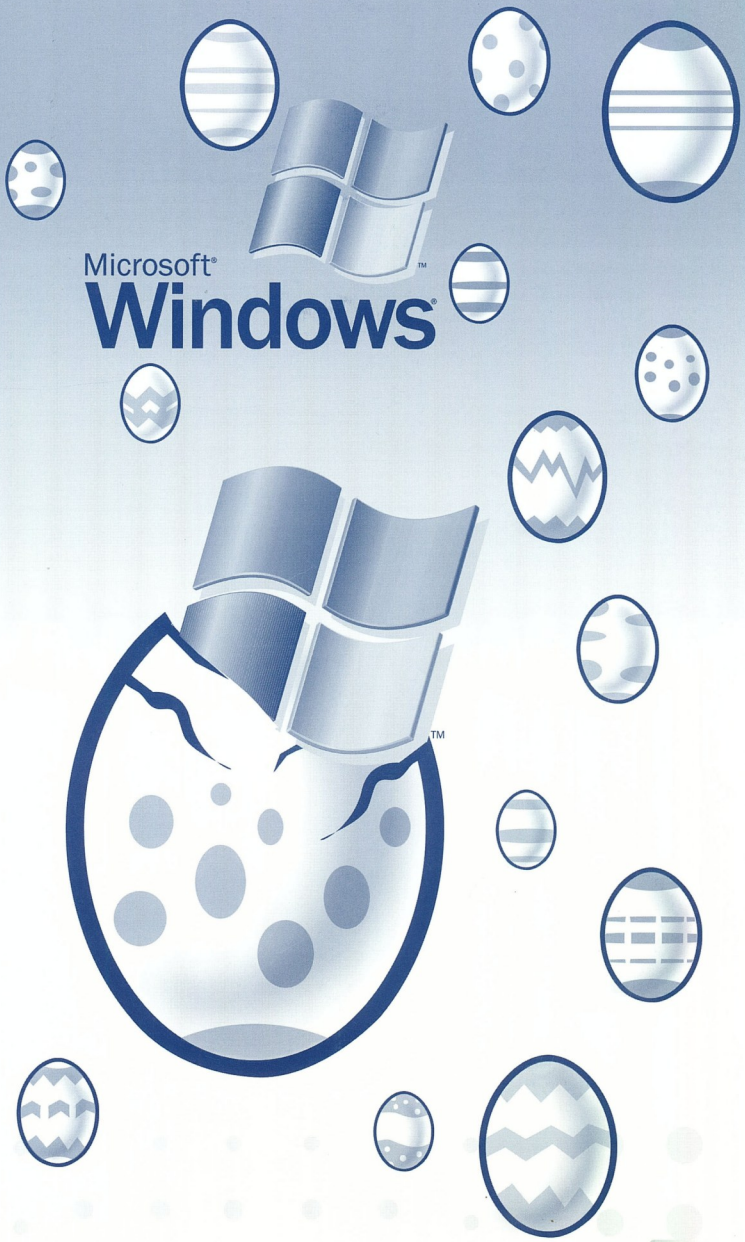
tech.net

working with windows



IV. / 04. szám
2003. április
1349 Ft

ISSN 15865185 04
9 771586 518005

Szerkesztőség:
Főszerkesztő: **Fóti Marcell**
marcellf@netacademia.net
Főszerkesztő-helyettes: **Fülöp Miklós**
mick@netacademia.net
A szerkesztőség címe:
1062 Budapest, Andrássy út 62.
Telefon: 472-1214
technet@netacademia.net
Nyilvános levelezési lista:
tech.net@technetklub.hu

Kiadja és terjeszti a
NetAcademia Kft.
Terjesztési, előfizetési információ:
Telefon: 472-1214
terjesztes@netacademia.net
Megjelenik havonta, ára 1.344 Ft

Előfizethető megrendelőlevélben a szerkesztőségnél:
1062 Budapest, Andrássy út 62.
Fax: 472-1215
http://technet.netacademia.net/subs

Hirdetésfelvétel: **Szívós Éva**
Telefon: 472-1214
Fax: 472-1215
info@netacademia.net

Nyomdai előkészítés:
Ars Luna Bt.
Vezető: Dobák Ildikó
Címlapgrafika: **Molnár Ferenc**

Nyomda:
AduPrint Kiadó és Nyomda Kft.
1061 Budapest,
Paulay Ede utca 55.
Felelős vezető: **Tóth Béláné**

ISSN 1586-5185

NetACADEMIA

A LEGJOBBAKAT TANÍTJUK.

NetAcademia tanfolyamok rendszergazdák számára

Április

XP Windows XP nagyvállalati környezetben *(április 22-23.)*
SETUP Felügyelet nélküli telepítés *(április 18.)*

Május

W2003 Windows 2003 Server Expert Workshop *(május 12-13.)*
NM TCP/IP + Hálózatmonitorozás *(május 14-15.)*
2072 Administering a Microsoft SQL Server 2000 Database *(május 26-30.)*

Június

2150 Designing a Secure Microsoft Windows 2000 Network *(június 2-6.)*
PKI Elektronikus aláírás és titkosítás *(június 16-17.)*

NetAcademia tanfolyamok fejlesztők számára

Április

DP Design Patterns – workshop *(április 28-30.)*
2389 Az ADO.NET programozása *(április 16-18.)*

Május – az XML hónapja

2500 Bevezetés az XML technológiákba .NET környezetben *(május 5-6.)*
2663 XML programozás a .NET Frameworkben *(május 7-9.)*
1905 XML alapú webalkalmazások fejlesztése *(május 12-16.)*

Június

2349 A .NET keretrendszer programozása C# és VB.NET nyelven *(június 2-6.)*
2310 ASP.NET webalkalmazások fejlesztése a Visual Studio .NET segítségével *(június 16-20.)*

Jelentkezés: http://www.netacademia.net

Köszöntő

Kezdők és profik

Amikor hosszas vajúdás után eldőlt, hogy kis informatikai csapatunk egy újabb munkatárssal bővülhet, egy komoly dilemmát kellett megoldanom: hogyan vegyünk fel profi szakembert úgy, hogy a bérkeretünk alapján csak egy kezdőt engedhetünk meg magunknak?

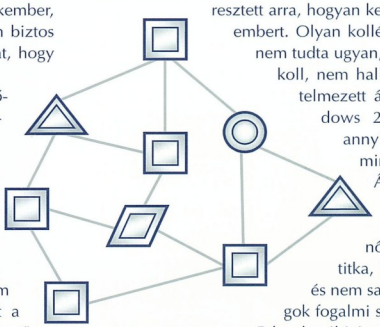
A főnökömmel vívott szópárbaj után úgy gondoltam, nincs sok esély a bérkeret növelésére, és a fából vaskarikát nekem kell elkészítenem. Arra az elhatározásra jutottam, hogy a rövid interjúk során kifürkészem, vajon ott lapul-e a profi szellemiség az egyébként kezdő pályázóban. Az egész szituáció arra ösztökélt, hogy gondoljam végig, vajon mit jelent az, hogy kezdő, és mit, hogy profi. Úgy tűnik, a mindennapi jelentéssel nincs probléma: a profi magabiztos, gyakorlott, és biztos kézzel kezd a feladat megoldásához. A kezdő épp ennek az ellentéte. De vajon mit kezdjen az ember a magabiztos kezdőkkel? És mennyire tekintsem a hosszú gyakorlat ellenére profinak azt, aki alapfogalmakkal sincs tisztában? És egyáltalán: ki meri-e jelenteni akármelyik profi szakember, hogy a rábízott rendszer bármely problémája esetén biztos kézzel dolgozik, és nem merül fel benne a gondolat, hogy ezt a hibát nem lesz képes megoldani?

Néhány nappal később egy nem informatikus ismerősömnek egy feladatot kellett elmagyaráznom, és megmutatni, hogyan kell azt elvégezni a számítógép segítségével. Minden egyes fázisnál görcsösen lelopta a képernyőt, majd kinyomtatta, így több mint negyven képpel rendelkezett, mire végeztünk.

Azon túl, hogy borzasztóan idegesített, rájöttem, ez az ember vélhetően még nagyon sokáig az információs társadalom analfabétája marad, hiába ül naphosszat a monitor előtt. A legfőbb probléma, hogy az ő gondolkodása lineáris: előbb az egyes lépés, azután a kettes lépés, és így tovább. Ezt láthatjuk a bal oldali ábrán. Az ő fejében műveletek kanyarognak, a tanulás számára betanulás, a kivételek kezelhetetlenek, egy új verzió pedig arra kényszeríti, hogy újra elmagyaráztassa magának a nyilvánvalót, új képeket készítsen és nyomtasson. Ez az ember nem veszi észre, hogy valójában nem a feladatot kell betanulnia, hanem azokat a virtuális objektumokat és tárgyakat megértenie, amikkel találkozik: ablak, menü, kérdő mondat *(párbeszédpanel)*, eszköztár, merevlemez, stb. Fogalmi szinten kellene kezelnie a munkáját. Végeredményben a megértés előfeltételét, a gondolkodást kellene elsajátítania. *(Hosszan lehetne értekezni arról, hogy a magyar iskolarendszer vajon mennyire jó, amikor mindenre megtanít minket, csak épp a gondolkodásra nem. Tonnányi információ ömlik az emberek fejébe, és ki, de amikor le kell ülni a PC elé, idegesek és zavartak, mert olyasmit kell csinálni, amit korábban nemigen vártak el tőlük: gondolkodni.)*

Aki gondolkodik, hamar rájön, hogy az informatika világa nem lineáris, hanem hálós szerkezetű, mint ahogy azt a másik ábrán láthatjuk. Ha ismerem az objektumokat, és értem a működésüket, akkor ismeretlen szituációban is eligazodom. Van egy egyszerű és ősi módszer a problémák megoldására: az állandó gondolkodás. Bár sohasem kellett Sun Solaris gép előtt dolgoznom, és kezdetben vélhetően suta lennék, ám az elsajátított fogalmak támogatnának, hamar kiismerném magam, és azt hiszem, nem kellene túl sok idő, hogy a SUN támogatta médiában írogassak a mostanihoz egészen hasonló gondolatokat.

Ennek a nem informatikus ismerősnek a viselkedése ráébresztett arra, hogyan keressem a megfelelő embert. Olyan kollégát vettünk fel, aki nem tudta ugyan, mi az, hogy protokoll, nem hallott még az alapértelmezett átjáróról, és a Windows 2000-ről leginkább annyit tudott, hogy más, mint a Windows 98. Ám csak fél év telt el, és megszerezte az első MCP minősítését. Az volt a titka, hogy gondolkodott, és nem sajnálta az időt a dolgok fogalmi szintű megértésére.



E lap hasábjai gyakran a profikat is próbára teszik, mert bizony új fogalmak tömkelege zúdul ránk. Mégis, nem titkolt vágyunk, hogy a gondolkodásra kész kezdőket is arra ösztökéljük, olvassanak és tanuljanak. Amikor a Windows 2000 közvetlenül a megjelenés előtt állt, Szalontay Zoli és Tarsoly Balázs előadásaiban az egyik dián megjelent néhány fontos szó: nagy és sokkoló. Arra utaltak, hogy rengeteget kell tanulnunk, hogy megértsük, mi és miért változott meg a szoftverben.

Szakmai szempontból kötelességemnek érzem, hogy a profik számára néha túlontúl triviális, ámde a kezdőknek kritikus fogalmakkal is foglalkozzam, hogy a magyar informatikai társadalmat minden szempontból jó szakemberek szolgálhassák. Ajánlom tisztelettel a Lexikon rovatot, a profikat pedig arra kérem, hogy adják a kezdők kezébe az újságot, biztassák őket a tanulásra, de legfőképp az ősi és jól bevált módszerre: a fegyelmezett, logikus gondolkodásra.

Lepenye Tamás
lepenyet@mal.hu

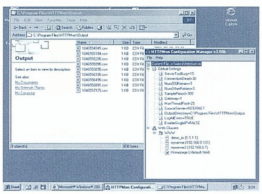
2003. 04. szám

Internet Information Services 6.0, I. rész

A Windows 2003 Server webkiszolgálójának újdonságai

Cikksorozatunkban a Windows Server 2003 IIS szolgáltatásának újdonságait mutatjuk be. Elsőként a legnagyobb átalakuláson átesett, legtöbb újdonsággal kecsegtető és kétségkívül legtöbbet használt szolgáltatás, a webkiszolgáló kerül bonckés alá.

4. oldal



Windows XP automatikus telepítése CD-ről

Aki már telepített operációs rendszert úgy, hogy órákon keresztül nyomkodta a next gombot, ismeri az érzést: milyen jó lenne ha a telepítés automatikusan megtörténne! Elég lenne 40 perc – 1 óra elteltével rápillantani a gépre és ellenőrizni, hogy minden sikerült-e?

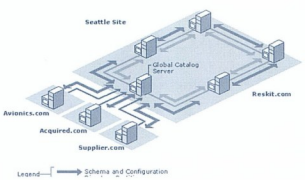
8. oldal

Gyógyszeresdoboz I.

Windows 2000 Server Resource Kit I.

Bár a horizonton már ott tündököl a Windows 2003 Server, azért még jó ideig nem merül feledésbe a Windows 2000 sem. Márpedig ha használjuk, lesznek ügyes-bajos problémáink. Ha problémáink vannak, jól jön a segítség. Például egy dokumentum és segédprogram gyűjtemény formájában a gyártótól? IGEN!

11. oldal



Netacademia Nagylexikon – Címtár

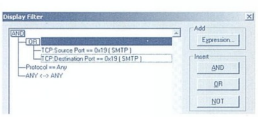
Ha van alkalmazás, amely a rendszergazdáké, akkor az a címtár. Tizenöt évvel ezelőtt még szinte ismeretlen volt ez a fogalom, tíz éve a Novell NDS megrökönyödést keltett újdonságaival, mára ez a technológia minden vállalati vagy intézményi informatikai rendszer alapjává vált. Egy jól konfigurált címtár olyan a rendszergazda számára, mint egy remekül hangolt hangszer. Ám a hozzá nem értés sok keserű órát és hiábavaló fáradozást eredményez.

15. oldal

NetMon a gyakorlatban

Hogyan nyerjünk információt a semmiből?

Nemrégiben érdekes probléma ütötte fel a fejét vállalati levelezésünkben: egy, azaz egyetlenegy cégnek nem mentek el a feladott leveleink. S hogy a dolog még furcsább legyen, ők is csak egyetlenegy helyről nem tudtak levelet fogadni: tőlünk. Hol a hiba? Nálunk? Vagy odaát? Majd a Network Monitor megmondja – ha megmondja!



19. oldal

Ki mivel .....? *Ha támad az IT világ fantomja*

A tervezés szerepe a kivitelezés tudatossá tétele. A tesztlaborokat pedig arra találták ki, hogy minden élesben kockázatos teendőt nyugodt körülmények közt, stresszmentesen végigpróbálhassunk. A feladat egyszerű: térjünk át Windows 2000-re. Rutinfeladat. Ráadásul másfél éves tervezés és real-life tesztelés után az ember biztosra mehet. Nemde? Nem.

21. oldal

Információbiztonság vagy magánszféra?

Security or Privacy?

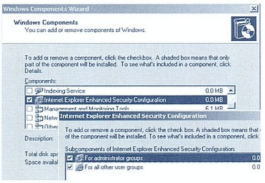
Napjaink egyik legfontosabb kérdésévé az információ biztonsága vált. Privacy or Security? Ez a kérdés az Egyesült Államoktól Európán át Ázsiáig foglalkoztatja az új információs technológiákkal dolgozókat, a felhasználókat, mindazokat, akik információt szolgáltatnak, vagy megszereznek.

23. oldal

Windows 2003: Secure By Default!

Avagy miért nem működik semmi a frissen telepített W2003-on?

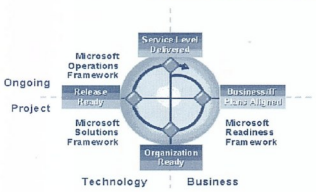
Windows 2003 kísérleteink során rendszeresen a legújabb buildekkel kínlódunk, hogy minél hamarabb észrevegyük, mi minden változik a redmondi fejlesztések során. A változtatások persze nem feltétlenül maradandóak, de az a „trükk", ami látszólag használhatatlanná tette a 3757-es buildet, valószínűleg végleges. Secure By Default!



25. oldal

A Microsoft operációs rendszerek biztonsági tényezői

III. rész – Gyenge pontok, megoldások



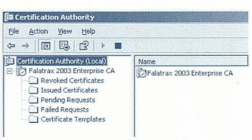
Még felsorolni sem könnyű, hogy hol és mennyi biztonsági szolgáltatást, funkciót halmoztak fel a Microsoft szoftvermérnökei a cég zászlóshajójában. Ezzel együtt nem szabad elhallgatnunk, hogy továbbra is vannak gyenge pontok a rendszerben, amelyek folyamatos törődést igényelnek. Dióhéjban áttekintjük ezeket, valamint a problémák leküzdésének eszközeit is.

27. oldal

Tanúsítványkiadók a Windowsban

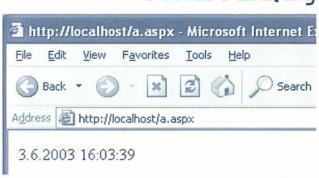
A Windows 2000 és Windows 2003 Server CA szolgáltatása

A tanúsítványkiadó szolgáltatás fontos, központi szerepet játszik a Windows 2000/2003 nyílt kulcsú infrastruktúrájában. Cikkünkben bemutatjuk a Certification Services szolgáltatását, majd természetesen kitérünk a Windows 2003 újdonságaira is.



33. oldal

Portál Para[digma] II



Második rész, avagy mi az a cache?

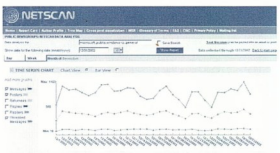
A tartalommenedzsment vagy CMS rendszereknél még vitatkozhatunk, de az igazi alkalmazásintegrációt megvalósító, főleg intranet portáloknál leszögezhetjük, hogy a statikus HTML oldalak előre legenerálása nem lehetséges. Az egyetlen út a teljesen dinamikus markup előállítás. Erről, és ennek optimalizálásáról szól a portál cikksorozat második része.

39. oldal

Netscan – ahol végessé válik a végtelen

Kiberszociológia a fejlesztők és a fogyasztók szolgálatában

A weben drága, nyelvi eszközökkel támogatott kereső-szűrő rendszerek segíthetnek a bennünket érdeklő információ kibányászásában. Egy Microsoft-kutatás most azzal kecsegtet, hogy a Usenet végeláthatatlan káoszából is sikerül automatikusan kiemelni a lényeget.



43. oldal

Internet Information Services 6.0,

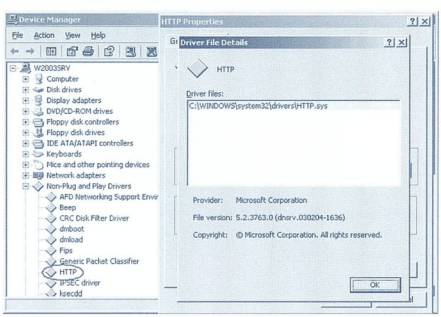
I. rész – A Windows 2003 Server webkiszolgálójának újdonságai

Cikksorozatunkban a Windows Server 2003 IIS szolgáltatásának újdonságait mutatjuk be. Elsőként a legnagyobb átalakuláson átesett, legtöbb újdonsággal kecsegtető és kétségkívül legtöbbet használt szolgáltatás, a webkiszolgáló kerül bonckés alá.

Az új operációs rendszer webkiszolgálója teljesen új alapokra épült. Az új szolgáltatás tervezésénél szemmel láthatólag a sebesség és a megbízhatóság volt a két fő vezérfonal; utóbbi nem is annyira egyszerű feladat, amikor a webkiszolgáló hemzseg az „idegen" kezek által készített dinamikus tartalomtól, bővítményektől.

A HTTP protokoll, mint új „eszköz"

Az IIS6 webkiszolgálójának lelkét az eszközkezelőben találjuk, ha bekapcsoljuk a „nem plug & play eszközök megjelenítése" opciót is:



A HTTP eszközmeghajtó az eszközkezelőben

Bizony-bizony, a webkiszolgáló magjából rendszerszolgáltatás helyett rendszereszköz lett! Egészen pontosan eszközmeghajtóvá, „drájverré" lépett elő. A rendszerszolgáltatások és az eszközmeghajtók közötti legfontosabb különbség, hogy míg a szolgáltatások az operációs rendszer felhasználói *(user)* módjában futnak, addig az eszközmeghajtók a kiemelt, kernel módban működnek. Az eszközmeghajtók *(vegyük például egy videókártya drájverét)* kernel módban képesek a lehető leggyorsabban hozzáférni az általuk kezelt hardverhez; nyilvánvaló tehát, hogy a http.sys is itt tudja leadni a legnagyobb teljesítményt. Természetesen nem a komplett webkiszolgáló került „le" kernelmódba, a http.sys feladata a beérkező HTTP kérések fogadása, azok sorbaállítása, majd „felküldése" a megfelelő kiszolgálókomponens felé. A *(felhasználói módban futó)* webkiszolgáló ekkor feldolgozza a kérdést, és a választ visszaadja a http.sys-nek, aki továbbítja azt a felhasználónak. A http.sys emellett tartalmaz egy beépített gyorsítótárat is, így bizonyos esetekben a beérkező HTTP kérést képes kiszolgálni anélkül, hogy arról a „webkiszolgáló" maga tudomást szerezne. *(Lásd ehavi Portál Para(digma) cikkünket.)*

A kernelmódban működő eszközmeghajtókra gondolva a gyakorlott Windows felhasználónak mindjárt a stabilitás jut az eszébe: bárki fel tud idézni legalább egy tucatnyi kék halált, amit egy-egy rosszul megírt nyomtató-, hálókártya-, vagy videómeghajtó okozott. Van-e félnivalónk az IIS6-tal kapcsolatban? Az egyébként 300kB körüli mérettel bíró http.sys nem tartalmaz „külső" kódot, a kéréseket feldolgozó és a webszolgáltatás tartalmát *(a válaszokat)* generáló részek továbbra is felhasználói módban futnak *(és halnak meg)*. Ráadásul ezen a téren is történt számos előrelépés, úgyhogy van remény! A http.sys kicsit részletesebb működésére néhány bekezdés múlva, pár új fogalom megismerése után még visszatérünk.

A Web Administration Service *(WAS)*

A webkiszolgáló második fő komponense a Web Administration Service *(WAS)*. Ez a komponens már a W3SVC rendszerszolgáltatás égisze alatt fut *(tehát külön processzként nem fogjuk megtalálni, az inetinfo.exe része)*, feladata kettős:

- Betölti és kezeli az IIS konfigurációs adatbázisát *(metabase.xml)*.
- Felügyeli a webkiszolgáló munkafolyamatainak működését.

A WAS tehát konfigurációs és felügyeleti feladatokat lát el, többek között még a http.sys működését is képes befolyásolni. A http.sys-hez hasonlóan a WAS sem tartalmaz „külső" kódot, tehát a konkrét webszolgáltatás nem az ő feladata. Ő a főnök, a munkát majd a beosztottak végzik el. Nemsokára velük is megismerkedhetünk.

Az „IIS 5.0 Isolation Mode"

Az IIS örök problémája az volt, hogy ha megengedi *(márpedig megengedi)*, hogy a webes kérések kiszolgálását scriptek, scriptmotorok, sőt, külső komponensek végezzék, ezzel kiteszi magát annak, hogy ezek a külső komponensek leállhatnak, lefagyhatnak. Az IIS5-ben a webes alkalmazások futtatását három „védelmi szinten" engedélyezhettük *(ez az „IIS 5.0 Isolation Mode")*:

- **Low *(IIS Process)*:** ebben az üzemmódban a komponenseket a webkiszolgáló processzében futtatjuk. Teljesítményszempontból ez a legjobb megoldás, mert

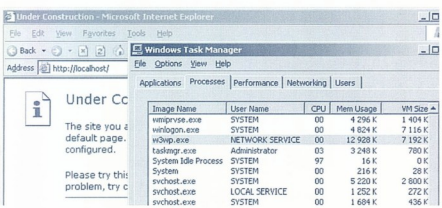
ekkor nincs szükség processzek közötti kommunikációra, stabilitási szempontból viszont a legrosszabb, mert egy leálló komponens magával rántja a teljes webkiszolgálót is.

- **Medium *(Pooled)*:** Minden webalkalmazás, amely ezen a szinten fut, egy közös, külső futtató processzen *(dllhost.exe)* osztozik. Teljesítményszempontból roszszabb, mint az előző változat, mert a webkiszolgálóval történő belső kommunikáció RPC hívásokat használ, ami idő- és erőforrásigényes dolog; stabilitási szempontból viszont jobb, mert a komponens lefagyása esetén a webkiszolgáló talpon marad *(igaz, hogy ezzel együtt a többi „Medium" szintű alkalmazás is lefagy)*.
- **High *(Isolated)*:** Ebben a módban minden webalkalmazás saját dllhost.exe-t kap. Stabilitási szempontból nyilvánvalóan a legjobb megoldás, mert ekkor az öngyilkosság végre mindenkinek a magánügye; teljesítményszempontból viszont valahol még az előzőnél is rosszabb, mert az RPC-hívások mellett a kiszolgálónak még a dllhost.exe-k tömegével is meg kell küzdenie. Egy kiszolgálón belül nem is ajánlott 2-3-nál több „High" szintű webalkalmazást futtatni!

A fenti problémák gyökere abban rejlik, hogy az IIS5-ben a beérkező kéréseket mindig a „nagy" webkiszolgáló processz, az inetinfo.exe kapta, és szolgálta is ki. Még ha úgy is döntöttünk, hogy a webes tartalom generálásának terhét levesszük az inetinfo.exe válláról, megtehettük, de a generált, kész válaszokat ezután is neki kellett visszaadnunk *(költséges RPC hívások segítségével)*.

A „Worker Process Isolation Mode"

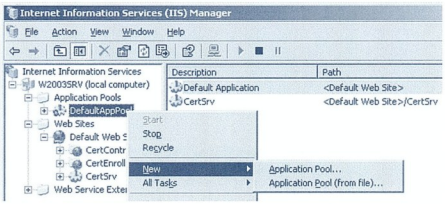
A fentiek fényében *(és a http.sys ismeretében)* nagyjából kitalálható az IIS6 újítása: itt ugyanis a kérések-válaszok kezelését az eszközmeghajtó végzi, őt pedig bármely processz körülbelül ugyanolyan hatékonysággal érheti el. Az IIS6-ban minden komponenst leválasztottak a webkiszolgáló magjáról *(a WAS-ról; olyan ez, mintha most mindenki „High" módban működne)*. A kérések kiszolgálását úgynevezett „Worker Process"-ek, tulajdonképpen mini webkiszolgálók végzik. Ők a dolgozók, a WAS beosztottai *(nevük w3wp.exe)*: a WAS létrehozhatja, leállíthatja, felügyeli őket, ők pedig dolgoznak: átveszik a http.systől a várakozó kéréseket, feldolgozzák azokat, majd a választ visszadják a http.sys-nek. Mi történik, ha meghal egy katona? Helyébe lép egy másik, a WAS ugyanis folyamatosan tartja velük a kapcsolatot, és ha ez a szomorú tény bekövetkezne, megteszi a szükséges lépéseket *(új WP-t indít)*. A WAS egyébként „nyugdíjazhat" is: bizonyos időnként a megfáradt dolgozót friss munkaerővel váltja fel. Ez a folyamat a *(WP-k szempontjából legalábbis)* nem túl humánus „Recycling" nevet kapta.



Munkában a Worker Process (w3wp.exe)

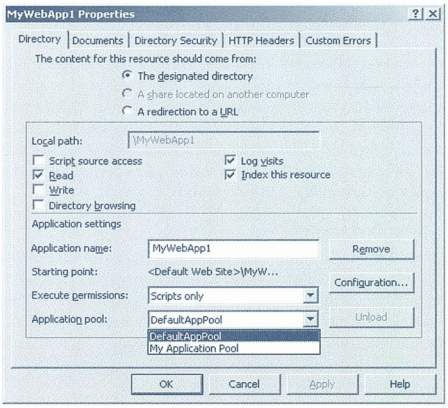
Az Application Poolok

A webkiszolgáló webalkalmazásait alkalmazáscsoportokba *(Application Poolokba)* rendezhetjük. Az Application Poolok logikai objektumok, több webalkalmazás kényelmes, közös felügyeletéhez használhatók. A frissen telepített IIS 6.0 egyetlen Application Poolt tartalmaz, ez a DefaultAppPool nevet viseli. Ebbe az alapértelmezett Application Poolba kerül be minden webalkalmazás, amelyeket nem rendeltünk másikhoz. Új Application Pool létrehozása egyébként néhány kattintás. Mindössze az új Application Pool nevét kell megadnunk, valamint azt, hogy a Pool az alapértelmezett beállításokat kapja, vagy másoljuk le valamelyik, már meglévő társa beállításait.



Az Application Poolok listája az IIS konzolban.

A webalkalmazásunkat pedig hasonlóan egyszerű módon rendelhetjük hozzá bármelyik meglévő Application Poolhoz:



A webalkalmazás beállításai között kiválaszthatjuk, hogy az alkalmazásunk melyik Application Poolhoz tartozzon

Az Application Pool, a Worker Process és a http.sys kapcsolata

Biztosan ismeri a Kedves Olvasó azt az örökbecsű feladványt, hogy mi az összefüggés a tömeg, az idő és a tér között? Nos az, hogy ha jó az idő, a tömeg lemegy a térre. Nos, a fenti három, kevésbé absztrakt fogalom ennél egy kicsit szorosabb kapcsolatban áll egymással:

- Alapértelmezésben minden Application Pool egy *(1)* darab Worker Process-t „használ". Ha a WP még nem fut, és az Application Poolba kérés érkezik, a WAS gondoskodik az elindításáról.

- Az Application Poolonkénti WP-k száma növelhető, ilyenkor gyakorlatilag több, azonos célú miniwebszervert futtatunk, amelyek a beérkező kéréseket egymástól függetlenül, párhuzamosan szolgálják ki *(mintha csak egy webfarmon lennénk; mivel azonban a webkiszolgálók itt egy gépen belül találhatók, ezt webkertnek [Web Garden] nevezték el).*
- A http.sys minden Application Poolnak egy-egy várakozási sort tart fenn, és a beérkező kéréseket a megfelelő várakozási sorba helyezi el.

A http.sys működése pontokba szedve

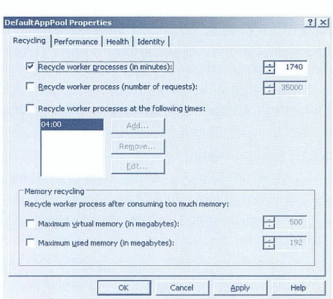
A hálózatról beérkező kérést tehát a http.sys a következőképpen dolgozza fel:

1. A kérés beérkezik a http.sys-hez.
2. A http.sys ellenőrzi a kérést, és ha az érvénytelen, azonnal hibaüzenettel válaszol.
3. Ellenőrzi, hogy a kérésre adott válasz megtalálható-e már a gyorsítótárában *(ez a „Kernel Mode Cache").* Ha igen, a választ azonnal elküldi a kérdezőnek.
4. Ha a válasz nincs a gyorsítótárban, a http.sys megkeresi a megfelelő várakozási sort *(ellenőrzi hogy a kérés melyik Application Poolba tartozó webalkalmazásnak szól),* és a kérést beleteszi a várakozási sorba.
5. Ha a várakozási sort éppen egyetlen WP sem szolgálja ki, a http.sys utasítja a WAS-t egy WP indítására.
6. A WP kiveszi a kérést a várakozási sorból, végrehajtja, majd a választ visszaküldi a http.sys-nek.
7. A http.sys a választ elhelyezi a gyorsítótárában és természetesen visszaküldi a kérdezőnek is.

Ha a kérést feldolgozó WP menet közben leáll, a WAS intézkedhet a probléma megoldásáról. Mindeközben a beérkező kérések a Application Pool várakozási sorába kerülnek, azaz nem vesznek el *(míg a várakozási sor be nem telik).* Sőt, ha a válasz a http.sys gyorsítótárában megtalálható, a felhasználó úgy jut hozzá a válaszhoz, hogy közben a webkiszolgáló tulajdonképpen nem élt!

Worker Process Recycling

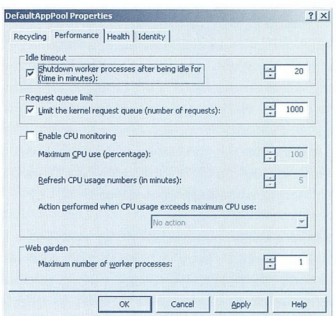
Itt az ideje, hogy megismerkedjünk az Application Pool beállításaival. Az alkalmazáscsoport tulajdonságlapjának első oldalán a WP-k „nyugdíjazásával" kapcsolatos beállításokat találjuk. Sokszor előfordul, hogy egy-egy webalkalmazás hosszabb futás során túl sok memóriát használ fel *(„folyik" a memória).* A probléma megoldása persze a webalkalmazás kijavítása, de vannak esetek, amikor erre nincsen lehetőség.



A munkafolyamatok újraindításának korlátai

A Recycling oldalon megjelölhetjük a WP-k újraindításának feltételeit:

- **Recycle worker processes *(in minutes)*:** eltelt idő alapján. Ez az opció alapértelmezésben is be van kapcsolva, az értéke pedig 1740 perc, azaz 29 óra *(?...).*
- **Recycle worker processes *(number of requests)*:** adott számú kérés kiszolgálása után.
- **Recycle worker processes at the following times:** minden nap adott időpontjaiban.
- **Memory recycling:** Recycle worker process after consuming too much memory: ha a WP virtuális illetve fizikai memóriahasználata eléri a beállított értéket.



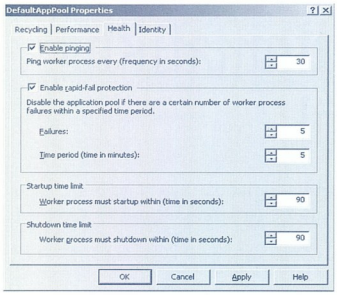
Az Application Pool teljesítménybeállításai

A következő, „Performance" oldalon az Application Pool teljesítménybeállításait találjuk:

- **Shutdown worker processes after being idle for *(time in minutes)*:** ha az Application Poolba nem érkezik kérés, a WAS az itt beállított idő eltelte után leállítja a WP-t. Alapértelmezésben be van kapcsolva, 20 perces értékkel.
- **Limit the Kernel Request Queue *(number of requests)*:** meghatározhatjuk a http.sys adott Application Poolhoz tartozó várakozási sorának méretét. Ha ez megtelik, a http.sys „HTTP Error 503: Service Unavailable" hibaüzenetet küld vissza. Értéke alapértelmezésben 1000.
- **Enable CPU Monitoring:** Ha bekapcsoljuk, az IIS az itt beállított időszak elteltével ellenőrzi a processz CPU-terhelését, és ha az meghaladja a beállított értéket, cselekszik: az Action performed... beállítástól függően ír az eseménynaplóba *(No Action)* vagy leállítja a WP-t *(Shutdown).*
- **Web garden**: Maximum number of worker processes: itt adhatjuk meg, hogy egy Application Pool kiszolgálásához a WAS mennyi WP-t indíthat el. Ha ez az érték több mint 1, máris van egy kiskertünk *(Web Garden).*

Kötelező rendszeres orvosi vizsgálat

A WAS, mint rendes munkáltató, tekintettel van a beosztottak egészségi állapotára *(még szerencse, hogy a WP-k nem igénylik az óránkénti tíz perces pihenőt :-)).*



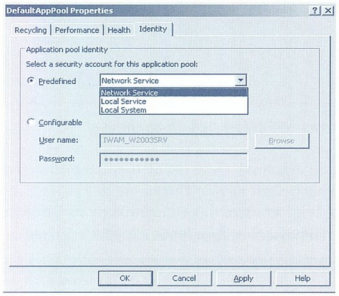
A Worker Process-ek „egészségügyi" beállításai

Az Application Pool tulajdonságlapjának „Health" oldalán ezeket a beállításokat találjuk:

- **Enable pinging:** ha engedélyezzük, a WAS adott időnként *(az eredeti beállítások szerint 30 másodpercenként)* „megpingeli" a WP-t *(egy belső – named pipe – kommunikációs csatornán keresztül adatot küld neki)*. Ha a folyamat nem válaszol, vagy a csatorna bármikor lebomlik, a WP halottnak számít, és a WAS helyette újabb példányt hoz létre.
- **Enable rapid-fail protection:** ha a WP adott időn belül *(5 perc)* egy adott értéknél *(5)* többször hal meg, az IIS nem próbálkozik tovább az újraélesztésével. Helyette a hiba kijavításáig leállítja az Application Poolt *(a http.sys ilyenkor az ide érkező kérésekre HTTP Error 503: Service Unavailable üzenettel válaszol)*.
- **Startup time limit:** az útjára indított WP-nek ennyi időn *(90 másodpercen)* belül kell válaszolnia a WAS-nak, különben az „halvaszületést" diagnosztizál.
- **Shutdown time limit:** a WAS által leállított WP-k ennyi időt *(90 másodpercet)* kapnak, hogy saját erőből befejezzék a működésüket és kilépjenek. A beállított idő eltelével a WAS könyörtelenül kilövi a processzt *(teheti, mert a WP-k a WAS gyermekprocesszei)*.

A WP-k processzeit futtató felhasználói fiók

Az Application Pool tulajdonságlapjának utolsó oldalán azt határozhatjuk meg, hogy a létrehozott WP-k mely felhasználói fiók nevében fussanak.



Az Application Pool tulajdonságlapjának „Identity" oldala

Alapvetően két lehetőségünk van: választhatunk az „előre definiált" fiókok közül, vagy megadhatunk saját, kézzel létrehozott felhasználói fiókot is. Az előre definiált beállítások között három rendszerszolgáltatás-fiókot találunk:

- Network Service
- Local Service
- Local System

Ezek „erőssége" a fenti lista szerint fentről lefelé nő, azaz a legkevesebb joga a rendszerben a Network Service fióknak van. Egyébként ez az alapbeállítás is.

Nézzük meg a wwwroot könyvtár alapértelmezett jogosultságait:

Új tag a wwwroot biztonsági oldalán: IIS_WPG

A jó öreg IUSR_számítógépnév bejegyzés mellett feltűnik egy új csoport is, ez az IIS_WPG. A csoportnak „gyárilag" az IWAM_számítógépnév felhasználó a tagja, de ide kell felvennünk mindenki mást is ahhoz, hogy megfelelő jogosultságokkal ruházzuk fel egy WP futtatásához. Visszatérve az Application Pool tulajdonságai közé: a WP-t futtató felhasználói fiók kiválasztásánál megadhatunk saját fiókot is *(az alapértelmezés itt természetesen az IWAM_számítógépnév)*. Ahhoz viszont, hogy a WP gondok nélkül futhasson a kiválasztott fiók neve alatt, a fiók tagja kell, hogy legyen az IIS_WPG csoportnak.

A következő számban a webkiszolgáló további újdonságaival…

…folytatjuk!

Fülöp Miklós
mick@netacademia.net

Kapcsolódó NetAcademia tanfolyamok:
Windows 2003 Expert Workshop

Windows XP automatikus telepítése CD-ről

Aki már telepített operációs rendszert úgy, hogy órákon keresztül nyomkodta a next gombot, ismeri az érzést: milyen jó lenne ha a telepítés automatikusan megtörténne! Elég lenne 40 perc – 1 óra elteltével rápillantani a gépre és ellenőrizni, hogy minden sikerült-e?

A válaszfájl

Az automatikus telepítéshez valahogyan előre meg kellene tudni adni a telepítő összes kérdésére a választ. Erre van is lehetőség: NT4 óta az összes Windows operációs rendszer fel van készítve arra, hogy a telepítéshez szükséges adatokat fájlból vegye. Szükség lesz tehát egy válaszfájlra, melyet tetszőleges szövegszerkesztővel – akár egy Notepaddel – elkészíthetünk, vagy használhatjuk a Windows XP CD-jén található varázslót is. Mivel a fájl meglehetősen sok bejegyzést tartalmaz, azt javasoljuk, hogy még a bátrabbak se essenek neki kézzel készíteni egyet, hanem először futtassák le a varázslót, és a kész fájlon fitogtassák ügyességüket Notepaddel.

A válaszfájl felhasználásának két módja ismeretes. Vagy hálózaton keresztüli automatikus telepítéshez használjuk *(ebben az esetben a WINNT32.EXE megfelelő kapcsolójával kell rárányítani a telepítő figyelmét fájlunkra)*, vagy CD-ről bootolva végzünk automatikus telepítést. Ez utóbbi eset annyiban különleges, hogy semmiféle módon nem lehet megváltoztatni, hol is keresse a telepítő a válaszfájlt – annak az A: meghajtóban lévő lemezen kell lennie. Sőt, a fájl neve is kötött: winnt.sif a becses neve.

Telepítés CD-ről

CD-s telepítés esetén tehát a kész válaszfájlt egy floppyra kell másolni winnt.sif néven, majd a Windows XP telepítő CD-jéről bootoljuk a gépet, és rendszerbetöltés kezdetekor helyezzük a meghajtóba az általunk elkészített lemezt. A gép rá fog nézni erre a lemezre és az ott található .sif fájlból egy olvasással betölti az összes információt. Ha ez megtörtént *(a floppymeghajtó kellemes berregése megszűnik és elalszik a hozzá tartozó szép zöld led)*, kivehetjük a floppyt, mert már nem lesz rá szükség.

Ha a válaszfájlban úgy határoztunk, hogy a partícionálás ne automatikusan történjen *(ha például nem a C:\ meghajtóra telepítünk)*, a telepítés elején még eldöntjük, melyik partícióra szeretnénk telepíteni a Windows XP-t, aztán csak magára hagyjuk a gépet, és egy órára elmehetünk kávézni. A partícionálásra még visszatérünk…

Mivel varázslóval egy kicsit gyorsabban elkészíthetjük a telepítéshez szükséges válaszfájlt, mi most ezen lépkedünk végig. Ez azért nem mentesít az alól, hogy tudjuk, mi kerül a winnt.sif-be.

A – belülről .INI szerkezetű – válaszfájlban a következő bejegyzéseknek *(szekcióknak)* feltétlenül benne kell lenniük:

[Unattended]	UnattendMode TargetPath
[GuiUnattended]	AdminPassword TimeZone
[Identification]	JoinWorkgroup
[Networking]	Csak akkor kell bejegyzés, ha be szeretnénk állítani a hálózat használatát
[UserData]	ComputerName FullName

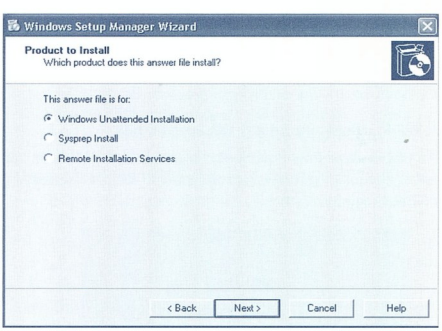
A varázsló használata közben nyomonkövetjük, hogy ezek a bejegyzések milyen értéket vesznek fel. A legtöbb lépés után megmutatom, hogy az adott lépésben mi kerül be a válaszfájlba.

Az egyes bejegyzések és lehetséges értékeik teljes készlete a deploy.cab fájlban található ref.chm súgófájlból kereshető ki.

A varázsló használata

Mint már említettem, a telepítővarázsló a Windows telepítőlemezén található, egészen pontosan a \Support\Tools\deploy.cab fájlban. Ebből bontsuk ki a setupmgr.exe-t és indítsuk el! A varázsló bejelentkező képernyőjén nyomjuk meg a jól ismert next gombot, majd a következő ablakban döntsük el, hogy egy új fájlt készítünk, vagy már egy létezőt szeretnénk módosítani.

Akár új fájlt készítünk, akár egy meglévőt szerkesztünk át, meg kell határozni a telepítés típusát. A Windows telepítéséhez válasszuk a Windows Unattended Installationt.



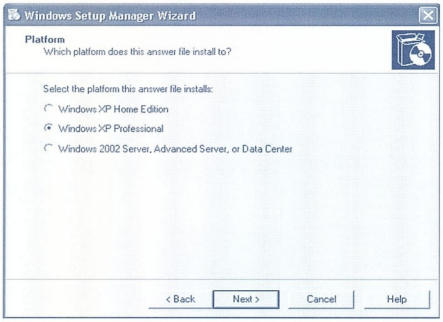
A Windows felügyelet nélküli telepítése

Ezen a ponton álljunk is meg: egy .INI szerkezetű válaszfájlt készítünk, tehát be kell tudnunk határolni, hogy ez a lépés mit is ír a fájlba, hogy későbbi módosításainkhoz ne kelljen a varázslót használnunk. Nos, ez a lépés ennyit ír a fájlba:

```
UnattendedInstall="Yes"
```

A másik két opció *(Sysprep Install és Remote Installation Services)* picit másféle fájlt csinál, legalábbis a fájl neve más lesz: A Sysprep kimenete Sysprep.inf, a RIS-ág bejárása után pedig remboot.sif nevű fájlt kapunk, melyek szintén .INI fájl szerkezetűek.

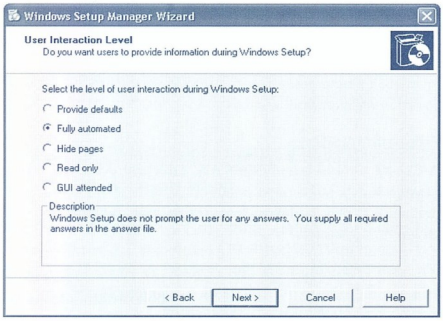
Most nem foglalkozunk sem a Syspreppel, sem a RIS-sel, haladjunk tovább, adjuk meg, mely platformot választjuk!



Telepítendő Windows verziójának kiválasztása

Miután kiválasztottuk a Windows megfelelő verzióját, döntsünk a telepítés módjáról is. Öt különböző választási lehetőségünk van:

- **Provide defaults:** a válaszfájl alapértelmezett válaszokat szolgáltat a telepítéshez, melyek megjelennek, és ezeket a felhasználó telepítéskor akár meg is változtathatja.
- **Fully automated:** teljesen automatikus telepítés. Ilyenkor semmiféle visszajelzést nem kapunk, minden értéket a válaszfájlból vesz a rendszer.
- **Hide pages:** a provide defaults módtól annyiban különbözik, hogy bizonyos ablakokat el lehet rejteni, és csak azok jelennek meg, amelyeket a felhasználók megváltoztathatnak.
- **Read only:** a telepítéskor megjelenő ablakokban minden adat megjelenik, melyet ebben a fájlban tárolunk, viszont a telepítés paraméterei nem változtathatók meg.
- **GUI attended:** csak a szöveges részek automatikusak.



Válasszuk a teljesen automatikus telepítést!

Ennek hatására a fájlba ez kerül:

```
UnattendMode=FullUnattended
```

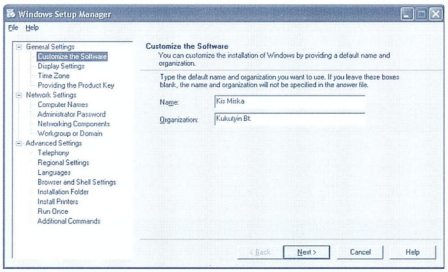
Ezek után adjuk meg a következő ablakban, hogy CD-ről történik majd a telepítés. Itt megadhatunk egy könyvtárat is, ahonnan majd a telepítést végezzük. Mi most maradjunk a CD-nél.

Pipáljuk be, hogy elfogadjuk az EULA-t, így a telepítés során már nem kell vele törődnünk.

```
OemSkipEula=Yes
```

Most már „csak" azokon az ablakokon kell végiglépkednünk, melyeket a telepítés közben már megszoktunk. Ha a teljesen automatikus telepítést választottuk két ablakkal ezelőtt, bizony lesznek olyan részek, melyeket kötelező kitöltenünk.

Ezek a név *(a szervezet maradhat üresen is)*, a termékkulcs, és a számítógépnév mezők.



Töltsük ki a név mezőt!

A fájlba ezután ez kerül:

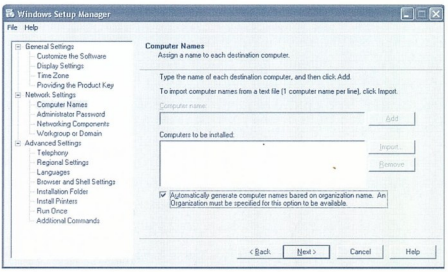
```
[UserData]
ProductID=AAAAA-AAAAA-AAAAA-AAAAA-AAAAA
FullName="Kis Miska"
OrgName="Kukutyin Bt."
ComputerName=kukutyin
```

Önmagában a termékkulcs automatikus bevitele miatt érdemes egy ilyen lemezt készíteni *(például Provide Defaults opcióval)*, mert így legközelebb nem kell begépelni a kulcsot. A Provide Defaults miatt kitöltve, de módosíthatóan jelenik meg minden opció!

Megadható egyszerre több számítógépnév is egy listában, de ezt a lehetőséget a CD-s telepítésnél nem tudjuk kihasználni. Egy pipa elhelyezésével utasíthatjuk a gépet arra is, hogy automatikusan generálja ezt az értéket a szervezet nevéből. Ennek feltétele, hogy az Organization mezőt előzőleg kitöltsük. Ekkor

```
ComputerName=*
```

kerül a fájlba.



A gép nevét automatikusan is generáltathatjuk

A többi mezőt hagyhatjuk változatlanul is, ilyenkor az alapértelmezett beállítások lesznek érvényben. Nézzünk néhányat közülük!

Mint a fenti táblázatban láthattuk, a válaszfájlban szerepelnie kell azon könyvtár nevének, ahova a telepítést végezzük. Ez Windows XP esetén a WINDOWS, melyet így állítunk be:

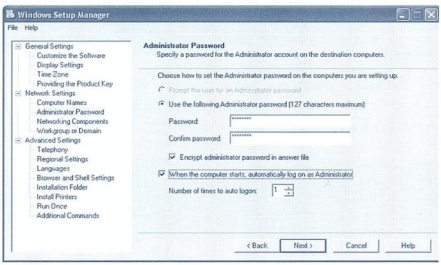
```
TargetPath=\WINDOWS
```

Itt jegyzem meg, hogy a Windows-könyvtár megváltoztatásával egy csomó hibásan megírt program képtelen lesz rendesen működni, így többek között a Nimda és a CodeRed férgek is elpusztulnak! Ezek bizony hibásan megírt programok!

Az időzónának és az Administrator jelszavának is szerepelni kell a winnt.sif-ben. Ha a varázslóban nem állítunk be semmit, a következő értékeket veszik fel:

```
AdminPassword=*
TimeZone=85
```

Ha mégis megadnánk az Administrator jelszavát, titkosíthatjuk is, valamint azt is szabályozhatjuk, hogy hányszor lépjen be automatikusan az Administrator.



Titkosítsuk a jelszót és engedélyezzük az automatikus belépést!

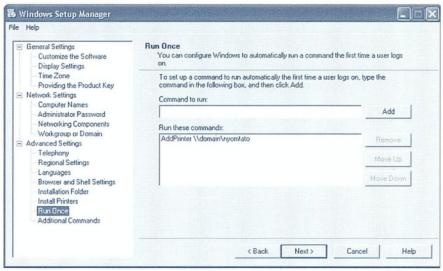
```
AdminPassword=e52cac67419a9a224a3b108f3fa6cb6d8846
↳ f7eaee8fb117ad06bdd830b7586c
    EncryptedAdminPassword=Yes
    AutoLogon=Yes
    AutoLogonCount=1
```

Ha megadjuk, hogy az Administrator x-szer lépjen be automatikusan, lehetővé válik tetszőleges parancsok utólagos futtatása. Ideális lehetőség automatikus Office-telepítés beindítására. Ezzel az operációs rendszer telepítéséhez minden adatot megadtunk.

Egyéb lehetőségek

Azokon a beállításokon kívül, melyekkel a telepítés során is találkozunk, van néhány, amit már most is megadhatunk, a későbbiekben pedig nem kell velük bajlódni. Ilyen például, hogy tetszőleges hálózati nyomtatót telepíthetünk, megadhatjuk, hogy az Explorerben mi legyen a kezdőoldal, vagy a kedvencek között milyen oldalak szerepeljenek. A telefon tárcsázási beállításai vagy a területi beállítások is ide tartoznak.

Bár a nyomtatók telepítésének beállítását külön ablakban elvégezzük, mégis ide kerül a nyomtató hozzáadása utasítás is:




```
[GuiRunOnce]
Command0="rundll32 printui.dll,PrintUIEntry
↳ /in /n \\domain\nyomtato"
```

A varázsló a végén létrehoz egy unattend.txt-t és egy unattend.bat-ot. Az unattend.txt tartalmazza bejegyzéseinket, míg az unattend.bat-ot használhatjuk hálózaton keresztüli automatikus telepítés beindítására. Ez utóbbi esetben szükségünk lesz egy hálózatot is használni tudó operációs rendszerre, ami lehet akár a MS-DOS is. Ha a hálózatról telepítendő oprendszert NTFS partícióra szeretnénk tenni, érdemesebb Windows PE-vel bootolni, ez ugyanis egy Windows XP-ből készített alap oprendszer: látja a hálózatot, a CD-meghajtót és kezeli az NTFS fájlrendszert is.

Partícionálás

Mint a bevezetőben említettem, érdemes lehet megakadályozni, hogy a telepítő maga kezelje a partíciókat. A kész válaszfájl automatikus partícionálást tartalmaz:

```
AutoPartition=1
```

Ez azt jelenti, hogy a telepítő a legelső szabad *(Windowst még nem tartalmazó)*, és elegendő üres helyet tartalmazó partícióra kerül. Ha nincs ilyen partíció, készít magának egyet. Ennek az eljárásnak a bölcsessége megkérdőjelezhető, ezért inkább írjuk át ezt az értéket 0-ra!

Felkészülés, rajt!

A CD-ről történő telepítéshez már csak egy dolog van hátra, nevezzük át a keletkezett unattend.txt szövegfájlt winnt.sif-re, másoljuk rá egy hajlékonylemezre és már indulhat is a cikk elején leírt telepítés.

Borsi Katalin
bobo@netacademia.net

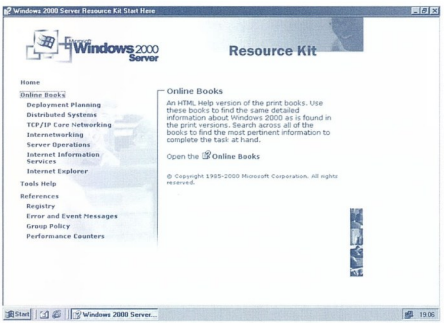
Gyógyszeresdoboz

Windows 2000 Server Resource Kit I.

Bár a horizonton már ott tündököl a Windows 2003 Server, azért még jó ideig nem merül feledésbe a Windows 2000 sem. Márpedig ha használjuk, lesznek ügyes-bajos problémáink. Ha problémáink vannak, jól jön a segítség. Például egy dokumentum és segédprogram gyűjtemény formájában a gyártótól? IGEN!

Annak ellenére, hogy kis hazánkban is sokan használják és kihasználják a Resource Kit csomag nyújtotta előnyöket *(a levlistákon visszatérő fordulat a „keresd meg a ResKit-ben", „van jó kis program erre a ResKit-ben" válasz)*, az angol elnevezést mégsem sikerült eddig frappánsan, esetleg – a tartalomnak megfelelően - sokatmondóan lefordítani magyarra. A Windows NT4 Workstation magyar nyelvű változatának kiadása után azért mégiscsak történt előrelépés: az első és tudomásom szerint a témakörben egyetlen magyarul megjelent háromkötetes könyv „Üzemeltetői Enciklopédia" névvel illette, még 1998. novemberében. Így vagy úgy, de érdemes legalább kipróbálni, vagy még jobb elmélyedni benne, esetleg még akkor is, ha nem egy konkrét problémára keresünk megoldást. Ebben a folyamatban szeretnénk segíteni ezzel a régóta vajúdó – legalább – kétrészes írással.

Egy kis áttekintés



Akad itt olvasnivaló bőven

Resource Kit sokféle van, sőt szinte azt lehet mondani, hogy termékenként *(pl. Windows NT4 Server és Workstation, Windows 2000 Server és Professional, XP, Exchange Server, Office, stb.)* létezik egy-egy. A **[reskitek]** címen megtekinthető az összes Resource Kit listája. Egy csomag általában alaposan megírt és részletes, mélyvíz jellegű könyve*(ke)*t plusz segédanyagokat, és egy halom, kisebb-nagyobb segédprogramot tartalmazó CD-ből áll. Egy példa: a Windows 2000 Serverhez 7 könyv *(kb. 7000 oldalon)*, közel 300 program *(a **[reskitprogs]** címen az összes alkalmazás neve megtalálható abc sorendben)* és számtalan egyéb technikai segédanyag tartozik. A Resource Kit fizetős termék, nem jár a szoftverhez *(mint pl. a Support Tools)*, ám nagyon sok részlet a könyvekből online megnézhető, és pl. a Windows 2000 Server esetén a parancssori eszközök közül jópár le is tölthető a **[reskitdown]** címről. A Microsoft Technet *(nem az ojságról van szó ☺)* előfizetők beépítve kapják, az MSDN előfizetők pedig ingyenesen letölthetik.
Ebben az írásban elsősorban Windows 2000 Server Resource Kit CD-n található szoftverkomponensekről lesz szó, de azért álljanak itt a csomagban megtalálható könyvek és segédkönyvek címei is:

- Deployment Planning Guide
- Distributed Systems Guide
- TCP/IP Core Networking Guide
- Internetworking Guide
- Server Operations Guide
- Internet Information Services Resource Guide
- Internet Explorer Resource Guide
- Technical Reference to the Windows 2000 Registry
- Error and Event Messages
- Group Policy Reference
- Performance Counters Reference.

A CD tartalma

Ha rákoncentrálunk a CD-re, az egyszerű és gyors telepítés után *(a full install kb. 60MB, de közel sem telepít fel mindent, ez szinte csak a tartalomjegyzék)* észlelhetjük, hogy a tartalmat nem túl jól elkülönítve, de azért valamennyire mégis követhetően három részre osztották:

- a CD APPS mappájában található általában komplexebb, grafikus felületű, Microsofttól és külső gyártóktól származó komponensek illetve alkalmazások, amelyek rövidebb-hosszabb listáját illetve leírását a CD HTML\start.htm weboldalán nézhetjük meg
- a programcsoportból a Tools pont alól indítható, kategorizált formában látható, kisebb és általában csak egy-egy célra használható programok,
- valamint a Documentation szekció, ahol a korábban felsorolt könyvek és segédkönyvek találhatóak .chm *(help)* formátumban.

A cikkben szereplő, kiemelt **[kulcsszo]** használata:
http://technet.netacademia.net/go?kulcsszo

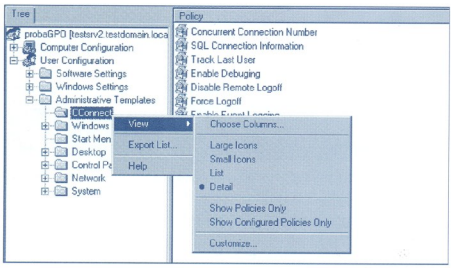
Nézzük először – és ebben a részben kizárólag – a nagyobb alkalmazásokat! A lista persze nem teljes, szubjektív üzemmódban szelektáltam a fontosabbakat illetve az érdekesebbeket.

Microsoft alkalmazások

Cconnect

Ugye másnak is problémát okoz, hogy bizonyos felhasználók több gépen is bejelentkeznek? Jó lenne, ha lehetne ezt valahogy szabályozni, ilyen esetben távolból kiléptetni, sőt, „ha lúd legyen kövér", ne is engedjük, hogy belépjenek! Erre való a Cconnect *(Con-Current Connection Limiter)*. A program egy kliens-szerver alkalmazás, ezért a beléptető szerverre és a megfigyelni szánt számítógép*(ek)*re is fel kell telepíteni a passzoló komponenst. A működéshez alapfeltétel egy - minimum 6.5-ös verziójú - SQL Server, mert a program egy táblába írja a userek belépéseit a különböző számítógépekre. Ezért létre kell hozni ezt a táblát Cconnect néven, majd érdemes egy megfelelő jogosultságokkal rendelkező külön usert is felvenni az SQL Server Security/Logins menüpontja alatt erre a célra *(pl. CCLogin)*, a többit a program elintézi.

Most már csak azt kellene megoldani, hogy minden gépen mindig működjön az ügyfél, be tudja magát írni az SQL táblába és ne is lehessen a kliensen leállítani. Könnyű dolgunk van, mert szerencsére használhatjuk a csoportházirendet. A programhoz mellékeltek egy sablonállományt *(Cconnect.adm)*, amelyet csak be kell importálnunk a vonatkozó szervezeti egység GPO-jába, a User Configuration\Administrative Templates alá. Ami itt egy kicsit fura: a beimportálás után a jobboldali keretben semmi nem lesz látható egészen addig, amíg a Cconnect tároló View menüpontja alatt a Show Policies Only pipát le nem vesszük. Cserébe viszont részletesen bekonfigurálhatjuk a program összes vonatkozó paraméterét.



A beimportált sablon trükkös

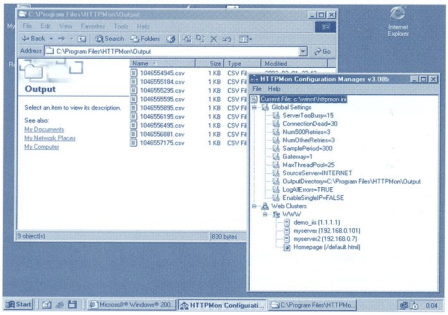
Érdekesség még, hogy elvileg Windows NT4 alatt is működik a kliens, amennyiben legalább SP4 és a Windows Script Host rajta van, és a WBEM és az MDAC v2 vagy magasabb verziója is rendelkezésre áll. A System Policy Editor és a mellékelt sablon segítségével a központi beállításokat szintén elvégezhetjük.

HTTP Monitoring Tool

Egyszerű, ám rendkívül hasznos eszköz, amelyet a webszervereink állapotának ellenőrzésére használhatunk. Rövid telepítés után a program könyvtárában négy mappa jön létre:

- Docs – a dokumentáció
- Output – az ellenőrzés végtermékei: a .csv file-ok
- Samples – mintaállományok a beállításhoz és a használathoz
- Source – a programkód, C nyelven.

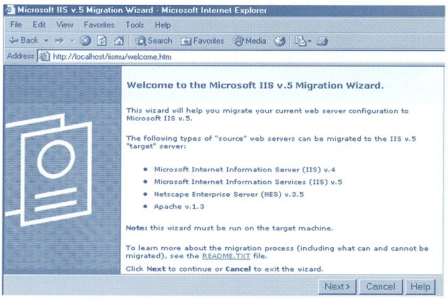
A httpmonconf.exe-vel tudjuk beállítani az ellenőrzés kritériumait *(Global Settings)*, valamint a Web Clusters alatt az ellenőrizni kívánt szerverek IP címét, a szerver és a kezdőlap nevét *(ennek sajnos egyformának kell lenni az összes szerver esetén)*. Ezután indítsuk el a szervizt, -mert a telepítés után ez nem történik meg – és várjunk 5 percet. Ha minden OK, kapunk egy .csv állományt, benne a szabványos http response kóddal, pl. a 200-ast, ha hiba nélkül működik a webszerver *(a Samples mappában találunk egy statuscode.htm állományt a legfontosabb kódokkal)*. A file többi oszlopában többek között a webszerver fejléce, a visszatérési időtartam, a próbálkozások száma és a mintavételezés időtartama szerepel.



A beállítások és az eredmény

Ha sok szervert akarunk bevinni, direkt szerkeszthetjük a \winnt\httpmon.ini állományt is, de ebben az esetben a változások azonnali érvényesítéséhez mindig újra kell indítani a szervizt.

IIS Migration Wizard



Indulhat a varázslás

A program – meglepő módon – nevében hordozza a felhasználási területét: egyszerű átállást biztosít Netscape 3.5.1, Apache 1.3 és az IIS4 webszerverekről. Ez ma már nem biztos, hogy említésre méltó, viszont két IIS5 között is képes erre, ami viszont még mostanság is jól jöhet. Háromféle telepítőcsomag áll a rendelkezésünkre, de amennyiben két IIS5 között akarunk migrálni, a célszerveren a IISv5MigrationUtility-TargetOnly_x86.exe

csomagot kell futtatni, amely létrehoz egy virtuális mappát a Default Web Site-on belül, ezért a folyamat innentől a böngészőből folytatódik, pl. a http://localhost/IISMU címen. Nézzük ennek lépéseit:

- Először meg kell adnunk a forrásszerver elérhetőségét *(dns/netbios név vagy IP-cím)*, és lehetőségünk van a folyamat lépéseinek lementésére *(aztán egy másik szerveren a visszatöltésére)* egy .cab file-ba.
- Természetesen megfelelő módon hitelesítetnünk kell magunkat *(név/jelszó/tartomány)*, de csak a jelenleg belépett felhasználó adatait használhatjuk, ergo adminként kell belépnünk.
- A következő lépésben a forrásszerver IIS-mappái közül egyesével kiválogathatjuk a migrálni kívánt beállításokat, tartalmat, MIME-típust, sőt akár az NTFS-jogosultságokat is.
- Elkezdődik a procedúra, a végén lehetőségünk van elmenteni *(a már említett folyamatleíró .cab file mellett)* az activity.log-ot, amelyből kideríthető: mi sikerült és mi nem.

És lőn! Az eredményt egy virtuális site formájában látjuk viszont a célszerver IIS Manager-ében. Ha például a forrásnál olyan partíción is tároltunk IIS komponenseket, amely a célszerveren nem létezik, a program ezt is megoldja az elérési út átírásával.

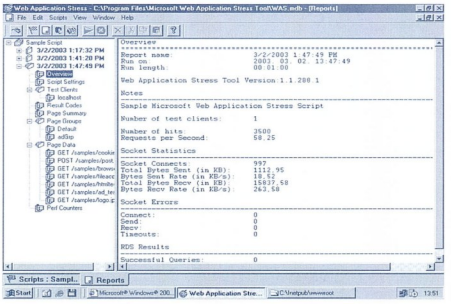
Ami még fontos: a készítők melegen ajánlják, hogy ha már befejeztük a migrálást, biztonsági okokból szedjük le a program összetevőit. ☺

Internet Explorer 5 Administration Kit

A rövidítve IEAK-ként ismert *(letölthető az **[ieak5]** címről is)* komponensnek komoly szakirodalma van, részletes ismertetése valószínűleg feleslegesen foglalna helyet, így röviden csak annyit róla, hogy a böngésző telepítőjének illetve beállításainak testreszabott preparálását lehet vele elvégezni. Többféle megoldás létezik: lehet teljes telepítőcsomagot gyártani vele amely „silent" módban a felhasználó beavatkozása nélkül rakja fel a programot, vagy lehet akár egy floppyn elférő, csak a beállításokat tartalmazó csomagot készíteni vele, amit pl. a már feltelepített IE-vel lehet „megetetni", ha nem akarunk komplett újratelepítést. Természetesen létezik változat már az IE6-oshoz is **[ieak6]**, de azt tudnunk kell, hogy valós létjogosultsága ezeknek a csomagoknak csak a nem tartományi és nem Windows 2000/XP gépek esetén van, hiszen ellenkező esetben a Csoportházirenddel teljesen ki lehet „radírozni" a felhasználók által végzett módosításokat, illetve tiltani a belekontárkodást.

Web Application Stress Tool

Ez a szoftver szintén az IIS szekció része, webszerverek tesztelésére szolgál. Különböző gyári, illetve akár általunk összeállított szkriptekkel aránylag kevés gépről *(vagy akár egyről is)* jelentős terhelést zúdíthatunk a webszerverünkre, azért hogy megtapasztaljuk teljesítményét, stabilitását, kapacitási határait. A használathoz először is másoljuk be a program mappájából a Samples mappát a webszerverünk gyökerkönyvtárába. A szoftver indítása után látható, hogy a baloldali keretben találhatóak az alapbeállítások, és a gyári szkriptek hét csoportba osztott konfigurálási lehetőségei. Ebbe a keretbe kerülnek az általunk készített saját szkriptjeink tárolói is.



A jelentés részletei

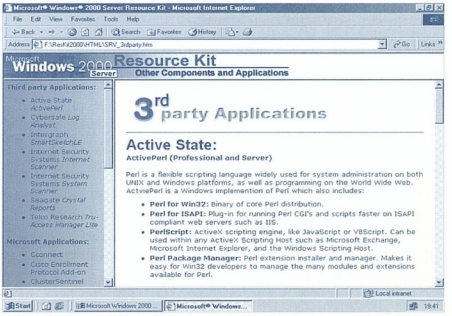
Mielőtt beindítjuk a vizsgálatot, célszerű átnézni ezeket a beállításokat és korrigálni az igényeinknek megfelelően. A Settings rész alatt a teszt erősségét *(szálak és socket-ek szorzata)*, időtartamát *(pl. napi rendszerességgel)* a kérések időbeli véletlenszerűségét, vagy éppen a sávszélességet állíthatjuk be. A Perf Counters menüpont alatt a figyelni kívánt webszolgáltatás különböző paramétereit *(Get Requests/sec, Post Requests/sec, stb.)* vehetjük fel. Érdekes lehetőség az ún. Page Groups opció, amellyel a weboldalakat a terhelés várható százalékos megoszlásának arányában csoportosíthatjuk.

Az összes eddigi jelentés kimenete egy helyen, a program egy új ablakában jelenik meg *(de a teszt után csak a View/Reports menüpontra kattintással lesz látható)*, illetve tudunk exportálni is, .csv formátumban.

Külső *(3rd party)* alkalmazások

Active State: ActivePerl

Ki ne ismerné – legalább hallomásból – a Perl-t, a platformfüggetlen szkriptnyelvet, amelyet webes alkalmazásoknál használhatunk, pl. ez *(a sokaknak szintén jó ismerős univerzális statisztikai eszköz)* az MRTG *(Multi Router Traffic Grapher)* alapja is.



A 3rd party alkalmazások is jól jöhetnek

Ezen a CD-n az Active Perl, a Windowsra írt változat alábbi komponensei találhatóak meg:

- Perl for Win32: a forráskód
- Perl for ISAPI: ISAPI bővítmény, IIS alá
- PerlScript: ActiveX szkript motor
- Perl Package Manager: a Perl komponensek telepítője és kezelője

Cybersafe Log Analyst

Ezzel a remek kis Resource Kit komponenssel az Event Viewer Security kategóriájának naplóállományait tudjuk igényes formában ábrázolni. Egy MMC modult ad hozzá a meglevőkhöz, amelyben megnyitható az aktuális biztonsági naplófile, vagy akár korábban vagy más gépen lementett *.evt állományokat is használhatunk elemzésre. A .11 különböző gyári sablon alapján részletesen beállítható formátumú, színes-szagos grafikonokat kaphatunk, amelyeket aztán ki is nyomtathatunk, sőt a program kifejezetten nyomtatóra van kihegyezve, mert feltelepített nyomtató nélkül nem is működik az ábrázolás.

Intergraph SmartSketchLE

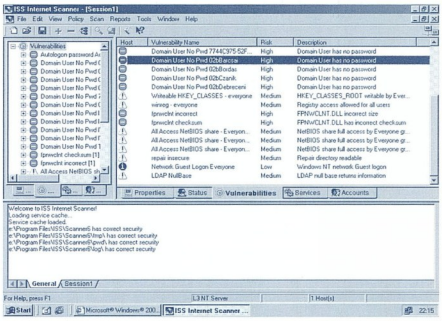
A következő program nem tipikus rendszergazdai eszköz, *(legfeljebb kikapcsolódásnak jöhet jól)*, hanem egy – leginkább a Microsoft Visio-hoz hasonlítható – technikai rajzoló és ábrázoló szoftver „csökkentett" tudású változata. A csökkentett jelző azért idézőjeles, mert iszonyú sok mindent lehet vele ábrázolni, rengeteg sablonnal rendelkezik és könnyedén kapcsolatot teremt a különböző CAD/Office programokkal.

Internet Security Systems Internet Scanner
Internet Security Systems System Scanner

A szintén ismerős ISS cég *(pl. az ISA szerver különböző filterei, bővítményei közül több is tőlük származik)* két szoftvere között sok szempontból nincs túl sok különbség. Mindkettő eléggé átfogó és erőforrásigényes program, ezért szerverekre a készítők szerint nem is ajánlott telepíteni, különösen az Internet Scannert nem. Mindkét programban hozhatunk létre saját policykat is *(ezeken alapulnak majd a vizsgálatok céljai és módszerei)* de létezik jó pár beépített gyári is.

Az Internet Scanner ismert biztonsági lyukakat keres, és elsősorban a hálózati problémák felderítésére használható. Több mint 500 exploit alkalmazását szimulálhatjuk *(NT és UNIX környezetet egyaránt ismeri)*, valamint ezek leírását és – többnyire elég részletesen – elhárításának módjait is tartalmazza. Mivel nem egy mostanában kiadott programról van szó, alapesetben kissé elavult pl. a hackertechnikák vagy a trójaiak területén, de az adatbázisa aktuálissá tehető egy ún. X-Press Update komponens segítségével *(kipróbáltam, működik)*. Első látásra igen bonyolultnak tűnik a kezelés, a sok lehetőség között el-eltévelyeg az ember és olyan érzése van, mintha pl. a Retina *(szintén kimagaslóan jó biztonsági analízis program)* és az MBSA *(Microsoft Baseline Security Analyzer)* lehetőségeit előzetesen összegyúrták volna, de ez erősen szubjektív vélemény.

A System Scanner inkább a futtató gépet vizsgálja, a humán erőforrás által okozott hiányosságokra koncentrál *(üres user jelszó, gyenge registry védelem, megosztások jogosultságainak laza kezelése, stb.)*. Ellenőrzés közben alaposan belenéz a registrybe, a filerendszerbe vagy a felhasználói adatbázisba és a kimeneteként szolgáló weboldalon, avagy állományban mazsolázhatunk a megtalált és azonosított problémáink között.



Itt akad probléma elég...

Seagate Crystal Reports 6

Ha nagyon részletes és sokoldalú naplóelemzést szeretnénk, ezt a programot célszerű kipróbálni. Rengeteget tud, avatatlan szemlélő már az elején is elakadhat, pl. a megfelelő típusú telepítés *(több hálózati variáció vagy pl. webszerver)* kiválasztásánál. Választhatunk legalább 8 féle riporttípusból *(levél, űrlap, stb.)* a forrás lehet továbbá többféle adatállomány, SQL adatbázis vagy akár egy webszerver naplófile is. A Windows 2000 mindhárom alap eseménynapló állományával *(System, Security, Application)* elboldogul. A riportok alapanyagának beolvasása után a szerkesztéshez számos, az adatbázisokkal kapcsolatos, illetve a külcsínyen javító formázó/grafikai eszközünk van.

Telco Research Tru-Access Manager Lite

Ez egy érdekes program *(bár nálam rejtélyes okokból rendszeresen szétfagyott)*, a halózatunkra belépő felhasználók tevékenységeinek „radarozására" szolgál. Össze kell kötni pl. az IAS-sal *(Internet Authentication Server)* és aztán naplózni lehet vele a csatlakozás idejét, mérni az eltöltött időt, vagy a felhasznált sávszélességet. A jelentéskészítésnél el tudja különíteni a befelé jövő analóg, digitális illetve VPN forgalmat, és lehetőség van a használt hálózati szolgáltatások kategorizálására is.

A következő részben a parancssori programok kerülnek fókuszba, addig is kellemes próbálgatást!

Gál Tamás
MCSA
gtamas@tjszki.hu

Netacademia Nagylexikon – Címtár

Ha van alkalmazás, amely a rendszergazdáké, akkor az a címtár. Tizenöt évvel ezelőtt még szinte ismeretlen volt ez a fogalom, tíz éve a Novell NDS megrökönyödést keltett újdonságaival, mára ez a technológia minden vállalati vagy intézményi informatikai rendszer alapjává vált. Egy jól konfigurált címtár olyan a rendszergazda számára, mint egy remekül hangolt hangszer. Ám a hozzá nem értés sok keserű órát és hiábavaló fáradozást eredményez.

Amikor fogalmakat kell tisztázni, mindig két kedvenc mondatomat idézem. Az egyik mondat a DNS zónákhoz kapcsolódik, és így szól: a zóna egy fájl. A másik a címtárakkal kapcsolatos: a címtár egy adatbázis. Szeretem ezeket a definíciókat, mert bár túl sokat nem mondanak, ám aki semmit nem tud, annak ezek világos, valamihez köthető megfogalmazások, hiszen aki rendszerüzemeltetésre adja a fejét, bizonyosan találkozott már valamilyen adatbázissal, mondjuk dBase vagy Access rendszerekkel. Egy adatbázisról lehet tudni, hogy valamilyen definiált struktúra szerint adatokat tartalmaz. Az adatok rögzíthetőek és kinyerhetőek, egy vagy több elemen műveletek végezhetők.
Egy címtár persze speciális adatbázis: speciális a célja, az adatszerkezete és adattárolási módja is. Ahhoz, hogy megértsük, miért olyanok a címtárak ma, amilyenek, és miért arra használjuk őket, amire, egy pillanatra tekintsünk vissza a múltba, amikor még nem volt szükség címtárakra, sőt, még igazán hálózatok sem léteztek.

Címtáralapok

A történet ott kezdődött, amikor lehetővé vált, hogy egyetlen számítógép egyszerre több felhasználót is képes legyen kiszolgálni. Valamilyen módot kellett találni arra, hogy a rendszer a felhasználókat megkülönböztesse egymástól. Valljuk be, ez azért nem túl nehéz feladat. A legelterjedtebb módszer mind a mai napig az, hogy a rendszer megkérdezi a felhasználó nevét, és hogy meggyőződjön arról, tényleg az illető ül-e a gép előtt, kér egy olyan titkot, amit csak a felhasználó tud – ez a jelszó. Ebben az esetben a címtár tulajdonképpen egy táblázat, amelynek minden sora *(rekordja)* egy nevet és a hozzá tartozó jelszót tartalmazza. Erre az alapra azután építkezni is lehet. Egy újabb mező felvételével megadható például a felhasználó saját könyvtára, a többször előforduló nevek azonosítása érdekében belső, egyedi azonosítót használhatunk, sőt, az elektronikus levelezés megjelenése után már az e-mail címet is felvehetjük a felhasználó tulajdonságai közé. Mindazonáltal a módszerünknek vannak korlátai is. Az egyik, hogy nem ad lehetőséget csoportosításra. Minden egyes rekord ömlesztve, egyetlen táblában található. Meg kell oldani a jelszavak biztonságos tárolását is, nehogy illetéktelenek hozzájussanak. Végezetül ki kell dolgozni valamilyen eljárásrendet arra vonatkozóan, hogy ki, mikor és milyen tulajdonságokkal bővítheti az adatbázist. Meg kell alkotni egy eljárást arra az esetre is, ha egy felhasználó adatait egyszerre többen akarják módosítani.
Az idők folyamán ezekre a kérdésekre adekvát válaszok születtek. A nagygépes rendszerek megbízhatóan és biztonságosan működtek, csakhogy időközben elérkezett a hálózatok kora. Tannenbaum mester óta tudjuk: a buta terminálokkal körülvett mainframe-ek tulajdonképpen nem tekinthetők valódi hálózatoknak, mert hálózat csak két vagy több számítási kapacitással rendelkező eszköz között jöhet létre.
Ha van egy mainframe eszközöm, az remek, ha van kettő, az még jobb, csak ezzel együtt a nyakunkba vesszük a több címtár problémáját is. Vajon az egyik rendszer felhasználója hozzáférhet-e a másik rendszerhez? Az üzleti élet gyakorlata szerint a válasz igen. De vajon honnan tudja ama második rendszer, hogy az elsőben regisztrált felhasználó valóban az, aki? A kérdésre kétféle technikai megoldás létezik. Az egyik bevezeti a címtár replikálásának vagyis másolásának fogalomkörét, a másik pedig a címtárak összekapcsolását jelenti. Ha másoljuk a címtárat az egyik gépről a másikra, elimináljuk ugyan a dupla címtár problémáját, ám helyette a másolás komplex kérdéskörével kell megbirkóznunk. Csak néhány kérdés a megválaszolandók közül: a címtár teljes tartalmát vagy csak egy részét *(partícióját)* másoljuk? Mekkora sávszélesség áll rendelkezésre a másoláshoz? Ha az adatok mennyisége jelentős a sávszélességhez viszonyítva, hogyan időzítsük a másolást, hogy az ne zavarja a többi alkalmazást? Hogyan győződjék meg a célállomás arról, valóban a forrásállomás küldi-e az adatokat, és azokat útközben nem módosították, illetve hogyan gondoskodjunk a titkos adatok, mint például a jelszavak biztonságáról?

> Mindaddig, amíg egyetlen géppel dolgozunk, a címtár valójában nem probléma, de már kettő gép esetén is elméletben és gyakorlatban megoldandó kérdések tömkelegével találkozunk.

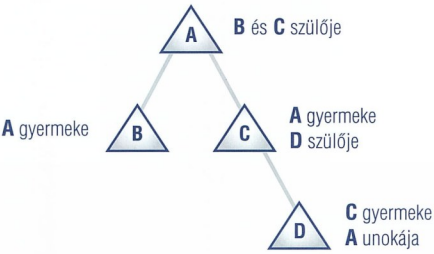
Nem válik könnyebbé a helyzetünk akkor sem, ha inkább a több címtár mellett döntünk. Ekkor azt kell a rendszer tudomására hozni, hogy bizonyos felhasználókat ne a saját címtárában keressen, hanem valahol máshol. Emellett a hitelesítést is annak a gépnek kell elvégeznie, amelyben a felhasználót regisztrálták. Vagyis: annak a rendszernek, amelynek az erőforrásait a felhasználó épp igénybe kívánja venni, meg kell bíznia abban, amelyben a felhasználót regisztrálták *(adatait rögzítették és tárolják)*. Összefoglalva: mindaddig, amíg egyetlen géppel dolgozunk, a címtár valójában nem probléma, de már kettő gép esetén is elméletben és gyakorlatban megoldandó kérdések tömkelegével találkozunk.

Vissza a jövőbe

Manapság nem egy, nem kettő, hanem több tucat, sőt akár több száz, ezer vagy tízezer intelligens állomás kapcsolódik össze hálózattá. Mindegyik tartalmazhat bizalmas információkat, mindegyiknek szüksége van a felhasználók azonosítására. A megoldás olyan címtárért kiált, amely a korábban vázolt kérdésekre és még sok minden másra is megnyugtató választ ad. Korunk vállalati címtárai elosztott, replikálható, hierarchikus, megbízási kapcsolatokra épülő, méretezhető, paraméterezhető, bővíthető, többféleképp használható, szabványos adatbázisok. Az alábbiakban minden egyes jellemzőt alaposan körüljárunk. A tulajdonságokat egy-egy konkrét példával illusztrálom, természetesen a Windows 2000 címtárát felhasználva.

Elosztott címtár

Az elosztott címtár annyit jelent, hogy a címtár teljes adattartalma egyik szerveren sem található meg, hanem csupán az adatok egy része. Ez logikus felépítés, hiszen ha van egy felhasználó Kuala Lumpurban, az vélhetően sohasem fog Budapesten dolgozni, tehát nincs szükség arra, hogy az adatait a budapesti szerver is tárolja. Persze kizárni semmit sem lehet, ezért ha mégis bekövetkezne ez a valószínűtlen esemény, a címtár egy másik tulajdonsága segíteni fog a probléma megoldásában.
A Windows 2000 beépített címtárában, az Active Directoryban egy tartomány jelent egy partíciót. A tartományon belül minden kiszolgáló törekszik a partíció valamennyi adatának frissen tartására, de más partíciók adatait nem tárolja.



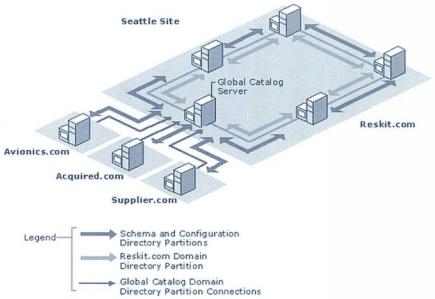
Az Active Directory elosztott és hierarchikus címtár

Hierarchikus címtár

Mivel a világ tele van hierarchiával, és egy adatbázis feladata nem más, mint adatok segítségével a valóság leképezése, ne csodálkozzunk, ha a címtárak hierarchikus szervezését lehetővé tették. Hogy ez a hierarchia mit is képezzen le ténylegesen, azt a fejlesztők már az üzemeltetőkre bízzák. Modellezhetjük hierarchiával a vállalati szervezetet, a földrajzi elhelyezkedést, a rendszeradminisztrátorok hatáskörét és így tovább. A hierarchia kialakítása nagy körültekintést igényel, ezért a Microsoft sokféle ajánlást tett közzé.
Az Active Directoryban kettős hierarchia érvényesül. Egyrészt tartományokat alakíthatunk ki, amelyek egymás alá, fölé vagy mellé rendelhetők. Másrészt egy tartományon belül szervezeti egységekkel valósíthatunk meg hierarchikus felosztást, amire részben a korábbi rendszerek miatt volt szükség. A szervezeti egység a Windows NT 4.0 erőforrástartományát váltotta fel. Ez azonban már nem része az alapismereteknek.

Replikálható címtár

Ha a címtárszolgáltatást több szerver együttesen nyújtja, meg kell oldani az adatok másolását, figyelembe véve a szerverek közötti sávszélességet, a másolandó adatok mennyiségét és a másolásra rendelkezésre álló időt. Gondoljunk bele, hogy bizonyos adatok másolására igen gyorsan sort kell keríteni, míg mások várhatnak. Ha egy felhasználó a kelleténél többször hibás jelszóval próbálkozott, és ezért az egyik címtárkiszolgáló a tulajdonságai közül megváltoztatta a kizárásra vonatkozót, ezt a módosítást gyorsan a többi szerver tudomására kell hozni. Ha viszont valaki másik irodába költözött, és ezt a címtárban feltüntetjük, technikai értelemben nem vészes, ha ez az információ csak másnap frissül a többi kiszolgálón.



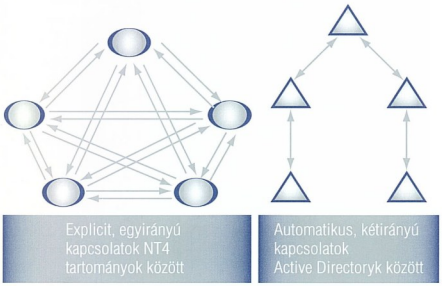
Egy nem is túl összetett replikációs rendszer

Az Acive Directory automatikusan hangolja az azonos, illetve eltérő szegmensben található címtárszerverek közötti adatmásolást. A telephelyeken átívelő replikáció időzíthető, az átvitel technológiája *(IP, SMTP)* pedig megválasztható.

Megbízási kapcsolatokra *(trust relationship)* épülő címtár

Kevesen gondolják végig, hogy minden operációs rendszer kétféle helyzetben működhet: önállóan vagy hálózatban. Ha önállóan működik, saját magának kell gondoskodnia arról, hogy a felhasználók azonosítását elvégezze. Ezt minden Linux/Unix operációs rendszer, vagy minden NT verzió elvégzi – szemben a Win9x változatokkal, amelyek felhasználó-hitelesítés nélkül is működnek. Ez azt jelenti, hogy ezer Windows 2000 munkaállomásnak ezer címtára van? Igen. Úgy válik használhatóvá ez az irdatlan mennyiségű adatbázis, ha nem használjuk őket! Egészen pontosan csak néhány, egyébiránt fontos funkciót használunk.
Amikor egy NT egy tartomány tagja lesz *(de nem tartományvezérlője!)*, a központi címtárba bekerül egy speciális felhasználó, amely a gépet reprezentálja. Ha egy tartományban létrehozott felhasználó szeretne bejelentkezni, a bejelentkezési kérelmet a gép a központi tartományvezérlők felé továbbítja,– mert megbízik bennük. Ettől kezdve a dolog egyszerű: ha csak a tartományban rögzítjük a felhasználókat *(vagyis egy „nagy", „közös" címtárban)*, a kicsi, helyi címtárakkal nem is kell foglalkoznunk. Az ezernyi gépet ezernyi megbízási kapcsolat fűzi egy központi címtárhoz, amelynek adatbázispéldányait a tartományvezérlők tárolják.
A fentieken túl egyéb megbízási kapcsolatok is ismertek. Jellemzőjük lehet, hogy milyen objektumok között hoznak létre

kapcsolatot. Két, azonos rendszerben *(pontos nevén: erdőben)* lévő Windows 2000 tartomány között automatikusan jön létre a kétirányú kapcsolat.



A megbízói kapcsolatok száma függ a tulajdonságaiktól

Két NT4 tartomány vagy egy NT4 és egy Windows 2000 tartomány között kézzel kell létrehozni a megbízási kapcsolatot, amely egy irányban érvényes, és csak a két tartomány között értelmezhető, vagyis nem tranzitív. A tranzitivitás fontos tulajdonság. A fenti ábra jobb oldalán a mindössze négy kapcsolat automatikus létrejötte után minden tartomány megbízik a másikban, vagyis elfogadja, ha egy felhasználóról egy tartományvezérlő azt mondja: „ő tényleg az, akinek mondja magát". Ugyanezt az eredményt NT4 tartományokkal húsz, kézzel létrehozott, egyirányú kapcsolattal érhetjük el. Ezt jelenti a gyakorlatban a tranzitivitás.

Méretezhető címtár

A méretezés nem a címtár logikai szerkezetére, inkább fizikai megvalósítási módjára utal. Adatbázisról beszélünk, amelyben adatokat tárolunk, rögzítünk, lekérdezünk stb., és nem mindegy, hogy ezt milyen gyorsan tehetjük. Tényleg nem mindegy? A mai számítási és tárolási teljesítmény mellett, ezer objektum *(felhasználó, számítógép, csoport stb.)* tárolása csupán 1-2 MB. Komoly méretezési számítások nélkül képesek lennénk akár a legnagyobb magyar nagyvállalat összes felhasználóját, csoportjait és egyéb, tárolásra érdemes objektumát egyetlen tartományvezérlőre pakolni, ezt csupán rendelkezésre állási és vonal-terhelési okokból nem tesszük. Úgy tűnik, a méret nem lényeg.

Azért van másfajta megközelítés is. Tegyük fel, hogy egy internetes áruházat üzemeltetünk, ekkor bizony a vásárlóinkról készítendő adatbázis több tízezer bejegyzést tartalmazhat. Az adatbázis frissítési, lekérdezési teljesítménye azonnal kritikus tényezővé lép elő. Szerencsére az Active Directory nagyon jól méretezhető. A tárolható objektumok száma korlátlan, és a mérések szerint a rendszer válaszidejét a növekvő objektumszám nem befolyásolja.

Az eredmények mögött egy kiforrott adatbázis-technológia áll. A Microsoft ugyanilyen *(vagy inkább nagyon hasonló)* JET adatbázis-motorral látta el korábban az Exchange rendszerét is. A rendszer jól hangolja magát, a műveletek tranzakciók *(tehát a módosítások biztonságos eljárással történnek)*, az adatbázis karbantartása, indexelése automatikus. A rendszergazdákat számos leírás segíti, ha különleges követelményeknek kell eleget tennie a címtárnak, hogy a válaszidő és a rendelkezésre állás megfelelő legyen.

Paraméterezhető címtár

Felhasználók, csoportok, telephelyek, vonalak, sávszélességek, tartományok, adatbázis-példányok – csak a legfontosabb tényezők, amelyeket eddig említettünk a címtárral kapcsolatban. Valamennyi objektumot, és azok egymáshoz való viszonyát is figyelembe kell venni, amikor címtárat hozunk létre, tervezünk, optimalizálunk. Minél nagyobb szabadságot kap a rendszer üzemeltetője a fenti objektumok paramétereinek meghatározásakor, annál jobb alkalmazást tud kiadni a keze közül. Ám ennél még többre van szükség. A címtárnak fel kell tudni ismerni a környezetét, és bizonyos határok között saját magát kell hangolnia a teljesítmény vagy a konzisztens adatbázis megőrzése érdekében. A korábban látott replikációs útvonal-struktúrának automatikusan át kell alakulnia, ha egy tartományvezérlő meghibásodik. Ha két telephely között megváltozik az őket összekötő adatcsatornák sávszélessége vagy a vonalat használó alkalmazások száma, a replikáció ütemezési beállításainak követnie kell a változást – ez azonban már a rendszergazdák hatásköre és felelőssége.

Bővíthető címtár

A címtárak objektumokat tárolnak. A telepítéskor az objektumok tárolt tulajdonságai *(és azok adattípusa)* meghatározottak. Ugyanakkor elvárás, hogy az objektumok száma bővíthető legyen. Ha szeretném nyilvántartani a hálózati aktív elemeimet, szükségem van arra, hogy létrehozhassak új objektum-típusokat, azokhoz pedig új tulajdonság-típusokat.

> Komoly méretezési számítások nélkül képesek lennénk akár a legnagyobb magyar nagyvállalat összes felhasználóját, csoportjait és egyéb, tárolásra érdemes objektumát egyetlen tartományvezérlőre pakolni.

A Windows 2000-ben a címtárat leíró adatszerkezetet sémának nevezzük. Ez a séma *(megfelelő jogosultság mellett)* bővíthető, kiegészíthető, a helyi igények szerint – bizonyos korlátok mellett – testre szabható. Így nincs „bedrótozva" a rendszer. Új adatok, új tulajdonságok segítségével a valóság újabb szelete képezhető le, ezáltal a címtár újabb értelmet kap. Az Active Directory sémabővítése jelenleg egyirányú. Ez azt jelenti, hogy objektumokat és objektum-tulajdonságokat létrehozhatunk, de nem törölhetünk. A fejlesztők gőzerővel dolgoznak azon, hogy a következő verziók ennél nagyobb rugalmasságot tanúsítsanak.

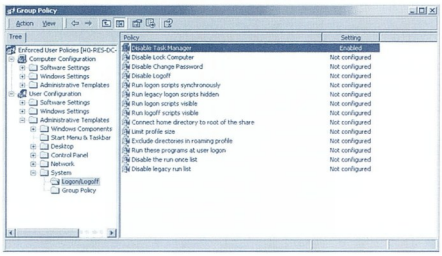
Több alkalmazást is segítő címtár

Ha bővíthető a címtár, új és még újabb alkalmazások vehetik azt birtokukba. A Microsoft több alkalmazását is integrálta az Active Directoryval. Az integráció legalább három szintet jelenthet.

Az első szinten az alkalmazások a címtár biztonsági funkcióit, elsősorban a felhasználók és egyéb objektumok hitelesítését, azonosítását veszik igénybe. Ennek az a következménye, hogy az alkalmazásokba nem kell beépíteni sem saját hitelesítő rendszert egyedi módszerekkel, sem pedig saját címtárat! Nem kell a tűzfalalkalmazás számára külön felvenni mind a 300 felhasználót még egyszer, elegendő a szabályokat felállítani a már létező címtár felhasználóira: a marketingesek internetezhetnek délelőtt is, a pénzügyesek azonban nem. Sok ilyen al-

kalmazás létezik már, és nemcsak a redmondiak fejéből. A Symantec pcAnywhere programja éppúgy integrált Windows alapú hitelesítést használ, mint a Deerfield cég Wingate proxyja.

A második szint, amikor az alkalmazás kibővíti az eredeti sémát, hogy újabb objektumokat vagy a meglévő objektumok újabb tulajdonságait tárolhassa. Tipikus példa az Exchange 2000 Server. Amíg nincs ilyen rendszerünk, addig nem rögzíthetjük, hogy a felhasználó postaládájának mekkora lehet a mérete *(hisz nincs olyan objektum sem, hogy postaláda)*. A sémabővítés segítségével csak a fantáziánk szab határt a különböző típusú alkalmazások létrehozásához. Lelki szemeim előtt egy olyan program lebeg, amely címtártulajdonságokon keresztül szabályozza, hogy a vállalati telefont ki és hogyan használhatja. Régi vágyam, hogy az antivírus szoftverek a saját megoldásaik erőltetése helyett címtárban definiált házirendek segítségével konfigurálják a PC-ken futó komponenseiket, és vírusadatbázisaik frissítésekor a címtár replikációs szerkezetére támaszkodjanak.



Csoportházirend – egy címtárral integrált alkalmazás

Az integráció harmadik szintje kicsit kilóg a sorból. Az ebbe a csoportba tartozó alkalmazások olyan szorosan integráltak az operációs rendszerbe, hogy lényegében annak részét képezik. Az alkalmazások között vannak nagyon egyszerűek, mint például az állomány- és nyomtatómegosztás, és vannak összetettek, mint a digitális tanúsítványok rendszere vagy a csoportházirendek. Közös bennük, hogy az adatokat a címtárban tárolják, gazdagítva az objektumok tulajdonságait, és eliminálva a második és harmadik címtár létrehozásának szükségességét.

Szabványos címtár

Egy adatbázis akkor hasznos, ha a benne tárolt információk rendelkezésre állnak. Az alkalmazások számára a rendelkezésre állás azt jelenti, hogy definiált módon férjenek hozzá az adatokhoz. Szerencsés, ha ez a hozzáférési mód általánosan elterjedt, és nem egyetlen címtárra specializált. A Windows 2000 az LDAPv3 szabványt használja az adatok lekérdezésére. Bármely alkalmazásnak elégséges ezt a bárki számára hozzáférhető, egyszerűen implementálható szabványos módszert használni, hogy értékes információkhoz jusson.

A Windows 2000 címtárához szorosabban vagy lazábban más standardok is kötődnek. A tartományok kialakítása a DNS szabványcsaládnál megszokott elvek szerint történik. A címtárban a tanúsítvány vagy a visszavont tanúsítványok listája a PKI szabványokban leírt tulajdonságokkal bír. A hitelesítési alrendszer, amely tekinthető az Active Directory egyik alkalmazásának, a Kerberost használja. Ez teszi lehetővé például azt, hogy nem Windows 2000 alapú hitelesítő rendszerekkel megbízási kapcsolatot lehessen létrehozni *(Kerberos Realm Trust)*.

Végül meg kell teremteni annak a lehetőségét is, hogy a rendszer más gyártók címtáraival együttműködjön. A vállalatfelvásárlások és cégegyesülések ugyanis különböző gyártók nagyméretű, összetett címtárait sodorják egymás mellé.

Zárszó

Milyen következtetéseket vonhatunk le a leírtakból?

1. A címtár az informatikai rendszer lelke és közepe. Új beruházáskor ennek az alkalmazásnak a tervezésével kell kezdeni a munkát. Csak akkor lehetünk hatékonyak, ha a címtár bevezetését alapos megfontolások előzték meg. Ha átveszünk egy IT-rendszert, az első dolgunk a címtár rendbetétele legyen. Minden más csak ezután következik.
2. A címtárak egyre szélesebb alkalmazásportfólióit integrálnak. Ez egyszerűséget, átláthatóságot, és a valóság jobb leképezését eredményezi. Törekedjünk a címtárral integrált alkalmazások vásárlására és bevezetésére, tartózkodjunk azoktól, amelyek ilyen lehetőséget nem tartalmaznak.
3. A címtárak jelentősége – elsősorban a fenti pontnak köszönhetően – folyamatosan nő, ezért nemcsak érdemes, hanem szükséges is a lehető legmélyebb ismereteket megszerezni az üzemeltetés egyszerűsítése, a biztonság növelése és a folyamatos átalakulások megkönnyítése érdekében.

Lepenye Tamás, MCSE 2000
lepenyet@mal.hu

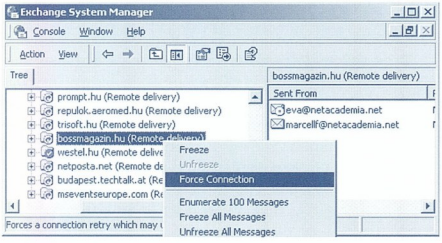
NetMon a gyakorlatban

Hogyan nyerjünk információt a semmiből?

Nemrégiben érdekes probléma ütötte fel a fejét vállalati levelezésünkben: egy, azaz egyetlenegy cégnek nem mentek el a feladott leveleink. S hogy a dolog még furcsább legyen, ők is csak egyetlenegy helyről nem tudtak levelet fogadni: tőlünk. Hol a hiba? Nálunk? Vagy odaát? Majd a Network Monitor megmondja – ha megmondja!

Képzeljék el azt a szerencsétlen szituációt, amikor az ÉN leveleim nem jutnak el egy informatikailag alultámogatott nyomdai partnercéghez. Ki a hibás, na ki? Adódik a feltételezés, hogy – miután mi, MCSE+MZ/X-ek űberokosok vagyunk – csak és kizárólag a partnernél lehet a hiba. Voltam olyan jó, hogy ingyen vállaltam a problémá**juk** megoldását, és mint kiderült, ezt nagyon okosan tettem: kisebb volt a végén az égés. De menjünk szép sorjában.

Az alábbi ábrán Exchange Serverünk SMTP várakozási sorai láthatók egy átlagos délutánon. Megfigyelhető, hogy bizony nem egy Queue várakozási állásponton van, vagyis elsőre nem sikerült oda levelet továbbítanunk. Ezek között húzódik meg szerényen a bossmagazin.hu, ami csak annyiban tér el a többitől, hogy az Exchange már értesítést küldött a feladónak, hogy 12 órája nem tudja kézbesíteni a levelet. *(Ilyet csak akkor tesz, ha már az x. kísérleten is túl van, és mégsem boldogul az adott címre menő levelekkel.)*

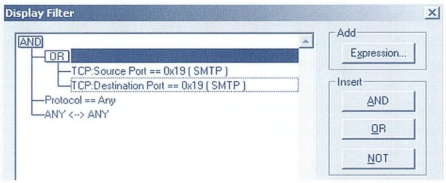


SMTP várakozási sorok az Exchange 2000-ben

Sajnos magának az Exchange Servernek sokszor halvány fogalma sincs arról, mi okozhatja a problémát, vagy ha tudja is, akkor sem mondja meg. A hiba elhárítása sem túl kifinomult: ha valami nem megy, később ismét megpróbálja. A századik kísérlet után azonban emberi beavatkozásra van szükség, ki kell derítenünk *(helyette!)*, hogy mi a baj.

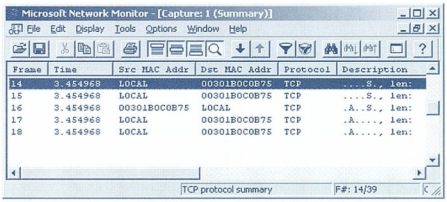
A nyomozást nagyban megkönnyíti, hogy nem kell kivárnunk, amíg ismét rájön a termékre a kézbesíthetnék, hanem az egér jobb gombjának egy jólirányzott kattintásával beindíthatjuk a küldési kísérletet *(a fenti ábrán: Force Connection)*.

Mielőtt azonban ezt tettem volna, elindítottam kedvenc végfelhasználói szoftveremet, a Network Monitort, hogy saját szememmel győződhessem meg a hálózaton zajló csetepatéról. Az elkapott hálózati forgalomból az alább látható szűrő segítségével választottam ki az SMTP-forgalmat *(korábbi NetMon cikksorozatomban már szóltam a szűrők készítésének technikáiról, itt csak a kész szűrőt mutatom meg)*:



NetMon protokollszűrő az SMTP-forgalom kiválasztására

A fenti szűrő csak azokat a csomagokat tartja meg, amelyekben vagy a feladó, vagy a címzett portszáma hexa 19, azaz 25, azaz SMTP. Magyarul azok a csomagok maradnak fenn a rostán, amelyek a 25-ös portra irányuló TCP-csatorna forgalmát adják. És itt ért az első nem várt meglepetés. Ugyanis ennyi csomag maradt:



A szűrőfeltételnek megfelelő 25-ös porti csomagok

Hát ez bizony semmi. Ötször leellenőriztem a szűrőket, de mindhiába: semmi más nem történt, minthogy felépült a TCP-csatorna, amiben aztán nem utazik semmi.

Ha valaki figyelgette már az SMTP hálózati forgalmat, tudhatja: a sikeres csatornaépítést az SMTP-kiszolgáló nem hagyja szó nélkül, hanem rögtön beleküldi saját marketingüzenetét, például:

```
220 mail.netacademia.net Microsoft ESMTP
↳ MAIL Service, Version: 5.0.2195.5329
↳ ready at  Mon, 3 Mar 2003 15:49:06 +0100
```

vagy:

```
220 ns1.ahol.hu ESMTP Sendmail 8.12.7/8.12.7;
↳ Mon, 3 Mar 2003 15:54:54 +0100
```

továbbá *(ez hosszú lesz!)*:

```
220-gil.axelero.hu -- Server ESMTP (iPlanet
↳ Messaging Server 5.1 HotFix 1.9
↳ (built Dec  3 2002))
↳ 220 Axelero Internet ESMTP open
↳ (WE DISALLOW ANY KIND OF SPAM ACTIVITY)
```

és:

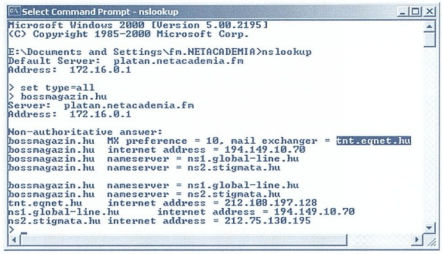
```
220 mail.interware.hu ESMTP Exim Mon,
↳ 03 Mar 2003 15:58:24 +0100
```

Az összes üzenetben ennyi a mondóka lényege: 220. Az üzenet további része egyfelől teljesen felesleges, másfelől veszélyes, hisz információt adnak Hacker Henrynek az általunk használt rendszerről, de már úgy megszoktuk, hogy ha nem lenne *(mint ahogy az én hibám esetében tényleg nincs)*, hiányérzet támadna bennünk. De nincs.

A néma szerver vallatása

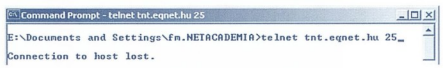
Gondolkodjunk, vajon miért nincs üzenet? Útválasztási probléma lenne? Az nem lehet, hisz a csatorna sikeresen felépül. A partner – talán – egy szerény gyártó SMTP-kiszolglóját használja? Ez sem valószínű. Esetleg egy köztünk lévő tűzfal nyeli el az üzenetet? Ez ellen nincs megdönthetetlen bizonyítékom, viszont értelmetlen lenne. Akkor marad ez: a partner SMTP-kiszolgáló némasági fogadalmat tett. Vajon miért? Ennek kell kideríteni az okát.

Következő lépésként megállapítjuk, hogy az ominózus email-cím postaládáit hol tárolják. Ott, ahová az MX-rekordok mutatnak. Ennek felderítéséhez NSLOOKUP-parancsot használunk, az alábbiak szerint:



DNS-névhez tartozó MX-rekordok kilistázása

Tehát valahányszor @bossmagazin.hu címre küldünk levelet, valójában a tnt.eqnet.hu szerverrel vesszük fel a kapcsolatot. Egy gyors telnet-próba *(parancssorból)* elég ahhoz, hogy tudjuk, jó helyen járunk:



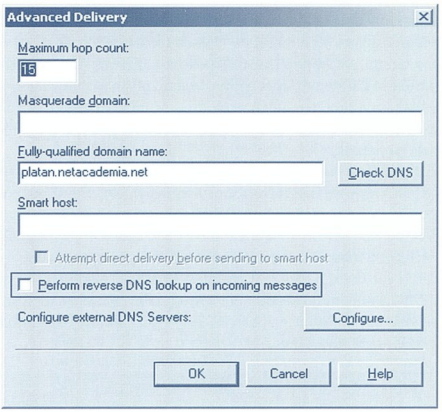
SMTP-kapcsolat parancssorból – így sem megy!

Ezen a ponton vettem fel a telefonkagylót, mert úgy éreztem, szoftveres úton nem tudok továbbjutni. Az Eqnetnél megértően fogadtak *(a sokadik őrült aznap, aki azt hiszi, ért hozzá…)*, és el akarták mesélni, mit kell tennem, hogy az Outlook Expresst használni tudjam. Néhány bűvös kifejezéssel *(MX-rekord, NSLOOKUP, RFC stb.)* meggyőztem az ügyfélszolgálatost, hogy valóban tudom, mit akarok, így perceken belül egy hozzáértő kollégához jutottam.

Innentől felpörögtek az események. A szakértő kolléga mindenekelőtt rákérdezett az én IP-címemre, amit készséggel megadtam, majd közölte, hogy ez a cím nincsen regisztrálva. Hiányzik hozzá a reverse-DNS! Az ő SMTP-szerverük nem válaszol „kalóz" IP-című gépeknek. S hogy miért „kalóz" a mi IP-címünk? Mert még „csak" 7 hónapja, hogy ezt használjuk *(akkor költözött az iroda)*, és ennyi idő alatt nem sikerült megcsinálni a megfelelő bejegyzést az in-addr.arpa zónába. Milyen jó, hogy kiderült! Gyors javítás, no meg a zóna TTL-jének leketyegése után ment minden, mint a karikacsapás.

Reverse DNS az Exchange Serveren?

De ha már előjött ez a probléma, a kisördög elkezdett bújkálni bennem: én is szeretném, ha így viselkedne az SMTP-kiszolgálóm! Nekiestem hát az Exchange alapértelmezett virtuális szerverének, felforgattam minden zugot, de csak ezt találtam:



Reverse lookup-beállítás az Exchange Serveren

Vigyázat, ez nem az! Itt arról van szó, hogy a beérkezett üzeneteken végezzünk-e névfeloldást abban az esetben, ha a feladó csak IP-címmel van megadva. Ez tehát feltételezi, hogy a levelet elfogadjuk. Az Eqnet szervere viszont az SMTP-parancsokat sem fogadta el.

Tanulságok

1. A fordított névlekérdezésen alapuló „biztonságot" gyakorlatilag az égvilágon senki sem használja, hisz több mint fél évig zökkenőmentesen használtuk mindenféle célra bejegyzetlen IP-címünket.
2. A Network Monitor a csendet ugyan nem mutatja meg, de a csend tényét igen, és ez néha a legtöbb információ, ami a rendelkezésünkre áll!

Fóti Marcell
marcellf@netacademia.net
A szerző a NetAcademia vezető oktatója
MCSE, MCT, MCDBA, MZ/X

Kapcsolódó NetAcademia-tanfolyamok:
Network Monitor Workshop

Ki mivel .....?

Ha támad az IT világ fantomja

A tervezés szerepe a kivitelezés tudatossá tétele. A tesztlaborokat pedig arra találták ki, hogy minden, élesben kockázatos teendőt nyugodt körülmények közt, stresszmentesen végigpróbálhassunk. A feladat egyszerű: térjünk át Windows 2000-re. Rutinfeladat. Ráadásul másfél éves tervezés és real-life tesztelés után az ember biztosra mehet. Nemde? Nem.

Fáradtság. Ez a szó sokunknak talán mást jelent, mint a hétköznapi ember számára. Rendszergazdák, programozók számára talán nem is jelent semmit. Annyira a hétköznapjaink részévé vált, hogy nem is vesszük sokszor észre. Gyakran már szinte csak akkor tudunk produkálni, ha nem alusszuk ki magunkat, vagy le sem fekszünk és mégis, másnap mintha picit jobban mennének a dogok. Persze ez csak a virtuális valóság.

Alapos tervezés, gondos tesztelés

Már másfél éve tervezgetjük, hogy egy körülbelül 50 felhasználót kiszolgáló szerveren lecseréljük az NT4-et Windows 2000-re. De nem volt, aki megtegye. Az elmúlt másfél évben Rendszegazdánkat a userek ügyes-bajos napi problémái és a kliensgépek folyamatos frissítése *(hw/sw)* foglalta el, én személy szerint – bár mindenesnek vallom magam – inkább a szoftverfejlesztésben vagyok járatosabb. Persze ezalatt az idő alatt az összes kapcsolatos előadáson jelen voltam, olvastam a dokumentációkat, könyveket, szívtam magamba a tudást, hogy ha majd eljön a Pillanat, a lehető legnagyobb hozzáértéssel tudjunk hozzálátni a dologhoz.

A Pillanat késett egy pár pillanatot, mivel a szerver folyamatosan üzemel hétvégén is, valamint – egyéb okokból – a világegyetem összetevőinek megfelelő konstellációja is kellett ahhoz, hogy végre bekövetkezzen. Most végre eljött az idő!

Mivel a meglévő tartományvezérlő alatti vas modernnek mondható *(tükrözött háttértár, 2 db PIII 800-as szerver proci + Asus alaplap)*, szerettünk volna egy tiszta op'rendszert tudni a hardveren, de úgy, hogy a korábbi címtártartalom megmaradjon. Tiszta vasra tiszta szoftvert! Tehát a cél egy újratelepített, de mégis a régi tartalmat hordozó Active Directory. A tanfolyamokon, előadásokon hallottak alapján a következő tervet eszeltük ki.

1. Egy NT4 Backup Domain Controllert telepítünk valami vasra. Ócskavas is lehet, ez a DC csak kisegítőszerepet lát el.
2. Szerepcsere. A korábbi PDC-t *(ez alatt van a jobbik vas)* detronizáljuk, s egy darabig az ócskavas lesz a PDC.
3. Az ócskavas-PDC-t Windows 2000-re frissítjük. Ezáltal ott a régi SAM tartalmát megőrző Active Directory születik.
4. A korábbi PDC-ről *(a jobbik vas)* lementjük az adatokat. Háromszor leellenőrizzük, majd Format C:
5. Új vasra Windows 2000 Server fel.
6. Ebből is tartományvezérlőt készítünk, azaz: DCPROMO.
7. A jelenleg az ócskavason, mint legelső Active Directory tartományvezérlőn tanyázó öt FSMO szerepet átvisszük a szép-új vasra.
8. Adatok visszatétele.
9. Ha minden OK, a kisegítő szerverről DCPROMO-val AD le, majd kikapcs.

(Lehet hogy kevesebb lépésből is menne, de ez tűnt az abszolute tiszta útnak.)

A Rendszergazda kolléga korábban ezeket a lépéseket az éppen keze ügyében levő tartalék gépeken leszimulálta. Az eredmény kielégítő volt, úgyhogy semminek nem láttuk akadályát. A Nagy Pillanatot megelőző napon előkészítettük az első három lépést.

Aznap este 8-kor amikor az utolsó kliens is felszabadult, nekiláttunk a feladatnak. A régi szerverről lemásoltuk az adatokat, majd aránylag gyorsan el is jutottunk a 6. lépésig.

Látványos kudarc

És itt kezdődtek a gondok. A DNS install, majd a DCPROMO lefutása után valami hihetetlen módon lelassult a gép. „Biztos az AD miatt meg csúnya Microsoft!" Nem baj, majd magához tér, DHCP service fel. Csak éppen nem akart elindulni.

Elkezdtünk izzadni. Az eddig magabiztosnak hitt kezünkben megremegett a csillagcsavarhúzó.

Begyakorolt mozdulatokkal nyúltunk az EventLog felé, ami bizony tele volt piros pontocskákkal. A bejegyzések DNS problémára utaltak. *(Mint mindig – a szerk. Az AD-problémák 137%-a DNS hibára vezethető vissza.)*

És valóban, a DNS-be csak félig kerültek bele az AD működéséhez szükséges bejegyzések! A megfelelő hibajavító eszközök közül a Visual Command Promptos megoldást választottuk *(Manual Wizard)*.

```
NET STOP NETLOGON
NET START NETLOGON
```

Ezután a DNS feltöltődött a kívánatos szeméttel *(SRV-rekordok stb.)*, tehát szemre a DNS-ben minden kijavult. Hittük mi. Sajnos a gép nem óhajtott kommunikálni a másik Domain Controllerrel, úgyhogy reboot.

Újabb izzadságszag terjengett a levegőben, mivel a gép újraindítás után is lassú volt – de az eventlog már megváltozott! Mint kiderült, autentikációs probléma miatt nem tudta rendesen lekommunikálni a dolgait a másik géppel. Ez alatt azt értsük, hogy – mit is? Káoszt! A két tartományvezérlő hátat fordított egymásnak! Pedig az Active Directory Users and Computersben látszott, hogy mindkét gép megjelent a szerverek között, mégsem akarták egymást szeretni!

Pedig hajszálnyira voltunk a céltól, majdnem minden működött, érdekes módon még menedzselni is hagyta az egyik a másikat. Hmm. Éjfélt ütött a BIOS órája, amikor úgy döntöt-

tünk, hogy hiányos ismereteinkkel nem jutunk tovább, nem próbáljuk meg hexeditorral meghekkelni a rendszert, hanem inkább leszedjük az AD-t, DNS-t és újra próbálkozunk.

A fáradtság sunyin, rejtőzködve vár az agyunk hátsó részében, hogy egy előre megjósolhatatlan pillanatban bevesse magát, és mint a legjobb trójai programok, szinte észrevehetetlenül tevékenykedjen.... a kárunkra. Aminek eredményeképpen megintcsak fáradtan kelünk fel reggel az ágyból. Ez az IT perpetuum mobile. Hát hajrá!

1. DCPROMO indít.
2. Nem, nem ez az utolsó DC!
3. Igen remove!
4. Admin login megadása, next, next, error....

Hehh?

Olyat mondott a DCPROMO, hogy nincs jogosultságunk az AD eltávolításához. *(Nem jó az a név+jelszó, amit megadtunk. Pedig igen, mert közben a DC-n ki-be jelentkeztünk ugyanazzal az accounttal).*
Végül egy sokatmondó 'The DSA object could not be found' üzenettel megmakacsolta magát. Többszöri próbálkozás után úgy döntöttünk, hogy az ócskavas-DC-n töröljük a Computer Accountot, ezen a szép, új vason pedig a Format C:-tól megismételjük az eljárást...
Lezúztuk *(rendszergazda módra, brute force)* a rendetlenkedő oprendszert:

```
Format C:
```

A második lépés kicsit váratott magára, mivel az AD nem engedte törölni az accountot, tekintve, hogy a *(megboldogult)* gép fénykorában Domain Controller volt.

Utánanéztünk a Microsoftnál, hogyan lehet kiszedni az AD-ból a beragadt gépaccountot, mivel egyéb *(user, appsettings, boss)* okokból ugyanolyan névvel kellene felinstallálnunk a gépet, mint régen!
Meg is találtuk a gyomlálás módját. Az NTDSUTIL nevű parancssori rémálomra van szükség, amiben addig kell pszeudorandom módon fel-alá mászkálni a menükben, amíg – ki tudja, miért – eltűnik a nemkívánatos account. A **[DCDEMOTE]** címen található KB-cikk öt képernyőn keresztül sorolja a lépéseket – de működik!
Végrehajtottuk a megadott lépéseket, majd újrainstalláltuk szegény leendő szervert. Ekkor már hajnalodott, úgy döntöttünk, hogy AD-t most nem csinálunk, a szükséges applikációkat viszont feltesszük rá.
Két nap problémamentes üzemelés után ismét megpróbálkoztunk az AD-vel. Varázslatos okokból kifolyólag problémamentesen felment. A szerepek átvétele után a „vadi" új, steril környezetre installált DC-nk gond nélkül fut. Azóta is.

Persze mi is rájöttünk, mi volt az az ok, ami másfél éves „előkészítés" után arcátlan módon betett nekünk.

A fáradtság.

Ugyanis NEM jutott eszünkbe, hogy amikor az átmeneti gépünkből Windows 2000 Active Directoryt készítettünk, azzal a céllal, hogy megmentsük a SAM tartalmát, ezzel **tényleg** megmentettük a SAM tartalmát: a DCPROMO bevette a régi NT4-es Backup Domaint Controllert az AD-be. Honnan tudhatta volna, hogy leformáztuk? Felvette, mint tartományvezérlőt!
Mi pedig erről egész egyszerűen elfelejtkeztünk. A tisztára formázott, frissen telepített Windows 2000-nek viszont *(business okokból)* ugyanazt a nevet adtuk, mint ami előtte ezen a vason volt. Valamilyen oknál fogva többé-kevésbé még a DCPROMO-zást is engedte, de végül csak beletörött a bicskája!
A tanulság, hogy a NAGY horderejű változtatásokat nem elég megtervezni, hanem szellemileg is ott kell lenni. Különben letámad az IT világ fantomja!

Két tartományvezérlő ugyanolyan névvel nem lehet ugyanabban a hálózatban. Még akkor sem, ha nem kapunk rá határozott hibaüzenetet. :-)

Antal István
aistvan@brendata.hu
Flórián István
fistvan@brendata.hu

Információbiztonság vagy magánszféra?

Security or Privacy?

Napjaink egyik legfontosabb kérdésévé az információ biztonsága vált. Privacy or Security? Ez a kérdés az Egyesült Államoktól Európán át Ázsiáig foglalkoztatja az új információs technológiákkal dolgozókat, a felhasználókat, mindazokat, akik információt szolgáltatnak, vagy megszereznek.

Mi szól az egyik és mi a másik mellett? A jognak először az egyén védelmét kellett körülhatárolnia, hiszen a jogfejlődés során mindig az egyén szabad mozgását korlátozó, az azon nagy mértékben uralkodó hatalomnak biztosítottak többletjogosítványokat. A fejlett demokratikus jogrendszerek számára az elsődleges feladat tehát az egyéni szféra, a magánélet területének meghatározása és védelme volt. Erre jó példa az Amerikai Egyesült Államok Alkotmánya, amely több kiegészítése révén érte el azt, hogy ma már a magánélet *(privacy)* védelme oly mértékű, hogy az állam Internetszabályozási törekvései sorban visszapatannak e burokról.

Nem maradt le azonban az európai jog sem. Az emberi jogok és alapvető szabadságok védelméről szóló Rómában, 1950. november 4-én kelt egyezmény 8. cikkelye biztosítja a magán- és családi élet tiszteletben tartásához való jogot. Az egyezmény kimondja, hogy mindenkinek joga van arra, hogy magán- és családi életét, lakását és levelezését tiszteletben tartsák. Ez az alapgondolat bekerült aztán az összes demokratikus európai alkotmányba, és ennek biztosítására büntető- és polgári jogi védelmet is nyújtanak a tiltó és szankcionáló szabályok.

Meddig szent a magánélet?

2001. szeptember 11.-e óta azonban megindult egy e jogfejlődéssel ellentétes tendencia is. A világot megdöbbentő terrortámadás előkészítésére az új típusú információs eszközöket, elsősorban a világhálót használták fel. A hasonló előkészületek felfedésére, azok esetleges megakadályozására azonban olyan fokú állami kontrollt kellene lehetővé tenni, amely már komoly mértékben ellene hatna az előzőekben ismertetett, a magánszféra védelmét biztosító jogi környezetnek.

Amióta emberiség létezik, azóta van információ, amelynek birtoklása mindig előnyt, hatalmat is jelentett. Kezdetektől voltak tehát törekvések arra, hogy a más által birtokolt információt megszerezzék, ha kell, jogellenes eszközökkel is. Az első, aki információt jogellenesen szerzett meg és használt fel, a mondabeli Prometheus volt, aki ellopta a tűzre vonatkozó ismereteket az istenektől, és megosztotta az emberekkel. Prométeuszt utolérte ugyan a rettenetes büntetés, ám az információ többé már nem volt titkos, kikerült a privilegizált körből.

A XXI. században az információ, a tudás a legértékesebb, legforgalomképesebb tényező. Az üzleti életben a birtokunkban lévő információ jelentős értéket képvisel, elvesztése, hamisítása, megszerzése a gazdasági életben katasztrofális következményeket vonhat maga után. Egy nemrégiben publikált amerikai jelentés szerint a 2001. évben 59 milliárd dollár veszteséget vallottak be nagyobb méretű cégek, amely a szellemi tulajdon ellopása miatt érte őket. Szellemi tulajdon alatt a jelentés készítői minden olyan információt értettek, amely nem nyilvános, és tulajdonosuk bizonyos intézkedéseket tesz védelmükre. Ez az iszonyatos veszteség, amelynek volumene évről évre nő, ugyancsak azt indokolja, hogy az információs technológiák biztonságosabbá tétele érdekében szülessenek bizonyos jogi lépések. Erről az oldalról közelítve a kérdést, már nemcsak az állami, hatalmi kontroll erősítése jelenik meg mint érdek, hanem bizony a magánszféra érdeke is biztonságért kiált, amely lehet egy személy, de egy kisebb – nagyobb létszámú csoport érdeke is.

Mielőtt azonban a védelem lehetőségeit és akadályait világítjuk át, nézzük meg, hogy kik is a modern kor Prométeuszai? Kiktől kell félteni manapság az információt?

„Prométeuszok"

Elsődlegesen a külső szereplőket vegyük szemügyre. Ezek legismertebb köre a hackertársadalom, akik jórész szórakozásból törik át a biztonságot, de náluk sokkal veszélyesebbek a céltudatos támadók, az ipari kémek, hírszerzők, a gazdasági versenytársak, a cég külső beszállítói, és végül nem jelentéktelen mértékben az egyre újabb hírekre kiéhezett, eszközeiben gátlástalan média. Amennyiben ezek eszköze az adott rendszerbe történő illetéktelen külső behatolás, ennek jogi rendezésére már előrehaladott lépésekkel találkozhatunk: a nemzetközi egyezmények által meghatározott együttműködés segíti a felderítést, és ezek alapján végrehajtott büntetőjogi szabályozás által meghatározott szankciók pedig komoly elrettentő erővel bírhatnak.

> A nem megfelelő védelemmel keletkezett kárért esetlegesen a vállalat felső vezetésének kell helytállnia.

A jog eszközei persze itt is korlátozottak, és akadályokba ütköznek. Ez méginkább így van a másik csoport, az információt belülről illetéktelenül megszerző, és onnan kijuttató körrel szembeni védelem területén. Ide tartoznak a cég alkalmazottai, a cég részére más jogviszony alapján munkát végzők, nem elhanyagolható veszélyt jelentenek a volt alkalmazottak, a beszállítók, és az ügyfelek.

A felmérések szerint a kutatás-fejlesztés eredményei, a pénzügyi adatok, az ügyfélre vonatkozó adatok azok, amelyek a

célpontot jelentik. Más csoportosítás szempontjából veszélyben vannak a vállalat saját üzleti titkai, a vállalat által kezelt személyes adatok *(saját munkavállalók, természetes személy ügyfelek)*, a vállalat által kezelt céges ügyféladatok, az üzleti partnerek üzleti titkai, valamint államtitkot, szolgálati titkot jelentő adatok.

És itt már azzal is számolni kell, hogy a nem megfelelő védelemmel keletkezett kárért esetlegesen a vállalat felső vezetésének kell helytállnia. Hiszen az e körbe tartozó információt külön jogszabályok védik, komoly, akár börtönbüntetést is jelentő szankciókkal sújtva azokat, akik megsegik az azok kezelésére, megőrzésére vonatkozó szabályokat.

A levelezés veszélyei

Az információ biztonságát fenyegető számos kritikus pontból ragadjuk ki a legkomolyabb veszélyt, a céges e-mail kommunikációt. E körben a tévedésből más címre elküldött e-mailektől kezdve, a jogosulatlanul kiküldött leveleken keresztül a spam-en át a rendszer biztonságát sértő e-mail küldéséig a legváltozatosabb példákat lehet sorolni. E-mailen keresztül a jogosulatlan adattovábbítással az adatvédelmi szabályok megsértésétől az üzleti titok, szolgálati titok, államtitok megsértésén át, a jó hírnév, személyiségi jog megsértéséig vezethet a skála, de akár a versenyjogi szabályokba ütköző magatartás is megvalósítható. Ha a cég alkalmazottja a jogszabályokba ütköző spam-et küld ki e-mailen keresztül, cégének akár komoly bírsággal vagy kártérítési igénnyel is számolnia kell. De felesleges ecsetelni, hogy milyen komoly kárt okozhatott pl. a Lockheed cégnek az az alkalmazott, aki 60.000 e-mailt automatikus olvasás-visszaigazolás kéréssel küldött ki, és emiatt a cég rendszere összeomlott, és 6 órán keresztül használhatatlanná vált.

A veszélyek elhárítására az információbiztonság technikai hátterének kiépítése mellett szükség van megfelelő szabályozásra is. Szabályozás alatt értsük ide a jogi és a céges szintű szabályozást is. A cikk elején kifejtettek arra utalnak, hogy alakulóban van az állami jogi szabályozásnak is a biztonság irányába történő elmozdulása, ám amíg a hagyományosan lassú jogalkotás lép, a cégnek addig is meg kell tennie mindent a saját szintjén, vagyis meg kell alkotnia a megfelelő szabályokat, amelyek kellően kontrollálják a vállalati kommunikációt. Ám itt rögtön a privacy korlátaiba ütközünk, hiszen a magánszféra körében kiemelt védelmet élvez a levelezés védelme is. A vállalati szabályozásnak tehát az adott jogi lehetőségek mellett meg kell találnia azt a vékony ösvényt, ahol a kecske és káposzta elv alapján a legmagasabb fokú biztonságot jelentő szabályozást alakítja ki úgy, hogy közben nem sérti a szükségesnél nagyobb mértékben a magánszférát.

Életből ellesett példák

A kérdés kényes voltára ugyancsak hozzunk fel egy példát, ahol a Chevron Corp. nevű cég férfi dolgozói vicces kör-emailjükkel, amelyben 25 érvet sorolnak fel amellett, hogy miért jobb a sör, mint a nő, 2,2 millió dollár kárt okoztak vállalatuknak. Ekkora összeg megfizetésére ítélte ugyanis a bíróság a céget, amiért nem akadályozták meg a női dolgozókat sértő e-mail terjesztését.

Az American Management Association, US News & World Report, ePolicy Institute 2001.-ben végzett felmérése szerint a vizsgált cégek

- 81%-a rendelkezik írott szabályzattal, de csak
- 62,8% ellenőrzi ténylegesen az Internet kapcsolatot, csak
- 50,6% követeli meg az alkalmazottak írásbeli nyilatkozatát a szabályzat megismeréséről, és
- 23,9% tart az alkalmazottaknak ePolicy képzést.

A joggyakorlat pedig számos esetben bebizonyította már azt, hogy egy bíróság szemében nem az ér valamit, hogy van-e szabályzat, hanem az, hogy ezt megismertették-e a dolgozóval, és hogy elvárható-e a dolgozótól annak betartása.

Mi tehát a megoldás?

Megfelelő vállalati biztonsági politika!
És melyek a megfelelőség kritériumai?

- Alkalmas – a magyar és európai jognak megfelelő – szabályzatok, amelyek a technikai megoldásokkal harmonizálnak.
- Megfelelő technikai megoldások, szűrők, szoftverek.
- A munkavállalók megismertetése a szabályzatokkal.
- A munkavállalók részére tréningek tartása, hogy megismerjék a biztonsági előírások betartásának módját.
- Szigorú ellenőrzés, számonkérés.

Mindezek elmulasztása esetén ugyanis komoly anyagi veszteséggel számolhat a cég. A cég tulajdonosai pedig ilyen esetben a vezetőség felelősségének kérdését is fel fogják vetni, hiszen annak kötelessége lenne az információbiztonság legmagasabb fokú biztosítása. Abban az esetben pedig, ha a vezetőség nem tett meg mindent ennek érdekében, jó eséllyel számolhatnak a felelősök azzal, hogy a cég által kifizetett nagy kártérítési összegek, vagy a veszteség az ő zsebükből kerül majd visszapótlásra.

> A szabályok megalkotásánál a privacy korlátaiba ütközünk, hiszen a magánszféra körében kiemelt védelmet élvez a levelezés védelmére is.

Benjamin Franklin egykoron azt mondta, hogy hárman akkor tudnak megőrizni egy titkot, ha kettő közülük halott. Nos a világ sokat változott a XVIII. században megfogalmazott gondolatok óta, de a titkok megszerzésének esélye a XXI. században az akkorinál jóval nagyobb. A titokgazdák fecsegése nélkül is megszerezhető információk védelmét ma már nem lehet pisztollyal megoldani, de a lehető legmagasabb biztonságot kellő előrelátással el lehet érni.

Dr. Mayer Erika
erika@drmayer.hu

Windows 2003: Secure By Default!

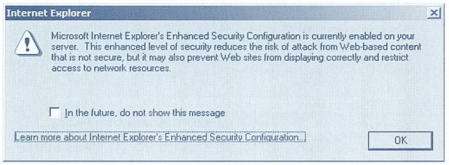
Avagy miért nem működik semmi a frissen telepített W2003-on?

Windows 2003 kísérleteink során rendszeresen a legújabb buildekkel kínlódunk, hogy minél hamarabb észrevegyük, mi minden változik a redmondi fejlesztések során. A változtatások persze nem feltétlenül maradandóak, de az a „trükk", ami látszólag használhatatlanná tette a 3757-es buildet, valószínűleg végleges. Secure By Default!

Miután rendesen végignyomkodtam a 3757-es *(nem publikus)* build telepítőjét *(Next Generation – akik a Next gombot nyomogatják)*, újraindítottam a gépet, és az aktiválási kérdésre igent mondtam *(ezt a béta „terméket" a Interneten keresztül aktiválni kell)*. Nem sikerült.

Néhány gyors mozdulattal ellenőriztem az IP-cím beállításokat, és mindent rendben lévőnek találtam. A szokásos reflexszel megpróbáltam UNC-névvel *(\\galagonya\viragzik)* rácsatlakozni a céges kiszolgálónkra, hogy onnan – mi mást? – Network Monitort telepítsek. Ez sem sikerült, mégpedig azért nem, „mert nincs szolgáltató, aki válaszolhatna a kérésre"!

A következő lépés a support.microsoft.com szokott lenni, így elindítottam az Internet Explorert. Érdekes látvány fogadott, az IE elindult ugyan, de mindjárt használhatatlanná tette önmagát az alábbi modális *(mellé-hiába-kattintasz)* ablakkal:



A gépen engedélyezve van az IE Enhanced Security Configuration. Akarom tudni a részleteket?

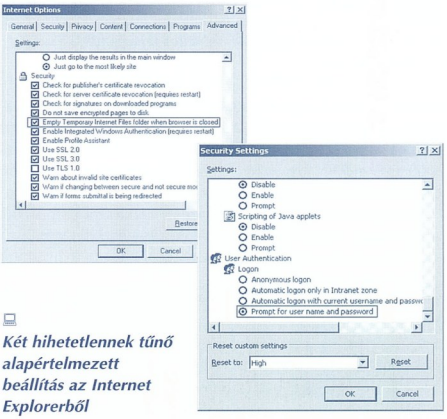
A felkínált „továbbtanulási" lehetőséggel élve elolvastam a mellékelt dokumentációt. Több mint érdekes! A változások lényege az, hogy

- alapértelmezésben az összes hálózati kapcsolat az Internet zónába tartozik;
- az Internet zóna biztonságát jelentősen megerősítve semmilyen autentikációs adat nem megy ki a gépből automatikusan. Windowsos megosztásokra *(UNC)* sem!

A szöveg elolvasása nélkül én bizony álmomban sem gondoltam volna, hogy az Internet Explorer biztonsági beállításai a Windowsos fájlmegosztások elérésére is hatással vannak. Pedig ez a helyzet!

Nézzünk meg két érdekes alapértelmezett beállítást a több tucatból, mert ezek a példák rávilágítanak arra, hogy a maradék negyvennyolc és fél további beállítás is valószínűleg váratlanul fog érni minket. A bal oldali azt mutatja, hogy a helyi gyorsítótárunknak befellegzett. Alapértelmezés szerint a Temporary Internet Files mappa kiürül, amikor bezárjuk a böngészőt! Ez azt jelenti, hogy vége a sütikorszaknak. Vége annak, hogy egy webhely illedelmesen, nevemen szólít, amikor visszatérek hozzá? Lehet, hogy vége, mivel a hétköznapi felhasználók nem fogják gépükön átállítani az alapértelmezett beállítást valami másra!

A Microsoft valószínűleg sejthette ezt, nem véletlen, hogy ASP.NET-ben kukimentes állapotátviteli lehetőség is van.



Két hihetetlennek tűnő alapértelmezett beállítás az Internet Explorerből

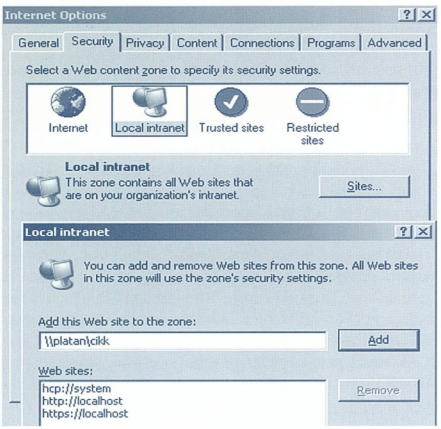
A jobb oldali pedig egy valódi forradalmi újdonságot mutat: végre meg merték tenni, hogy az Internet zóna biztonsága magas *(High)* legyen! Itt ülök a Windows XP-m előtt *(amelyen – rendes júzerként – sosem piszkáltam a biztonsági beállításokat)* és csodálkozva nézem, hogy alapértelmezésben az egész Internet csak közepes *(Medium)* biztonságot érdemelt! Ennek is vége. Most először merték kimondani *(és beállítani)*, hogy az ismeretlenbe ne küldjön az IE semmiféle nevet és jelszót a tudtom nélkül!

A dolog hátulütője, hogy így semmi nem működik. Fájlokat és nyomtatókat nem lehet elérni, tartományhoz nem lehet csatlakozni…

Kiútkeresés

Sok kiút van, ezek közül azokat vizsgáljuk meg, amelyek nem úznek csúfot a Tervezők akaratából *(vagyis például nem foglalkozom azzal a lehetőséggel, hogy az Internet zónát hogyan lehet alacsony [Low] biztonságú fokozatba taszítani).*

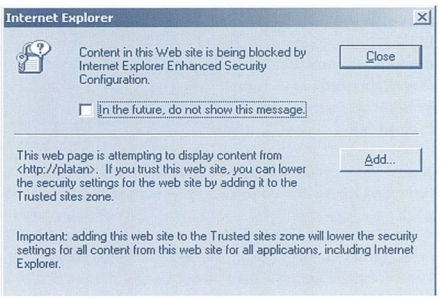
A dokumentáció szerint az a helyes eljárás, ha az elérni kívánt szolgáltatásokat felvesszük a megfelelő zónákba. Tehát a helyi erőforrások elérése érdekében a helyi hálózatot, vagy annak egyes gépeit felvesszük a Local Intranet zónába, így:



Az elérni kívánt hálózati helyeket felvesszük a Local Intranet zónába

Az UNC útvonal megadása után egyébként egy file:// URL kerül be a listába. Ezek után a fájlmegosztás elérése zökkenőmentes – arra az egy gépre, amit felvettem a listába. *(Érdemesebb egész DNS-tartományokat felvenni, mert így egyszerre egy sereg gép bekerül a Local Intranetbe.)*

Ez a megoldás kielégítőnek tűnik, következzen tehát a második „rituálé", a leveleim elolvasása OWA segítségével. Bepötyögöm a szokásos //galagonya/exchange URL-t, és ezt kapom jutalmul:



A weblap tartalmát megette az IE Enhanced Security Configuration

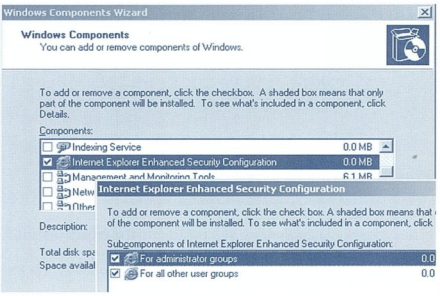
Az üzenet értelme röviden az, hogy a weblap által nyújtott tartalom olyan elemeket *(OWA esetén JScript, de lehetne ActiveX komponens is)* tartalmaz, amelyek a jelenlegi biztonsági beállítások szerint károsnak minősülnek, így – biztos ami biztos – az egész lapot elveszi tőlem. Viszont kaptam egy „Add" nyomógombot, hogy ha megbízom a webhelyben, hozzáadhassam a Trusted Sites zónához.

Ami azért furcsa, mert az imént vettem fel ugyanezt a gépet a Local Intranet zónába. Igen ám, de először is FILE:// URL-lel *(ez meg itt HTTP://)*, másodszor pedig nem a webhelyek *(sites)* közé, pedig az OWA az egy webszájt.

Nem sokat teketóriáztam, felvettem mégegyszer ugyanazt a gépet. Így vettem észre, hogy a windowsupdate.microsoft.com gyárilag bent van a megbízható webhelyek listájában! Így ha a terméket aktivizálni nem is, de peccseket letölteni alapból lehet ;) Hopp, az oca.microsoft.com is bent van *(OCA = Online Crashdump Analysis)*! Kék halál dumpfájlokat is felküldhetünk! A Local Intranet és a Trusted Sites közötti megannyi apró beállítási eltérést a legelső figyelmeztető ablakból nyíló HTML-oldalon olvashatjuk el részletesen.

Kompatibilitás, a biztonság halála

Ha valaki olyan környezetben dolgozik, ahol még mindig vannak Windows 95-ök és hasonló őslények, sajnos a zónákba pakolással sem fog ereményt elérni, mert *(a dokumentumból ki nem derülő változtatás miatt)* az IE Enhanced Security nem kompatibilis, csak az NT4 SP3-nál ifjabb rendszerekkel. Az együttélés kivitelezhetősége érdekében létezik még egy drasztikus módszer, ez pedig az IE Enhanced Security eltávolítása a rendszerből. Ezt a Windows-komponensek szokásos telepítési helyén, az Add/Remove programsban tehetjük meg. Érdemes megfigyelni, hogy külön eltávolítható a júzerek és a rendszergazdák számára. Ez némi átgondoltságra utal, mi is legyünk megfontoltak, és – ha valóban nincs más módszer – csak a felhasználóknál lazítsunk a hurok szorításán. Az Administrator csak hadd szenvedjen!



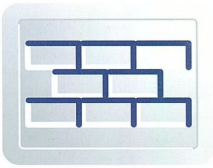
Az IE Enhanced Security eltávolítása

A legfontosabbat majd' el felejtettem: a termékaktiváláshoz egy pillanatra „Low"-ba kell tenni az Internet zónát, mivel a varázslóból nem derül ki, pontosan melyik URL-t kellene felvenni a Trusted Sites-be. Nesze neked biztonság. De hát ez még béta.

Fóti Marcell
marcellf@netacademia.net
A szerző a NetAcademia vezető oktatója
MCSE, MCT, MCDBA, MZ/X

Kapcsolódó NetAcademia tanfolyamok:
Windows 2003 Expert Workshop

A Microsoft operációs rendszerek biztonsági tényezői



III. rész – Gyenge pontok, megoldások

Még felsorolni sem könnyű, hogy hol és mennyi biztonsági szolgáltatást, funkciót halmoztak fel a Microsoft szoftvermérnökei a cég zászlóshajójában. Ezzel együtt nem szabad elhallgatnunk, hogy továbbra is vannak gyenge pontok a rendszerben, amelyek folyamatos törődést igényelnek. Dióhéjban áttekintjük ezeket, valamint a problémák leküzdésének eszközeit is.

Nagy forráskód

A Windows 2000 megjelenése előtt több statisztikai jellegű adat is napvilágot látott. Az egyik ilyen adat arról tudósított, hogy a Windows NT 4.0-hoz képest a rendszer 80%-át újraírták. Ezért – bár látszólag csak egy újabb verzióról van szó – valójában egy új termék született. Néhány tanácsadó cég óva intette a felhasználókat a korai alkalmazástól, mondván, egy „1.0-ás verzió" rengeteg gyermekbetegséggel és biztonsági hiányossággal küzd. Szárnyra kapott olyan adat is, miszerint a Windows 2000 legalább 45-50.000 hibát rejt magában. A számítások alapja az a statisztikai megfigyelés volt, miszerint 1000 programsorra átlagosan 2-3 hiba jut. A Windows 2000 pedig 15.000.000 sorból áll. Köztudott továbbá, hogy egy szoftvertermék sohasem hibátlan állapotban kerül a piacra, hanem akkor, amikor a hibák előfordulási valószínűsége egy bizonyos szint alá csökken.

A Microsoft, – amellett, hogy cáfolta a híreket – nagyon komoly gondot fordított arra, hogy a lehető legkevesebb hibával – főleg biztonsági hibával – rendelkező terméket bocsásson ki. A megjelent termékben a tényleges hibák száma ugyan messze nem érte el a fent említett negatív várakozásokat, és lényegesen kevesebb volt, mint a korábbi NT verzióban, az elmúlt két évben mégis több száz szoftverfoltot és három nagyobb javítócsomagot kellett kiadnia a gyártónak. A lényeges javulás ellenére ez mégis túl sok volt ahhoz, hogy érdemben javítson a cég termékeinek megítélésén. Úgy tűnik, hogy a hibák statisztikai tényezőként való kezelése az igazi probléma. Amennyiben ez az elv fennmarad, az egyre nagyobb forráskód statisztikailag növekvő veszélyt jelent majd.

Kényelem kontra biztonság kontra kompatibilitás

A biztonsági tényezők alkalmazásakor sokszor felhívják a szakemberek a figyelmet, hogy a biztonság növelése gyakran a kényelem és a használhatóság rovására történhet. Biztonságosabb a jelszóval történő azonosítás, mintha ilyen nem történik, de meg kell jegyezni a titkos jelsort *(kényelem csökkenése)*, és ha tévesen adjuk meg a minket azonosító adatokat, a rendszer kizárhat minket *(a használhatóság korlátozódik)*.

Az üzemeltetési tapasztalatok azonban azt mutatják, hogy egy harmadik dimenzió is befolyásolja a biztonsági funkciók bevezetését, ez pedig a kompatibilitás. A rendszer egyes elemeit más-más csoportok fejlesztik, akik eltérő sebességgel képesek a központi biztonsági igényekhez alkalmazkodni. A fürtszolgáltatás például nem képes a Kerberos hitelesítésre, így akik ezt a szolgáltatást igénybe veszik, nem tudnak tisztán Kerberos rendszert bevezetni. Sőt, a helyzet még ennél is rosszabb volt, mert a 3-as javítócsomag megjelenése előtt a fürtszolgáltatás NTLM hitelesítést használt, tehát még az NTLMv2-es, erősebb szabványt sem lehetett alkalmazni. Hasonló probléma a DNS zónák biztonságos frissítése és a gyermektartományok létrehozása, amiről korábban már szóltunk. Ugyancsak kompatibilitási gond a NAT és az RRAS közös szerveren való használata. A NAT megvalósítása során ugyanis olyan tervezési döntés született, mely szerint a betárcsázáskor használt interfész külső „lábnak" minősül, így a betárcsázók nem képesek a NAT szolgáltatáson keresztül elérni külső IP címtartományokat. A tervezési problémát a Windows .NET kiszolgáló orvosolja ugyan, de a hiba jól jelzi a különböző programozócsoportok közötti összhang hiányát.

A Microsoft biztonságot növelő tevékenységei

Megértve azt, hogy a biztonság nem egy állapot, hanem inkább egy folyamat, a világelső szoftvergyártó számos olyan, teljesen ingyenes szolgáltatást kínál a termékeihez, amelyek hatékonyan segíthetik az üzemeltetőket a Windows rendszerek biztonságosabbá tételében. Vizsgáljuk meg, mint tesz a Microsoft a rendszerein „kívül és túl", hogy az ügyfelei elégedettebbek legyenek, és nagyobb biztonságban érezzék magukat.

Tervezés

Azok a hibák a legveszélyesebbek, amelyeket a tervezéskor követnek el, mert ezek alapvetően meghatározzák a szoftver tulajdonságait, és nehezen javíthatók. A Microsoft úgy igyekszik ez ellen védekezni, hogy a szoftvertervezés eljárásrendjébe beillesztette a biztonsági funkciók implementálását. Számos esetben nem a spanyolviasz feltalálásával foglalkozik a cég, hanem átveszi az Internetes közösség által fejlesztett szabványokat *(Kerberos, IPSec, L2TP stb.)*, amelyek többnyire a biztonsági kérdéseket is megfelelő módon kezelik. A sok saját szabvány erőltetése helyett így egy átláthatóbb, és összekapcsolható architektúra jöhet létre.

A Windows 2000 fejlesztőcsapat átszervezése

A Windows NT-t kezdetben 4-5 ember fejlesztette. A kis létszám és a kicsi kód nagyon gyors fejlesztési periódust tett lehetővé. A Windows 2000-et viszont több ezer ember készítette, az ő munkájuk megszervezése egész más elveket kívánt, mint amit korábban a cég alkalmazott. A fejlesztő szervezetet többször átalakították úgy, hogy egyrészt a fejlesztési periódus sebessége ne csökkenjen, másrészt a szervezet működése hatékonyan se-

gítse a hibás kód elkerülését vagy a hibák gyors észlelését és kijavítását. A kezdeti állapotokról, majd a változásokról Mark Lucovsky előadásában lehet bővebben olvasni a **[windev]** számmal jelölt címen.

.NET

Ha nem is kizárólag a biztonságos rendszerek fejlesztése miatt, de részben ilyen szándékkal is készült a .NET keretrendszer. A .NET mindenekelőtt intelligens XML webszolgáltatások platformja kíván lenni. Azonban biztonsági szempontból is sok lehetőséget kínál a felhasználók és a fejlesztők számára. Ezek a következők:

- **Szerepalapú biztonsági beállítások.** A Microsoft szándékai szerint .NET környezetben a felhatalmazás *(authorization)* és a hitelesítés *(authentication)* a felhasználók személyazonosságán és szerepén alapul. A hitelesítés magában foglalja a megfelelő adatok vizsgálatát *(pl. felhasználónév és jelszó)*, és annak megállapítását, hogy a felhasználó valóban az-e, akinek állítja magát. Miután a hitelesítés megtörtént, a .NET alkalmazás megállapítja, hogy a felhasználó milyen szerepkörrel rendelkezik *(pl.: pénzügyi igazgató, számviteli előadó stb.)*, és milyen műveletek megengedettek a számára.
  A .NET keretrendszer minden általánosan ismert hitelesítési módszert támogat, beleértve az alap *(basic)*, a kivonatolt *(digest)*, az NTLM, a Kerberos és az SSL/TLS ügyféltanúsítványokat. Lehetőség van a Microsoft .NET Passport hitelesítési módszer használatára is. A kiválasztott módszertől függetlenül a fejlesztők számára konzisztens programozási modell áll rendelkezésre, hogy a biztonsági funkciókat beépítsék alkalmazásaikba.
- **Evidence-based és code access security.** A rendszergazdák előírhatják, hogy mely erőforrásokat érhet el egy bizonyos kód attól függően, hogy honnan származik *(egy adott könyvtárból, az Internetről vagy Intranetről)* vagy milyen jellemzői vannak, például rendelkezik-e egy bizonyos digitális aláírással stb. Így az adatok jobban védhetők az ismeretlen vagy csak részben megbízható kóddal szemben. *(Erről Soczó Zsolt részletes cikket írt a 2002. decemberi számban.)*
- **Kriptográfia.** A .NET keretrendszer függvényeket tartalmaz a titkosításhoz, a digitális aláírások, jegyek és véletlen számok előállításához. Támogatott algoritmus a DES, a 3DES, az RC2, az RSA, a DSA és az XML digitális aláírás specifikációk.

Online biztonsági szolgáltatások

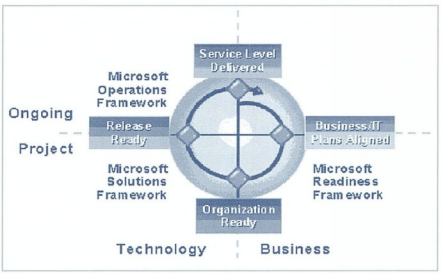
A Microsoft sok éve üzemeltet a saját honlapján belül egy biztonsági oldalt, amely megfelelő kiindulópont a felhasználók, rendszergazdák, fejlesztők és az üzleti döntéshozók számára, hogy megtalálják a rendszerük biztonságosságát technikai értelemben növelő erőforrásokat, eszközöket. Tulajdonképpen egy online szolgáltatás gyújtópontjába kerülünk, ahonnan az igényeink és helyzetünk szerint kerülhetünk más, konkrét problémákon segítő lapokra.

Biztonsági hírlevél *(Security bulletins)*.

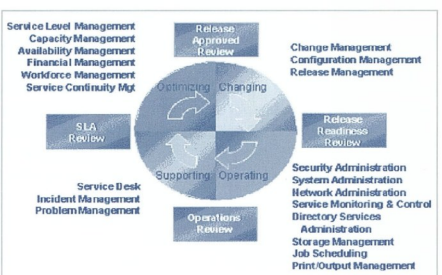
A Microsoft biztonsági hírleveleken keresztül tájékoztatja ügyfeleit a feltárt biztonsági hiányosságokról. A hírlevélnek kódja van, amely a kiadás évét és egy sorszámot tartalmaz. Az MS02-071 azt jelzi, hogy a 2002-ben felfedezett 71-dik biztonsági hibáról van szó. A hírlevelek szerkezete kötött. Tartalmazza a biztonsági hiba leírását, a hiba el nem hárításából adódó lehetséges következményeket. Meghatározza, hogy mely termékeket érinti a hiba, és azt, hogyan lehet ellenőrizni, hogy érintettek vagyunk-e. Ezután a hírlevél instrukciókat sorol, amelyekkel a hiba megszüntethető. Ez lehet egy rendszerparaméter módosítása, egy biztonsági folt letöltése és telepítése vagy a Windows Update weboldalra való ellátogatás.
A biztonsági hírlevél kiadásával egyidejűleg a Microsoft megjelentet egy tudásbázis-cikket is, amely szintén tartalmazza a fent már leírtakat, és esetleg további technikai részletekkel szolgál. Ezt a cikket frissítik, miután a javítás bekerül egy nagyobb javítócsomagba. A tudásbázis cikknek is van egy kódja *(pl.: Q328310)*, amelyre más cikkek hivatkozhatnak. A cikket keresőszavakkal és kifejezésekkel látják el, hogy később csak a jelenség megfogalmazása után is rá lehessen találni a cikkre, mint megoldásra.
A hírlevelet természetesen elő lehet jegyeztetni, így a biztonsági rés felfedezése után azonnal értesülhetünk a tényekről, és tanácsokat kapunk, hogy mit tegyünk védelmünk érdekében.

Módszertanok és biztonsági eljárás-ajánlások

Külön említést érdemel, hogy a redmondi szoftvercég a saját termékeinek bevezetéséhez, üzemeltetéséhez és karbantartásához saját *(a legszélesebb körben elfogadott brit ITIL módszertanra épülő)* módszertant fejlesztett ki, amely három keretrendszerből áll.



A Microsoft Readiness Framework *(MRF)* keretrendszerben lehet felkészíteni a szervezetet valamely technológia befogadására.
A Microsoft Solution Framework *(MSF)* a tényleges bevezetéshez nyújt segítséget. A Microsoft Operation Framework *(MOF)* a mindennapi működéshez ad szervezési támogatást.



Ez utóbbi keretrendszer önmaga is négy részre oszlik: a váltás *(változás)*, működtetés, támogatás és optimalizálás negyedekre. Az egyes negyedek már olyan konkrét szolgáltatásfunkciók-

ra bomlanak, mint például a konfigurációkezelés, hálózatadminisztráció stb.
Miután a rendszer életciklusa szerint megtörtént az egyes funkciók meghatározása, a modell felállít egy olyan szervezetet, amelyben funkciók szerint vannak elkülönítve a feladatok. Ezt team-szerep fürtnek *(team-role cluster)* nevezi a modell. Hat szerepet különböztet meg: kibocsátás, infrastruktúra, támogatás, működtetés, kapcsolattartás és biztonság.

Minden egyes szerephez felelősségi köröket lehet rendelni. Miután e modell szereplőit és feladatait definiáltuk, el kell végezni egy megfeleltetést. Az egyes szerepek ugyanis a MOF-modellben több negyedhez is tartoznak. Könnyű belátni, hogy a biztonsági feladatokat mind az üzemeltetés, mind az optimalizálás során el kell látni. A modell tehát hozzárendeli a szerepeket a működési ciklus negyedeihez és a szolgáltatásfunkciókhoz. Ám ennél többet is megtesznek a modell készítői. A MOF részletezi azokat a szerepeket, amelyeket más szerepekkel együtt be lehet tölteni, és azt is meghatározza, hogy melyek az egymást kizáró szerepek. A kisebb szervezetek, amelyek kénytelenek összevonni bizonyos felelősségi köröket, világos eligazítást kapnak, hogy mely feladatokat lehet azonos munkavállalóra bízni, és melyeket nem.

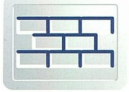
A keretrendszer segít megtervezni a támogató szervezet felépítését, a felelősség és munkamegosztás optimális kialakítását. Mindehhez azonban egy újabb dimenziót is figyelembe kell venni, mégpedig a termékeket. A termékek tulajdonképpen eloszlanak az egyes negyedekben. A Windows 2000-nek vannak változáskezeléssel kapcsolatos szolgáltatásai és biztonsággal kapcsolatos funkciói mind a bevezetés, mind az üzemeltetés során. A Microsoft az üzemeltető és a biztonsági szerepekre gondolva számos szoftveréhez megjelentetett úgynevezett „Operations Guide"-ot, vagyis üzemeltetési ajánlásokat, amelyek a termékekből fakadó, de a keretrendszer által definiált feladatokat összegzik. Ilyen például az „Exchange 2000 Server Operations Guide" vagy a „Windows Operation Guide". Emellett a biztonsági szerepekhez elkészült már a „Security Operations Guide for Exchange 2000" és a „Security Operations Guide for Windows 2000 Server". Ha egy speciális Microsoft-termék esetén a felhasználó nem talál ilyen átfogó üzemeltetési ajánlást, még mindig lehet támaszkodni az ún. legjobb megoldásokhoz *(Best Practices)*, amelyek a legnagyobb felhasználóknál szerzett tapasztalatokat foglalják össze.
A működési keretrendszer, a szerepek definiálása és a termékekhez tartozó üzemeltetési ajánlások vagy legjobb megoldások együttesen gyakorlatilag „hézagmentes" megoldáshoz vezetnek, az üzemeltetés *(és így a biztonság)* minden aspektusát kézben lehet tartani.

Biztonsági eszközök

Miután létezik a módszer, és megvan a lehetőség, hogy az optimális üzemeltető szervezetet kialakítsuk, továbbá instrukcióink is vannak, hogy milyen módon tegyük biztonságosabbá a rendszerünket, el kell végezni a technikai biztonsági elemzést. A felkészülés után tehát a „légy biztonságosabb" *(Get Secure)* felszólításnak tehetünk eleget. A Microsoft *(jelenleg még)* sokféle eszközzel segíti a rendszer biztonságosabbá válását. Ezek között vannak dokumentumok és segédprogramok. A dokumentumok egy-egy termékre vonatkozóan tesznek biztonsági beállítási javaslatokat, baseline-okat. A Microsoft korábbi termékeinél a könnyű üzemeltethetőség kedvéért minimális biztonsági alapszintet határozott meg. Amikor tehát a termék a piacra került, képes lett volna ugyan védekezni bizonyos támadások ellen, de ezek a védelmi képességek ki voltak kapcsolva, mert az üzemeltetők többsége semmilyen paramétert sem állított át a gyári beállításokhoz képest. A szoftvercég megértette álláspontja helytelenségét, és a meglévő termékek esetén a biztonsági beállítási javaslatok kiadásával próbálta ellenállóbbá tenni szoftvereit a támadásokkal szemben.
A rendszergazdákat azonban nem csak írásokkal segítik. A következő felsorolás a legfontosabb segédprogramokat sorolja fel, amelyekkel biztonságosabbá tehetők a Windows rendszerek.

IIS Security Lockdown Tool. Egy webszerver akkor tekinthető biztonságosnak, ha csak a szükséges funkciók működnek a rendszeren. A „szükséges" meghatározását segíti az IIS Lockdown eszköz, amely jól konfigurálható módon képes lezárni azokat a funkciókat, amelyek nem használtak, és amelyek potenciális veszélyt, támadási felületet jelenthetnek. Az eszköz használata azért veszélyes, mert minden elővigyázatosság ellenére képes használhatatlanná tenni a már működő webes alkalmazásokat. A végleges beállítás előtt ezért célszerű egy tesztrendszeren elvégezni a szerver lezárását.

Security BaseLine Analyser. Egy grafikus felületen keresztül helyi és távoli kiszolgálók biztonsági hiányosságait igyekszik felderíteni a program. Egy Internetről letöltött XML-állomány képezi a vizsgálat alapját, amely az aktuális hotfixek listáját és egyéb biztonsági ismereteket tartalmaz. A kiszolgáló fájljait összehasonlítva az XLM-adatbázis tartalmával megmutatja, hogy milyen javítások hiányoznak a rendszerből. A javítófoltok hiánya mellett bizonyos beállításokat is ellenőriz a program, és ha azokat nem találja kellően biztonságosnak, jelzi az elké-



szült jelentésben. Az ellenőrzés után részletes leírást kapunk arról, hogy a program mit ellenőrzött, mi lett az eredmény, és hogyan lehet a biztonsági rést eltávolítani.

HFNetCheck. Az előző segédprogram parancssori változata. Windows NT 4.0 és Windows 2000 operációs rendszereket vizsgál, továbbá IIS4, IIS5, SQL7, SQL2000 valamint az Internet Explorer 5 és 6 alkalmazások biztonsági hiányosságait is képes megállapítani. Ugyanúgy egy Internetről letöltött XML-állományt használ. A grafikus felülettel szemben ezt az eszközt be lehet illeszteni egy olyan programkörnyezetbe, amely automatikusan végigfésüli a hálózat valamennyi Windows számítógépét. A vizsgálatról ugyanúgy készülhet jelentés.

Outlook E-Mail Security Update. Egy speciális javítócsomagról van szó, amely a Microsoft egy levelezőprogramjának, az Outlooknak a biztonsági hibáit javítja. Az eszköz célja elsősorban a kártékony csatolt állományokat tartalmazó levelek szűrése és a további fertőzések megakadályozása.

URLScan Security Tool. Egy olyan ISAPI szűrőről van szó, amely elemzi az Internet szervernek küldött kéréseket, és megpróbálja kiszűrni a potenciálisan támadó jellegűeket. Az eszközt egybeépítették az IISLockdownnal, de külön is használható.

Event Comb. Egy többszálú alkalmazás, amely sok távoli kiszolgáló eseménynaplójában képes egyszerre eseményeket keresni. A program lehetővé teszi, hogy egy vagy több eseményazonosító vagy egy eseményazonosító intervallum felhasználásával keressünk. A keresés szűkíthető csak egy bizonyos eseménynaplóra *(pl. csak a rendszernaplóra)*, egy bizonyos eseménytípusra *(pl. csak figyelmeztetésre)*, eseményforrásra, stb.

QChain. Ezzel a segédeszközzel elérhető, hogy a javítófoltokat sorban, egymás után telepíthessük, és csak a legutolsó után kelljen a rendszert újraindítani. Ellenkező esetben minden javítófolt után le kellene állítani a rendszerünket.

Képernyővédő a biztonságról.

Egy forráskóddal együtt letölthető képernyővédő, amely a felhasználók és rendszergazdák számára 10-10 megszívlelendő mondatot tartalmaz. A vállalati hálózatban való szétterítése növelheti a felhasználók és az üzemeltetők biztonság-tudatosságát.

Patch menedzsment

Ha már elértünk egy biztonsági szintet, azt meg is kell tartanunk. A Microsoft ezt a fázist a „maradj biztonságban" *(Stay secure)* jelszóval illeti. Három ingyenes eszköz segíti a Windows felhasználókat.

Windows Update website. Online, bárki által használható szolgáltatás. A webhelyet elérve néhány telepítendő ActiveX-komponens kerül a gépünkre. Ezek átvizsgálják a rendszerünket, és megállapítják, hogy milyen alkalmazásokat telepítettünk, valamint, hogy a Microsoft alkalmazások esetén milyen javítófoltok vagy csomagok hiányoznak. A vizsgálat eredményeképp kapunk egy listát, amelyben a webhely javaslatot teszt, milyen csomagokat töltsünk le feltétlenül. Emellett lehetőségünk van nem kritikus alkalmazások frissebb verzióira is áttérni *(pl. MSN Messenger)*. Az alkalmazások mellett az eszközmeghajtók átvizsgálását is elvégzi a program, és ha van frissebb meghajtó, akkor azok telepítését is javasolja.
A szolgáltatás megjelenésekor vegyes érzelmeket váltott ki. Voltak, akik egyszerűen kémprogramnak titulálták a megoldást, mondván, nem lehet tudni, hogy a cég mit tesz a megszerzett adatokkal. Voltak technikai kifogások is, mert a webhely csak a jól letesztelt javítófoltokat tartalmazza, vagyis nem teljesen naprakész. Sokan azt kifogásolták, hogy kritikus biztonsági hibának minősített az alkalmazás olyan szituációt, amikor valaki nem a legfrissebb böngészővel rendelkezett. A nagyvállalatok azt kifogásolták, hogy a webhely feltételezi a rendszeradminisztrátori jogkört a helyi gépen, különben a kritikus javítócsomagok és az eszközkezelő frissítések meghiúsulnak. Végül az ellenkezés hullámai elcsitultak, és mára havonta mintegy 300 millió letöltés történik csak erről a helyről. Ha más nem, ez világosan mutatja, hogy a Microsoft fontos lépést tett a végfelhasználók biztonságának növelése érdekében.

Office Update. A Windows Update az operációs rendszer és kiegészítőinek frissítését végzi el. Az Office alkalmazások egy hasonló webhellyel rendelkeznek, és az Office 2000 illetve Office XP csomagok esetén tesznek javaslatot a frissítések végrehajtására.

Software Update Services. A nagyvállalati igények kielégítésére született meg a Windows Update webhely „letölthető" változata. A rendszerüzemeltetők kialakíthatnak egy olyan intranet lapot vagy hierarchikus rendszert, amelyről az ügyfélgépek automatikusan letölthetik a javítócsomagokat és eszközfrissítéseket. A megoldás egyik előnye, hogy sávszélességet lehet megtakarítani azzal, hogy minden gép csak a hozzá közel elhelyezkedő szerverhez fordul a frissítésekért, és nem közvetlenül az Internetre.

Megvásárolható alkalmazások

Bár mindeddig csak a Windows 2000-ről és a hozzá kapcsoló ingyenes kiegészítőkről volt szó, mégis érdemes néhány mondat erejéig megemlíteni a Microsoft azon termékeit, amelyeket ugyan külön kell megvásárolni, mégis hatékonyan segíthetik a biztonságosabb Windows környezetek kialakítását.

Systems Management Server *(SMS)* – Eredetileg a változáskezelés és a felhasználó-támogatás eszközeként definiálták ezt a terméket, de a nemrég megjelent „Feature Pack" kiegészítéssel a ma ismert egyik leghatékonyabb folt- és javítócsomag terítő rendszerré vált, megtartva természetesen az eredeti funkciókat is.

Microsoft Operation Manager *(MOM)* – A hatékony, tudásbázis alapú szerver-menedzsment funkció mellett rendelkezik eseménynapló-konszolidátor képességekkel is. A naplóesemények elemzésére és automatikus beavatkozásra képes termékről van szó. Kiegészítő modulokkal *(NetIQ)* hatékony biztonsági audit rendszer építhető vele.

Internet Security and Acceleration Server *(ISA)* – A Microsoft intelligens, modulokkal bővíthető, címtárral integrálható tűzfalmegoldása. A dinamikus csomagszűrés és címfordítás mellett ellátható alkalmazásszűrőkkel, behatolásérzékelő modulokkal. A védelmi funkciók mellett gyorsítótár szerepet is képes játszani egyedül, többen *(redundánsan)* és hierarchiában.

Megbízható számítástechnika kezdeményezés *(Trustworthy Computing)*

A szoftveróriás igen jelentős mennyiségű pénzt költ olyan kutatásokra, amelyek a hibátlan szoftverkód elérését célozzák. Sőt, a cél ennél is ambiciózusabb: a hardver- és szoftvergyártókat, integrátorokat és megoldásszállítókat összefogva „kellően megbízható" informatikai rendszereket szeretne készíteni a Microsoft. A „kellően megbízható"-t úgy kell elképzelni, mint az elektromosságot. Ha nem vigyázunk, megcsaphat minket az áram, de a megfelelő *(nem túl sok)* szabály ismerete mellett az elektromos eszközök megbízhatóan működnek, és segítenek a mindennapjainkban. Sajnos korunk informatikai infrastruktúrája és alkalmazásai még nem tartanak itt. Vannak ugyan nagy rendelkezésre állású rendszerek *(pl. repülőgépjegyfoglalás)*, azonban óriási, centralizált felügyelő apparátussal, jórészt elszeparált, kizárólag egy célra fenntartott eszközökön működnek. Heterogén környezetben, decentralizált felügyelet mellett, nagyszámú, különböző gyártótól származó eszköz esetén még nem vagyunk képesek olyan „egyszerű" kívánságokat teljesíteni, mint például: „a szolgáltatás a szükséges helyen és időben álljon rendelkezésre és működjön".
A megbízható számítástechnikáról írt tanulmány *(white paper)* három perspektívát vázol fel, amelyben a megbízhatóság, bizalom értelmezhető **[trustgoals]**. Az első a „célok" *(goals)*. A célok a felhasználók szemszögéből a következőképp értelmezik a bizalmat:

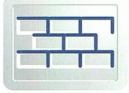
Célok	Ami alapján a fogyasztó úgy dönt, hogy a rendszerben megbízik
Biztonság *(Security)*	Az ügyfél *(felhasználó)* elvárhatja, hogy a rendszerek ellenállóak legyenek a támadásokkal szemben, és a rendszer, valamint a rábízott adatok bizalmassága, integritása és rendelkezésre állása biztosított legyen.
Bizalmasság *(Privacy)*	Az ügyfél *(felhasználó)* képes ellenőrzést gyakorolni a saját adatai felett, amelyek csak megfelelő információs elvek mellett használhatóak.
Megbízhatóság *(Reliability)*	Az ügyfél *(felhasználó)* rábízhatja magát a termékre, mert az elvégzi a feladatát, ha szükséges.
Az üzletmenet folytonossága *(Business Integrity)*	A termék szállítója felelősen viselkedik probléma esetén

A második dimenzió az eszközök *(means)*, amelyekkel a szállítóknak élniük kell ahhoz, hogy a célokat meg lehessen valósítani. Ha a célok azt vázolják, hogy mit kell megvalósítani, akkor az eszközök azt definiálják, hogy a célok hogyan érhetők el.

Eszközök	Üzleti és mérnöki teendők, amelyek lehetővé teszik a rendszerszállítók számára a célok elérését
Tervezett biztonság *(Secure by Design)* Alapértelmezett biztonság *(Secure by Default)* Biztonságos üzembeállítás *(Secure in Deployment)*	Azok a lépések, amelyek biztosítják az adatok és a rendszerek bizalmasságát, integritását és rendelkezésre állását a szoftverfejlesztés minden fázisában, a tervezéstől a telepítésen át a karbantartásig.
Korrekt információs elvek	A végfelhasználók adatait sohasem gyűjtik és osztják meg másokkal a tulajdonos engedélye nélkül. A magánélethez való jogot az adatok korrekt információs elvek szerinti kezelésével tiszteletben tartják.
Rendelkezésre állás *(Availability)*	A rendszer rendelkezésre áll, és használható, ha szükség van rá.
Kezelhetőség *(Manageability)*	A rendszert könnyű telepíteni és kezelni a méretéhez és komplexitásához mérten.
Alaposság *(Accuracy)*	A rendszer pontosan látja el a funkcióit. A számítások eredménye hibamentes, az adatok védettek a sérülésektől és megsemmisüléstől.
Használhatóság *(Usability)*	A szoftver könnyen használható, és megfelel a felhasználók igényeinek.
Készségesség *(Responsiveness)*	A vállalat elfogadja a problémákkal kapcsolatos felelősséget, és megfelelő intézkedéseket tesz a hibák kijavítására. Az ügyfél megfelelő segítséget kap a termék tervezéséhez, telepítéséhez és üzemeltetéséhez.
Áttekinthetőség *(Transparency)*	A szállító folyamatainak áttekinthetővé kell válniuk az ügyfél számára. Az ügyfél tudja, hogy a problémája *(megrendelése, tranzakciója stb.)* milyen státuszban van a szállítónál.

A harmadik perspektíva a végrehajtás *(execution)*. A végrehajtás során egy szállító a folyamatait úgy szervezi, hogy a megbízható számítástechnika kritériumainak megfelelő rendszereket készítsen. Három aspektust kell megemlíteni a megvalósítás elemzésekor: a szándék *(intents)* vállalati szabályokat, elvárásokat jelent a termék-életciklusra vonatkozóan. A megvalósítás *(implementation)* az az üzleti eljárás, amely a szándékok szerint jár el. Végül a bizonyosság *(evidence)* olyan mechanizmus, amelynek segítségével meg lehet győződni arról, hogy a megvalósítás a szándék szerint történt.

A megbízható számítástechnikát legalább két időmenzióban kell kezelnünk. Az első időmenzió a jelen. Már most szükség



van a célok elérésére vagy legalábbis megközelítésére, de a mai rendszereket nem a fenti elveket figyelembe véve készítették. Ez azonban nem azt jelenti, hogy nem lehet semmit tenni. A meglévő eszközök és szoftverek javítása a „megbízható számítástechnika" szellemében és szempontjai szerint sokat segíthet a ma szakembereinek és felhasználóinak, hogy közelebb kerüljenek céljaikhoz. A mai rendszerek javítása azonban egyúttal tapasztalatszerzés is. Megvizsgálva a jelenlegi eszközöket megállapíthatjuk, hogy mit vagy mit nem javíthatunk, az információkat pedig visszacsatolhatjuk a következő generációs eszközök fejlesztési ciklusába. Egy figyelemreméltó akció volt a jelen idődimenzió kezelésére, amikor a Microsoft 2002 elején valamennyi Windows mérnökét – több mint 8500 szakembert – küldött biztonsági technológia továbbképzésre, felfüggesztve a fejlesztéseket is **[backtoschool]**. A monitorok elé visszatért mérnökök azután átvizsgálták a teljes rendszerkódot, és jóval megbízhatóbbá tették a Windows 2000-et a harmadik javítócsomag segítségével. Hasonló akciók történtek a .NET keretrendszer, az Office és a Visual Studio termékcsaládoknál is.

A másik idődimenzió a jövő, a Microsoft szerint egy évtized, vagy még annál is több. A szoftveróriás célja olyan rendszerek építése, amelyek minden ízükben biztonságosak. Olyan biztonságot vizionál a cég, amely magától értetődő és megkérdőjelezhetetlen. Olyan biztonságot, amely láthatatlan, de létező és működő természetesség. A elérendő célokat fázisokra bontották. A következő fázisban „megbízhatónak készült" *(Designed for Trust)* termékeket készül kiadni a redmondi vállalat. A Windows XP SP1-gyel és a .NET Server megjelenésével egy robosztusabb biztonsági keretrendszert szeretne adni a Microsoft a felhasználóknak. A következő fázis a „megbízhatónak tervezett" *(Architected for Trust)* szoftverek megjelenése lesz, de ehhez már új generációs hardvereszközökre is szükség lesz. A projekt neve Palladium. Ennél távolabbra egyelőre a Microsoft sem lát.

A „megbízható számítástechnika" egy olyan kezdeményezés, amely túlmutat a biztonsági kérdéseken, persze magában foglalva azokat. Túlmutat a Microsoft termékein is, és egy egész iparág problémáját öleli fel. Végetül túlmutat jelen cikk keretein is. Megvalósításával szemben azonban – ismerve a jelenkor technológiai korlátait – bizonyos szkepticizmus tapasztalható. Személyesen azt gondolom, hogy akkor tekinthetjük sikeresnek a fenti kezdeményezést, ha a végfelhasználói szerződésből kikerülnek azok a pontok, amelyek a felelősség elhárítását szolgálják arra az esetre, ha a szoftver nem az elvárt módon viselkedik, nem áll rendelkezésre, és ebből a felhasználónak kára származhat.

Zárszó

A PC megjelenésétől a megbízható számítástechnikáig igyekeztem megrajzolni azt az utat, amelyet a világ legnagyobb szoftvergyártó cége bejárt, s a múlt idő talán nem is helyénvaló. A Microsoft a múltban több hibát is elkövetett, rontva ezáltal a saját hitelét, egyúttal a teljes iparágba vetett bizalmat is kikezdve. A hibák mellett azonban gyakran úttörő munkát végzett. Versenytársaival szemben, akik többnyire monolit rendszerben gondolkodtak *(azonos hardvergyártó, azonos szoftvergyártó, azonos alkalmazásszállító)*, a heterogén, többszereplős, elosztott architektúra mellé tette le a voksát. Ma már tudjuk, hogy az információs rendszerek győztese, az Internet is ilyen. A biztonsági kihívások pedig épp az ehhez hasonló rendszereknél a legkomolyabbak. A Microsoft meg kíván felelni a felhasználók elvárásainak, és súlyához mérten veszi ki a részét az informatikai szolgáltatások elterjesztésében és megbízhatóságának növelésében.

Lepenye Tamás, MCSE 2000
lepenyet@mal.hu

Tanúsítványkiadók a Windowsban

A Windows 2000 és Windows 2003 Server CA szolgáltatása

A tanúsítványkiadó szolgáltatás fontos, központi szerepet játszik a Windows 2000/2003 nyílt kulcsú infrastruktúrájában. Cikkünkben bemutatjuk a Certification Services szolgáltatását, majd természetesen kitérünk a Windows 2003 újdonságaira is.

A tanúsítványkiadó szolgáltatás feladata *(PKI tanfolyam 5 percben)*

A CA *(Certification Authority – „hitelesítő szervezet")* feladata az, hogy nyílt kulcsú *(PKI)* tanúsítványokat adjon ki, miután ellenőrizte a tanúsítványt kérő személy valódi kilétét. A CA által kiadott tanúsítványt a CA saját digitális aláírásával hitelesíti. A CA digitális aláírása persze ugyanígy egy tanúsítványon alapul; ezt a tanúsítványt a szervezet vagy egy másik CA-tól kapta, vagy önmagának adta ki. Ez utóbbi esetben a CA úgynevezett „Root CA", mert ha a tanúsítványok digitális aláírásait követve felépítjük a tanúsítvány hierarchikus hitelesítési útvonalát, ez a kiszolgáló lesz a hitelesítési lánc végén. A tanúsítvány definíció szerint csak akkor érvényes, ha a hitelesítési lánc végén található Root CA-ban a felhasználó explicit megbízik. *(Erre való a felhasználó tanúsítványtárában található Trusted Root CA's logikai tárolóhely, lásd korábbi számunkat).*

Ha a CA nem „root", hanem a hitelesítési láncban valahol „középen" szerepel, „subordinate" CA-nak hívjuk. Ezen belül attól függően, hogy csak más CA-knak osztogat tanúsítványt, vagy végfelhasználókat is kiszolgál, „intermediate" vagy „issuer" CA-nak nevezzük. Lényegében, legalábbis számítástechnikai értelemben nincs sok különbség aközött, hogy a CA kinek és milyen típusú tanúsítványt osztogat. Annál inkább jogilag!... Miután a CA az igénylőnek kiadta a tanúsítványt, feladata még az is, hogy a nyilvános kulcsokat tartalmazó tanúsítványokat közzé tegye, **terjessze**. Windowsos, vállalati környezetben a tanúsítványok központi terjesztési pontja az Active Directory, máskor, tartományi környezeten kívül telepített kiszolgálók esetén lehet, hogy a kiadott tanúsítványok mindössze a fájlrendszer egy kijelölt mappájába kerülnek bele, és a rendszer másra bízza azok terjesztését.

A harmadik nagyon fontos feladat pedig a tanúsítványok életútjának kezelése *(szükség szerint azok megújítása)*, illetve **visszavonása**. Egy tanúsítvány visszavonására akkor kerülhet sor, ha kiderül, hogy valaki szert tett a tanúsítványban található publikus kulcs privát párjára, és így illegális módon használhatná azt, vagy például ha a privát kulcsot eredeti tulajdonosa elveszítette, és emiatt a tanúsítvány amúgy is használhatatlanná vált. A CA-k az általuk kiadott tanúsítványokat tehát visszavonhatják, és a visszavont tanúsítványok listáját mindig nyilvánosan elérhetővé teszik. Az, hogy egy tanúsítvány szerepel-e ezen a listán, a tanúsítvány elfogadása előtt az azt ellenőrző felhasználó *(vagy szoftver)* feladata *(lenne)*.

A Windows CA szolgáltatás telepítési módjai

A Windows CA szolgáltatását kétféleképpen telepíthetjük:

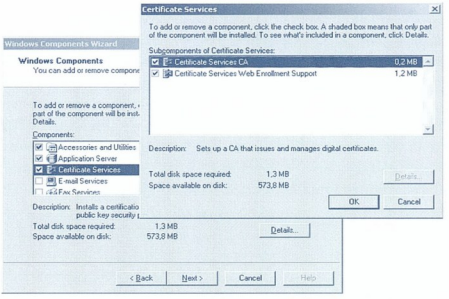
- Stand Alone *(önálló)*, vagy
- Enterprise *(vállalati)* módban.

A két üzemmód közötti legfontosabb különbségek az alábbiak:

- **Stand-Alone CA:** Használatához Active Directory nem szükséges; a tanúsítványok „kiadását" *(a kérések elfogadását vagy elutasítását)* a rendszergazda végzi; a kiadott illetve visszavont tanúsítványok listája a fájlrendszerbe kerül.
- **Enterprise CA:** Csak Active Directory tartományi környezetbe telepíthető; a tanúsítványkérések kiszolgálása vagy elutasítása az Active Directoryban tárolt információk alapján automatikus; a kész és a visszavont tanúsítványok listáját a CA az Active Directoryba teszi közzé.

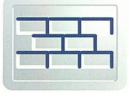
A Certificate Services telepítése

A tanúsítványkiadó szolgáltatás a többi komponenshez hasonlóan, a Windows Components Wizard segítségével telepíthető.



A Certificate Services telepítési opciói

A Certificate Services opció kiválasztása után a Details... gombra kattintva láthatjuk, hogy a telepítés két részből áll: az első maga a CA szolgáltatás, a másik pedig a „Certificate Services Web Enrollment Support", azaz a CA webes felülete. Ennek telepítéséhez természetesen a számítógépen futó IIS-re van szükség. A webfelület egyébként a CA-tól független számítógépre is telepíthető; ha pedig bármikor szeretnénk azt újratele-



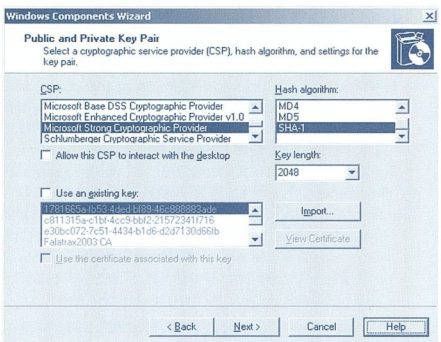
píteni *(például mert az IIS-t újra kellett telepítenünk, vagy bármilyen más okból a webes struktúrából eltűntek a Certificate Services virtuális mappái)*, a

```
certutil - vroot
```

parancs segítségével ezt később is megtehetjük. De térjünk vissza a telepítéshez! A telepítővarázsló első oldalán a telepítendő CA típusát kell kiválasztanunk. A CA-nk lehet:

- Enterprise root CA
- Enterprise subordinate CA
- Stand-alone root CA
- Stand-alone subordinate CA.

Miután eldöntöttük, hogy milyen CA-t telepítünk *(a továbbiakban – hacsak külön nem említjük – Enterprise root CA-val foglalkozunk majd)*, válasszuk ki a „Use custom settings to generate the key pair and CA certificate" jelölőnégyzetet *(Windows 2000-ben ezt az opciót kicsit máskép hívják, de felismerhető)*. Ha megtettük, a következő oldalon a CA saját kulcspárja generálásának módjába és körülményeibe avatkozhatunk be.



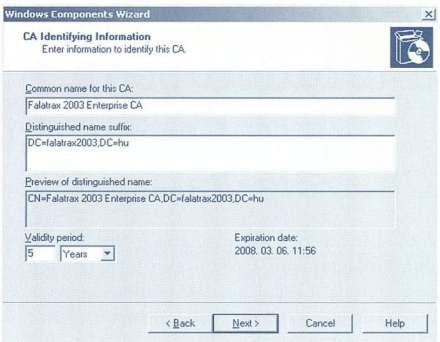
A CA kulcspárjának beállításai

Kiválaszthatjuk például a használni kívánt titkosító modult *(Cryptographic Service Provider, CSP)*, ami a Microsoft beépített, szoftveres *(Base, Strong, Enhanced)* moduljai mellett bizonyos hardveres CSP-k lehetnek. Utóbbi választását jól gondoljuk át, mert ebben az esetben a CA-nak minden olyan műveletnél, amikor a privát kulcsára szüksége van, a hardvereszközhöz kell nyúlnia, és ezt nem minden CSP támogatja. Hardvereszköz híján választhatunk a Microsoft Base, Strong és Enhanced Cryptographic Providerek közül. A három modul a beépített titkosító algoritmusokban és a leghosszabb kezelhető kulcsméretben különbözik egymástól, a csúcs – és persze az ajánlott választás – mindenképpen a Microsoft Strong vagy az Enhanced CSP. Amint az ábrán is látható, a CA kulcsának alapértelmezett hossza a Windows 2003 Serverben már 2048 bit.

Az oldal alján kijelölhetnénk, hogy melyik esetleg már létező kulcspárt és tanúsítványt szeretnénk használni ahelyett, hogy újat generálnánk. Erre a lehetőségre akkor lehet szükség, ha a tanúsítványkiszolgálót valamilyen oknál fogva újra kellene telepíteni. Ugyanitt importálhatunk is tanúsítványt.

A következő oldalon meg kell adnunk a CA nevét *(Windows 2000-ben kicsit részletesebben, de tulajdonképpen adminisztratív adatokról van szó)*, valamint a CA tanúsítványának érvényességi idejét.

Az alapértelmezett érvényességi idő Windows 2000 esetén 2, Windows 2003 esetén 5 év.



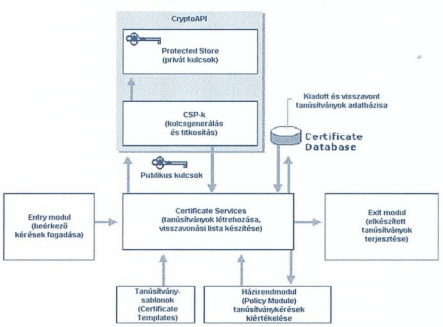
A CA tanúsítványán szereplő név és az érvényességi idő beállítása

Erre figyeljünk, mert a CA mindig legfeljebb annyi ideig érvényes tanúsítványt hajlandó kiadni, amíg a saját tanúsítványa még érvényes, a CA tanúsítványának lejárta pedig automatikusan érvényteleníti az összes általa kiadott tanúsítványt is! *(Lejárat előtt a CA tanúsítványát természetesen majd meg lehet újítani)*.

A következő oldalon megadhatjuk a CA adatbázis és logja helyét *(ami egyébként belső felépítésében hasonló az Active Directory adatbázisához)*. Ugyanitt – megintcsak leginkább katasztrófa utáni helyreállításkor – megadhatjuk, hogy a CA használjon egy már korábban meglévő adatbázist. A varázsló ezután befejezi a telepítést és el is indítja a szolgáltatást *(újraindításra nem lesz szükség)*.

A Certificate Services logikai felépítése

A Certificate Services blokkdiagramja a következő:



A Certificate Services blokkdiagramja

A különböző modulok feladata *(és tulajdonképpen a tanúsítványok kiadásának és kezelésének módja)* az alábbi:

- **Entry modul:** fogadja a beérkező tanúsítványkéréseket *(legyen az webes, vagy közvetlen eljáráshívással – RPC – érkezett)*.
- **Tanúsítványsablonok:** Enterprise CA esetén az Active Directoryban jópár különböző célra használható, előre

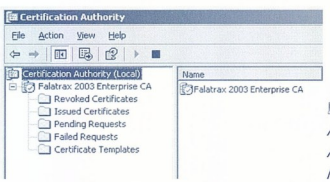
elkészített tanúsítványsablont találunk *(ezekre még visszatérünk)*. Stand-alone CA-k esetén tanúsítványsablonok nincsenek, ott a tanúsítványt kérő személy konkrétan adott célra kéri a tanúsítványokat.

- **Policy Module:** A policy modul a rendelkezésre álló információk *(bejelentkezési adatok, Enterprise CA esetén a kért tanúsítványsablonon definiált jogosultságok, stb.)* alapján eldönti, hogy a kért tanúsítvány kiadható-e.
- **Certificate Services:** A kérést a szolgáltatás a Policy Module utasításai alapján kiszolgálja vagy elutasítja. Szükség esetén létrehozza a tanúsítványt, tárolja azt a saját adatbázisában. Ugyanez a komponens generálja a visszavont tanúsítványok listáját is.
- **CryptoAPI, CSP-k, Protected Store:** a tanúsítványok létrehozásához szükséges konkrét kriptográfiai műveleteket *(pl. a digitális aláírást, de szükség esetén akár kulcsgenerálást is)*, a CryptoAPI, azon belül is a kiválasztott CSP modul végzi. A CA privát kulcsa védett helyen, a Protected Store-ban található.
- **Exit Module:** feladata az elkészült tanúsítványok közzététele *(miután a tulajdonosa már megkapta azt)*. Enterprise CA esetén a kész tanúsítványt az Active Directoryba küldi *(hozzárendeli a tanúsítványt kérő felhasználóhoz)*, Stand-alone esetén pedig szükség esetén fájlba menti az.

Bár a „gyári" Windowsban csak egy-egy policy és exit modul található, ilyeneket saját célra is fejleszthetünk. A Policy modul ugyanis lecserélhető, az Exit modulok közé pedig saját modulokat vehetünk fel – mondjuk olyat, ami a kész tanúsítványt még e-mailben is elküldi a kért címre.

A Certificate Services kezelői felülete

A Certificate Services kezelőfelületét *(MMC konzolját)* az Administrative Tools → Certification Authority paranccsal nyithatjuk meg.



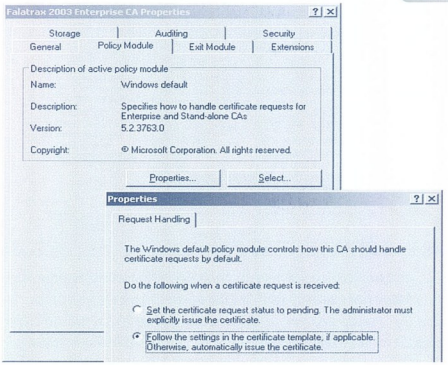
A Certification Authority MMC modul

A felügyeleti eszköz Windows 2000-ben és a Windows 2003-ban is nagyon hasonlóan néz ki. Röviden összefoglalva a konzolfa elemeit:

- **<CA név>:** a Certificate Services biztonsági mentése, visszaállítása, a főbb beállítások.
- **Revoked Certificates:** a visszavont tanúsítványok listája.
- **Issued Certificates:** a kiadott tanúsítványok listája.
- **Pending Requests:** a folyamatban lévő tanúsítványkérések *(Enterprise esetén szinte mindig üres, Stand-alone módban itt kell a tanúsítványokra kattintva választanunk az „Issue" [kiadható] vagy a „Deny" [elutasítandó] lehetőségek közül)*.
- **Failed Requests:** az elutasított tanúsítványkérések listája *(Stand-alone módban a rendszergazda, Enterprise módban pedig maga a policy modul tehet ilyet!)*.
- **Certificate Templates *(Windows 2000-ben: Policy Settings)*:** a CA által kiadható tanúsítványtípusok listája.

A CA szolgáltatás beállításai – a Policy modul

A Policy modul beállításait a CA tulajdonságai között találjuk:



A Policy modul beállításai

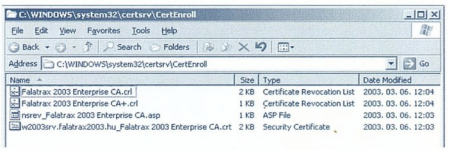
A Select... gomb hatására választhatunk a rendszerben található Policy modulok közül *(ha van telepítve más is, mint a „gyári")*. A Properties... gomb pedig a modul beállításait mutatja: itt választhatunk, hogy a tanúsítványt a rendszer a tanúsítványsablonok alapján mindig adja ki *(ez az alapértelmezés)*, vagy a sablonoktól függetlenül bízza ezt a rendszergazdára *(Windows 2003-ban Enterprise CA esetén is; ez az opció Windows 2000-en Enterprise CA esetén szürke)*.

A CA szolgáltatás beállításai – az Exit modul

Az Exit modul feladata a tanúsítványok közzététele, és mint azt már említettem is, Exit modulból egynél több is lehet a rendszerben. A beépített Exit modul beállításai között a tanúsítványok Active Directoryba illetve a fájlrendszerbe történő közzétételét engedélyezhetjük, illetve tilthatjuk le. Az Active Directoryba való közzététel az Enterprise CA-k esetén alapértelmezés, Windows 2003-ban ki sem kapcsolható. A fájlba való közzététel pedig a Stand-alone CA-k alapbeállítása, de persze Enterprise CA-k esetén is bekapcsolható.
A fájlba mentett tanúsítványok helye a

```
%windir%\system32\certsrv\CertEnroll
```

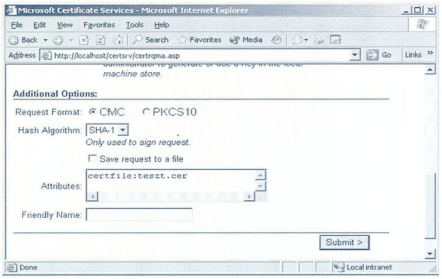
könyvtár. Ebben a könyvtárban alapértelmezésben a CA saját tanúsítványa és a visszavont tanúsítványok listája található *(Ez a mappa egyébként a http://.../certsrv/CentEnroll/ címen is elérhető, ha engedélyeztük-telepítettük a Web Enrollment supportot)*:



A CA tanúsítványa és a CRL-ek a fájlrendszerben

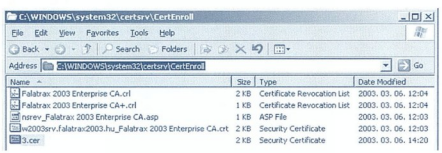
A felhasználói tanúsítványok is ide kerülnek *(ha engedélyeztük)*, de nem automatikusan: a tanúsítványkérésnek tartalmaznia kell tanúsítvány fájlnevét, certfile:filenév.cer formában. Ezt vagy a PKCS#10 fájl generálásakor kell megadnunk a megfele-

lő helyen, vagy – kicsit előreszaladva – a webes tanúsítványkérés legvégén, mint speciális opciót:



A tanúsítvány szerveren történő fájlba mentéséhez az Attributes mezőt kell kitöltenünk

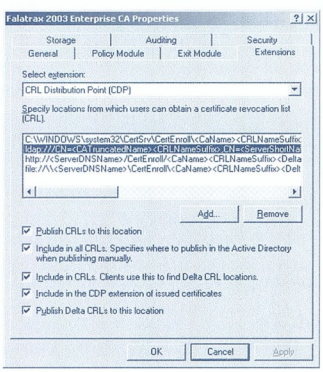
Figyelem, a „Save request to a file" **nem** erre vonatkozik! A certfile: opció hatására az Exit modul elmenti a kész tanúsítványt. Itt van egy kis különbség a Windows 2000 és Windows 2003 CA között: míg a Windows 2000 a kért fájlnéven tárolja azt, a Windows 2003 a kérés belső sorszámából generálja a fájlnevet:



A fájlba mentett tanúsítvány Windows 2003-ban

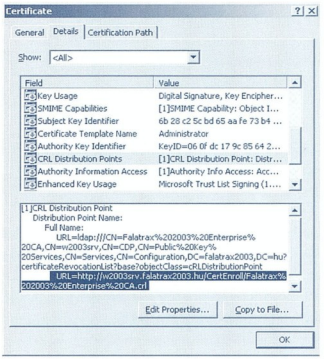
A visszavont tanúsítványok listájának és a CA saját tanúsítványának elérési útja

A tanúsítványkiszolgáló tulajdonságlapjának következő oldala az „Extensions". Az itt következő beállításokat Windows 2000-ben a Policy modul beállításai között találjuk meg.
A dialógusablakban a visszavont tanúsítványok listájának *(CRL)*, valamint a CA saját tanúsítványának *(Authority Information Access, AIA)* címtárbeli *(ldap://)*, *webes (http://)* és akár fájlrendszerbeli *(file://)* elérési útját állíthatjuk be.



A CRL és az AIA elérési útjai

Ahol bekattintjuk a „Publish CRLs to this location" opciót, oda a rendszer mindig „menti" az elkészült visszavonási listákat; ahol pedig az „Include in the CDP extension of issued certificates" mező is be van jelölve, a cím bekerül a kiadott tanúsítványokba is *(ez persze ajánlott, különben a felhasználók nem tudnák, hogy hol keressék a CRL-t illetve az AIA-t)*. Windows 2000-ben ezt az adott cím előtti pipa bejelölésével tehetjük meg.

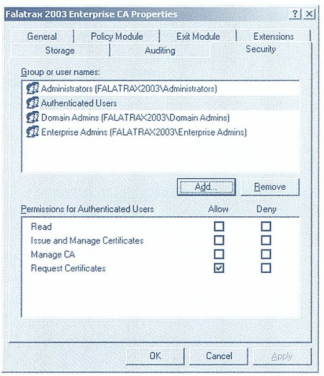


A CRL elérési útja egy kiadott tanúsítványban

Ha egy valós CA által kiadott valós tanúsítvány CRL Distribution Points mezőjéből kimásoljuk a http:// címet, azt egy böngészővel megnyitva – ha minden jól megy – visszakapjuk a tanúsítványt kiadó CA érvényes CRL-jét.

Biztonság és naplózás

Az „Auditing" oldal új a Windows 2003 Server Certificate Services tulajdonságlapján. Itt igényeink szerint bekapcsolhatjuk a CA-val végzett komolyabb műveletek naplózását. A bejegyzések a biztonsági eseménynaplóba kerülnek, de ehhez először a Group Policy-ban globálisan is engedélyeznünk kell az „Audit object access" naplózási házirendet. A „Security" oldalon pedig a tanúsítványkiadó szervezet szolgáltatásaihoz való hozzáférési jogokat adhatjuk meg:



Egy Windows 2003 Enterprise CA alapértelmezett biztonsági beállításai

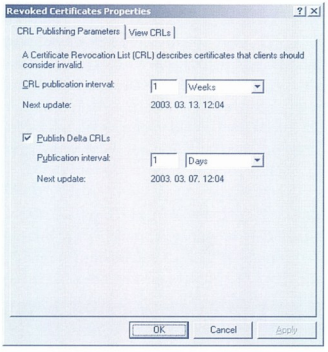
A jogosultságok magukért beszélnek:

- **Issue and Manage Certificates:** tanúsítványok kiadásának és kezelésének joga.
- **Manage CA:** a tanúsítványkiadó szervezet beállításainak módosításához szükséges jog.
- **Request Certificates:** tanúsítványkérési jogosultság.

Windows 2000-ben a jogosultságok kicsit mások, ott csak általában „Manage" és „Enroll" jog létezik *(utóbbi szükséges a tanúsítványkéréshez)*. Természetesen a „Request Certificates" jog megléte még nem garancia arra, hogy a felhasználó a tanúsítványt meg is kapja; ehhez előtte még a Policy modulnak is lehet egy-két szava.

A visszavont tanúsítványok listájának *(CRL)* közzététele

A CRL-ek kezelésének beállításait a „Revoked Certificates Properties" dialógusablakban találjuk, amit a konzolfában a „Revoked Certificates" sorra jobb gombbal kattintva, majd a „Properties" sort választva csalogathatunk elő. Windows 2000-ben itt beállíthatjuk a CRL-ek frissítésének időszakát *(CRL Publication Interval; ez alapértelmezésben hetente történik)*, illetve megtekinthetjük az éppen érvényes visszavonási listát.
A Windows 2003 a „hagyományos" CRL-ek mellett ún. különbségi, azaz Delta CRL-t is kezel. A Delta CRL kisebb és gyakrabban *(alapértelmezésben naponta)* készül, mint a „nagy" CRL, és csak az utolsó teljes CRL kiadása óta történt változásokat tartalmazza. Ha van Delta CRL, akkor a „nagy" CRL-ben a „Freshest CRL" mezőben a CA feltünteti a Delta CRL elérési útját. Ha a felhasználó rendszere kezeli a Delta CRL-eket, erről a címről letöltheti azt; ha pedig nem, akkor egyszerűen figyelmen kívül hagyja és a teljes CRL-t használja majd.
Windows 2003-ban tehát a tulajdonságlapon a CRL mellett a Delta CRL frissítési időszakát is beállíthatjuk, valamint megtekinthetjük.

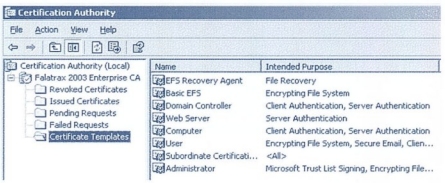


A tanúsítványvisszavonási listák beállításai Windows 2003-ban

Egy kiadott tanúsítványt egyébként úgy tudunk visszavonni, hogy az „Issued Certificates" listában az adott tanúsítványra bökünk és az All Tasks → Revoke Certificate parancsot választjuk.

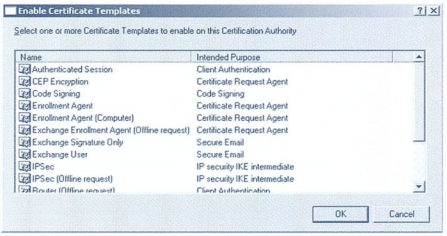
A kiadható tanúsítványsablonok listája

A CA konzolfa utolsó eleme *(persze csak az Enterprise CA-kban)* a „Certificate Templates" elem *(Windows 2000-en ez a „Policy Settings" nevet viseli)*.



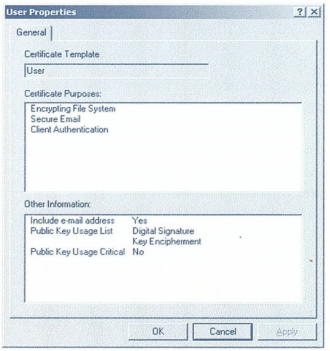
A kiadható tanúsítványok listája

A listában azokat a tanúsítványsablonokat találjuk, amelyeket az adott Certificate Service kérésre kiszolgálhat. Ilyen előre elkészített tanúsítványsablon azonban a listában láthatónál sokkal több van definiálva az Active Directoryban. Ha új kiadható tanúsítványtípust szeretnénk felvenni, kattintsunk jobb gombbal a Certificate Templates sorra, majd válasszuk a New → Certificate *(Template)* to Issue parancsot.



A kiadható tanúsítványsablonok kiválasztása

Az erre megjelenő ablak tartalmazza az összes definált *(és a CA listájában még nem szereplő)* tanúsítványsablont, amelyek közül kattintással választhatunk.
Ha a Certificate Authority MMC konzolban kettőt kattintunk valamelyik tanúsítványsablonra, megjelenik az adott tanúsítvány főbb beállításait tartalmazó ablak:

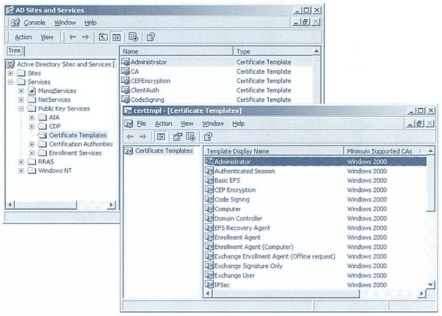


A tanúsítványsablon jellemzői

Ebben az ablakban látható, hogy az adott sablon alapján kiállított tanúsítvány milyen célokra használható *(az ábrából kiderül, hogy a „User" típusú tanúsítványok például titkosított fájlrendszer, titkosított levelezés és felhasználóazonosítási célra)*, valamint néhány más információ is *(például, hogy a tanúsítvány tartalmazza a felhasználó e-mail címét)*. Ezek a beállítások mind-mind az Active Directoryba „beégetett" tanúsítványsablon-objektumokból származnak. Ugyanitt definiálható az is, hogy egy-egy tanúsítványsablon által ki jogosult tanúsítvány kérésére – lássuk, hogyan férhetünk hozzá az Active Directory-beli tanúsítványsablon-listához!

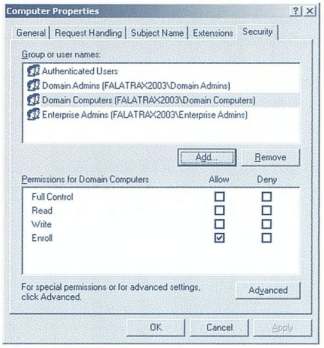
Tanúsítványsablonok az Active Directoryban

Ezen a ponton a Windows 2000 kicsit nehézkes volt, amin a Windows 2003-ban már segítettek a fejlesztők. Mivel a tanúsítványsablonok az Active Directory ún. rendszerbeállításokat tartalmazó partíciójában találhatók, Windows 2000 alatt azokhoz úgy férhetünk hozzá, ha elindítjuk az Active Directory Sites and Services eszközt, majd ott kiválasztjuk a View → Show Services node opciót. Ekkor a Services → Public Key Services → Certificate Templates sorban megtaláljuk a tanúsítványobjektumokat. Maguk az objektumok egyébként persze a Windows 2003 Active Directoryban is ugyanitt találhatók, csak kezelésükhöz készült egy különálló eszköz *(MMC modul: Certificate Templates)*. Ezt a modult a Certificate Authority konzolból egy lépésben is elindíthatjuk, ha az előbb tárgyalt Certificate Templates soron a „Manage" parancsot választjuk.
Windows 2000-ben, illetve Windows 2003-ban a Windows 2000-kompatibilis *(szürke)* sablonokra kettőt kattintva a sablonok beégetett, módosíthatatlan tulajdonságait láthatjuk.



Tanúsítványsablon-kezelés Windows 2000-ben (hátul), és a Windows 2003-ban (elöl)

Az egyetlen „dinamikus", általunk is módosítható oldal a Security, ez határozza meg, hogy az adott tanúsítványsablon alapján ki kérhet tanúsítványt:



A tanúsítványsablon jogosultságlapja

Az adott típusú tanúsítvány kéréséhez legalább Read és Enroll jogra van szükség. Az ábrán látható Computer tanúsítványt például a Domain Computers csoport tagjai *(azaz a tartományi számítógépek)* kérhetik *(ők Read jogot az Authenticated Users csoporton keresztül kapnak)*.

A Windows 2003 Server Enterprise Edition Certificate Services újdonságai

Az eddigiekhez képest sokkal több újdonságot tartalmaz a Windows 2003 Server Enterprise Edition tanúsítványkiszolgálója *(szerkeszthető tanúsítványok, automatikus tanúsítványkérés felhasználók részére is, kulcsarchiválás, stb.)*. Ezekre a következő számban térünk vissza.

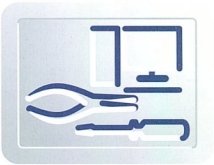
Folytatjuk …

Fülöp Miklós
mick@netacademia.net

Kapcsolódó NetAcademia tanfolyamok:
- Windows 2003 Expert Workshop
- PKI Workshop

Portál Para[digma] II

Második rész, avagy mi az a cache?



A tartalommenedzsment vagy CMS rendszereknél még vitatkozhatunk, de az igazi alkalmazásintegrációt megvalósító, főleg intranet portáloknál leszögezhetjük, hogy a statikus HTML oldalak előre legenerálása nem lehetséges. Az egyetlen út a teljesen dinamikus markup előállítás. Erről, és ennek optimalizálásáról szól a portál cikksorozat második része.

Az egyszerű információs weboldalakkal ellentétben egy igazi portál interaktív felhasználói élményt nyújt, azaz a megjelenő markup erősen függ a felhasználó tevékenységétől, a megjelenő információ rendszerint személyre és az adott pillanatra szabott. Nem tudhatjuk, hogy a felhasználó mit fog beírni a keresőbe, vagy melyik ügyfél adatait kéri le a CRM-ből. Nincs mit tenni, a portáloldalt a lekérés pillanatában kell előállítani.

A dinamikus oldalak előállítása több processzoridőt vesz igénybe, egyes adatok fájlokból, adatbázisokból vagy egyéb viszonylag lassú rendszerekből kerülnek kinyerésre. Ha ezt megtetézzük rengeteg felhasználóval, akiknek nincs jobb dolguk, mint a mi portálunkat terhelni, a portál reakcióideje megnőhet, illetve kevesebb felhasználót képes kiszolgálni. A válaszadás ideje nem lehet nagy, mert az rontaná a felhasználói élményt, bár azt is hozzá kell tenni, hogy még mindig jobb egy viszonylag lomha kereső, ami mondjuk két percig keres végig ezer dokumentumot, vagy tízezer ügyfelet, mint ha azt mi magunk tennénk jó néhány órán vagy napon át. A teljesítmény fogalma tehát igen relatív.

Mindazonáltal, ha már portált építünk, az legyen fürge, de ne tévesszük össze a teljesítményt a skálázhatósággal. Több processzor vagy szerver hozzáadásával nő a kiszolgálható felhasználók száma, de a teljesítmény nem. Egy ideig gyorsabbnak érezzük a rendszert, de csak akkor, ha a lassúság oka a sok felhasználó által okozott túlterheltség volt. A portálok teljesítményének finomhangolása sok gyakorlatot és türelmet igényel, a legjobb eredményt pilotokkal és mérésekkel lehet elérni, illetve nem árt jól kitesztelt portálmotorra építkezni.

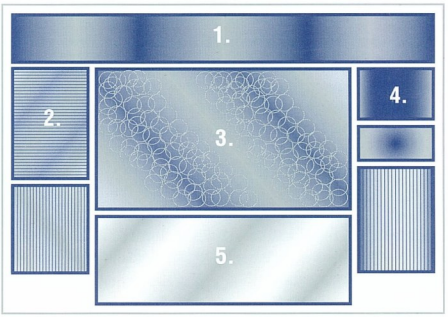
Átmeneti tár

Ha a portálunkra érkező kérések nagy része ugyanazt az eredményt állítja elő, akkor ugyanannak a markupnak az újból és újból történő előállítása csak fölöslegesen terheli a szervert. Bevett szokás, hogy a fáradságos CPU idővel generált HTML oldalakat az előállítás után a memóriában lévő átmeneti tárba teszik, majd ha újból olyan kérés érkezik, amire a válasz megegyezik valamely átmenti tárban tárolt válasszal, újragenerálás helyett az átmeneti tárból szolgálják ki a kérést. Az átmeneti tár közismertebb angol neve a cache, ami nem tévesztendő össze a cash, azaz készpénz szóval, bár ha úgy vesszük némi közük van egymáshoz, hisz az elsővel megtakaríthatunk a másodikból, és a kiejtésük is teljesen megegyezik.

Ez szépen is hangzik, csak éppen azt nem könnyű eldönteni egy kérésről a válasz előállítása nélkül, hogy a válasz vajon ugyanaz lesz-e, mint az előbb volt. A dinamikusan előállított oldalak tartalma ugyanis időben változik, például egy híroldal idővel frissebb híreket is meg kell jelenítsen, mint ami az átmeneti tárban van. Erre az lehet a megoldás, hogy az átmeneti tárat időnként töröljük, és a következő kérésnél a kimenetet újra előállítjuk és újabb időre az átmeneti tárba tesszük. Ezt az időt érvényességi vagy lejárati időnek, *(angolul expiry time)* szokás hívni. A másik megoldás, hogy a cache törlését valamitől – például egy új hír megjelenésétől – függővé tesszük.

Szintén probléma, hogy a portáloldalak egy-egy darabját szinte mindig egyedileg kell előállítani, így az egész oldal nem tehető az átmeneti tárba. Az alábbi ábra egy tipikus intranet vagy bármely más portál sematikus képét mutatja, például:

(1) Fejléc;
(2) Menü, navigáció;
(3) Friss hírek, letölthető dokumentumok;
(4) Kereső;
(5) Akár egy CRM vagy ERP felhasználói felülete is lehet a portálba ágyazva.



Dobozkák – a portál felhasználói interfésze

A legtöbb portál, de főleg az intranet és extranet portálok igazából felhasználói alkalmazások prezentációs rétegei, felhasználói felületei. Egy alkalmazás felhasználói felülete azonban állandóan változik, reagálva a felhasználó által bevitt adatokra. A portál annyiban komplikáltabb, hogy itt nem egy, hanem több kicsi alkalmazás kisebb-nagyobb felhasználói felülete jelenik meg dobozkák formájában egy HTML oldalra bepakolva. Ezeket a különböző gyártók a webpart, portlet, gadget, placeholder és egyéb nevekkel illetik.

Minden dobozka más és más forrásból származó tartalmat vagy felhasználói felületet jelenít meg. Ha az egész oldalt, minden dobozkát állandóan frissítenénk, nemcsak a portált, de a mögötte rejlő alkalmazásokat is könnyen leterhelnénk. Az egész oldalt nem tehetjük egyben az átmeneti tárba, mert szinte minden dobozkának más és más az élettartama. A híreket

elég tíz percenként frissíteni, a navigációt akár órákig is lehet tárolni, hisz olyan ritkán változik; a kereső vagy a mindig változó fórum viszont egyáltalán nem, vagy csak másodpercekre cache-elhető.

A fórumban lévő témák listája elsőre úgy tűnik, cache-elhető, de ha vannak például zárt fórumok, azaz felhasználónként dől el, hogy a jogosultságok alapján mely fórumok jelennek meg a listában, azt sajnos egyedileg, felhasználónként kell előállítani – nem lehet átmeneti tárolást alkalmazni, legfeljebb mélyen a fórum kódjában. Az átmeneti tárolás ugyanis nem csak generált HTML-re alkalmazható, tárolhatunk XML-eket, ADO Resultsetet vagy bármi más objektumot a memóriában, így csökkenthetjük a terhelést az adatbázisokon vagy más külső adatforrásokon, már csak azért is, mert a webszervereket sokkal könnyebb és költségkímélőbb skálázni.

Fontos megemlíteni, hogy egy olyan portálon, ahol például a híroldalra másodpercenként száz lekérdezés érkezik, ott az egyperces átmeneti tárolás is hatezer *(!)* lekérdezéstől mentesítheti a híreket tároló adatbázist.

Az átmeneti tárolás alaplogikája igen egyszerű, de van egy-két hátránya is, többek között sok memóriát igényel és igen nehéz kitesztelni. Ha sok mindent helyezünk az átmeneti tárba, könnyen betelik a fizikai memória, és ettől a portálunk nemhogy gyorsabb, de lassabb és instabil is lesz. Tovább nehezíti a dolgunkat, hogy terheléselosztás, azaz NLBS esetén a webszerver farm minden tagjának a memóriájában egymástól függetlenül lesz átmeneti tár.

Mielőtt azonban mélyebben belemennénk az átmeneti tárolás rejtelmeibe, nézzük meg, milyen eszközök állnak a rendelkezésünkre ASP.NET esetén, és mit nyújtanak a portálmotorok.

ASP.NET

Az ASP programozóknak nem sok eszköz állt rendelkezésükre átmeneti tár megvalósításához. A session vagy application objektumokat ugyan lehetett cache-nek használni, de ehhez programozásra volt szükség, márpedig a cache beállítása általában az üzemi környezetben szerzett tapasztalatok alapján, a programozás végén, vagy akár utána, az üzemeltetés részeként következik.

Az ASP.NET fejlesztők szerencsésebb helyzetben vannak, az új eszköz ugyanis támogatja a HTML vagy egyéb markup egyszerű deklaratív átmeneti tárolását, illetve rendelkezésükre áll egy Cache API is, amivel saját cache logikát valósíthatnak meg bonyolultabb feladatokhoz.

ASP.NET deklaratív cache

A legegyszerűbb eset, ha az egész ASPX oldal kimenetét szeretnénk egy időre átmenti tárba helyezni.

```
<%@ Page Language="vb"%>
<%@ OutputCache Duration="10" VaryByParam="None"%>
```

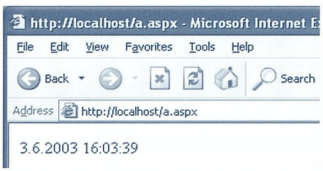
A fenti sort az ASPX oldal tetejére biggyesztve a szerver az oldal első futása után tíz másodpercig minden kérést az átmenti tárból szolgál ki, és csak a tíz másodperc leteltével futtatja le újra az oldalt. Ezt legegyszerűbben úgy próbálhatjuk ki, hogy kííratjuk az aktuális időt. Tegyük a következő scriptet egy ASPX oldal fejlécébe

```
<html>
<head>
<script runat="server">
        Sub Page_Load()
                time.Text = Now()
        End Sub
</script>
</head>
```

majd adjuk hozzá a következő vezérlőt a HTML-hez, ebbe fogja a script kiírni a pontos időt.

```
<body>
        <asp:Label id="time" runat="server"/>
</body>
</html>
```

Ez a bonyolult „alkalmazás" így néz ki a böngészőben:



Cache-elt oldal a böngészőben

Ha az oldalt frissítgetjük, látható, hogy csak minden tizedik másodpercben frissül a pontos idő. Sajnos ez a megoldás egy portálnál nemigen használható, de szerencsére tud az ASP.NET ennél többet is. Az OutputCache direktívának a következő paraméterei vannak:

Attribútum	Magyarázat
Duration	Az átmeneti tárolás ideje másodpercben, megadása kötelező
Location	Web Form-ok esetében beállítható az átmeneti tárolás helye, ami lehet a server, vagy akár a kliens is
VaryByControl	Ha több User Controlt használunk az oldalon, megszabhatjuk, hogy a tartalmuk különböző ideig tárolódjon
VaryByCustom	A Global.asax egyik metódusának fölülírásával saját logikát adhatunk a cache kezelésére
VaryByHeader	A cache alkalmazását függővé tehetjük a http-kérés fejlécének tartalmától is
VaryByParam	A cache alkalmazását függővé tehetjük a kérésben érkező GET és POST paraméterektől is

A VaryBy kezdetű paraméterek a legérdekesebbek, hisz ezek mögött a gyakorlatban is jól használható funkcionalitás rejtőzik. Ez a paraméter lehetővé teszi, hogy olyan oldalakat is programozás nélkül cache-eljünk, amelyek querystring paraméterek, form mezők értéke, http-fejléc értékek vagy böngészőfajták alapján dinamikusan generálódnak.

Az alábbi példa egy online katalógusoldal, amely a TermekLista nevű táblázatba tölti a Kategoria paraméterben érkező kategóriájú termékeket.

```
<asp:DataGrid id="Termeklista" runat="server"/>
```

Ha ennek az oldalnak az elejére a következő direktívát tesszük,

```
<%@ OutputCache Duration="600"
↳ VaryByParam="Kategoria" %>
```

akkor a

```
http://www.myshop.hu/termek.aspx?Kategoria=konyv
```

lekérésre az oldal a könyvek listáját adja. Mivel az ASP.NET-nek megmondtuk, hogy az oldalt cache-elni kell, így a HTML bekerül az átmeneti tárba. Azonban megadtuk, hogy vegye figyelembe a Kategoria nevű paramétert, a

```
http://www.myshop.hu/termek.aspx?Kategoria=cipo
```

lekérdezésre nem a cache-ből fogja venni a HTML-t, hanem ismét lefuttatja az aspx-et. Ha ezek után az előző lekérések bármelyikét kapja a szerver, a választ tíz percig a cache-ből fogja venni. Tehát a szerver a VaryByParam értékének megadott paraméter minden értékére külön cache-el. Ha a következő példához hasonlóan több paramétert adunk meg,

```
termek.aspx?Ital=bor&Evjarat=2003
termek.aspx?Ital=bor&Evjarat=1974
termek.aspx?Ital=pezsgo&Evjarat=2003
termek.aspx?Ital=pezsgo&Evjarat=1974
```

annak minden kombinációjára külön cache-el, feltéve, hogy minden paraméter nevét megadtuk a direktívának.

```
<%@ OutputCache VaryByParam="Ital;Evjarat" ... %>
```

Ezzel a megoldással azonban vigyázni kell, mert a kombinációk száma rohamosan *(exponenciálisan)* nő, és ez rengeteg memóriát felemészthet.

A Windows Server 2003-ban *(korábbi nevén Windows .NET Server)* található Internet Information Services 6 használata esetén a Location paraméter Server értékre állítása azt eredményezi, hogy a lekérés el sem jut az aspx alkalmazáshoz, a http.sys közvetlenül az átmeneti tárból szolgálja ki a lekéréseket.

```
<%@ OutputCache Location="Server" ... %>
```

A cache kezelését nemcsak direktívákkal, hanem kódból is szabályozhatjuk a Response objektum metódusaival, a szintaxis más, a hatás ugyanaz.

```
Response.Cache.SetExpires(
↳ DateTime.Now.Addseconds(10))

Response.Cache.SetCacheability(
↳ HttpCacheAbility.Public)
```

A leghasznosabb azonban, hogy ha User Controlokat alkalmazunk, azokat külön-külön is cache-elhetjük. Az oldalunkat tehát vezérlőkből építjük fel, és minden vezérlőnek külön állítjuk be, hogy kerüljön-e átmeneti tárolásra, vagy ne. A módszer szinte azonos az előbbi, oldalakra vonatkozó megoldással, itt is az OutputCache direktívát kell alkalmazni, íme:

```
<%@ Control Language="VB" %>
<%@ OutputCache Duration="120" ... %>
```

ASP.NET cache API

Az előbbi megoldások egyszerűek, de igazán bonyolult oldalakat, mint például egy komoly portált, már nehézkesen lehet ezekre építeni. Komolyabb feladatokra azonban itt a Cache osztály, amely HTML vagy egyéb markupon kívül akár objektumokat is képes tárolni. Az alábbi egyszerű példa egy XML stringet próbál meg kiolvasni a cache-ből. Ha sikerül, kiírja, ha nem, újra előállítja a GetXML() függvénnyel, majd az eredményt beteszi az átmeneti tárba, és ezután ki is írja. A példa nem állít be érvényességi időt a cache-nek, ez a legegyszerűbb alkalmazás.

```
Dim XML As String = Cache("myXML")
If XML Is Nothing Then
       XML = GetXML()          'nics bent, betöltjük
       Cache("myXML") = XML 'betöltjük a cache-be
       Response.Write(XML)   'kiírjuk
Else
       Response.Write(XML)   'bent volt, kiírjuk
End If
```

Az átmenti tár törlése nem csak időhöz köthető, bár ez a leggyakoribb. Az ASP.NET Cache osztálya támogatja a cache-ben tárolt objektumok függőségének kezelését. Egy objektum függhet egy fájltól, egy másik cache-elt objektumtól és az időtől. A következő példa a fájltól való függést mutatja be. A cache-ben tárolt XML katalógus akkor törlődik, ha az XML fájl a fájlrendszerben megváltozik. Deklarálunk egy XML dokumentumot

```
Dim xml As XmlDocument()
```

… és betöltjük az XML-t:

```
xml.Load(Server.MapPath("katalogus.xml"))
```

Most feltételezve, hogy az XML az ASPX fájlunkkal egy könyvtárban van, beállítjuk a függőséget a fájlra:

```
Dim fuggoseg As New CacheDependency(
↳ Server.MapPath("katalogus.xml "))

Cache.Insert("katalogus", xml, fuggoseg)
```

Mint látható, létrehoztunk egy példányt a CacheDependency osztályból, majd ezt adtuk át a Cache objektum Insert metódusának, létrehozva a függőséget.

Az időtől való függőségnek két altípusa van, az abszolút *(absolute)* és a csúszó *(sliding)*. Az abszolút egy megadott idő után törli a cache-t akár van kérés, akár nincs. A csúszó ezzel ellentétben minden az időn belül érkező kéréskor újraindítja az idő mérését. Az előző példát módosítva létrehozunk egy tíz másodperces csúszó időintervallumot, majd a Cache objektum Insert metódusánál ezt adjuk meg függőségnek.

```
Cache.Insert("katalogus", xml, Nothing,
↳ DateTime.MaxValue, TimeSpan.FromSeconds(10))
```

A következő példában a katalógus adatait az árlista és a vevőlista frissességétől tesszük függővé.

```
Dim fuggosegek(1) As String
fuggosegek(0) = "arlista"
fuggosegek(1) = "vevok"
```

```
Dim fuggoseg As new
↳ CacheDependency(nothing, fuggosegek)
Cache.Insert("katalogus", xml, fuggoseg)
```

Az átmeneti tárat természetesen mi magunk is törölhetjük a következő utasítással:

```
Cache.Remove("katalogus")
```

Az @OutputCache direktívával ellentétben a Cache API tehát nemcsak HTML kimenet cache-elésére alkalmas, hanem bármilyen objektumot képes a memóriában tartani, és akkor törölni, amikor szükséges.

Egyéb ASP.NET cache tulajdonságok

A Cache osztály egyszerű, de igen hasznos interfész az ASP.NET belső cache mechanizmusához. Az interfészen keresztül szabályozhatjuk, hogy meddig legyen egy objektum az átmenti tárban. Amikor azonban a memória fogyni kezd, a rendszer automatikusan törölni kezdi az átmeneti tárat, hogy memóriát szabadítson fel. Így elképzelhető, hogy olyan objektum is törlődik onnan, amelyről minden okunk megvan feltételezni, hogy ott van. Először ezeket a legkevésbé használt objektumokat törli, azok közül is a kevésbé fontosakat. Azt, hogy mi számít fontosnak, a függőségek beállításánál lehet megadni a következő módon:

```
Cache.Insert("katalogus", xml, Nothing, d, t,
↳ CacheItemPriority.High, Nothing)
```

A metódushívás utolsó előtti paramétere megadja a cache-elt objektum fontosságát, ami lehet normál, alacsony, magas és a törlés tiltása is lehetséges.

Az Insert metódus utolsó paraméterének megadásával pedig egy függvényt adhatunk át, amely akkor kerül meghívásra, ha a rendszer törli az objektumot a memóriából.

Portálspecifikus problémák megoldása

A fent bemutatott megoldások a legtöbb esetben jól működnek, azonban a fejlesztő itt olyan oldalakat vagy vezérlőket is elhelyezhet az átmenti tárba, amiket nem szabadna. Jó példa erre egy login dobozka egy Internet portálról, ahol ha a felhasználó beírja azonosítóit, megjelenik egy rövid üdvözlet, ami a nevét is tartalmazza. Ha ezt a logikát egy szerveroldali vezérlővel oldjuk meg, és a vezérlő kimenetét vagy az egész oldalt cache-eljük, akkor a cache törléséig minden felhasználó ugyanazt az üdvözletet kapja, azaz ha Józsi kérése töltötte fel a cache-t, akkor minden felhasználó az „Üdvözöljük Józsi!" üzenetet kapja. Ez a példa különösen kellemetlen, mert a felhasználók egy ilyen malőrben biztonsági rizikót látnak, és rögtön arra gondolnak, hogy ha én most Józsiként vagyok bejelentkezve, biztos más is be tud jelentkezni az én nevemben. Így lehet egy jelentéktelen cache beállításból akár még botrány is.
Egy megfelelő portál keretrendszer lehetőséget biztosít egy modul vagy dobozka fejlesztőjének, hogy szabályozza modulja cache-elhetőségét, megszabhasson minimum vagy maximum érvényességi időt, vagy éppen a fenti példa esetében tilthassa a cache beállítását.
Általában az is igény, hogy ne csak programozók, hanem egyszerű felhasználók is tudják az átmeneti tárolást befolyásolni. Egy hírportál szerkesztősége pontosan be tudja állítani oldalról-oldalra, hogy milyen gyakran frissülnek az ott található tartalmak. Egy intranet képzett felhasználója pedig testreszabhatja a saját oldalát, a cache-elés optimális beállításával meggyorsíthatja az oldal betöltését, ezzel növelve munkája hatékonyságát.
Ez a szabadság természetesen megköveteli a cache alkalmazásának fejlesztői korlátozását, illetve a hozzáférés programozói és deklaratív beállításának lehetőségét, röviden legyen a beállítás jogosultsághoz köthető.
Ha már a jogosultságnál tartunk, érdemes megemlíteni, hogy a jogosultsági beállítások miatt a legtöbb alkalmazás teljesen testreszabott tartalmat jelenít meg, aminek a markup szinten történő cache-elése gyakorlatilag lehetetlen, nem beszélve arról, hogy a jogok megvonása esetén az adott információnak még a cache-ből sem szabad hozzáférhetőnek maradnia.

NLBS

Ha jól skálázható és nagy rendelkezésre állású portált fejlesztünk, számolnunk kell webfarm építésével, ami a cache területén is tartogat nekünk kellemetlen meglepetéseket. Az alapproblémát az okozza, hogy az ASP.NET cache rendszere egy gépen fut, egy gép memóriájával dolgozik. A fájltól függés igen ritka, hisz az effajta NLBS-ben futó szerverfarmok általában adatbázisokból, vagy más, a Cache API számára ismeretlen rendszerekből táplálkoznak. Így megtörténhet, hogy az egyik gépen már frissebb adat van a cache-ben, mint a másikban, mert a cache törlése nem egy időben történik *(mivel a leggyakoribb az időtől való függés)*. Ilyenkor a felhasználó azt tapasztalhatja, hogy a tartalom az oldal újratöltésével régebbi lesz, mint előtte volt, hisz a http kérések a különböző érvényességű választ visszaadó szerverek között véletlenszerűen oszlanak el.
Sokszor problémát okoz, hogy a portálalkalmazás törölni szeretné a cache-t, de azt csak azon a gépen tudja megtenni, amelyikre az adott http kérés beérkezett. Erre jó példa lehet egy CMS, ami egy új hír felvitelénél törli a híreket a cacheből, így a felvitt hír azonnal megjelenik. Sajnos jelenleg erre az ASP.NET nem nyújt támogatást NLBS esetében, de léteznek olyan portál keretrendszerek, amik mindezt tudják.

Ha ezután cache-elős alkalmazás készítésére és tesztelésére adjuk a fejünket, legyünk türelmesek. Készítsük fel alkalmazásainkat a cache tesztelésére olyan debuginformációk megjelenítésével, amelyek segítik a tesztelést. Ilyen például az időpecsét, ami kiválóan jelzi a cache működését.

Bíró Tamás
MCSD
bt@sensenet.hu

Netscan – ahol végessé válik a végtelen

110001
001010
100111

Kiberszociológia a fejlesztők és a fogyasztók szolgálatában

A weben drága, nyelvi eszközökkel támogatott kereső-szűrő rendszerek segíthetnek a bennünket érdeklő információ kibányászásában. Egy Microsoft-kutatás most azzal kecsegtet, hogy a Usenet végeláthatatlan káoszából is sikerül automatikusan kiemelni a lényeget.

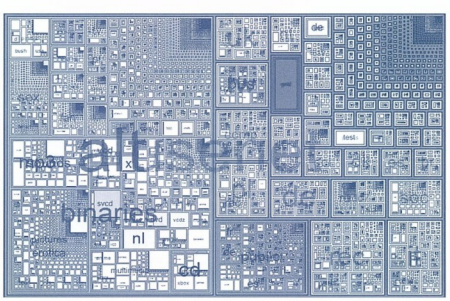
Amikor manapság az Internet kerül szóba, szinte mindenkinek a web és az e-mail jut az eszébe. Pedig van az online kommunikációnak egy olyan ága, amely már akkor létezett, amikor a webről még csak nem is álmodtunk, s amelyet ma is százezrek, ha nem milliók használnak világszerte. Természetesen a hírcsoportok világára, a Usenetre gondolunk.

Cikkünk írása idején több mint 78 ezer hírcsoport *(newsgroup)* működik a legkülönbözőbb témákban, az Office-fejlesztés kérdéseitől a pornográfia kifinomult műfajaiig. Persze az itt megjelenő publikációk nem hírek – legfeljebb pletykák, még inkább kérdések és válaszok, vélemények és ellenvélemények. Ember legyen a talpán, aki átlátja ezt a zűrzavart, s külön tudja választani a búzát az ocsútól, az értékes információt a megalapozatlan találgatástól vagy éppenséggel a szándékos félretájékoztatástól.

Ember? Inkább gép – mondják a Microsoft kutatóműhelyében, a Microsoft Researchnél *(közkeletű rövidítésével az MSR-nél)*.

Szociológiai Petri-csésze

Marc Smith, a laboratórium munkatársa nem kisebb fába vágta a fejszéjét, mint hogy megpróbálja objektív módon, automatikusan mérni, kifejező paraméterekre leképezni a Usenet-aktivitást, kiszűrni annak – a felhasználó szempontjából – valóban lényeges elemeit. Mindezt nem csupán néhány témakörre összpontosítva: a Netscan elnevezésű projekt egy olyan szerverről táplálkozik, amelyre közel 50 ezer hírcsoport anyaga fut össze.



Ilyennek látta a Usenet egészét a Netscan 2002. december 1-jén. Az egyes négyszögek egy-egy hírcsoportnak felelnek meg, s annál nagyobbak, minél több üzenetet publikáltak bennük. Ha az előző hónaphoz képest nőtt az üzenetek száma, a program zöldre, ha csökkent, pirosra festi a mezőt. A színezés erőssége a változás nagyságával arányos.

„Számomra az internet hatalmas szociológiai Petri-csésze. – mondja a kutató. – Napról napra félmilliárdnyian lépünk ki az online világba, s hagyjuk hátra lábnyomainkat a digitális hóban. Szociológus az emberi magatartásról ennél tökéletesebb felbontású anyagot még nem kapott. Népszámlálással a földkerekség népességének kevesebb mint egy százalékát érjük el. A Netscannel minőségileg másfajta adatokból indulhatunk ki. Nem mintát veszünk: a teljes egészet láthatjuk, tanulmányozhatjuk."

Jellemzők...

Smith rendszere két fő modulból áll. Az egyik a gyűjtő és -tároló, a másik a tulajdonképpeni feldolgozó, elemző egység. Az előbbi letölti a hírszerverre befutó anyagokat, az üzenetfejekből kiemel és adatbázisba ír bizonyos jól megválasztott információ-elemeket. A szoftver természetesen nemcsak folyamatosan építi az adatbázist, hanem annak karbantartásáról is gondoskodik. Az elemző modul az adatbázisból kimutatásokat, statisztikákat generál, illetve a felhasználó által választott időintervallumot alapul véve, a kért jellemzők szerint értékeli a kijelölt hírcsoport*(ok)* forgalmát, tartalmát.

Milyen paraméterekkel méri a Netscan a Usenet-aktivitást? Figyeli, rögzíti, hogy az egyes szerzők

- hány üzenetet publikálnak,
- hány különböző üzenetfonalba *(threadbe)* kapcsolódnak be,
- írásainak hány százalékára érkezik válasz,
- hány más szerző üzeneteire válaszolnak,
- hány naponként publikálnak újabb üzenetet.

... és tanulságok

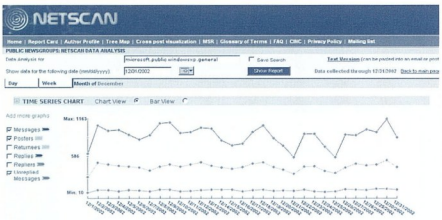
Mióta a rendszert 1999-ben elindították, több száz millió üzenetfej adatait vette fel az adatbázisba és értékelte ki. Így szignifikáns magatartásformákat és trendeket lehet már kimutatni, mind a szerzőkre, mind pedig az üzenetfonalakra és a hírcsoportokra vonatkozóan.

Objektív, statisztikai elemzéssel olyan eredményeket kaptak, amelyek – meglepő vagy nagyon is érthető módon – egybees-

tek azzal, hogy a hírcsoportok aktív levelezői, olvasói szubjektíve mikor ítéltek megbízhatónak valamely szerzőt vagy szavahihetőnek, hasznosnak valamely írást.

Általában azok az üzenetek bizonyultak a legértékesebbnek, amelyeknek szerzői

- rendszeresen *(de nem feltétlenül nagy terjedelemben)* publikálnak,
- általában válaszolnak mások üzeneteire, de azt röviden teszik,
- valamennyi *(vagy a legtöbb)* üzenetláncba bekapcsolódnak – de nem sokszor, általában mindegyikbe alkalmanként csak egyszer, legfeljebb párszor.



Netscan napi statisztikai grafikon a microsoft.public.windowsxp.general hírcsoport 2002 december havi aktivitásáról. A bal oldali panelen választhatjuk meg, hány paraméter értékei jelenjenek meg az ábrán. Esetünkben (felülről lefelé haladva) az üzenetek összes számát, a publikálók számát és a válasz nélkül hagyott írások számát tüntettük fel

Több mint elmélet

Ha valaki ezek után azt hinné, hogy Smith munkája csupán szociológusok érdeklődését kielégítő kísérlet, az alaposan téved. A kibertér szociális felmérésének nagyon is gyakorlati, az üzletmenetre kiható haszna van. A Netscant magánál a Microsoftnál összesen négyezernél többen használják; naponta a cég ötven-hetven munkatársa fordul a rendszerhez.
A termékmenedzserek számára például rendkívül fontos *(lenne)* a szoftverjükkel foglalkozó hírcsoport*(ok)* figyelése, hiszen az üzenetekből rengeteg visszajelzést kaphatnának. Gyakran ezeken a fórumokon írnak először a programhibákról, arról, milyen funkciókat látnának szívesen, vagy éppenséggel mit éreznek haszontalannak. Mégis, eddig a termékmenedzserek nemigen engedhették meg maguknak a usenetezést, hiszen csak az őket érintő hírcsoport átolvasása napi jó másfél órájukba kerülne. A Netscan gyökeresen változtatott ezen a helyzeten.
„Én most csak annyit mondok nekik: szerda reggelente frissül a Netscan rendszer tartalma az előző hétről begyűjtött információkkal. Akkor szánjanak rá háromnegyed órát – s napi 90 helyett heti 45 perc alatt átfuthatjátok mindazt, amire szükségetek lehet" – magyarázza Smith.

Reggeli előtt Aurora?

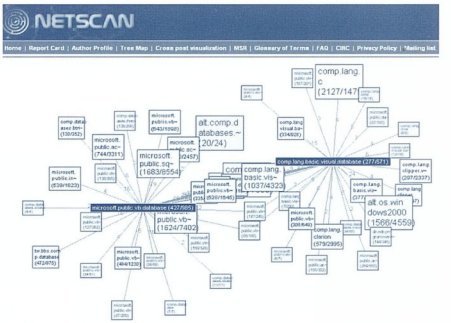
Ám a rendszer kidolgozója más alkalmazási lehetőségeket is lát. Kézenfekvő például a levelezési listák, chat fórumok, belső, céges vitafórumok aktivitásának feldolgozása, tanulmányozása. Ezen messze túlmutat azonban az a Netscanre épülő projekt, amelynek Smith az Aurora nevet adta.

Hiszen a Usenet *(és az Internet)* nemcsak szoftverről szóló információval van tele. Írnak az emberek mindenfajta termékről, ami csak a boltokba kerül, a halkonzervtől a fogkeféig.
Mi lenne, ha a szupermarketben bevásárlás közben hordozható szkennerrel leolvasnánk a kiszemelt termék vonalkódját, s az így a számítógép számára beazonosított termékre elindítanánk egy keresést a Netscan rendszerben?
A gondolatot tett követte. Smith a kéziszámítógépét vonalkódolvasóval szerelte fel, s munkatársaival kidolgozta az ötlet megvalósításához szükséges szoftvert. Attól nem kellett tartania, hogy eláraszták a tévinformációk vagy a félrevezető pletykák. A Netscan erénye éppen az, hogy csak azokat a dokumentumokat jeleníti meg, amelyek megbízhatónak minősített szerzőtől származnak.
Nem kellett sokat várni a mindennapokban kamatoztatható eredményre. A kutató kedvenc gabonapelyhéről kiderült: allergiásoknak veszélyes lehet, mert – noha a dobozon nem tüntették fel – tojást és mandulát is tartalmaz. Arról a zöldborsókonzervről pedig, amelyet rendszeresen vásárolt, az a hír járta, hogy rovarirtó-nyomokat fedeztek fel benne.
A sikeren felbuzdulva a munkacsoport vonalkóddal látta el a Microsoft képzőművészeti gyűjteményének darabjait. Ettől fogva a rendszertől az alkalmazottak megtudhattak minden háttérinformációt az alkotásokról, sőt, azt is, kollégáiknak melyik mennyire tetszett.
Smithék most azt tervezik: vonalkóddal látják el a Microsoft telephely sok egyforma épületének folyosóit is. Így, ha valaki eltéved, a Netscan megmondhatja neki, hogyan juthat célba, sőt, akár arról is felvilágosítást adhat, hol van a legközelebbi nyomtató.

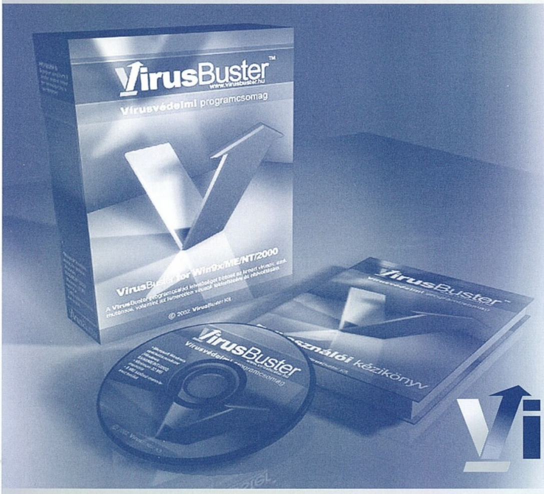
Mikolás Zoltán
mikolas@ebookone.com
A szerző e-tartalom tanácsadó, szakújságíró

Kapcsolódó URL:

Netscan (Smith, Marc. 2001. Netscan: A tool for measuring and mapping social cyberspaces) – http://netscan.research.microsoft.com



A Netscan a hírcsoportok közötti kapcsolatok grafikus megjelenítésére is képes. Két fórum között akkor rajzolódik ki vonal, ha keresztpublikálás történik, azaz ugyanazt az üzenetet szerzője egyszerre két vagy több hírcsoportba is elküldi. A gráf szemléletes képet ad a témák (illetve az őket érdeklődéssel kísérő társadalmi csoportok) egymáshoz kapcsolódásáról, ami egyébként szinte bizonyosan észrevétlenül maradna



Szeletelő

Helyes megfejtés esetén a kiemelt négyzetekbe került betűket felülről lefelé folyamatosan összeolvasva Szmolinka Rezső rejtvényszerkesztő, a FeleSkandi rejtvényújság kiadójának gondolatát ismerheti meg. A kiadó elérhető: Telefon/fax: 238-0109, crossword@mail.datanet.hu
A helyes megfejtéseket az (1) 472-1215-ös faxszámra vagy az info@netacademia.net címre várjuk.

Szabad terjesztésű szoftver ------------------>					21				27	24	20	31	27
Nyalánk ------------------>					14	16	27	21	21	8			22
Delegáció ------------------>				4	28	23	16				21	14	10
Csak részben önműködő szerkezet ------------------>			25				1	13	29	19	20	13	20
A társaság vendéglátója ------------------>					5	6					8	16	20
Lumbágó ------------------>					12	16	27	10				9	20
A számítógépes képrögzítés egyik fajtája --------------->	17	12	30	27	23				25	12	4	1	21
Certificate ------------------>				12	10				23	2	6	7	3
Reklámok kölcsönösen ingyenes megjelenítése a honlapokon --->			9	20			27	31	26	21	27	31	27
Függőleges tabulálás ------------------>			2	27	31	13	12	26					9
Két szám hányadosának szófaji megnevezése ------------------>			13	11				8	6	19	7	14	2
Isolation ------------------>			27	23	4	28	23			15	13	14	21
Examination ------------------>									10	6	23	20	13
Zeneker vezénylése ------------------>					16	12	31	12				6	21
Ellenőrző összehasonlítás ------------------>		2	27	31	12	25	12	26			12	29	7
Kizárólag az interneten működő vállalkozások neve ---------->			16			26	29	19	26	14	10	27	4
Érvényesítés ------------------>							12	16	20	13	12	29	7
Munkaőrült ------------------>				24	29	31	4					12	26
Error message ------------------>							20	28	8	27	7	27	13
Egységes halmazba rendezett tényszámok ------------------>					20	16	20	13	9			12	21
Működésbe hozás ------------------>					20	4		12	2			6	21

Microsoft .net

Microsoft .NET fejlesztői tavasz

Biztosan induló Microsoft .NET fejlesztői tanfolyamok a SZÁMALK Továbbképzésnél

Szeretne többet megtudni a Microsoft új fejlesztőeszközéről, a Visual Studio.NET-ről? Megismerné az adatkezelési technikákat, webes és Windows alapú alkalmazások fejlesztését? Ráadásul mindezt párhuzamosan két programnyelven?

Március végétől minden hónapban Microsoft .NET fejlesztői tanfolyamok a C# vagy Visual Basic .NET programozási alapismerettel már rendelkező szakemberek részére.

A közeljövőben induló képzések

2071 – SQL adatbázis alapok, lekérdezések (Querying SQL Server 2000 Databases)	március 31 – április 1.	79.000 Ft	**–10.000 Ft**
2389 – .NET adatkezelési technikák (Programming with ADO.NET)	április 2–4.	129.000 Ft	**–10.000 Ft**
2555/2565 – Windows alkalmazásfejlesztés (Developing Microsoft .NET Windows Applications)	április 22–24.	129.000 Ft	**–10.000 Ft**
2310 – Webes alkalmazásfejlesztés (Developing Microsoft .NET Web Applications)	május 26–30.	149.000 Ft	**–15.000 Ft**
2524 – XML és webszolgáltatások fejlesztés (Developing XML and Web Services)	június 30 – július 2.	129.000 Ft	**–10.000 Ft**

Minél TÖBB tanfolyamra jelentkezik, annál OLCSÓBBAN juthat hozzá!

Egyszerre több fejlesztői kurzusra történő jelentkezés esetén az ötnapos tanfolyamok díjából 15.000 Ft, a többiből 10.000 Ft kedvezményt érvényesíthet tanfolyamonként!

A kurzusok biztosan indulnak. Ne hagyja ki!
Jelentkezzen, amíg a szabad helyek tartanak!

Microsoft CERTIFIED Technical Education Center

SZÁMALK TOVÁBBKÉPZÉS

SZÁMALK

☎ 203-0304/3050 (Simon Ferenc) **www.szamalk.hu/tisza** 1115 Budapest, Etele út 68.

Windows 2003 Server Expert megrendelőlap

Címzett: NetAcademia Kft.
Faxszám: (1) 472-1215

fax

Windows 2003 Server Expert Workshop (2 nap)

A Windows 2003 technológia újdonságainak felfedezése.

Szakítson időt a Windows 2003 újdonságainak felfedezésére! Nálunk két nap alatt megtudhatja mindazt, amit egyedül másfél év alatt lehet összeszedni!

Témakörök:

Az Active Directory újdonságai

Erdők közötti tranzitív trust; tartományok átnevezése; sémabővítés deaktiválása, felülírása; kezdeti replikáció mentésből (DCPROMO); Global Catalog-mentes bejelentkezés; Active Directory Application Partition és az Active Directory-integrált DNS újdonságai; Forest és Domain Functionality Level, frissítéshez: ADPrep; LDAP-újdonságok: TLS-támogatás, TTL a bejegyzéseken, Digest autentikáció, Virtual List View, elmenthető lekérdezések; Active Directory Migration Tool, User State Migration Tool;

A Group Policy újdonságai

Resultant Set of Policy; Új házirendcsomagok: NetLogon, Credential Manager, Terminal Services, kliensoldali DNS-beállítások; Software Restriction Policies; Administrative Templates Web View; A GPO korlátozása WMI-szűréssel; Group Policy Management Console: GPO mentés-visszaállítás, export-import

Fejlesztői újdonságok rendszergazdáknak

Hasznos programocskák készítése (Visual) Notepad segítségével: .NET Framework; ASP.NET; ADO.NET

Rendszerújdonságok

GUID Partition Table; Headless Server; Automated System Recovery; Volume Shadow Copy; új WMI Providerek: a replikációhoz, a trust-kapcsolatokhoz, DFS-hez, nyomtatókezeléshez; Parancssori WMI (WMIC.EXE)

Biztonsági újdonságok

EFS: több felhasználó, titkosítás WebDAV-megosztáson
Certificate Services újdonságok: auto-enrollment felhasználók számára is, Certificate Mapping, módosítható tanúsítványsablonok, delta CRL-támogatás, központi kulcstárolás, kulcsvisszaállítás
Software Update Services

IIS 6 újdonságok

http.sys és Web Administration Service; Worker Process Isolation Mode; Web Garden
Demand Start; URL Authorization; XML Metabase; POP3 és SMTP szolgáltatás

Nálunk mindent kipróbálhat!

Jelentkezési adatok:

A WorkShop ára **140.000,-Ft**, mely díj magába foglalja a tananyag és az ebéd költségét.

Cégnév: ..... Jelentkezők száma: .....

Cím: .....

Email: ..... Telefon: .....

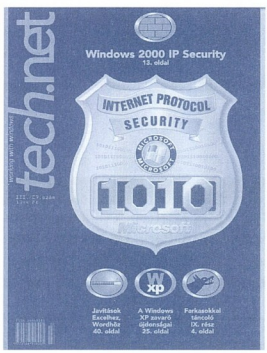
Időpontok: ☐ 2003. április 10-11. ☐ 2003. május 12-13. ☐ 2003. június 23 - 24.

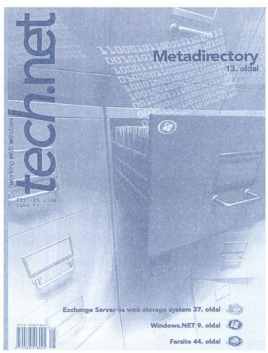
Információ: **http://www.netacademia.net/workshop/w2003/**
Elérhetőségeink: **Tel.: 472-1214 Fax: 472-1215 info@netacademia.net**

tech.net magazin előfizetési szelvény

fax

C mzett: NetAcademia Kft.
FaxszÆm: (1) 472-1215





http://technet.netacademia.net/subs/szamrend.asp • e-mail: terjesztes@netacademia.net • fax: (1) 472-1215

Előfizetem a **tech.net** magazint:

..... példányban egy évre (14.784 Ft)

..... példányban fél évre (7.392 Ft).

Az előfizetés kezdete: .....

Megrendelő neve: .....

Cég neve: .....

Számlázási cím: ☐☐☐☐ .....

.....

Postázási cím*: ☐☐☐☐ .....

.....

E-mail cím: .....

Telefon: .....

Fax: .....

A fizetés módja: ☐ csekkel ☐ átutalással

Kelt: ...../ ...../ ..... Aláírás: .....

*Amennyiben a számlázási cím nem egyezik meg a szállítási címmel, kérjük az alábbi részt is töltse ki!