a SYN Attack Monitor használhatatlan lenne, hanem az, hogy ez a támadásfajta már nem létezik, senki sem használja, hisz ennél lényegesen körmönfontabb denial of service támadások léteznek manapság. Röviden a következőről van szó: amikor egy TCP csatorna felépül, akkor mindkét kommunikáló fél lefoglal 8-8k puffert a memóriában a csatorna gyors kiszolgálása érdekében. A korai TCP implementációk már a kapcsolatfelvétel legelső csomagjának megérkezésekor lefoglalták a puffert, amit ügyes hackerfiúk úgy fordítottak a TCP ellen, hogy kismillió csatornaépítést kezdeményeztek távoli gépeken – ám a háromutas kézfogás (*chip-chip-choka, lásd NetMon sorozat 2. rész*) sohasem jött létre, mert például hamis IP címről eregették a SYN jeleket. A kis buta webkiszolgáló meg addig foglalgatta a 8k-s puffereket, amíg a teljes fizikai memóriát el nem emésztette, s meghalt.

Tekintettel arra, hogy ma már egy hűtőgépben is óvatosabban kódolják a TCP stacket, ez a támadástípus kihalt-nak tekinthető.

Mít nem tud a Monitor Control Tool?

Sajnos a beépített monitorkészlete – érthető okokból – nem naprakész. Nem találunk a gyári készletben például olyan elemzőket, melyekkel tipikus hackertámadások felismerésére nyílna lehetőség: ezek ugyanis általában igen jól felismerhető mintázatot követnek. Az SDK publikus. Ki írja meg nekem azt a Monitor Control Toolt, amely lefűleli az IIS elleni CMD támadásokat?

Fóti Marcell
marcellf@netacademia.net
MCSE, MCT, MCDBA, MZ/X

A cikkben szereplő URL-ek:

[1] <http://technet.netacademia.net/download/netmon>



David Solomon, az Inside Windows NT és 2000 szerzője. Mondanom sem kell talán, hogy a kerneltől mindent tud. Könyvének írásakor nemhogy a Windows forráskódjához hozzáfért (*hisz az sokak számára, különösen egyetemi körökben elérhető*), hanem a fejlesztőkkel is konzultálhatott: mi miért van? Ebből a forrásból táplálkozva lenyűgöző előadást tartott például az operációs rendszer feladatütemezőjéről.

Előadásai: Windows 2000 / XP Process & Thread Scheduling Internals és Windows 2000 / XP Memory Management Internals
Don Box, a Developmentor közösség igazi mentora. Saját bevallása szerint több, mint nyolc évet áldozott életéből a COM, DCOM, COM+ technológiák terjesztésére, s most nesze neked, itt a COM utódja, a .NET. Don Box mindig is kiváló tanítómester volt, de amit a Tech.Ed-en véghezvitt, azt hódolói nem fogják elfelejteni soha: a SOAP-ról tartott előadását stílszerűen egy fürdőkádban ülve tartotta meg!



◉ **Don Box** a SOAP-ról csak kádban ülve tud elmélkedni

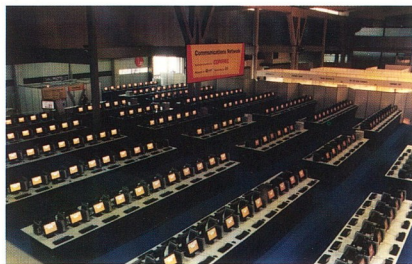
Előadásai: .NET Literacy I. – CLR as a better COM, .NET Literacy II. – SOAP/XML as a better DCOM és Using the SOAP Toolkit
 A fenti előadók „műveiből” külön cikkeket is fabrikálunk, hisz mindegyik egy-egy külön kis világot tárt fel előttünk.

Kiállítók

A nyüzsgő sokaság persze vonzotta a hardver- és szoftverárusokat is, több, mint 200 cég kínálhatta a kiállítócsarnokban portékáját Inteltől IBM-ig, NetIQ-tól Citrixig, RSAtól BindViewig. Gyakorlatilag a teljes informatikai piac képviseltette magát. A szemfellebb konferenciavándorok egész halom marketingvaccal megakodva térhettek haza: póló minden mennyiségben, világító frizbi korlátozott számban állt rendelkezésükre.

CommNet

Internethoz szokott szakembereink nem biztos, hogy túléltek volna egy teljes hetet elszakítva az éltet adó bitektől. Ezt a kockázatot volt hivatva kiiktatni a Communication Network összesen 450, hálózatba kötött munkaállomása, melyekhez oda lehetett sündörögni a szünetekben, és szítpantani egy slukk Interneten.



◉ CommNet

A fenti képen a CommNet még üres: egy nappal korábban ott voltak, hogy ezt a képet elkészíthessem! Később öröm volt látni az egymáson taposó addikt személyeket! Tettelességig azonban sohasem fajultak a dolgok, mert a szervezők gondoltak a teljes függőségben szenvedőkre is (*akik nem bírják ki egy percig sem Internet nélkül*). Mintegy hat-ezer, ingyen „kölcsonözhető” wireless (drótnélküli) hálózati kártya, és Compaq iPAQ kézisámítógép állt rendelkezésre, hogy ha az illető már majd be ... Internetezze, kultúraltan el tudja végezni a dolgát.



◉ Casa Milo

Labok

Párhuzamosan, egyszerre hat teremben (*ügynevezett laborgyakorlatok keretében*), és egy folyosói szekcióban (*Games Area*) lehetett egérrel bibelődni. Volt VisBas for Windows CE, C++ for Windows CE, 64 bites Whistler Server, Visual Studio .NET, BizTalk Server és Analysis Services gyakorlat, amelyek egyikére sem jutottam be/el. Valahányszor rámjött az egértoltagási reflex, ezeket a labokat dugig tömve találtam.

Fóti Marcell
marcellf@netacademia.net

A teljes élménybeszámoló, több mint 200 fényképpel
 [1] <http://barcelona.netacademia.net>



És most következzenek az előadások (27. oldal)

Az ASP-ről jogászszemmel



Az ASP üzletág

Ma már nem szükséges milliárdos nagyságrendű bevételt elérni ahhoz, hogy olyan költséges és komplex alkalmazásokat – mint például a CRM, ERP, e-commerce alkalmazások – bevezethessük. Ezek eddig csak olyan nagyvállalatok számára voltak elérhetőek, akik rendelkeztek az installáláshoz és a bevezetéshez szükséges anyagi és humán erőforrásokkal. Ma már nem kell ezeket megvásárolni, lehetőség van arra, hogy weben keresztül bérelhessük az új üzleti megoldásaink alapját képező szoftvereket – az Application Service Providerok végzik a bérbeadást.

Az ASP (alkalmazás távszolgáltató) üzletág 2000. évben produkált növekedésével bebizonyította, hogy már nemcsak elméleti alternatíváját képezi a hagyományos tulajdonjogon alapuló informatikai rendszereknek. Egyre vonzóbbá válik a vállalkozások számára egy külső alkalmazás-szolgáltató koncepciója, amikor szembesülnek az IT infrastruktúra megszerzésének, valamint a szoftverek, azok fenntartásának és verziófrissítésének valóságos költségeivel.

ASP vs outsourcing

Az ASP (Application Service Provider) egy olyan cég, amely kulcsrakészes alkalmazáscsomagokat (pl.: SAP) szolgáltat, általában az Interneten keresztül, díj ellenében; az alkalmazásokat kezeli, fenntartja, fejleszti; a szerverek és a szoftverek tulajdonjoga őt illeti. Az outsourcingtól elhatárolva a következő különbségek ragadhatóak meg:

- míg az outsourcingnál a hardver és a szoftverlicenc a felhasználó cége, addig az alkalmazás távszolgáltatás esetén ezek az ASP tulajdonát képezik.
 - outsourcing esetében testreszabott megoldásokat kap a felhasználó, az ASP kulcsrakészes, kevésbé változtatható megoldásokat szolgáltat. Ez egyébként az ASP üzleti modell legkritikusabb sajátossága. Egyik nagy problémájának tartják, hogy az egyedi igényeket nem tudja kezelni, mert a kedvező árszínvonalat akkor már nem tudná biztosítani. Ezen kívül az ASP-ek másik nagy dilemmája, hogy ha egyéges szolgáltatásokat nyújtanak, a piaci versenyt csupán az árak fogják meghatározni és nem jelent majd gondot a felhasználóknak, hogy bárkikor másik ASP-t választhatnak, hiszen nagyjából mindenhol ugyanazt kapják.
 - mivel a rendszer üzemeltetéséről az ASP gondoskodik, a felhasználó cég kezében kevés irányítási jogosultság marad pl.: a hibajavítási folyamatokba nem lát és nem is szólhat bele, szemben az outsourcinggal, ahol a kontroll erősebb.
 - az outsourcingnál a szolgáltatáscsomagokat a felhasználó cég állítja össze, ami igen hosszadalmas, fárasztó, komoly erőforrásokat igényel munka, az alkalmazás távszolgáltatásnál ez a feladat az ASP-re hárul.
- A különbséget jól érzékelteti a mindennapi életben oly

gyakran előforduló eset, amikor a szoftverlicenccsökkentés hiánya miatt a céget felelősségre vonják. Alkalmazás távszolgáltatás esetén az ASP felelősségét lehet megállapítani.

ASP modellek

Az alkalmazás távszolgáltatás számtalan formában működhet, alkalmazási szintek alapján három modellt határozhatunk meg:

- I. az egyszerű, kevésbé összetett szolgáltatásokat a weben keresztül érheti el a felhasználó, a fizetés általában átlánydíjas formában történik. (pl.: interneten elérhető jogtár)
- II. a komplexebb szolgáltatások komoly rendszerbevezetést, implementációt igényelnek (pl.: levelező rendszer-nél ügyfélszoftver telepítése, oktatás stb.),
- III. a vállalati profilhoz konfigurálják a szolgáltatásokat (pl.: iktórendszer)

A két utolsó esetben a díjfizetés egyedi számlázási rendszer keretében történik (pl.: mennyiségi alapon)

Az ASP és a felhasználó közötti kapcsolat technikai megoldásai igen sokfélék lehetnek. Bár az Internet volt, ami hozzájárult az alkalmazás távszolgáltatás nagyarányú elterjedésének, ma már VPN (virtual private network), és beépített vonal alapú elérések is jó minőségű alternatívát nyújtanak nagyobb felhasználók részére. Esetlegesen kevésbé költségigényes megoldásként thin clientek alkalmazása is felmerülhet kisebb felhasználók számára.

A díjfizetési módok is igen sokféle alapon valósulhatnak meg. Legáltalánosabb a havi díj megállapítása, amely magában foglalja a hardvert, a szoftverlicencet, a hálózati infrastruktúrát, a szakmai személyzetet, a működtetést, a fenntartást, a verziófrissítést és minden egyéb járulékos költség egy hónapra lebontott összegét.

A számlázás alapja lehet még a felhasználók száma, tranzakciók mennyisége, a tárcapacitás, a felhasználó cég ügyfeleinek száma, a szolgáltatás elérhetőségének ideje; opcionális ártnyezők lehetnek a mentések mennyisége, és a visszaállítás sebessége.

Előnyök és hátrányok az ASP-nél

Elsődlegesen azt szokták hangsúlyozni, hogy az ASP modell az IT kiadásokat 63%-kal csökkentti. Ez elsőre irreálisnak tűnik, de tény, hogy az ASP kulcsrakészes szolgáltatásainak igénybevételével megszorítható a testreszabás költsége, ami a szoftverbevezetési projektek költségvetésének 50%-át teszi ki. Nem kell beruházni a hardvert, nem kell kifizetni a szoftverek licencdíjait, ezenkívül a költségek, kiadások prognosztizálhatóvá válnak. Mivel az ASP a nagyobb beruházási méretek miatt nagyobb kedvezményeket kap, ezért a fajlagos beruházásai kisebbek; emellett több felhasználónak szolgáltató, ezért a költségek is megszokatlan. Így az ASP modellben a felhasználó több év alatt fizeti ki azt az összeget,



amit egyébként a rendszer bevezetése jelentene számára. Ezen felül a bevezetési idő, ami egyébként a szerződéskötéstől számítva kb. fél év, az alkalmazás távszolgáltatás esetében maximum 16 hét.

Az ok amiért a nagyvállalatok vonzódnak tartják ezt a megoldást az, hogy az IT részleg létszáma jelentősen csökkenthető, így az IT szakemberek alkalmazásának összköltsége kevesebb. Kisebbségi cégek pedig olyan profi szakembergárdát tudhatnak maguk mögött, akiket alkalmazottként megfizetni egyébként nem tudnának.

Mivel az ASP kötelezettséget vállal, hogy egyes alkalmazások verzióját rendszeresen frissíti, külön költség nélkül, mindig a legújabb technikai megoldások előnyeit élvezheti a felhasználó. Felmerül a kérdés, hogy a vállalat bizalmas adatait más cég szerverén tárolni mennyire biztonságos, épp ezért különös figyelmet kell fordítani a hozzáférési jogosultságok rendezésére. Ha cégen belül ebben a körben mulasztás történik, a bizalmas adat akkor sem kerül teljesen illetéktelen kezekbe, míg ha ugyanez előfordul alkalmazás távszolgáltatási rendszerben, kivülálló harmadik személy nyerhet betekintést. Azt szokták modani: ha ASP-t alkalmazunk, a biztonságot is outsourcing-oljuk.

Összességében mégis elmondható, az ASP szerverén nagyobb biztonságban vannak az adatok, hiszen az ASP-nek mind hardver-, mind szoftver szinten komoly beruházásokra van lehetősége a már említett költségmegtakarítás miatt. Kevés cég engedhetné meg magának például a 24 órás rendszerfelügyeletet, a jogosulatlan hozzáférést folyamatosan figyelő és jelentő szoftvereket, többszintű háttértároló rendszert, és a katasztrófavédelem érdekében különböző fizikai helyen tárolt adatokat. Tehát az ASP teljes körűen nagyobb biztonságot tud elérni az összes rendszerem tekintetében: hálózat, operációs rendszer, alkalmazás, integráció.

Másrészt az alkalmazás távszolgáltatás a változáskezelést igen megnehezíti. Minden kívánt változtatás igen lassan fog realizálódni, ha egyáltalán megvalósul, viszont a költségek ebben az esetben is tagadhatatlanul kisebbek.

Alkalmazás távszolgáltatás esetén lehetőség van rá, hogy az addigi befektetések, a hagyományos infrastruktúra, ne vesszenek kárba. Sőt a legtöbb ASP külön díj ellenében vállalja a korábbi adatok új alkalmazásba történő migrálását.

Nem szabad elfelejteni, hogy az ASP üzletág piaca még igen fiatal, kevés a tapasztalat, így a kockázati tényezők az általánosan elfogadhatónál magasabbak. Összegezve tehát, az alkalmazás távszolgáltatás nem „A” megoldás, csak egy alternatíva, ami nagyon hatékony és gyümölcsöző lehet, ha megfelelő környezetben alkalmazkodik. Kisebbségi vállalatoknak akkor érdemes ebbe az irányba fordulniuk, ha olyan alkalmazásra van szükségük, amelynek erőforrásigényét egyedül nem képesek finanszírozni (pl. SAP). Ha a nagyobb vállalatok beruházás helyett az alkalmazás távszolgáltatás mellett döntenek, akkor lesznek elégedettek, ha ezzel a döntéssel költséget tudnak megtakarítani, vagy pedig olyan speciális megoldásokra van szükségük, amelyek bevezetésére csak rövid idő áll rendelkezésre, de nem okoz gondot az, hogy a kontroll kiesik a kezükből. (pl. Call Center)

SLA

A hálózaton keresztül nyújtott szolgáltatások drámai növekedése felhívta a figyelmet a SLA-ek (Service Level Agreement)

fontosságára. Ezek olyan szerződések, megállapodások, amelyek az ASP és a felhasználó között jönnek létre, és mérhető paraméterekkel határozzák meg azt a szolgáltatási szintet, amelyet az ASP biztosítani fog. Ilyen paraméterek például:

- havi rendelkezésreállítás
- tervezett leállások ideje
- válaszidők
- felmerülő problémák kijavításának maximális határideje

Egyik jelentős tényező, amely az SLA-ekre irányította a figyelmet, az e-commerce térnyerése. Ezen kívül a SLA számos más területen is bizonyít, így biztosítani tudja a hálózati infrastruktúra és a felhasználó vállalat számára létfontosságú alkalmazások (pl.: ERP) optimális teljesítményét annak ellenére, hogy ezek egy kivülálló cég (pl.: ASP) szerverén találhatóak. Összefoglalva azt mondhatjuk, az SLA megjelenése azért jelentős az IT kapcsolatokban, mert bonyolult szolgáltatások esetében a minőségi mutatókat mérhetővé és kezelhetővé teszi, így azok számonkérhetővé válnak.

Szemléletváltás

Egykor csupán a rendszergazdák törődtek a szolgáltatási szintek garantálásával és menedzsmentjével. Statisztikák alapján elemezték, hogy a hálózat minden egyes eleme megfelelően működik-e. Egyetlenegy szempontot hagytak csak figyelmen kívül: azt, hogy a végső felhasználónak mi a véleménye...

Az IT szakemberek és a végső felhasználók igen eltérően ítélik meg, hogy egy szolgáltatás mitől jó. Ennek oka, hogy mindkét fél másképp határozza meg, mi az elégséges szolgáltatási szint, és melyek a kulcsfontosságú mutatói az adott szolgáltatási szintnek.

Az a szemléletváltás, amely a végső felhasználók elvárásait állította az adott szolgáltatás értékelésénél a középpontba, az SLA-et is megváltoztatta. Korábban az SLA-k a hálózati infrastruktúra teljesítményére összpontosítottak, pl.: az adatsomag az A router-ről 200 ms-on belül jusson el a B-re. Ez is értékes információ, de a mai SLA-k az alkalmazások teljesítményére koncentrálnak. Ma már az a legfontosabb, hogy a végső felhasználónak mi a véleménye, hiszen a hálózati alkalmazások működése (pl. reakcióideje) kihat a dolgozók, így összegezve az egész vállalat teljesítményére.

Az SLA-k jelentősége a felhasználó és az ASP szempontjából 1. Végső felhasználó

Amíg a rendszeradminisztrátor a rendelkezésre álló adatok alapján úgy találja, hogy a rendszer működése kielégítő, addig a végső felhasználók gyakran panaszkodnak: „Ma igen lassú a program...” Ez igen fontos információ, hiszen a felhasználó, aki napi 8 órát ül a gép előtt biztos hogy bármely jelentéssel jobban meg tudja mondani, mikor lassú a hálózat. Tehát az SLA egyik tartalmi eleme olyan monitorozó és jelentéskészítő rendszer meghatározása, amely a végső felhasználók szempontjait helyezi előtérbe. Ma már számtalan monitorozó szoftver található a piacon, és ezek alkalmazása elengedhetetlen.

2. ASP

- Az SLA megkötés előtti tárgyalások segítenek, hogy a vevő tökéletesen tisztában legyen az ASP üzleti modelljével, szolgáltatásaival. Ez csökkenti a későbbi konfliktushelyzetek kialakulását, és ezzel a kapcsolat hosszabb és produktívabb lesz.



- Az SLA biztosítja a vevő lojalitását, kézzelfogható emelkedetetőt ad számukra, hogy mit veszítenek, ha szolgáltatót váltanak.

- Az SLA segít az ASP-nek a cél szem előtt tartásában, vagyis pontosan meghatározza mi az elvárható szolgáltatási szint, amit biztosítani kell, hogy annál se kevesebbet se többet ne szolgáltatson. Ezen kívül könnyebbé teszi az üzleti kockázat menedzsmentjét, mert a jövőre nézve is meghatározza az ASP teendőit. Összefoglalva, egy megfelelően megalkotott SLA egyaránt hasznos az ASP-nek és a végső felhasználónak. Az ASP számára előnyt jelent, hogy a SLA-ben konkretizálhatja az elvárható szolgáltatási szinteket, megelőzve a technikailag vagy anyagilag irreális elvárásokat. Az ügyfél pedig igényeinek megfelelő és garantált szinten kapja a szolgáltatásokat.

Milyen a jó SLA?

Számos ismertetőjele van egy eredményes SLA-nek, nevezetesen hogy a benne foglalt feltételek elérhetőek, megvalósíthatóak és mérhetőek legyenek. Egy SLA négy legfontosabb alkotóeleme a hálózat hozzáférhetőségének, a felhasználó elégedettségének, a hálózat teljesítményének és a szolgáltatások rendelkezésre állásának mutatószámai. Fontos hogy megtaláljuk a megfelelő mérhető paramétereket és négy szempont monitoringrendszerének meghatározásakor. Ezeket a mérőszámokat két területre bonthatjuk le: a hálózati szolgáltatás és az alkalmazásslolgáltatás területeire. A hálózati szolgáltatások magukba foglalják a hozzáférhetőséget, a várakozási időt és az összeköttetés-letévesítési hibákat. Az al-

kalmazások szempontjából két lényeges mérőszám van: a rendelkezésre állás és az alkalmazás válaszideje. (p.: *egy ERP alkalmazásnál (pl. SAP) az átlagos elvárható válaszidő 8-10 s.*) Ma az ASP-ek piacán a siker feltétele a képesség a különböző forrásokból származó információk feldolgozására. Felértékelődött az adatgyűjtés jelentősége. Az SLA monitoringeszközök két szempont alapján gyűjtik és rendszerezik az adatokat: az egyik a real-time adatgyűjtés a másik pedig történeti. A real-time monitoring alkalmas a felmerülő problémák azonnali észlelésére, és lehetőséget ad az azonnali kezelésre, mielőtt az SLA-ben foglalt hibahatárokat átlépnék. A történeti trendanalízis lehetővé teszi az SLA igazolását éppúgy, mint a jövőben problémaként jelentkező tendenciák kimutatását.

Az SLA teljesíthetőségének komoly akadálya, hogy az adott alkalmazás szolgáltatásának mérhető feltételeit igen nehéz maradéktalanul számba venni, így gyakori, hogy a szolgáltatóval szembeni elvárások mércéjéig irreálisn magasra helyezik. A múltban sok SLA hatástalannak bizonyult a mérhetőégi eszközök hiányának köszönhetően. Összefoglalva tehát: a SLA-ben meghatározott monitoring rendszer nem az ujjal egymásra mutogatás eszköze. Azon kívül, hogy mérhető paraméterekkel határozza meg a szolgáltatást, proaktív módon segít a problémák felismerésében, azonosításában, mielőtt azok komollyá válnának, és jelentősen lecsökkenti a problémamegoldás idejét.

*dr. Illés Blanka
Nádas & Mayer Ügyvédi Iroda*



ADA STRA RT

<http://vilagokorseg.hu>





Mi fontos, és mi nem fontos egy négynapos konferencia alkalmával? Találkoztam magyarokkal, akik igencsak fanyalgotak az előző cikkben feltárt előadásarányok miatt. Seholy egy RAS, NTFS, DHCP, Office, vagy SQL nyelv előadás. A .NET előadásokra meg nem mennek be, mert „jól” tudják, hogy a .NET egyenlő az SQL 2000-be beépített XML támogatással. Valóban, a tavalyi slágerek teljes egészében hiányoztak. Aki lerágott csontért jött Barcelonába, az nem találta meg a számítását. Természetesen minden résztvevő maga állította össze a saját előadáslátogatási forgatókönyvét, amely alapján a későbbiekben tájékozódott, ment egyik helyszínről a másikra. Ez a cikk az én forgatókönyvem alapján készült. Megpróbálom a kedves olvasót elmeríteni a Tech.Ed forgatagában.



Akik ott voltak, tudhatják: hétfőn még nem voltak előadások. Nos, földi halandóknak nem is! De nem úgy a Microsoft Certified Trainereknél! A hétfői nap az MCT Day nevet viselte, és a redmondi tankönyvírók, valamint az oktatási üzletág vezetői nyilatkoztak meg számunkra. Egyetlenegy mondanivalójuk volt: a tantermi oktatásnak befellegzett. Arccal az e-learning felé!

Gonoszul arra gondoltam, mint esett atomjaira eddigi számtalan papír- és tanteremments ötletünk, kezdve a MOLI-val (emlékszik rá valaki? *Microsoft Online Institute volt a megboldogult*), befejezve az egy kopasz CD-ből álló tankönyvvel. (*Mastering sorozat. A végén szegényszemre nyomtatásban is megjelent.*)

Nem akartam nyilvánosan megszegényíteni a főfő redmondi oktatásgurukat, így csak Önöknek mondom: az e-learning ezer sebből vérzik.

Első gond: senki nem hajlandó fizetni érte. Második gond: a redmondi e-learning megoldásban a problémákör egyedi kezelését igénylő szoftver helyett egy ötletmentes, sivár balmenüs HTML oldalt mutogattak. Pont, mint az msdn.microsoft.com. De ha pont olyan, mi szükség rá? Visszatérünk az első pont-ra: az msdn ingyenes... Harmadik, és legsúlyosabb gond: már ma is minden tudás forrása fellelhető az Interneten. Aki ez alapján nem tud tanulni, az ez alapján NEM tud tanulni. Hihetetlenül egyszerű érvelés nem? Sokadszor emlegetem, de én a WMI-vel heroikus küzdelmet vívtam, mire enyém lett. Legalább 6 éjszakai óram ráment arra, hogy a logikáját úgyahogy (egyébként mint utólag kiderült, hibásan!) felfogjam (WMI? CIM? WBEM? SWbemObject? SwbemLocator? Associator-soft? Jaj!). Később megkértem TP-t, hogy mintegy két mondatban foglalja össze nekem, így megfelelő magyarázattal, interaktív formában további öt perc kellett, hogy kitisztuljon a látásom. No de nem ragozom ezt a témát tovább, nyomuljon mindenki tetszése szerint az e-learningben – én a magam részéről a tech.net magazin webes megjelenését éppúgy kiegészítő eszköznek tartom, mint holmi e-learning webeket.



Este meglátogattam a zenélő szökökutát a Montjuïc hegy lábánál. Miközben a csodálatos színeket és formákat fotóztam, lelopták az oldalamról a telefonomat, amit később megtaláltam a park egyik padján. Maig nem értem, ha egyszer ellopták, minek tették le? Talán a Katalán hackerek nincsenek a magyar kollégáik szintjén? Minden védelem kikapcsolható, a telkő értékesíthető!



A Tech.Ed első napja. Az első néhány előadás még nem szekciókra bontott, hanem tömegtájékoztató jellegű volt. Ilyenkor szoktak a nagy emberek előadni. Tavaly például maga Bill Gates tartott egy roppant lapos előadást a népek – már nem is emlékszem a mondanivalójára. Talán nem is volt neki. Nem volt dotnetje, csak néhány ősi, bevezetett terméke (*SQL 2000, Exchange 2000*). Ezenkívül a Whistler újdonságaival vakították a népet, de csak szóban, mert Beta még nemigen volt. De idén! A bevezető előadást maga Anders Hejlsberg tartotta, aki a .NET stratégia kulcsfigurája!

Keynote 1.: a .NET stratégia

Aki azt hiszi, hogy a .NET nem más, mint XML mindenfelé: téved. Aki azt hiszi a .NET nem más, mint egy még újabb programozási nyelv: téved. A .NET egy forradalom.

Mi a Windows a DOS-hoz képest? Egy csomó szolgáltatás beemelése az operációs rendszerbe, hogy ne kelljen minden alkalmazásfejlesztőnek egérkezelést, nyomtatási funkciókat, autentikációt, grafikus képernyőkezelést fabrikálni. Volt is nagy fujolás az egérdrájverírók részéről! De ma már el sem tudjuk képzelni az életünket magától működő egerek és nyomtatók nélküli.

Mi a .NET a korábbi világához képest? A .NET koncepció új-ból összegegyűjtötte, melyek azok a szolgáltatások, amelyekre mindenkinek szüksége van, de az operációs rendszer egyelőre nem tartalmaz rá megoldásokat, így mindenki fúr-farag valami tákolmányt. A barkácsolásnak vége!

E kiválóan eltalált marketingzsócoka egyszerre fedi

☞ az operációs rendszer(ek) objektumorientált átfaragását (*.NET framework, mely a későbbiekben nem utólag telepítendő réteg lesz, hanem bekerül az oprendszembe*), hogy a szolgáltatásokat egységesen, és könnyéden el lehessen érni – nem tizenyolc API függvényhívással, hanem egy objektum egysoros megszólításával. A különbség hasonlatos az ODBC és az ADO programozhatóságához. Míg senki emberfia nem képes ODBC-t programozni, addig az ADO hozzá képest gyerekjáték. A fejlesztők megszabadulhatnak az API hívásoktól és a WM_LOPIKULA Windows üzenetek meghívásától különböző ravasz LPARAM és WPARAM paraméterekkel.

☞ az úgynevezett DLL Hell megszüntetését. Végre nem lesz dicséretes cselekedet a Windows rendszerkönyvtáiraiba haji-

- gálni a DLL-jeinket. A verziózás bevezetésével lehetővé válik, hogy egy rendszerben együtt éljen egy DLL újabb, és régebbi változata. Mindenki azt használja, amelyiket akarja
- ☞ a többplatformos (*telefon, hűtőszekrény stb.*) alkalmazás-fejlesztés lehetőségének megteremtését. A Visual Studio.NET-tel fejlesztett alkalmazások ugyanis nem Intel gépi kódra fordulnak le, hanem egy köztes, platformfüggetlen nyelvre (*IL, Intermediate Language*). A futatható gépi kódot Just In Time fordítók (*Jitter*) generálják a szükségesség függvényében
 - ☞ az összes programozási nyelv felemelését (*hajrá Cobol!*) arra a szintre, hogy teljesértékűvé váljon a XXI. század követelményszintjén. Az ígéretek szerint összesen huszonhat nyelven lesz elérhető a .NET Framework
 - ☞ webszolgáltatások gép-gép közötti igénybevételét, egységes szolgáltatáskatalógust (*UDDI*)
 - ☞ a C# nyelvet (*lásd tech.net magazin, 2000 szeptembere*) egyszerű bejelentkezést nemcsak a vállalati hálózatokban, hanem az Interneten is (*Hailstorm*)
 - ☞ kiváló stabilitású alkalmazásokat azon sokmillió programozó kezéből, akiket pl. vājárból és virágkötőből képeztek át egy kéthetes gyorstalpalón, így kizárólag memóriafogyasztó (*memory leak*), félrepointeröző (*Dr. Watson!*) alkalmazásokat lehet tőlük várni
 - ☞ Valóban 24/7 üzemmódban futó alkalmazásokat, pontosan az erőforráslopás megszüntetésével
 - ☞ Komponensszintű jogosultságkezelést. A .NET komponensek ugyanolyan finoman szabályozható jogosultsággal rendelkeznek, mint Jüzer Judit és Marketing Matyi. Vége az ActiveX komponensek bizalmi alapon történő, mindent-vagy-semmit vak futtatásának!

Az új környezet véget vet a hulladék alkalmazások problémájának. Automatikus erőforrásfelszabadító (*garbage collection*) algoritmus a még a halott alkalmazás kezéből is kivészi az esetet, és visszaadja jogos tulajdonosának, a Windowsnak! Erre a napra még egy-két előadás mellé beterveztem egy 64 bites Whistler gyakorlatot, de az érdeklődők hatalmas tömege miatt nem jutottam be a terembe.



A keddi nap estéjén úgynevezett County Drinks állófogadásra vehettünk részt. A magyar csapatot stílszerűen a Microsoft Magyarország látta vendégül egy, a Barcelona fő sétálóutcájából (*La Rambla*) nyíló étteremben. Seafod, pezszió!

És büszkeséggel tölthette el kicsi szivecskénket, hogy a feltörekvő keleteurópai országocskák közül kizárólag a mi Microsoftunk tartotta érdemesnek ezt a gesztust kuncaftjai és partnerei felé. A roppant fejlett csehek a következő népekkel kényserlültek egy tálból cseresznyézi: Belorusszia, Bulgária, Horvátország, Kenya, India, Lengyelország, Románia, Oroszország, Szerbia, Szlovákia, Szlovénia, Dél Afrika, Ukrajna, Jugoszlávia.



Az első előadás ezen a pompás, napfényben fürdő napon az SQL Server futtatása Datacenter Serveren, ezután Mark Russinovich előadásai következtek. A Datacenter előadás több információt nyújtott a hardverről és az operációs rendszerről, mint magáról szegény SQL Serverről, ennek ellenére hasznos

volt. Azt állítja a prospektus, hogy a Datacenter Server maximum 64 gigabájt memóriát képes kezelni. Igen ám, de mind a jelenlegi Windows 2000 operációs rendszer, mind pedig az Intel processzorok 32 bitesek, melyekkel az álmohatár 4 giga! Itt valami nem stimmel, valaki 36 bites! Talán csak nem a processzorok? De bizony! Az Intel egyezmény már megjátszotta ugyanezt a trükköt 16 bites processzoraival. Ha jól emlékszem, a 286-os processzorok címbusza nem 16, hanem 20 bites volt, s ez lehetővé tette, hogy szerencsétlen lapozgatók üzemmóddal (*LIM EMS*) ugyan, de el lehessen érni a többletmemóriát a 16 bites alkalmazások címtartományában. Ugyanez van most is. A Pentium processzorok újabb nemzedékei nem 32, hanem 36 címbittel rendelkeznek, s csakúgy mint régen, most is lapozgatással, ablakozással válik elérhetővé ez a többletmemória az alkalmazások számára. A technológia neve Physical Address Extension, s a Windows 2000 számára úgy válik elérhetővé a többletmemória, ha közöljük vele a BOOT.INI-ben, hogy /PAE. A 286-os hasonlat tökéletesen leírja a valóságot: anno az EMS meghajtó tette lehetővé a memória elérését az alkalmazásokból, manapság az AWE (*Address Window Extension*) API-t kell használniuk az alkalmazásoknak, ha 4 gigánál több memóriára vágnak. Az SQL Server talán az egyik legjobb példa AWE alkalmazásokra; aki ismeri az SQL Server memóriakezelését tudhatja, hogy saját cachealgoritmusával igen jól kihasználja a rendelkezésre álló teret!

A Toor of the Sysinternal Tools

Mark Russinovich első előadása a Sysinternals eszközkészlet ingyenes tagjaival foglalkozott (*lásd 31. oldal, ugyanebben a számban*). Ki ne ismerné az NTFSDOS-t és a hasonló eszközöket? Ami nagyon furcsa számomra, hogy odaengedik a mikrofonhoz azt a személyt, aki visszafajította az NTFS fájlrendszer kódját, és az alapján írt egy DOS alá, CONFIG.SYS-be illeszthető NTFS meghajtót, mellyel az NTFS jogok megkezelésével, megvalósásával mindent le lehet szedni a jól védett lemezeiről (*jó, jó, az EFS ellenél!*). Később azt a büntetett is elkövette, hogy – elunva a javítócsomagok állandó kódváltoztatásait az NTFS.SYS-ben – fogta az eredeti NTFS.SYS-t, és készített neki egy keretrendszerrel, hogy azt higgye, NT-n fut, így már a gyári eredeti meghajtóval hatolhatunk az NTFS-en – Windows 95 alá! (*Meg tudom érteni beűhődését. Gyakorlatilag minden SP tásajított valamikézt az NTFS implementációján, s ő kezdette a munkát előlöl...*)

Elsősorban a monitorozóeszközökről beszélt, kiemelve, hogy olyan funkcióhoz jutunk a RegMon, FileMon párossal, melyet az NT valójában nem is támogat. A FileMon például úgy működik, hogy – az NTFS visszafordításának tapasztalatával felvértezve, és az NT-t becsapva – Mark beilleszt az eredeti NTFS főle egy ál-fájlrendszerrel, és az ezen áthaladó műveleteket képes kijelezni nekünk.

A RegMon demója felhívta figyelmünket arra a szomorú tényre, hogy vannak alkalmazások, melyek állandóan tekerik a registryt. Mivel a registry egy roppant lassú adatbázis, az ilyen alkalmazások jelentős terhet jelentenek a rendszer számára. Van egy szívem csücske eszköze is a srácnak: TDIMon. Ez a jószág olyasmi, mint a Network Monitor, de annak hálózati korlátai nélkül: akár a lokális gépre irányuló „hálózati” fogalmat is elkapja, mert nem az Ethernet kártya szintjén sniffel, hanem a Transport Driver Interface alatt. Innen a név: TDIMon.

Mark Russinovich után egy kis Mark Russinovich következett: bevezetés a kék halál rejtelmeibe. Tekintettel arra, hogy ez ügyben másfél évvel ezelőtt már volt alkalom elemzőcikket írni, erről a témáról átengedem a szót Fülöp Miklósnak.

Solving DLL Hell with the .NET Framework

Ugye mindannyian áltak már tanácsalannal egy telepítőprogram ama ravasz kérdése előtt, hogy talált-e példányt a rendszerben az általa telepíteni kívánt DLL-ből, lecserélje-e? Erre a kérdésre nincs helyes válasz. Kizárólag a csillagok állása alapján lehetne megjósolni, hogy vajon mikor lesz kisebb az okozott kár nagysága: ha igen, vagy ha nemet kattintunk. Ez a becslés sem lehet eléggé jó. A széteső sales alkalmazás működésképtelensége miatt elmaradt haszon hosszú távon sokkal nagyobb kárt jelent, mint hogy a főnök képtelen tortadiagramokat rajzolgatni a lefagyó Excelével. Bár ki tudja...

Az előadó Mike Pelton még jobb példával szolgált. Egyszer írt egy grafikus alkalmazást, melyben az egyik művelet a dokumentációtól eltérően makacsul fejfel lefelé rajzolta ki a megjelenítendő ábrát. Mit tett Mike? Az összes Y koordinátát megszorozta -1 -gyel, és az ábra talpra állt. Később a felhasználót lecserélte egy másik alkalmazással. Mit tesz Isten, az addig jól működő alkalmazás hirtelen fejfel lefelé kezdett rajzolni. Hiába, a fejlődés megállíthatatlan. A DLL fejlesztői egy következő verzióban kijavították azt a hibát, amelyre Mike az egész alkalmazását építette. Szelavi.

A .NET Framework az ilyen típusú kavarodásoknak remélhetőleg egyszerű s mindenkoran elejét veszi, mert a Microsoft végre leszámolt azzal az illúzióval, hogy minden egyes DLL (és COM objektum) a világ összes alkalmazása számára közös. Ott kezdődik a történet, hogy a COM objektumokkal ellentétben a .NET objektumokat nem KELL beregisztrálni sehova. Nincs GUID, nincs registry. Az ilyen komponensek az adott alkalmazások lokális kiegészítői, s bántatlanul átvészelik, ha más alkalmazás ugyanolyan nevű, de különböző verziójú DLL-lel fertőzi a rendszert.

A valóban közös (shared) objektumok természetesen közös helyre kerülnek ugyan, de akárhány verzió élhet párhuzamosan! Miután GUID-t nem használunk többé, valamilyen más módszert kellett kidolgozni az esetlegesen merő véletlenségből azonos nevű DLL-ek kavarodása ellen. A digitális aláírás minden kétséget kizáróan megállapíthatóvá teszi az aláíró személyét: digitális aláírás lesz a DLL-eken!

Microsoft Windows Forms

Mi a szösz már megint? Mit nem tud a VisBas? Minek még egy fejlesztőeszköz?

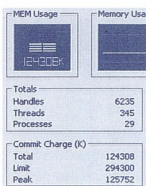
A VisBaszal ezen a szinten semmi baj, a Windows Forms pedig nem fejlesztőeszköz, hanem egy keretrendszer. Sokkal nagyobb baj van például a Visual C++ vizualításával! Aki próbálta, tudja: attól vizuális, hogy mindkét szemünkkel látjuk – de ebben ki is merül ebbéli tulajdonságainak köre. Ha C++-ból szeretnénk dögs felhasználói felületet alkotni, bizony Windows API hívásokkal kell bajlódjunk. A Windows Forms nem más, mint a .NET keretrendszer azon objektumai, melyeket a felhasználói felület kihasználására alkottak meg a Microsoftnál. A Windows Forms teljes egészében nyelv számára elérhetővé teszi a VisBas és Delphi (!) háza táján kifejlesztett roppant gazdag elem-tárházat. Tulajdonképpen itt csíphetjük nyakon a Delphis Anders „felvásárlásának” kézzelfogható hasznát. A Delphi operációs rendszer szintre megy, a komponensek bekötöznek a Windowsba.



Későn száll alá a nap a júliusi Barcelonában, irány Gaudi legnagyobb, lenyűgöző alkalmazása: a Sagrada Família székesegyház! Több mint 100 éve építik, de a kiállított tervek tanúsága szerint eléggé sehol sem tartanak. Áll ugyan négy-négy torony a leendő épület két végében, de a templomnak belseje még nincs. Gaudi műveiről vagy két terecs filmet csatogtattam el, melynek nyomait a weben lehet felfedezni a [1] címen.



Csütörtökön David Solomon előadásait hallgattam. Elsőként a Windows 2000 és XP memóriakezeléséről hallhattunk érdekes tényeket. Tudták például, hogy TaskMan ide, PerfMon oda, gyakorlatilag nem lehet kimérni a szabad memória mennyiségét, mert nem tiszta ez a fogalom? Ha szorul a hurok, és fogyóban a memória, akkor a foglalt tartományt szép lassacskán, LRU algoritmus mentén kicsorgatja a memrelemezre. Az operációs rendszer mindent gyorsítótáraz (kessel), alkalmazásokat, adatokat egyaránt. A csorgatás tehát átmege a cache memórián, így ha hirtelen vissza kell lapozni, van, hogy a blokk még mindig fizikai memóriában van. Most tessenek megmondani: a kesbbe kicsorgatott, halálós ítéletet kapott memóriablokkok, melyek még valójában nem kerültek lemezre, szabadnak, vagy foglaltnak tekinthetők? A TaskMan által mutatott MEM Usage nincs köszönőviszonyban a teljes memóriafelhasználással, ez ugyanis a Commit Charge Total, az összes olyan memória, ami majd valaha kilapozásra kerül! Kizárólag Kernel Debuggerrel kapható meg a fizikai memória teljesen helyes kiosztása. Apró pap lapozás: az előadás végeztével



megkérdteztem a jeles szakembert, vajon a fix lapozófájlme-rebbe vetett babonás hitnek van-e alapja. Nos, íme a mindent és mindenkit megnyugtató válasz (amit egyébként mindig is tudtam és hirdetek)... „It doesn't matter”. A memóriakezelés további rejtelmeiről egy külön cikkben emlékezem majd meg, D. Solomon előadásanyagát ugyanis - későbbi okulás céljából - gondosan letöltöttem magamnak.

Windows 2000/XP Process and Thread Scheduling Internals

Solomon úr mindent tud a Windowsról – és persze elvárható az Inside könyvek szerzőjétől. Leleplező előadásában szomorúan értesülhettünk arról, hogy Scheduler valójában nincs, illetve van ugyan, de már évek óta nem az útezői a feladatokat. Erre mondjuk korábban én is gondoltam, mert a Windows időselelei (kvantumjai) hihetetlenül vastagok: hardverfüggetlen tíz vagy húsz milliszekundumosak, azaz másodpercenként maximum ötvenen juthatnának szóhoz, azok azonban – a processzor órajeléhez viszonyítva – egy örökkévalóságig (négy százezer órajel!) futhatnak. Hát nem így van. A kvantum hossza jó volt tíz évvel ezelőtt, de ma már hihetetlenül hosszú. S hogy akkor mi a csudátér nem porogtít fel a redmondí fiúk? Mert NEM szoftveresen áll be az időhossz, hanem a beépített hardver timer megszakítására csücsülnek rá – az meg ugyan áttállítható, de mivel osztott erőforrás, ezzel más folyamatok borulnának fel. A kvantumtól,

vagyis a hardverkorlátoktól oly módon szakadtak el, hogy az időzítés levált a vekkerről, egy kvantum alatt akár több ezer processz is szóhoz juthat. Ennek az az ára, hogy miután ilyen frekvenciájú megszakítás nincs, nem a Scheduler ütemez, hanem...? Az oprendszer API hívásai!

Úgy képezzük el, hogy amikor egy alkalmazás API-t hív, akkor nemcsak a kért szolgáltatást kapja meg a Windowstól, hanem mellesleg – ha már nála van a gyeplő – Windowsunk belekezd a többi várakozó szál futtatásába. Mire lihegve visszaér hozzánk a kívánt szolgáltatással, és kódunk tovább futhat, addigra ezer másik alkalmazásnak adott esélyt. Vagy más hasonlattal élve: lebukunk API-ért a víz alá, de amíg odalent vagyunk, seregnyi másik alkalmazás bukkan fel levegőt venni (egyprocesszoros rendszerben egyszerre mindig csak egy van a víz színe fölött). Csak a Windowson múlik, hogy nem fülünk meg odalenn: egyszer csak – kezünkben az API hívás eredményével – feljöhethünk levegőt venni.

De mi a helyzet azokkal az alkalmazásokkal, akik valamilyen okból kifolyólag nem API-znak? Hát azok biza átmenetileg megfoghatják a rendszert, de maximum csak egy kvantumnyi ideig, akkor ugyanis eljön értük a Scheduler, és elviszi őket a víz alá, s legközelebb majd jó sokára enged fel, de akkor övé megint egy teljes kvantum, hisz megállapítottuk: nem API-zik. A Scheduler szerepe a mai Windowsban tehát a renitens alkalmazások processzorabálásának kordában tartása. Task Managerrel zárva a magyarázatot: ha egy task Not Responding, az nemcsak annak lehet a jele, hogy lefagyott, hanem annak is, hogy vesztettül dolgozik, nem ér rá a Windowssal kommunikálni.

About Modern Cryptography & Security

Rafal Lukawiczki nem kevesebbre vállalkozott, mint hogy másfél órában elmagyarázza az összes titkosítóalgoritmus működését, hekkelésüket és a megfelelő övintézkedéseket. Nem sikerült neki, nem ért bele az idejébe. Egyenesen James Bond nyomozást ígért – de az RSA algoritmusig nem jutott már el. Előadásából azonban nem hiányoztak a kriptográfia körüli pletykák és botrányok, s élesen kikelt a „Security by Obscurity”, vagy magyarul hókuszpókusz biztonság ellen. Ígazat adok neki. Aki ismeretlen titkosítóalgoritmussal lep meg minket az egészen biztosan szélhámos. Egyszerű gyakoriságelemzéssel visszafejthetők a sarlátának által készített „szupertitkos” algoritmusok. Ami manapság nem fejthető vissza, arról viszont azt kell tudnunk, hogy matematikailag NEM bizonyított a visszafejthetlenségük (például RSA). Ha egy kis indiai matematikus egyszer nekifeszül, és kitalálja, hogy mi módon állítható elő a publikus kulcs alapján a privát, akkor lesz nekünk nemulass! Báj-báj digitális aláírás!



Péntek ismét a .NET napja. A színpadon Don Box, a Developer ura, tucatnyi sikerkönyv szerzője, a COM szakértője, evangélistája. Elsőként a Common Language Runtime-ról beszélt. A CLR egy objektumorientált infrastruktúra, melyben objektum.módotus formában hívhatók

az oprendszer funkciói, nem kell hateraz soros API-nyagga-tó kóddal vacakolni mondjuk egy PKI funkció eléréséhez. A CLR forradalmi, a CLR jó, a CLR átfogó, a CLR a programozók megmentője – ám a CLR mögött továbbra is ott a jó öreg Windows API, amivel nekünk ugyan már nem kell foglalkozni, de jó ha tudjuk: ott van! (Utálom a WM_AKARMI üzenetküldést wPammal és lPammal. Ha egy Windows üzenetben három paramétert kell átadni, akkor is csak kettőt lehet! Vajon miért, ha a C nyelv – melyben írták – megengedi változó számú paraméter átadását? Mert a Windows 1.0 még Pascal nyelven készült, s abban nincs ilyen lehetőség. Később áttértek C-re, de addigra egy csomó dolog készen volt – s mindörökké úgy is marad. Hajrá kompatibilitás!)

SOAP/XML as a better DCOM

Ez az a híres előadás, amit hősünk egy fürdőkádban ülve vitt véghez. A SOAP egyébként egy XML alapú protokoll távoli objektumok megmocantására, metodusainak meghívására, tulajdonságainak átállítására. Egy végtelenül leegyszerűsített, istentelen ponyodalmaítól megfosztott aszinkron, platformfüggetlen RPC. A SOAP üzenet nem más, mint egy text-fájl, mely leírja, hogy a távoli objektum úgy táncoljon, ahogy én füttyölök. Az objektum futhat kenyérpírtón és Linuxon – egyre megy. Az XML formátumú textfájl el tudják olvasni, s hasonló formában válaszolni is tudnak. Hol van már a Windowshoz kötöttség! A Microsoft új világot alkot! Don Box egyébként mindent megengedhet magának: a SOAP Toolkit előadását egyfelől egy darab PPT nélkül tartotta meg, másfelől Macintosh géppel fabrikálta a példa-programjait. Pár éve ez lehetetlen lett volna. De mivel demonstrálható jobban a SOAP keresztplatformossága, mint egy testidegen objektumszerverrel? És ment!

Zárszó

A konferencia önmagában is jó volt. A cikkírás kedvéért végiggondolva viszont olyan részletek váltak számomra világossá, amelyek elsikkadtak volna, ha csak a konferenciadőt fordítottam volna gondolkodásra. Olyan ez, mint amikor sokad-szor nézünk meg egy filmet: újabb és újabb részletek, poénok bukkannak elő. Remélem kis összefoglalómmal sikerül a kiutaz-tó részvevőknek átadnom valamennyit e fílingből. Aki pedig nem volt velünk, jövőre talán elcsábul. Hogy jövőre mi lesz az újdonság? Ezt talán ma még csak Bill tudja...

Fóti Marcell

marcellf@netacademia.net

A cikkben szereplő URL-ek:

[1] <http://barcelona.netacademia.net>



Szerszámosláda – SysInternals különkiadás

A SysInternals céget Mark Russinovich alapította 1996-ban, akkor még NInternals néven. Az alapötletet néhány egyszerű, és mégis nagyszerű eszköz elkészítése volt, ami segítheti a rendszergazdákat a mindennapi munkában. A SysInternals ma több, mint 70 eszközzel, rengeteg technikai információval áll a Windows NT/2000/xp illetve 95/98/Me felhasználók és üzemeltetők rendelkezésére – ingyenesen, és a legtöbb eszköz forráskódjával együtt! Cikkünkben bemutatjuk a legjobb SysInternals eszközöket, akit ennél több érdekel, keresse fel a SysInternals webhelyet a [1] címen. (Némelyik eszköz bővebb funkcionalitású verziója a *Winter-nals* [2] webhelyén vásárolható meg.)

A fájlrendszer-hozzáférés naplózása (FileMon)

A mindenkori kedvenc (és az elsők korai remek) a fájlrendszer elérését naplózó FileMon, illetve a RegMon, ami mindezt a registryvel tudja. Ha valami nem, vagy nem úgy működik, mint ahogy elvárnánk, és fájlrendszer-elérési hibára gyanakodnánk (akár a hozzáférési jogok tekintetében is), próbáljuk ki a FileMon-t! A legújabb verzió a [3] címről tölthető le, és – mint a legtöbb SysInternals eszközt – megtelepíteni sem kell. kattintsunk a filemon.exe-re!

Time	Process	Request	Path	Result
92	notepad.exe	FSCTL_IS_VOLUME_MOUNTED	W:\A	SUCCESS
93	notepad.exe	FASTIO_QUERY_OPEN	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
94	notepad.exe	FSCTL_IS_VOLUME_MOUNTED	W:\A	SUCCESS
95	notepad.exe	FSCTL_IS_VOLUME_MOUNTED	W:\A	SUCCESS
96	notepad.exe	FASTIO_QUERY_STANDARD	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
97	notepad.exe	IRP_MJ_CREATE	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
98	notepad.exe	IRP_MJ_CLEANUP	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
99	notepad.exe	IRP_MJ_CLOSE	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
100	notepad.exe	FSCTL_IS_VOLUME_MOUNTED	W:\A	SUCCESS
101	notepad.exe	FASTIO_QUERY_OPEN	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
102	notepad.exe	FSCTL_IS_VOLUME_MOUNTED	W:\A	SUCCESS
103	notepad.exe	IRP_MJ_CREATE	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
104	notepad.exe	IRP_MJ_CLEANUP	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
105	notepad.exe	IRP_MJ_CLOSE	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
107	notepad.exe	FSCTL_IS_VOLUME_MOUNTED	W:\A	SUCCESS
108	notepad.exe	FASTIO_QUERY_OPEN	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
109	notepad.exe	FSCTL_IS_VOLUME_MOUNTED	W:\A	SUCCESS
110	notepad.exe	FASTIO_QUERY_STANDARD	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
111	notepad.exe	IRP_MJ_CREATE	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS
112	notepad.exe	FASTIO_QUERY_STANDARD	W:\Win2K\PRD\System32\PD\IOCLS...	SUCCESS

☞ A FileMon figyelje a lemezműveleteket

Indítás után azonnal indul a naplózás, amit a nagyító (balról a második) ikonra kattintva kicsit felüggeszthetünk – legalább addig, amíg egy szűrőt be nem állítunk. A szűrő-diálógust a tölcser ikonra kattintva csalogathatjuk elő, és így fest:

Filemon Filter

Enter multiple filter match strings separated by the "*" character. "?" is a wildcard.

Include: notepad.exe;jar.exe

Exclude:

Highlight: jar.exe

Log Reads: Log Writes:

Buttons: Apply, Reset, Cancel

☞ A szűrő segítségével beállíthatjuk, kit (és kit nem) szeretnénk látni.

A szűrőben a processzek neveit (*ha több van*), pontosvesszővel választjuk el (*szóköz ne írjunk a pontosvessző után*). Az Include sorban megjelölt processzek lemezhöz-férései naplózva lesznek, az Exclude sorban nem. A Highlight sorban megjelölt processzek sorait a FileMon kiemeli (*a kiemelés színét az Edit menüben módosíthatjuk*). Ha egy sorra duplán kattintunk, a FileMon megpróbálja a hivatkozott fájlt vagy könyvtárat a Windows Explorerben megnyitni. Érdemes még megjegyezni a radit (*balról a negyedik*) ikont, ami törli a naplót.

A szűrőkhöz még hozzátartozik, hogy a Drives menüben kiválaszthatjuk, hogy mely meghajtók kezelését szeretnénk naplózni. Alapértelmezésben ez az összes fix meghajtót jelenti, de ha akarjuk, bekapcsolhatjuk a cserélhető lemezes egységeket, sőt, named pipe-ok és mail slot-ok naplózását is. A napló egyes oszlopainak jelentése a következők:

- ☞ #: a naplóbejegyzés sorszáma
- ☞ Time: a naplóbejegyzés időpontja
- ☞ Process: a processz neve és azonosítója (PID)
- ☞ Request: a végzett művelet leírása
- ☞ Path: az elérni kívánt fájl/könyvtárobjektum neve
- ☞ Result: a művelet eredménye (*sikeres, sikertelen, hibáüzenet*)
- ☞ Other: egyéb információk

A szerző, Mark Russinovich barcelonai demonstrációját követve próbáljuk ki a FileMon használatát mi is a gyakorlatban! Hozzunk létre egy fájlt, mondjuk teszt.txt néven, majd a fájl biztonsági beállításai közül vegyük el a saját hozzáférési jogainkat! Ha most ezt a fájlt megpróbáljuk megnyitni, a Notepad a következő hibát jelzi:



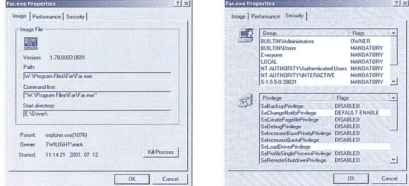
☞ Valaki kivette volna a lemezeket a vicsneszterből!...

Ez nem valószínű. Keressük meg, mi a valódi probléma! Indítsuk el a FileMon-t, állítsuk be a szűrőt a notepad.exe-re, és próbáljuk meg ismét megnyitni a fájlt. A hibáüzenet megjelenése után a nagyító ikon segítségével állítsuk le a további naplózást, ami nekünk kell, az már valahol a naplóban bújik. Az Edit / Find... menü segítségével keressük meg azt a sort, ahol a Notepad megpróbálja megnyitni a teszt.txt fájlt! Az eredmény valószínűleg egy olyan sor lesz, ahol a Result rovatban az „ACCESS DENIED” felirat virít – és máris kiderül a valódi probléma: nincs jogosultságunk a fájlt megnyitásához. Ne ijedjünk meg attól, hogy a Request oszlopban itt IRP_MJ_CREATE szerepel, ez nem a fájl létrehozását jelenti. Az Other oszlopban láthatjuk, hogy a művelet „Open”, azaz fájlt megnyitása lett volna.

A registry-kezelés naplózása (RegMon)

A RegMon [4] a FileMon-hoz nagyon hasonló eszköz. Hasonlóképpen néz ki, hasonlóak a parancsok is, csak a fájlrendszer helyett a registry naplózására használjuk. Ha beállítjuk a szűrőt a notepad.exe-re, és indítunk egyet, láthatjuk, hogy a Notepad csak az elindulásához több, mint 80 registry-műveletet végez (műveletnek számít *egy-egy kulcs megnyitása, adatainak kiolvasása, és a kulcs lezárása*). Láthatjuk például, hogy a Notepad ablak megjelenésének helyett a

létecné tartalmát. Ha egy processzre jobb gombbal kattintunk, a felugró menüben beállíthatjuk a prioritást, illetve – ha akarjuk – leállíthatjuk a processzt (*vigyázzunk! ez a Task Manager-rel ellentétben tényleg leállít mindent!*). Ha kettőt kattintunk a processz nevére, előbújik a tulajdonságlapja:



☞ *Egy processz tulajdonságlapjának két oldala: rajta a verziósám, a processzt futtató felhasználó, illetve annak jogai és privilégiumai*

HKEY_CURRENT_USER\Software\Microsoft\Notepad\1\WindowsPosX

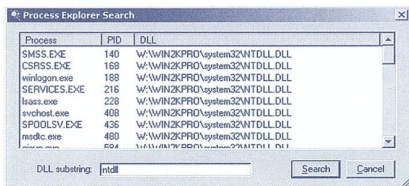
és társai kulcsok értékeiből olvassa ki; de ugyanígy láthatnánk azt is, hogy a gyengébb védelmi szoftverek mit és hogyan próbálnak kiolvasni a registryből, hogy eldöntsék, lejárt-e már a harmincnapos próbaidő ... Az ilyenféle információk felhasználásának módját az Olvasóra bízom. (Kihmm... :-). A RegMon még két funkciójára hívnam fel a figyelmet: először is, egy-egy naplósorra kattintva (vagy az *Edit menüből a Regedit Jump parancsot kiválasztva*) megnyílik egy RegEdit, és az aktuális kulcsra-értékre ugrik. A másik pedig, hogy az operációs rendszer indítását is lehet naplózni, ha az Edit menü Log Boot parancsát választjuk. (Tippelj meg, hány registry-művelet kell például egy Windows 2000 indításhoz, próbáljuk ki, majd lépődjünk meg az eredményen :-).

A FileMon és a RegMon pici kis eszközmeghajtók telepítésével működik, ennél fogva használatukhoz adminisztrátori jogosultságra van szükség. Ha viszont a rendszergazda már futtat egy FileMon-t, illetve RegMon-t, bármelyik felhasználó is képes azok elindítására (a felhasználók által indított példány képes használni a már meglévő eszközmeghajtókat).

A Process Explorer

A régebben HandleX, ma már Process Explorer néven ismert jóság [5] már komoly rendszergazdai eszköz, amelyet a Task Manager folytatásának is nevezhetnénk.

A Process Explorer ablakának alsó részében vagy a kiválasztott processz által birtokolt rendszererőforrások kezelőinek (handle) listáját, vagy pedig az általa betöltött (és ebből következik, hogy „fogott”) .dll-ek és fájlok listáját láthatjuk. Hogy éppen mit, azt a View menü View Handles / View DLL-s menüpontjainak kiválasztásával határozhatjuk meg (ugyanazt teszi az eszköztár harmadik ikonja is). Természetesen minden erőforrás illetve .DLL nevére is kattinthatunk mind jobb, mind bal gombbal, ilyenkor az aktuális információkat láthatjuk. Kíváncsiak vagyunk arra, hogy miért nem tudjuk törölni mondjuk az ntldr.dll-t? Kattintsunk a Search menü Find... menüpontjára!

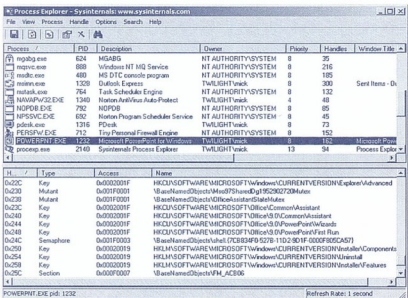


☞ *Ki tartja fogva az NTDLL.DLL-t?*

A Process Explorer minden modern Windows platformon (95-től XP-ig) működik.

Naplózzuk a hálózatkezelést is?

Nem probléma! A FileMon és a RegMon unokatestvére a TDIMon [6], aminek használatát a FileMon és RegMon megismerése után gyerekjáték lesz elsajátítani.



☞ A Process Explorer fő ablaka

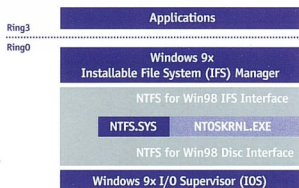
A felső részben láthatjuk a futó processzek nevét, ikonját, leírását (az .exe fájlból szedi ki), a processzt futtató felhasználó nevét, a processz prioritását, az általa kézbén tartott rendszererőforrások számát (Handles), az esetleges ablakfej-



Emellett az NTFSDOS Professional tartalmaz egy másik eszközt is: az NTFSCHK a Windows NT/2000 CHKDSK programjához hasonló, DOS-ból futtatható segédprogram, ami az NTFS partíciók ellenőrzésére használható.

Sajnos ingyenesen az NTFSDOS Professional is csak olvasható üzemmódban érhető el, akinek teljes funkcionális kell, az a Winternals [11] webhelyén megrendelheti.

Nem csak DOS floppyról, de Windows 95/98-ból is elérhető az NTFS partíciók. Az NTFSDOS for Windows 98 [12] hasonló trükköt alkalmaz ehhez mint az NTFSDOS Professional: a Windows 98-on belül olyan környezetet épít fel az NT-ből vett eszközmeghajtónak és társainak, hogy azok azt hiszik, hogy NT környezetben futnak!



☞ *Hála az NTFS for Windows 98-nak, az NT rendszerkomponensek nem tudják, hogy ők tulajdonképpen egy Windows 98-at szolgálnak ki*

Az NTFSDOS for Windows 98 nevével ellentétben minden Windows 95 és Windows 98 verzió működik, és hasonlóan testvéreikhez, az ingyenes változat mindössze az olvasásra képes – de néha ez is több, mint a semmi.

Nézzük a másik oldalt: mit tehetünk akkor, ha NT alól szeretnénk elérni FAT32 partícióit? A SysInternalsnál találunk egy FAT32 eszközmeghajtót (*Fat32 for Windows NT 4.0* [13]), bár az ingyenes változat szokás szerint csak olvasni tud.

Ha már a fájlrendszerek el (*nem*) érése nál tartunk: ha baj van, a HDV-j segítségül egy Windows 2000 telepítő lemezt! A címről telepítés nélkül futtatható a Recovery Console, ami számos hasznos rendszerszükszög (például *chkdsk*) segítségével esélyt ad arra, hogy felélesszük a félélt rendszerünket. Ne higgyük, hogy csak a Windows 2000-ek megmentésére van esélyünk – a Recovery Console tökéletesen működik NT4 esetén is, ha arra előzőleg telepítettük az SP6 javítócsomagot.

Ha a biztonsági beállítások azt megengedik, még fájlokat is másolgathatunk a gépen belül – például NTFS-ről egy könyvtárban elérhető FAT partícióra. Mások viszont sokkal egyszerűbb, ha a halott operációs rendszert tartalmazó merevlemez egyszerűen egy működő Windows NT 4.0 vagy Windows 2000 mellé csatlakoztatjuk.

A PSTools segédprogramcsalád

Ha a [14] címről letöltjük a PSTools segédprogram-csomagot, számos ügyes kis parancssori (*PS* :) eszközhöz jutunk. A psexec segítségével például programot futtathatunk egy távoli gépen (természetesen a megfelelő jogosultságok birtoká-

ban). Ha kell, a futtatandó programot a távoli gépre másolhatjuk (*-c kapcsoló*), persze ha a program a távoli gép részét képezi, erre nincs szükség. A psexec érdekessége, hogy használatához az áldozat gépre semmit sem kell telepítenünk! A psfile nevű eszközzel gyorsan ellenőrizhetjük hogy gépünkön (*vagy akár egy távoli gépen*) ki és milyen célből tart nyitva fájlokat, könyvtárakat.

```

C:\> Command Prompt
Microsoft Windows 2000 (Version 5.00.2195)
(C) Copyright 1985-2000 Microsoft Corp.

M:\> psfile \\tuilight

PsFile v1.01 - local and remote network file lister
Copyright (C) 2001 Mark Russinovich
Sysinternals - www.sysinternals.com

Files opened remotely on tuilight:

[27] M:\JINZHIFRO
User: RICK
Locks: 0
Access: Read

M:\>

```

☞ A psfile listázza a nyitott fájlokat

A psinfo hasznos információkat (verziószámok, processzorok száma, próbaváltozat lejárati ideje, stb.) listáz a helyi vagy egy távoli számítógépről.

A pskill a Resource Kit-ben található hasonló eszköz (segítségével a rendszerből bármelyik folyamat kilőhető, még az is, amelyeket jogosultság hiányában a Task Manager segítségével nem tudunk eltávolítani). A különbség az, hogy a pskill ugyanezt távoli gépen futó processzekkel is képes megtenni! A pslist a processzekkel, végrehajtási szálakkal, azok idő, memória és erőforrás-felhasználásával kapcsolatos kernel-szintű információk listázására való, természetesen távolról is működő, nagyon komoly mutatókat felvonultató kis eszköz. A psloggedon az adott (*helyi vagy távoli*) számítógépre (interaktív vagy hálózati kapcsolat keresztül) bejelentkezett felhasználók neveit listázza ki.

A psloglist a számítógép rendszernaplójának listázására és törlésére használható. A pservice a (*megintcsak helyi vagy távoli számítógépen futó*) rendszerszolgáltatások listázására, elindítására, leállítására, újraindítására való.

```

C:\> Command Prompt
Microsoft Windows 2000 (Version 5.00.2195)
(C) Copyright 1985-2000 Microsoft Corp.

M:\> pservice \\tuilight start u3wsc

PsService v1.01 - local and remote services viewer/controller
Copyright (C) 2001 Mark Russinovich
Sysinternals - www.sysinternals.com

SERVICE_NAME: u3wsc
SERVICE_PATH: World Wide Web Publishing Service
Provides Web connectivity and administration through the Internet Information
Services engine.
TYPE:                : 28 WIN32_SHARE_PROCESS
START_NAME            : 2  SYSTEM_PROCESS
WINDIR_EXIT_CODE     : 0  (0x0)
SERVICE_EXIT_CODE  : 0  (0x0)
CHECKPOINT           : 0x0
WAIT_HINT             : 0x0x0

M:\>

```

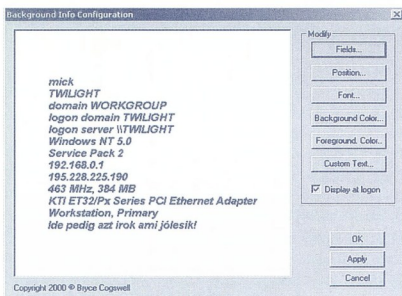
☞ Egy parancs, és már indul is a webkiszolgáló a távoli gépen

A psshtutdown a Resource Kit-ben található shutdown-hoz hasonló funkcionális eszköz: nevéhez híven számítógépek leállítására, újraindítására használható, természetesen távolról is. A psgetsid a számítógép belső biztonsági azonosítójának kiírására szolgál. Aki akarja, megváltoztathatja ezt az azonosítót a [15] címről letölthető NewSID nevű programmal. Ugyanezen a címen olvasható, miért lehet erre szükség.



Ki vagyok én?

A Windows NT-n és Windows 2000-en működő ügyes kis BGInfo [16] automatikus háttérkép generál, amin feltünteti az általunk fontosnak tartott rendszerinformációkat (gépnév, IP címet, verziószámokat, processzortípust, memóriaméretet, satöbbi).



☞ Rendszerinformáció egy pillantásra

Ezt a háttérképet a bejelentkező ablak hátterének állítja be, így a gép bekapcsolása után egy pillantással megbizonyosodhatunk annak legfontosabb beállításairól.

További eszközök

Jócskán van még bemutatnivaló a SysInternals webhelyén, és jócskán marad még mit böngészni cikkünk elolvasása után is. Addig is azonban nézzünk bele a „vegyes cuccok” felíratú fiókba a [17] címen! Miután a gyűjteményt letöltöttük és telepítettük, például a következő eszközökkel lehetünk gazdagabbak:

- ☞ autologon – az automatikus bejelentkezés gyors és egyszerű konfigurációja
- ☞ autoruns – az automatikusan elinduló programok listája
- ☞ efsdump – a Windows 2000-ben titkosított fájlok adatainak kielvasása (a fájlt titkosító és az ún. Recovery Agent tanúsítványa)
- ☞ satöbbi – a gyűjtemény egyre csak bővül

Végül...

A SysInternals kikapcsolódásképpen és a kollégák szívritmusszabályozása érdekében ajánlja Kék Halál Képernyővédjét [18], amely minden bizonnyal a legkedveltebb eszköz mind közül. Ez a hasznos kis eszköz képernyővédőként kedves kékhálál-képernyőket szimulál, teljes gépújraindítással, induláskori chkdsk futtatással (hibákkal! :-)), a rendszer feloldói processzor, memória és eszközmeghajtóadatait felhasználva. Igazi kis gyöngyszem! :) Tessék kipróbálni, jó szórakozást!

Fülöp Miklós

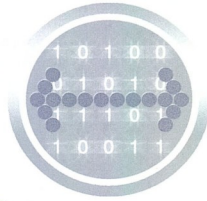
mick@netacademia.net

A cikkben szereplő URL-ek:

- [1] <http://www.sysinternals.com>
- [2] <http://www.winternals.com>
- [3] <http://www.sysinternals.com/ntw2k/source/filemon.shtml>
- [4] <http://www.sysinternals.com/ntw2k/source/regmon.shtml>
- [5] <http://www.sysinternals.com/ntw2k/freeware/procepx.shtml>
- [6] <http://www.sysinternals.com/ntw2k/freeware/tdimon.shtml>
- [7] <http://www.sysinternals.com/ntw2k/freeware/pagedefrag.shtml>
- [8] <http://www.sysinternals.com/ntw2k/freeware/contig.shtml>
- [9] <http://www.sysinternals.com/ntw2k/freeware/NTFSDOS.shtml>
- [10] <http://www.sysinternals.com/ntw2k/freeware/ntfsdospro.shtml>
- [11] <http://www.winternals.com/products/repairandrecovery/ntfsdospro.shtml>
- [12] <http://www.sysinternals.com/ntw2k/freeware/ntfswin98.shtml>
- [13] <http://www.sysinternals.com/ntw2k/freeware/fat32.shtml>
- [14] <http://www.sysinternals.com/ntw2k/freeware/pstools.shtml>
- [15] <http://www.sysinternals.com/ntw2k/source/newsid.shtml>
- [16] <http://www.sysinternals.com/ntw2k/freeware/bginfo.shtml>
- [17] <http://www.sysinternals.com/ntw2k/source/misc.shtml>
- [18] <http://www.sysinternals.com/ntw2k/freeware/bluescreensaver.shtml>



ASP suli (VII.rész)



E hónapban folytatjuk az IIS beépített segédobjektumainak bemutatását. Míg előzőleg az egyszerűbbekkel ismerkedtünk meg, mára a bonyolultabb, sokrétűbb, nagyobb tudású objektumok maradtak. A példaprogramok, mint mindig, az [1] címen található.

A Tools objektum

Kezdjük mindjárt egy olyan segédobjektummal, amelybe néhány, egymástól teljesen független szolgáltatást szűfoltak bele (mint egy svájci biciskába). A `.FileExists()` metódus például egy adott fájl meglétének gyors ellenőrzésére használható (hasonlóképpen, mint a `FileSystemObject` ugyanilyen nevű függvénye) – ld. `tools.asp`:

```
Set oTools = Server.CreateObject("MSWC.Tools")

Response.Write ".FileExists("".ms.jpg"") = " & _
oTools.FileExists("ms.jpg") & "<br>"
```

A különbség annyi, hogy ez a függvény csak relatív fájlnevhivatkozásokat fogad el, azaz segítségével még véletlenül sem lehet kimenni az adott web hatásköréből. Visszatérési értéke 0, illetve False, ha a fájl nem létezik. A `.Random()` függvény egy véletlenszerű értéket ad vissza, ami egy -32768 és 32767 közötti egész szám lesz. Ha csak pozitív számokat szeretnénk, használjuk az abszolútérték (`Abs`) függvényt, ha pedig egy adott tartományba eső számot, akkor a maradékos osztást (`Mod`). Például:

```
oTools.Random()           ' értéke - 32768..32767
Abs(oTools.Random())      ' értéke 0..32768
Abs(oTools.Random()) Mod 100 ' értéke 0..99
```

Ha azt akarjuk, hogy a véletlenszerű szám valóban véletlenszerű legyen (ha már igazán véletlen ügysem lehet), akkor a `Tools` objektumot ne közvetlenül az oldalon, hanem már a `global.asa`-ban hozzuk létre, mert különben a visszaadott véletlenszám nagyjából mindig ugyanaz lesz. Írjuk tehát a következő sort a `global.asa`-ba (de ha megteszük, az előző példa `Set oTools...` kezdetű sorára nincs szükség):

```
<OBJECT RUNAT=Server SCOPE=Application ID=oTools
PROGID="MSWC.Tools"> </OBJECT>
```

A `Tools` objektum harmadik metódusa a `.ProcessForm`, ami egy kitöltött Form adatainak gyors feldolgására használható. A függvény lényege, hogy az elküldött form adatait egy előre elkészített sablonfájlba illeszti, majd az eredményt elemi a kívánt fájlba. A sablon és az eredményfájl lehet tulajdonképpen bármi, az egyszerű szövegfájltól az `.asp` kódig. A `pform.asp` bemutatja a függvény használatát. Miután a kér-

dőv adatai megérkeztek, egyszerűen hívjuk meg a `.ProcessForm` függvényt és adjuk át a sablon és a kimenő fájl nevét:

```
If Request.Form("btnSubmit") <> "" Then
oTools.ProcessForm "pformout.asp",
    "pformtemp.asp"
End If
```

A sablonfájl tartalma a következő (`pformtemp.asp`):

```
<html><head></head>
<body>
<h1>ProcessForm template</h1>
<%
    Response.Write "Ez a kód bentmarad az
    eredményfaljban.<br>"
%>

<% 'Ez viszont vegrehajtotdik!
    Response.Write "A = " &
    Request.Form("valueA") & "<br>"
    Response.Write "B = " &
    Request.Form("valueA") & "<br>"
    Response.Write "C = " &
    Request.Form("valueA") & "<br>"
%>

</body></html>
```

Figyeljük meg, hogy a megszokott `<%` és `%>` kódhatárolók mellett most feltűnt a `<%%` és `%%>` is. A különbség az, hogy a hagyományosan jelölt kódreszletek változatlanul bekerülnek az eredményfájlba, míg az utóbbiakkal jelölték a `.ProcessForm` hívása során kiértékelődnek (magyarul: ezek közé kell írunk az adatok feldolgozásához szükséges kódot). Az eredményfájl ilyen lesz:

```
<html><head></head>
<body>
<h1>ProcessForm template</h1>
<%
    Response.Write "Ez a kód bentmarad az
    eredményfaljban.<br>"
%>
A = értéKA<br>B = értéKA<br>C = értéKA<br>
</body>
```

Az eredményfájlról két dolgot fontos megjegyezni: az első, hogy az aktuális felhasználónak írással kell rá rendelkeznie, különben a `.ProcessForm` feldolgozása hibáüzenettel leáll. A második pedig az, hogy a kimenő fájl egyelőre felülírja az esetleg már meglévő, ugyanolyan nevű fájlt (a `.ProcessForm` har-

madik, nem használ paramétere pontosan ennek az elkerülésére lenne használható, de ez a funkció nincs implementálva). Hogy ezek után ki mire tudja használni ezt a szolgáltatást, az Olvasóra bízom :-)

A Content Linking Component

Sokszor van szükség arra, hogy egymás után következő oldalakat kapcsoljunk össze valamilyen navigációs rendszerbe. A Content Linking Component segít nekünk ebben: egy szöveg-fájlból felolvasott „tartalomjegyzék” alapján képes megmondani, hogy az aktuális után vagy előtt álló oldalnak mi a címe, illetve rövid leírása. Ezzel egyszerűen készíthetünk tartalomjegyzéket és navigációs gombokat, mint ahogy én azt a `clink.asp` példaprogramban tettem. Tessék kipróbálni! :-)

A konfigurációs fájl tartalma az alábbi (`clink.txt`):

```
1.asp -> Egy - megérett a meggy
2.asp -> Kettő - csipkebokor vessző
3.asp -> Három - te vagy az én párom ->
    Megjegyzés: vagy 'te leszel a párom'
4.asp -> Négy - megszépiett a légy
5.asp -> Öt - érik a tök
6.asp -> Hat - hasad a pad
7.asp -> Hét - zsemlet és a pék
8.asp -> Nyolc - üres a polc
9.asp -> Kilenc - kis Ferenc
10.asp -> Tíz - tiszta víz
10b.asp -> Ha nem tiszta, vidd vissza
10c.asp -> A kis cica megjósza.
clink.asp -> Tartalomjegyzék
```

Mint látható, soronként egy oldalt írunk le, minden sor három részből áll, a részeket tabulátor karakterek választják el egymástól (itt nyílal (->) jelöltük). Az első rész az oldal URL-je, ami lehet relatív, vagy abszolút, sőt, akár paramétereket is tartalmazhat, pl.:

```
http://www.google.com/search?q=netacademia
```

A második elem az oldal rövid leírása (*description*), amit egy opcionális harmadik elem, egy megjegyzés követhet, persze tabulátorjal után. A megjegyzést a Content Linking Component nem veszi figyelembe.

Nézzük meg, hogy készül a tartalomjegyzék (`clink.asp`):

```
Set oLinks = Server.CreateObject("MSWC.NextLink")

For i = 1 To oLinks.GetListCount("clink.txt")
    Response.Write "<a href=" &
    oLinks.GetNthURL("clink.txt", i) & "'>"
    Response.Write
    oLinks.GetNthDescription("clink.txt", i)
    Response.Write "</a><br>"
Next
```

Az első sorban az objektum létrehozásának módja látható. Ezt az objektumot nyugodtan létrehozhatjuk a `global.asa`-ban vagy igény szerint, a felhasználás helyén is, mert semmilyen állapotfüggő összetevőt nem tartalmaz (még a konfigurációs fájl nevét

is át kell adni minden egyes hívásnál). A `GetListCount()` metódus visszaadja a konfigurációs fájlban található szabványosan definiált sorok számát. A `GetNthURL()` és a `GetNthDescription()` a konfigurációs fájlban található *n*-edik sor URL-jét illetve leírását adják vissza, paraméterként természetesen a konfigurációs fájl nevét és az indexértéket várják. A konfigurációs fájl neve mindig relatív fájlnev kell, hogy legyen. Ennél több nem is kell a tartalomjegyzékhez.

A navigációs funkciók használatához nézzük meg az egyik „be-ső” oldal kódját (*1.asp-10c.asp-ig mind ugyanaz egyébként*). Először is, kiírjuk az aktuális oldal leírását (a vers aktuális sorát):

```
i = oLinks.GetListIndex("clink.txt")
Response.Write
oLinks.GetNthDescription("clink.txt", i)
```

Ehhez tudnunk kell, hogy az aktuális oldal hányadik sor a konfigurációs fájlban. Ezt az értéket a `GetListIndex()` függvény adja vissza. Ha az aktuális fájlnev nem szerepel a konfigurált oldalak között, a függvény visszatérési értéke 0. Miután ez megvan, a tartalomjegyzékből már ismert módon kiírhatjuk a vessort. Lássuk most a navigációs panelt:

```
Response.Write "<center><a href=" &
oLinks.GetPreviousURL("clink.txt") &
">"
Response.Write "&lt;&lt;&lt; Előző"
Response.Write "</a> | " &
oLinks.GetNthURL("clink.txt", i) & " | "
Response.Write "<a href=" &
oLinks.GetNextURL("clink.txt") & "'>"
Response.Write "Következő &gt;&gt;&gt;</a>"
```

Kezdjük a közepéről: a szokásos módon, az index ismeretében kiírjuk az aktuális oldal URL-jét (`GetNthURL()`). A navigációhoz szükséges függvények pedig (`GetPreviousURL()`, `GetNextURL()`) visszaadják a sorrendben előző, illetve következő oldal URL-jét. Figyeljük meg, hogy ezek a függvények csak a konfigurációs fájl nevét várják, az aktuális oldal indexét nem, ehelyett felismerik, hogy éppen hol tartunk a tartalomjegyzékben. Bár nem használtuk, az előző és következő oldalak leírása is rendelkezésre áll a `GetPreviousDescription()` és a `GetNextDescription()` függvények segítségével.

A Content Rotator komponens

A Content Rotator komponens – nevéhez híven – HTML tartalmat „general”: egy konfigurációs fájl tartalmából választja ki, hogy a sok variáció közül éppen mit ad vissza. A konfigurációs fájl tartalma ehhez hasonló (`controt.txt`):

```
% // Súlyozás nélkül az alapértelmezés 1


% #4 // Kétszer annyi kék, mint piros
<font color="blue">KÉK</font>

% #2
<font color="red">PIROS</font>
```



A fenti listában három variáció látszik. Mindegyiket %% jel előz meg, a jel után opcionálisan a rész súlyozása következhet (# jel, majd egy számérték 1-10000 között). Az egyes variáció megjelenésének valószínűsége függ ettől az értéktől: minél nagyobb, annál többször jelenik meg az azt követő HTML kódresztlet. A %%-al kezdődő sor végére // jelek után megjegyzést is írhatunk.

Ha a konfigurációs fájlt elkészítettük, jöhet a feldolgozás (controt.asp):

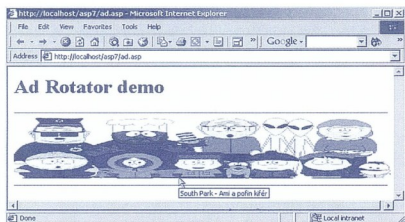
```
Set oContent =
Server.CreateObject("MSWC.ContentRotator")
Response.Write
oContent.ChooseContent("controt.txt")
```

A nagyon egyszerű példában az objektum létrehozásának módja látható (természetesen ezt is létrehozhatjuk már jó előre, a global.asa-ban). A kiválasztott tartalom a ChooseContent() függvény segítségével érhető el, amelynek paraméterként a konfigurációs (relatív) fájlnevet kell átadnunk (ebből következik, hogy ugyanaz a Content Rotator komponens vígan kiszolgál több konfigurációs fájlt is). A komponens másik metódusa a főleg a fejlesztés időszakában használt GetAllContent(), ami a konfigurációs fájlban definiált összes tartalmat visszaszítja. (Ez látható a controt.asp oldal alsó részén):

```
Response.Write
oContent.GetAllContent("controt.txt")
```

Az Ad Rotator komponens

Az Ad Rotator komponens az előző bonyolultabb változata (illetve tulajdonképpen a Content Rotator az Ad Rotator egyszerűsített változata). Az Ad Rotator feladata, hogy az internetről már jól ismert reklámbannereket jelenítsen meg, kezelje és naplózza az átkattintásokat.



Egy hirdetés az Ad Rotator komponens jóvoltából

Kezdjük most is a konfigurációs fájllal. A példaprogramban is használt fájlt (adrot.txt) tartalma a következő:

```
REDIRECT ad_redir.asp
WIDTH 600
HEIGHT 100
BORDER 0
*
banner1.gif
http://www.snoopy.com
Snoopy - Home of Peanuts on the web
20
```

```
banner2.jpg
-
South Park - Ami a pofin kifér
10
banner3.gif
http://www.garfield.com
Club Garfield
20
```

A fájl egy globális konfigurációs fejlécből és a tartalomtól áll, amelyet egy *-ot tartalmazó sor választ el egymástól. A globális konfigurációban a REDIRECT parancs után megadhatjuk, hogy a rekláma való kattintás hova irányítsa a felhasználót (nyilván arra az .asp oldalunkra, ami naplózza a kattintásokat, majd elvégzi az átirányítást az igazi címzethez). Ez az URL lehet relatív és abszolút cím is. A WIDTH és HEIGHT a reklámlablakok méretét állítja be, ha nem adjuk meg, az alapértelmezést használja (440x60 pixel). Végül a BORDER parancs a képek köré rajzolt keret vastagságát jelenti, az alapértelmezés 1.

Ezután az egyes hirdetések következnek, mindegyik hirdetés négy sorból áll:

- ☞ Az első sor a hirdetés fájlneve (URL, ahol a kép található). Lehet relatív és abszolút URL is.
- ☞ A második sor a hirdetés cél URL-je (ide jut majd a felhasználó, ha rákattint a képre, persze csak az átkattintás naplózása után). Ha nem akarjuk megadni, írjunk egy - (minusz) jelet. Ekkor a hirdetés nem lesz kattintható.
- ☞ A harmadik sor a kép alternatív szövege, ez jelenik meg, ha a kurzorral a kép felett időzünk egy picit
- ☞ Végül a negyedik sor a hirdetés súlya. Minél nagyobb ez az érték, annál nagyobb valószínűséggel jelenik meg az adott hirdetés. Ez az érték nem lehet nagyobb 10000-nél.

A hirdetés megjelenítése ezeketán pofonegyszerű (ad.asp):

```
Set oAdRotator =
Server.CreateObject("MSWC.AdRotator")
oAdRotator.Border = 0
oAdRotator.Clickable = True
oAdRotator.TargetFrame = "_top"
Response.Write
oAdRotator.GetAdvertisement("adrot.txt")
```

Az objektum létrehozása után még három paramétert beállíthatunk:

- ☞ A Border paraméter a hirdetések köré rajzolt keret vastagságát határozza meg (minden hirdetésre vonatkozik)
- ☞ A Clickable paraméter azt határozza meg, hogy a hirdetésekre lehet-e kattintani
- ☞ A TargetFrame pedig a cél keret nevét tartalmazza. Itt használhatjuk a keret igazi nevét, vagy a HTML szabályainak megfelelően a _top, _new, _self, _parent, _blank szavakat is.

Végül a GetAdvertisement() függvény visszaad egy hirdetést (kész HTML kódot), a paraméterként kapott konfigurációs fájlból választva (azaz, egy Ad Rotator komponens megintcsak több konfigurációs fájlt is „kiszolgálhat”). A visszakapott kód az előző példában, Garfield esetében például a következő:

```
<A HREF="ad_redir.asp"
```

```

url=http://www.garfield.com&image=banner3.gif"
TARGET="_top"><IMG SRC="banner3.gif"
ALT="Club Garfield" WIDTH=600 HEIGHT=100
BORDER=0</A>

```

Az Ad Rotorator tehát nem más, mint egy ügyes HTML generátor *(jó, vannak sokkal ügyesebbek is, de ez kéznél van)*. Látható, hogy az átkattintás során a kliens az `ad_redirect.asp` oldalra jut, ahol az url paraméterben megkapjuk a kívánt cél URL-t *(ide irányítjuk át a klienst, miután naplóztuk az átkattintást)*, az image paraméterben pedig a kép fájlnevét *(ha véletlenül ugyanarra az URL-re több reklámkép is hivatkozna)*. Az `ad_redirect.asp` tartalma pedig például a következő lehet:

```

<@
  On Error Resume Next

  Set oFSO = Server.CreateObject
    ("Scripting.FileSystemObject")

  Set oFile = oFSO.OpenTextFile
    (Server.MapPath("./adlog.txt"), 8, True)

  oFile.WriteLine Now & " " &
    Request.QueryString("image") & " - " &
    Request.ServerVariables("REMOTE_ADDR") & " - " &
    Request.ServerVariables("HTTP_USER_AGENT")

  oFile.Close

  Response.Redirect Request.QueryString("url")
>

```

Az igazi lényeg az utolsó sorban van: a `Response.Redirect()` metódus segítségével a hívást átirányítjuk az igazi címre, amit az url paraméterként megkaptunk. Előtte azonban még naplózunk a kattintás tényét: a `FileSystemObject` segítségével megnyitjuk az aktuális könyvtárban található `adlog.txt` fájlt, és bejegyezzük a kattintás időpontját, a kép fájlnevét, valamint a kliens IP címét és a böngésző típusát. Figyeljünk rá, hogy ez a naplózás a felhasználó jogosultságaival fut, azaz hibát fog okozni, ha az anonymus felhasználónak nincs írásjoga a naplófájlra. Az esetleges hibákat az `On Error Resume Next` parancsnak köszönhetően átugorjuk, így a hiba tényét a naplózás elmaradása jelenti majd. Éles helyzetben ennél azért egy kicsit bombabiztosabba építsük fel a naplózást *(például használjuk a Counter vagy a PageCounter komponenseket)*!

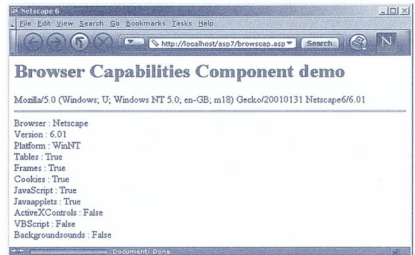
Böngészőtulajdonságok lekérdezése

Örök téma. Milyen jó lenne, ha már a kiszolgáló-oldalon, az `.asp` futtatásakor tudnánk, milyen böngészővel nézik éppen az oldalunkat... Hiszen tudjuk! Minden böngésző illendően bemutatkozik, amit mi bármikor lekérdezhetünk a

```
Request.ServerVariables("HTTP_USER_AGENT")
```

függvény segítségével. Az ezáltal visszaadott szövegből általában lehet következtetni a böngésző típusára és verziójára, illetve néha a kliens gépen futó operációs rendszerre is. Igaz viszont az is, hogy a bemutatkozás a böngészőn múlik, és mindenki úgy

mutatkozik be, ahogy neki jól esik – ha úgy gondolja, hazudik. Mindenesetre, az IIS tartalmaz egy olyan komponenst, ami a `HTTP_USER_AGENT` értékéből következtet a böngésző bizonyos tulajdonságaira. A komponens neve `Browser Capabilities Component`, és nem csinál mást, mint azt, hogy a böngésző azonosító szövegét összehasonlítja egy saját kis adatbázissal. Ez az adatbázis a `browscap.ini` nevet viseli, és egy egyszerű szövegfájl, amit valószínűleg az Olvasó is megtalál a gépén, ha telepítette az IIS-t *(a |WINNT|System32\inetSrv könyvtárban érdemes keresni)*. Miután a böngészők gyakran frissülnek, a `browscap.ini`-t is érdemes gyakran frissíteni, a [2] címen például egy egészen friss példány található. Miután ezt bemásoltuk a megfelelő helyre (felülírva az eredetileg telepített, tavaly februárban aktualizált verziót), már érdemes kipróbálni az IIS-beépített `Browser Capabilities` komponensét *(browscap.asp)*. Ha a komponens nem ismeri fel a böngészőt *(ami azért előfordulhat, nálam például az Opera 5.01-gyel nem bírt)*, akkor a `Browser` jellemző értéke „Unknown” lesz. Ilyenkor nincs más választásunk, mint fogni a `HTTP_USER_AGENT` értéket, abból kitalálni, hogy mégis ki lehet a drót túlsó végén, és kézzel felvenni a `browscap.ini` fájlba.



☞ Az új `browscap.ini` már felismeri a NetScape 6.01-et

Két nagyon fontos hiányossága van ennek a felismerési módszernek: egyrészt, az adatbázis nagyon lassan, de elévül, és kereshetünk, vagy készíthetünk frisseket. A másik pedig, hogy az adatbázisban található adatokból nem derül ki például az, hogy a felhasználó milyen felbontású, hány színű képernyő előtt ül.

Megint a süтик segítenek

Létezik egy szakállas, régi trükk a böngészőtulajdonságok ellenőrzésére: ha a kliensoldalon működik a JavaScript-támogatás és nincsenek letiltva a cookie-k, megtehetjük, hogy egy elküldött *(akár HTML)* oldal a kliensoldalon már rendelkezésre álló adatokkal feltölt egy cookie-t, amit aztán már kiszolgálóoldalon is kiolvashatunk. A cookie előkészítését a `detect.htm` oldal végzi:

```

<HTML>
<HEAD></HEAD>

<SCRIPT language="javascript">
function setcookie()
{
  str = "width=" + screen.width + "&";

```




```
str = str + "height=" + screen.height + "&";
str = str + "colordepth=" +
screen.colorDepth + "&";
str = str + "appName=" + navigator.appName + "&";
str = str + "platform=" + navigator.platform;

document.cookie = "BrowsCap=" + str;
document.location = "detect.asp";
}
</SCRIPT>

<BODY onload="setcookie();">
</BODY>
<HTML>
```

Amint azt talán már észrevette a Kedves Olvasó, most tulajdonképpen nem ASP scriptet írunk, hanem kliensoldalon futó javascript kódot. Az oldal betöltődésekor a <BODY> elem onload paraméterének hatására meghívódik a setcookie() függvény, ami a screen és navigator objektumok segítségével fontos jellemzőkhöz fér hozzá (a képernyő mérete, a színmélység, vagy a böngésző belső neve, verziószáma). Vigyázzunk, hogy most olyan kódot kell írunk, amit minden böngésző támogat, hiszen kliensoldali scriptekről van szó. Éppen ezért – bár a DHTML screen és navigator objektum Internet Exploreren a példánál jóval több jellemzőt tartalmaz – bányunk óvatosan a lekérdezésekkel. A legtisztább az, ha saját kezűleg próbáljuk ki, hogy egy-egy adat rendelkezésre áll-e minden böngészőben. (Az igazi webfejlesztő különben is legalább 3-4 különböző böngészőben ellenőrzi a munkáját!). Miután az adatokat egy szöveges változóba gyűjtöttük, a dokumentumban létrehozunk egy „BrowsCap” nevű cookie-t:

```
document.cookie = "BrowsCap=" + str;
document.location = "detect.asp";
```

Végül a javascript segítségével kliensoldali átirányítást végzünk. A document.location paraméter beállításának hatására a böngésző a detect.asp oldalra ugrik. Ott pedig nincs más dolgunk, mint kiolvasni az adatokat a cookie-ból:

```
Response.Write "Width=" &
Request.Cookies("BrowsCap")("width") & "<br>"
Response.Write "Height=" &
Request.Cookies("BrowsCap")("height") & "<br>"
Response.Write "ColorDepth=" &
Request.Cookies("BrowsCap")("colordepth") &
"<br>"
Response.Write "AppName=" &
Request.Cookies("BrowsCap")("appName") & "<br>"
Response.Write "Platform=" &
Request.Cookies("BrowsCap")("platform")
```

Most nézzük meg a detect2.htm – detect2.asp párost! A cookie-feltöltő html fájl csak annyiban különbözik az előzőtől, hogy a detect2.asp-re irányítja át a klienst. A detect2.asp viszont meglepetést tartogat:

```
Set oBrowsCap =
Server.CreateObject("MSWC.BrowserType")
Response.Write "width=" & oBrowsCap.width & "<br>"
Response.Write "height=" & oBrowsCap.height &
"<br>"
Response.Write "colordepth=" &
oBrowsCap.colordepth & "<br>"
Response.Write "appName=" &
oBrowsCap.appname & "<br>"
Response.Write "platform=" &
oBrowsCap.platform & "<br>"
```

Nem arról volt szó, hogy a Browser Capabilities komponens nem tud ilyen, kliensfüggő információkat (mint például a képernyő mérete) visszaadni? Most pedig – legalábbis ha a kiszolgáló IIS5 – mégis működik!

Bizony, ez az IIS5 egyik új szolgáltatása. Míg a detect.htm-ben bárhol hívhatjuk a cookie-t, amibe az adatokat gyűjtjük, a detect2.htm-ben jelentősege van a névnek. A Browser Capabilities komponens ugyanis ellenőrzi, hogy van-e BrowsCap nevű cookie a kérdésben és ha igen, az adatokat beolvassa belőle! Kellemes meglepetés, ugye?

Zárzó

Ennyit az IIS beépített objektumairól. Legközelebb a Collaboration Data Objects (CDO) objektumcsalád elemeiről lesz szó – levélküldés (sőt, -fogadás) ASP alól! Miután a CDO már nem ASP specifikus objektummodell, a következő rész nyílt nap lesz az ASP suliban: szívesen látunk minden érdeklődőt, akit érdekel a CDO használata. Addig is pihe-nésképpen mindenki olvassa el az előző néhány számban megjelent MIME sorozatot – szüksége lesz rá!

Fülöp Miklós
mick@netacademia.net

A cikkben található URL-ek:

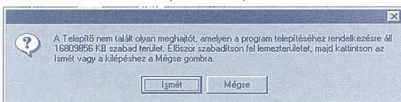
- [1] <http://technet.netacademia.net/feladatok/asp/7>
- [2] <ftp://ftp.g-world.com/browscap.ini>
- [3] <http://msdn.microsoft.com/scripting/windowshost/doc/wsObjWshShell.htm>

A fejlődés megállíthatatlan



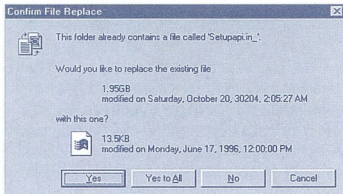
A fejlődés megállíthatatlan

Nem múlik el nap anélkül, hogy az IT ipar vezető cégei valami újdonsággal ne rukkolnának elő. Bizony az örült tempó miatt előfordul, hogy olyan funkciók és korlátlan lehetőségek kerülnek egy-egy szoftverbe, amelyről nem hogy a tanácsadó cégek, de maguk a fejlesztők sem tudnak. A rejtett képességek alaposan meg tudják növelni a szoftverek méretét! Rózsa Gáboréknak például gondot okozott egy hétköznapi szoftver telepítése, mert nem állt rendelkezésre 16 GB szabad terület a lemezen: (30204-ből)



☞ **Előre meg kellett volna bizony tervezni ezt a szoftver telepítést!**

Mint a helyszíni nyomozás megállapította, felelőtlen pazarlás okozta a gondot, hisz merevlemezükön két gigabájtos tömörített .INI fájlokat találtunk:



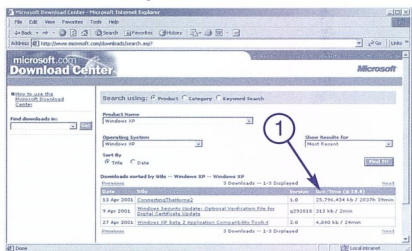
☞ **Inkább a kisebbet kérjük, ha lehet!**

Ezeket a NetAcademia szakemberei lecserélték 13,5 kilobájtos fájlokra, és mindjárt lett hely dögvel! Szakácsi Róbert nem gondolta végig, mire vállalkozik, amikor nekiállt letölteni a Windows XP ConnectionTheHome 1.0 komponenst a Microsofti letöltési oldaláról. Ez a szoftver az alábbi ábra tanúsága szerint 25 gigabájt méretű (az (1) pontról pontosan leolvassa: 25.496.734 kb), amelynek letöltése biza' 2037 óráig és 39 percig tartana, ami közelítőleg



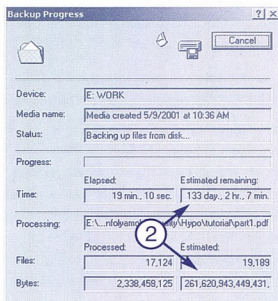
2001. szeptemberben a fenti településeken várható szoftvervazia

85 nap, egy teljes vakáció. A NetAcademia szakemberei lebeszéltek erről a megoldásról.



☞ **Ezt a komponenst amerikai sávszélességeken sem lenne semmi letölteni. Hát még 28.8-as modemmel! Ejnye!**

És végül egy esettanulmány a NetAcademia berkeiből: mi nem fillérezünk, ha hardverről van szó! Kicsit hosszú ideig tart ugyan 261 exabájtos merevlemezünk (2) lementése, de valamit valamiért! A dicsőséges 133 nap...



☞ **Jobb helyeken nem spórolnak a tárolókapacitással...**

Fejtörő

1. Lehet-e az SQL Servert terheléselosztó fűrtben (NLBS) használni?
2. Milyen új lehetőséget kínál a Whistler Server a több erődből álló vállalati címterek használatához?
3. Mi az a FAZAM?
4. Kiláthatnak-e a munkaállomások az Internetre, ha nem telepítjük rájuk az ISA ügyfélpogramját?
5. Mit csinál az operációs rendszer feladatütemezője (Scheduler)?
6. Mire jó a ContentLinking komponens?

.NET testközelben!

A NetAcademia júniustól elindította intenzív .NET felkészítő tanfolyamait, ahol elsőkézből megtanulhatja a legújabb technológiákat.

2124 – Introduction to C# Programming for the Microsoft .NET Platform

5 napos intenzív kurzus, ahol részletekbe menően megismerjük a C# lelkivilágát.

2063 – Introduction to ASP.NET

3 napos „átképzés“ ASP fejlesztőknek, ADO.NET-tel fűszerezve.

1913 – Exchanging and Transforming Data Using XML and XSLT

5 napban az XSLT-ről, alfától-omegáig.



A legjobbakat tanítjuk.

Bővebb információk:

<http://www.netacademia.net>

Az MGH Magyarország Lap- és Könyvkiadó Kft-nél nemcsak a

BYTE
MAGYARORSZÁG -ra

fizethet elő, hanem több mint
50 féle nemzetközi kiadványból is válogathat!

Access-VB-SQL Advisor
A/C Flyer
Architectural Record
Aviation Week
Business & Commercial Aviation
Business Security Advisor
Business Week
Design.Build
Dr. Dobb's Journal
e-BUSINESS ADVISOR
Electrical World
ENR
FileMaker Pro Advisor
FoxPro Advisor
Harvard Business Review
Healthcare Informatics
The Hollywood Reporter

Hospital Practice
Infoconomist
Information Week
Internet Week
Lotus Advisor
MSDN Magazine
Network Computing
Network Magazine
New England Journal of Medicine
Overhaul & Maintenance
Physician & Sportsmedicine
Postgraduate Medicine
Power
tele.com
Unicenter TNG Advisor
Websphere Advisor
World Aviation Directory

Newsletters:

Aviation Newsletters
Energy & Business Newsletters
New England Journal of Medicine N.
Platts Newsletters
UDI Newsletters

Advisor Archival CD's:

Access-VB-SQL Advisor
Lotus Advisor
Business Security Advisor
e-Business Advisor
FoxPro Advisor
FileMaker Pro Advisor

Bővebb felvilágosítás:

Csobán Gyula (csoban@byte.hu)

Telefon: 303-8937, mobiltelefon: 70/315-3979 (üzeneutrőgzítő is)



INGYENES BELÉPŐJEGY A tech.net KLUB MESTERKURZUSRA

Regisztráció: <http://technet.netacademia.net/brainstorm>

A belépőjegy elsődlegességet biztosít a rendezvényen való részvételre, de a korlátozott résztvevői létszám miatt a **regisztráció kötelező!**

Helyszín: **Central European University**
(1106 Budapest Kerepesi út 87.)

Időpont: **2001. szeptember 28.** péntek du 14 óra
Bővebb információt honlapunkon talál, illetve a következő elérhetőségeinken kérhet:

Tel.: 263-2732
Fax: 261-7145
info@netacademia.net



Belépődíj: 5000,-Ft+ÁFA, de **tech.net** előfizetők részére a belépés ingyenes!