Windows 2000 • Domain Naming System
Encrypting File System
BackOffice • Biztonságos Exchange levelezés
Transact SQL
Dupla KV
Developer • XML
Office 2000 • Server Extensions

tech.net

A Microsoft Magyarország szakmai magazinja

2000.11.

I. Évfolyam 3.szám / Ár: 899 Ft

KÖSZÖNTŐ
Fóti Marcell főszerkesztő

Technoblabla

Három kiadott lapszámmal a hátunk mögött *(szeptember, .net különszám, október)* igen jelentős múlttal rendelkező lappá váltunk. Rengeteg témáról írtunk az eddig megjelent több mint 150 oldalon. Egyik közeli ismerősöm *(aki annyira közeli, hogy őszinte mer lenni)* meg is kérdezte, hogy ha ilyen ütemben haladunk, vajon mikorra fogyunk ki a témából? Mit lehet még mondani a Window 2000-ről, és a .net Enterprise családról?

Nos, a közeljövőben szeretnénk írni a Lightweight Directory Access Protocolról *(LDAP)*, az Encrypting File Systemről *(EFS)*, a hálózati adatfolyam-titkosítás szabványos megoldásáról, az IPSecről, szeretnénk egy kicsit felderíteni az NTFS rejtelmeit, hisz sokat változott az elmúlt verzió óta. Míg a Windows 2000 bétaváltozatainban még szerepelt az Automatic System Recovery *(ARS)*, mellyel elvileg egyetlen varázsflopi segítségével vissza lehetett volna állítani elhalt operációs rendszerünket, a végleges termékben már megint az ügyetlen Emergency Repair Disk *(ERD)* szerepel, de most már használható módon. Nincs többé SP issue!

Napjaink nyitott hálózataiban roppant fontos a megfelelő titkosítási algoritmusok használata. A nyílt kulcsú titkosítás nem képzelhető el a Ronald Rivest, Adi Shamir és Leonard Adleman matematikusokról elnevezett RSA algoritmus nélkül, melyet érdemes legalább egyszer az életben végigküzdeni, mert utána az ember büszkén nézhet magára: ezt is értem!

Biztosan sokan találkoztak már a 169.254.0.0 IP címtartományba eső címekkel, melyet az operációs rendszer „csak úgy" magáévá tett. Itt az automatikus IP cím felvételel *(APIPA)* van dolgunk, de hogyan is működik?

Sokakat fog érinteni az X.500 alapú címtárak replikációja, hisz az Active Directoryt sok vállalatnál össze kell kötni Novell NDS-sel, vagy valamilyen UNIX-on futó X.500 címtárral. Ilyenkor jön jól az LDIF protokoll ismerete.

A távoli eljáráshívás, Remote Procedure Call *(RPC)* ugyan nem új technológia, de mélyebb ismerete meg nem árthat, hisz könnyebben fogjuk venni az „RPC Service is unavailable" néven ismert akadályt.

Nagyvállalati környezetben a Windows for Workgroupsnál bevezetett egyenrangú hálózat remekül szolgálta ki a 10-12 fős munkacsoportokat, hisz minden gépen meg lehetett osztani a fontos könyvtárakat, és végre meg lehetett szabadulni a flopis adatátviteltől. Ugyanakkor ha egy vállalatnál nem 12, hanem 612 gép osztja meg erőforrásait, azt káosznak hívjuk. Ezen a problémán segít a vállalati megosztott erőforrások egységes faszerkezetbe rendelésével, felfűzésével a Distributed File System *(DFS)*, mely ráadásul hibatűrő képességgel is felruházhatóvá teszi a fájlkiszolgálókat. Hogyan?

Ugyanide tartozik a megosztásoknál elsődlegesen használt Server Message Block *(SMB)* protokoll, mely nemcsak Windowsos környezetben használható, hanem UNIX-okon is, hisz a Samba Serverek SaMBa kiszolgálók. Még Windows 2000 tartományba is betehetjük őket, ha...!?

A Windows 2000-rel elérkeztünk a PnP operációs rendszerek korába. Ha ez csak annyit jelent*(ene)*, hogy végre szabadon klónozhatjuk a tökéletesre csiszolt operációs rendszer tőpéldányt, már megérte *(volna)*. De ennél sokkal többről van szó. A PnP összenő az ACPI-val, hibernálást, standby üzemmódot, és egyéb hasznos szolgáltatásokat kínálva mindenkinek. Már csak egy lépés hiányzik a gyönyörhöz, az OnNow technológia, ami lehetővé fogja tenni, hogy a bekapcsolás után ne kelljen öt percet várni, amíg a gép magához tér, hanem azonnal használatba lehessen venni, mint most egy televíziót.

Security Configuration Manager. Ezzel lehet tűélesre állítani a vállalat számítógépein a biztonsági beállításokat. Olyannyira, hogy ha ügyetlenek vagyunk, eljuthatunk a használhatatlanul biztonságos számítógépig, mely süket és vak, de feltörni sem lehet, az biztos!

Ezzel szinte egy időben, mondhatni az előzőekkel párhuzamosan fog elkészülni APM, WMI, SMP, NCP, QoS, L2TP, HTTP, IMAP, TCP/IP, SCE, IGMP, LanMan, ADSI, 4GT és UDP, BAP, NFS, LDIFDE, NBTSTAT, COM+, S/MIME, RSVP, CHAP, RIS, HMAC elemző cikkünk.

De természetesen nem fogunk megfeledkezni olyan fontos technológiákról sem, mint a RAID, ActiveX, SMTP, USB, ARP, SSL, TLS, DDNS, AWE, TGT, SAM, XML.

Műhelymunka szintjén még az idén terítékre kerülhet a SIS, IrDA, TAPI, PPP, MD5, KDC, SNMP, MMC, ICMP, WSH, PAE, bár valószínűleg cikk formájában nem fogunk elkészülni vele.

Azután jó néhány cikket, mi több egész cikksorozatot érdemel az ISAKMP, KMS, NTLM, RIP, NAT, Kerberos, WINS, ACPI, PPTP, LPD, VBS, NDIS, ADO, OSPF, NWLink, ASP, MSMQ, SCSI, GPO, RADIUS, MAPI.

Ezekkel folyamatosan fogunk megjelenni a megfelelő rovatok hasábjain.

És végül, de nem utolsó sorban írni fogunk az SHA, GC, POP3, FSMO, PAP, WebDAV, SID, BOOTP, VPN, IXFR, USN, FTP, CSVDE, MFT, X.500, DHCP, TGS, SNTP, GRE, SLIP, HTML, GDI, UTF-8, JAVA, CA, RAS, NetBIOS, IAS, BIND, PKCS technológiákról, valamint az Extensible Authentication Protocol (EAP)-ról.

RFC rovatunk pedig már van is :-)

Ezek lennének rövid-, közép-, és hosszútávú céljaink. Tehát a kérdés nem az, hogy miről fogunk írni, hanem inkább az, hogy meddig tudjuk 48 oldalba zsúfolni havi 64 oldalnyi anyagunkat?

Fóti Marcell
MCT, MCSE, MCDBA
marcellf@netacademia.net

Microsoft

tech.net

A Microsoft Magyarország Szakmai Magazinja

Szerkesztőség
Főszerkesztő: **Fóti Marcell**
marcellf@netacademia.net
Főszerkesztő-helyettes: **Fülöp Miklós**
mick@netacademia.net
Szerkesztőség címe:
1105 Budapest, Ihász utca 13.
Tel.: 263-2732
technet@netacademia.net
Nyilvános levelezési lista:
technet@lyris.netacademia.net

Kiadja a **Microsoft Magyarország**
1031 Budapest, Graphisoft park 3.
Tel.:437-2800
A kiadásért felel:
Arany Tóth László
v-laarto@microsoft.com

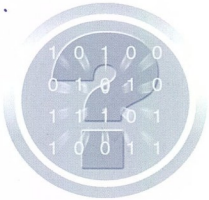
Terjeszti a **NetAcademia Kft.**
Terjesztési, előfizetési információ:
Tel.: 263-2732
terjesztes@netacademia.net
Megjelenik havonta, ára 899 Ft

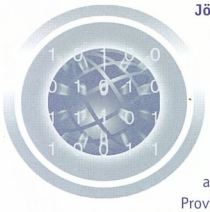
Előfizethető megrendelőlevélben a szerkesztőségnél:
1105 Budapest, Ihász utca 13.
Fax: 261-7145
http://technet.netacademia.net/subs

Hirdetésfelvétel:
Bársonykalapács Marketing
Felelős: **Udvarev Rita**
Tel./Fax: 214-0923
velvethammer@ahol.com
1027 Budapest, Fő utca 67. V. 1.
Grafikai tervezés, kivitelezés, nyomdai előkészítés:
Bársonykalapács Marketing
Művészeti vezető: **Balogh Zoltán**

Nyomda:
Partner's 2000 Kft.
1124 Budapest, Sion lépcső 7.
Felelős nyomdász: **Galambos Sándor**

ISSN 1586-5185





Jövőre megváltozik a Solution Provider program.

Olyannyira, hogy a neve is más lesz: Microsoft Certified Partner lesz az új hivatalos megnevezés. Az új csatlakozási feltételekkel párhuzamosan a Microsoft ígéretei szerint a partnereknek nyújtott szakmai támogatás is megváltozik, hogy minél több partnercég vehessen részt a .NET stratégia sikerében. Az eredeti Solution Provider program 1992 óta működik, napjainkban világszerte 31 ezer cég él a partnerség adta lehetőségekkel.

Minden előzetes várakozással és híreszteléssel ellentétben megjelent a Windows NT 4.0-ra telepíthető Active Directory ügyfélszoftver!

A kiadott, letölthető szoftver a következő fontos szolgáltatásokkal rendelkezik:

- Telephelyek figyelembe vétele *(Site Awareness)*, mellyel a munkaállomás mindig a legközelebb eső tartományvezérlőnél jelentkezik be
- Jelszóállítás tetszőleges DC elérhetősége esetén *(nem vár a PDC-re, ha az nem elérhető)*
- Active Directory Services Interface *(ADSI)* parancsfájlok futtatási lehetősége
- DFS hibatűrést kihasználó ügyfél, mely képes áttérni a DFS által megadott tartalék fájlmegosztás használatára, ha az elsődleges kiesik
- Active Directory Windows Address Book *(WAB)* tulajdonságlapok kezelése. A címjegyzéken véghezvitt módosítások *(megfelelő jogosultságok esetén)* bekerülhetnek az Active Directoryba
- Az Active Directory sémamódosítások által igényelt felhasználói felület módosítások *(display specifiers)* is rátelepíthetők, így NT 4.0-ról is elvégezhető a címtár karbantartása.
- NT LAN Manager version 2 hitelesítés. Lásd cikkünket a 29. oldalon

Ugyanakkor meg kell jegyeznünk, hogy e kiegészítő telepítése után sem válik a víz vérré – nem lesz az NT 4.0-ból hirtelen Windows 2000 Professional. Nem lesz elérhető továbbra sem:

- A Kerberos bejelentkezés
- A csoportos házirend *(Group Policy)* nem értékelődik ki
- Nincs Intellimirror
- Nincs IPSEC és L2TP támogatás

Ezen hiányosságok ellenére is nyújt azonban jó pár fontos megoldást, így érdemes lehet letölteni az alábbi címről *(lapzártakkor Beta változatban)*:

http://www.microsoft.com/WINDOWS2000/adclients/default.asp

Hackerek hatoltak be a Microsoft belső hálózatába

A hazai sajtóban is megjelent a hír a Microsoft belső hálózatán észlelt rendellenes ténykedésről. A Microsoft biztonsági csapata október 14-én észlelte a behatolást, és figyelemmel kísérte a hacker lépéseit, melyek arra utaltak, hogy kísérletet tesz a belsőbb hálózatok elérésére is. A nyomozás folytatódik, de az már bizonyos, hogy a behatolást nem valamelyik Microsoft kiszolgálótermék biztonsági hibája, hanem emberi mulasztás okozta. A támadó úgynevezett féregprogramot *(QAZ worm, közismert fájltörlő és jelszólopó program)* küldött el az egyik gyanútlan alkalmazottnak, aki a csatolt fájl elindításával megfertőzte saját gépét. A worm ezután a felhasználó jogosultságaival élve tevékenykedett a hálózaton. A korábbi bejelentésekkel ellentétben nincs bizonyíték arra, hogy a forráskódokhoz bárki hozzáfért volna. A Microsoft szorosan együttműködik az USA hatóságaival a nyomozás lefolytatásában.

Már megint új kiszolgálótermék: TAHOE

A magyarul kicsit tahó hangzású terméknév mögött igazi újdonság rejlik: valódi portál és dokumentumkezelő alkalmazás van születőben! Félre veled NTFS! A terméket a későbbiekben részletesen is ismertetjük, most azonban lássunk egy-két régen várt szolgáltatást!

- Portálfunkciók: szabad keresés, és navigálás a publikált dokumentumok között. Ha ez idáig nem túl meggyőző, hadd említsem meg az előfizetési/feliratkozási lehetőséget, amely arra szolgál, hogy a Tahoe emailben kiértesítsen, ha egy mappában változás történik, például új dokumentum születik.
- Dokumentumkezelés: verziókövetés, dokumentum-életciklus kezelés, check-in, check-out műveletek a dokumentumtár egységességének, ellentmondás-mentességének fenntartására. A dokumentumok tetszőleges tulajdonságainak *(szerző és egyebek)* megjelenítése a könyvtárlistákban

A Tahoe Server alatt ugyanaz a Web Storage Engine dolgozik, mint amelyre az Exchange 2000 is épül.

Mi lesz az SMS sorsa?

A ma elérhető Systems Management Server utódaként fog megjelenni valamikor 2001 közepe táján a Microsoft új rendszerfelügyeleti terméke, a Microsoft Operations Management nevű termék, mely a ma elérhető szolgáltatások körét kiegészíti a realtime adatokon alapuló elemzés képességével. Ehhez a Microsoft megállapodást kötött a NetIQ céggel, akiknek már van Windows 2000-re készített realtime felügyeleti megoldásuk. A termék piacra lépése azért esik a távoli jövőbe, mert nem csupán a különféle termékekben meglévő felügyeleti szolgáltatások összeboritásáról van szó, hanem ezek integrációjáról és a .NET kiszolgálókkal történő illesztéséről is. A .NET Management Services segítségével a meglévő rendszerfelügyeleti funkciók *(WMI,MMC)* elérhetőek lesznek hálózaton, XML felületen keresztül is. Ezáltal az SMS utóda fejlesztési platformmá válik, melyhez az eddigieknél lényegesen egyszerűbb lesz külső gyártóknak kiegészítéseket írniuk. Ennek a folyamatnak az elősegítésére jött létre a Microsoft Management Alliance szervezet, melynek feladata technikai- és marketingtámogatás nyújtása a rendszerüket a .NET Management Services felé megnyitó alkalmazásfejlesztők számára.

...és még egy új kiszolgálótermék: a Mobile Information Server

Erről a kiszolgálótermékről egyelőre nem lehet sokat tudni, de az bizonyos, hogy célja a vállalati hálózatok elérhetőségének biztosítása mobil kommunikációs készülékek segítségével. Hogy mik ezek az eszközök, azt ne firtassuk, s hogy mik is lennének a telefonon elérhető szolgáltatások, azt is inkább csak találgassuk: ugye jó lenne az elektronikus levelezés használata? Hát a névjegyek és naptári bejegyzések elérése? A madarak csicsergéséből az Outlook Mobile Access *(OMA)* szavakat vélem kihallani :) De amint a Microsoft weblapján található technikai leírás mondja: az OMA csak a jéghegy csúcsa. Kár, hogy nem látunk a vízfelszín alá!

Whistler

Október 31-én a Microsoft bejelentette, hogy bétaváltozatban elkészült a Windows 2000 utóda, a jelenleg Whistler kódnéven futó Windows.NET. Tudjuk, hogy sok vállalatnál éppen csak belefogtak a Windows 2000-re történő átállásba. Számukra talán megnyugtató hír, hogy a terméket csak 2001 második félévére ígérik, és ez az időpont szokásosan csúszni fog. A Beta 1 általában inkább csak gondolatfoszlánynak tekintendő, mintsem terméknek. Élesen emlékszem a Windows NT 5.0 Beta 1-re *(a Windows 2000 „leánykori neve")*, mely annyira hasonlított a Windows NT 4.0-ra – hogy az is volt, csak rá lehetett telepíteni az Active Directory egy korai verzióját.

És egy hír saját házunk tájáról

NetAcademia – AdAstra együttműködés

A NetAcademia Kft. Sikeresen leszerződtette a jövő évadra az AdAstra Rt. munkatársait, hogy egy kis színt vigyenek a lap egysíkú illusztrálásába. Az AdAstra egy kitalált cég, ahol a dolgozók mindennapját áthatja az informatika. A „kollégákat" Pepe, a http://rajz.lap.hu főszerkesztője alkotta számunkra. A decemberi számtól kezdve minden hónapban bepillantást nyerhetünk a cég zaklatott informatikai életébe. A vállalat szervezeti felépítését itt tekinthetik meg: **http://technet.netacademia.net/adastra**

WINDOWS 2000
Domain Naming System

Windows 2000 DNS

Ahhoz, hogy a hálózati számítógépek TCP/IP nyelven beszélgethessenek egymással, mindegyikükhöz egyedi IP címet kell rendelnünk. Míg a gépek számára ezek használata a legkevésbé sem jelent gondot, az embereknek hamar meggyűlik vele a baja. A felhasználók szemszögéből a kommunikáció akkor hatékony, ha a számítógépekre nevekkel hivatkozhatnak, miközben azok továbbra is az IP címüket használják egymás közt.

A mai Internet őse, az **ARPANET** kezdetben néhány számítógépből álló hálózat volt, melyek nevét és IP címeit egyetlenegy szövegfáljban, az úgynevezett **HOSTS.TXT**-ben tartották nyilván. A Network Information Center *(NIC)* frissítette az összes számítógép nevét és címét a változásokról beérkező elektronikus levelek alapján. Az **ARPANET** felhasználói ezután **FTP** protokollal letölthették a legfrissebb **HOSTS.TXT** fájlt.

Az ARPANET növekedése rávilágított arra, hogy ez a módszer nem méretezhető:

- A HOSTS.TXT fájl terjesztése az ARPANET-hez kapcsolódó gépek számának négyzetével arányos sávszélességet igényelt. Mivel a gépek száma exponenciálisan növekedett, nyilvánvaló lett, hogy ezt a terhelést egyetlen számítógép nem bírja el.
- Az ARPANET-en két számítógép nem vehetett fel azonos nevet. A gépek számának növekedésével a statikus, hierarchiát nélkülöző HOSTS.TXT fájl miatt megnőtt az azonos nevek kiadásának kockázata, és a központi nyilvántartás is egyre nehezebbé vált.
- A hálózat természete megváltozott: az ARPANET egykori nagy, időosztásos számítógépeit felváltották a munkaállomásokból álló hálózatok, amelyek mindegyikének egyedi nevet kellett viselnie. Ennek kezelése központilag nem lett volna megoldható.

Jobb megoldást kellett találni. Számos indítvány született a hierarchikus névteret használó, elosztott névszolgáltatásra. Megszületett a 882-es és a 883-as RFC, amelyek leírják az általános erőforrásadatokat tartalmazó, elosztott adatbázisra épülő tartomány névrendszer *(Domain Name System – DNS)* felépítését. A további fejlődés miatt kiadták az 1034-es és az 1035-ös RFC-t, amelyek az Interneten ma is használt DNS leírását tartalmazzák. A DNS folyamatosan fejlődik – e sorok írásakor is több fejlesztési javaslat tárgyalása folyik.

A Windows 2000 DNS áttekintése

A számítógépek közötti kommunikációt elősegíti, ha a számítógépeknek ugyanabból a névtérből adunk nevet, amely egyben meghatározza a névadás szabályait és a nevek IP címmé történő átalakításának módját. Ahhoz, hogy a számítógépek megszólíthassák egymást, a neveket IP címekkel kell helyettesíteniük, amit a névlekérdező szolgáltatás segítségével végeznek el.

A Windows 2000 két fő névteret és névátalakítási módszert használ: a Windows Internet Naming Service *(WINS)* szolgáltatással megvalósított NETBIOS-t és a cikk témáját adó DNS-t. A Windows 2000 más névtereket is támogat, például a Novell Netware-t és Banyan Vines-t.

Mi a DNS?

A DNS az IETF névszolgáltatási szabványán alapuló szolgáltatás, amellyel a hálózati számítógépek megvalósíthatják a DNS tartománynevek bejegyzését és átalakítását. Ezeket a neveket használjuk például az Internethez csatlakozó számítógépek erőforrásainak kereséséhez és használatához is.

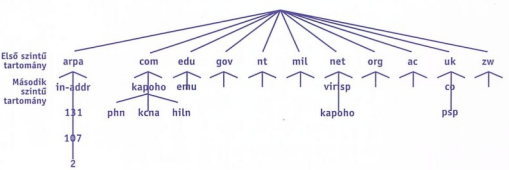
A DNS három fő alkotórészből áll:

- A tartomány névtér és a kapcsolódó erőforrásrekordok elosztott nevekre vonatkozó adatbázist alkotnak.
- A DNS névkiszolgálók tárolják a tartománynévteret és az erőforrásrekordokat, továbbá válaszolnak a DNS ügyfelek kérdéseire.
- A DNS ügyfelek részét képező DNS lekérdezők *(resolver)* felveszik a kapcsolatot a névkiszolgálókkal, és névlekérdezéseket küldenek, hogy hozzájussanak az erőforrásrekordokhoz – az IP címekhez.

A legfontosabb alapfogalmak

Tartománynévtér

A tartománynévtér hierarchikus, faszerkezetű. A DNS névtérben a tartománynévtér fájának minden csomópontja és levele egy névvel ellátott tartományt jelöl. Minden tartománynak lehetnek gyermektartományai. Az alábbi ábrán az Internet tartománynévterének szerkezete látható.



Az Internet tartománynévtere

Az ábra szerint a DNS fa minden csomópontjának saját neve, illetve az 1034-es RFC szóhasználatával élve címkéje van. A DNS címkék hossza 1–63 karakter lehet – kivétel a gyökértartomány *(root)* neve, melynek hossza nulla karakter.

Egy csomópont tartománynéve a fa gyökerétől a csomópontig tartó útvonal címkéiből tevődik össze, amelyeket megállapodás szerint balról jobbra, a gyökértől legtávolabbi címkével kezdve olvasunk, az eredményt pedig teljes tartománynévnek *(Fully Qualified Domain Name – FQDN)* hívjuk *(például www.falatrax.hu)*.

A tartománynevekben szerepelhetnek kis- és nagybetűk is, de az 1034-es RFC-nek megfelelően ezt egyetlen DNS művelet sem veszi figyelembe. A www.falatrax.hu, WwW.FalATraX.Hu és a WWW.FALATRAX.HU tehát a tartománynév-műveletek szempontjából azonosak.

Windows 2000

Elsőszintű tartománynév

Az elsőszintű tartománynevek közvetlenül a gyökér alatt helyezkednek el. Az előző ábrán is látható, hogy több ilyen tartomány létezik, de a továbbiak létrehozása – legalábbis az Interneten – nem egyszerű feladat. Az elsőszintű tartománynevek három csoportba sorolhatók:

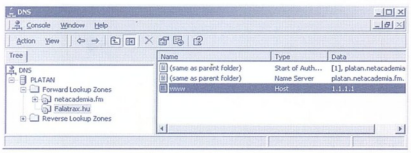
- Az „ARPA" különleges tartomány, ma már csak fordított névlekérdezésre használjuk.
- A hárombetűs tartományneveket az 1591-es RFC definiálja. Jelenleg az alábbi táblázatban látható hét név létezik, de a fokozott igény miatt a számuk a jövőben várhatóan nőni fog.
- A kétbetűs tartománynevek megegyeznek az International Organization for Standardization (ISO) országneveivel, és elsősorban az Egyesült Államokon kívüli szervezetek használják. Az egyetlen kivétel az Egyesült Királyság, amelynek jele az ISO szerint GB, az Interneten mégis a .uk honosodott meg.

tartománynév	felhasználási kör
com	üzleti tevékenységet folytató szervezetek *(a Microsofté például microsoft.com)*
edu	oktatási intézmények, elsősorban négyéves főiskolák és egyetemek *(a Carnegie Mellon University-é például cmu.edu)*
gov	USA szövetségi kormányügynökségek *(az FBI-é például fbi.gov)*
int	nemzetközi egyezmények alapján létrejött szervezetek *(például a NATO-é nato.int)*
mil	katonai célú tartomány az Egyesült Államokban *(a US Air Force-é például af.mil)*
net	Internettel foglalkozó szervezeteket, Internet- és hálózat-szolgáltatókat stb. takar *(az InterNIC-é például internic.net)*
org	„egyéb" kategória, például nem kormányzati vagy nonprofit szervezetek *(Reikiről például a reiki.org címen olvashatunk)*

Az Internet hárombetűs tartománynevei

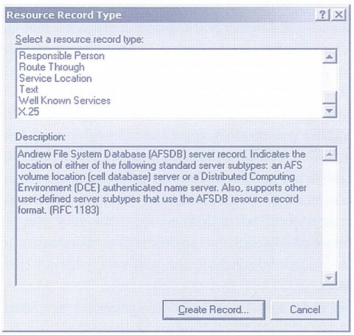
Erőforrásrekordok

Az erőforrásrekordok a DNS adatbázisban tárolt tartományok adatait tartalmazzák, amit DNS ügyfelek használnak. Egy gép címrekordjából például kiderül az IP címe. Az alábbi ábrán a www.falatrax.hu névhez tartozó IP címet (1.1.1.1) láthatjuk Windows 2000 DNS Szerveren. Az **A** rekordok Host néven szerepelnek.



A DNS kiszolgálók azoknak a tartományoknak az erőforrásrekordjait tárolják, amelyekért felelősek, illetve amelyekre vonatkozó kérdéseket képesek megválaszolni. Ha egy DNS kiszolgáló a DNS névtér bizonyos részéért felelős, akkor az erre vonatkozó adatok hitelességét a kiszolgáló rendszergazdájának kell biztosítani. A hatékonyság növelése érdekében a DNS kiszolgálók a tartományfa bármely részének erőforrásrekordjait képesek ideiglenesen tárolni.

Az 1035-ös, 1036-os és még néhány későbbi RFC több erőforrásrekord-típust határoz meg. Bár ezek többségét már nem használjuk, a Windows 2000 még támogatja alkalmazásukat. Az alábbi ábrán az igen híres Andrew File System Database *(AFSDB)* erőforrásrekord látható, több másik ismeretlen rekorddal egyetemben, amelyeket valószínűleg újszerű célból, kollégáink megtréfálására tudunk csak felhasználni :-)



Az alábbi táblázat felsorolja azokat a rekord-típusokat, amelyek leginkább elképzelhetők egy Windows 2000 hálózatban.

rekordtípus	felhasználás
A: cím *(host Address)*	Munkaállomás IP címe.
CNAME: alias	Egy munkaállomáshoz több név is *(Canonical NAME)* tartozhat.
MX: levelezés-szervező *(Mail eXchanger)*	Az elektronikus leveleket levelezéskiszolgálóhoz, illetve ha nem elérhető, pótkiszolgáló*(k)*hoz irányítja.
NS: névkiszolgáló *(Name Server)*	Felsorolja a tartományért, illetve a delegált altartományért felelős DNS kiszolgálókat.
PTR: mutató *(PoinTeR)*	Fordított névátalakítás az inaddr.arpa tartománnyal.
SOA: felelősség kezdete *(Start Of Authority)*	Meghatározza a zóna elsődleges DNS kiszolgálóját és egyéb jellemzőket is.
SRV: szolgáltatás-azonosító *(SeRVice locator)*	Megmutatja, hogy adott szolgáltatás melyik kiszolgálón fut. Az Active Directory SRV bejegyzéseket használ a tartományvezérlők, a globális katalógus és az LDAP kiszolgálók azonosítására.

A leggyakoribb erőforrásrekord-típusok

Bár erőforrásrekordok a DNS fa bármely csomópontjához hozzárendelhetők, egyes típusok több tartományban egyáltalán nem szerepelnek *(PTR bejegyzés például csak az in-addr.arpa alatti tartományokban található)*. A felsőbb szintű tartományoknak *(például microsoft.com)* saját erőforrásrekordjaik lehetnek *(például MX bejegyzés a Microsoftnak küldött levelek irányítására)*, illetve további altartományok kapcsolódhatnak hozzájuk, amelyeknek szintén lehetnek saját erőforrásrekordjaik *(az eu.microsoft.com altartománynak például lehet egy www.eu.microsoft.com címrekordja)*.

Alias

Aliasok segítségével ugyanaz a tartomány többféleképpen is elnevezhető. CNAME bejegyzések használata a következő esetekben javasolt:

- Az **A** bejegyzéssel megadott számítógépet ugyanabban a zónában kell átnevezni. Ha például a gep.falatrax.hu nevet masina.falatrax.hu-ra kell változtatni, akkor készíthetünk egy CNAME bejegyzést, miszerint a gep.falatrax.hu a masina.falatrax.hu-ra mutat.
- Egy közismert szolgáltatás, például ftp vagy www, több kiszolgálón elosztva fut, így az általános névnek több gépre kell mutatnia. Lehet, hogy a gep.falatrax.hu és a masina.falatrax.hu nevekhez például egy közös www.falatrax.hu aliast szeretnénk rendelni. A felhasználók az utóbbit használják, és fogalmuk sem lesz arról, hogy a kéréseiket melyik kiszolgáló teljesítette.

DNS lekérdezés

A DNS ügyfél lekérdezi egy DNS kiszolgálótól a megadott tartomány egy vagy több erőforrásrekordját, például a falatrax.hu tartomány **A** címrekordjait. Ha a tartomány és a kért erőforrásrekord létezik, a kiszolgáló DNS válaszüzenetben küldi vissza a kérésben szereplő adatokat. A DNS válaszüzenet tartalmazza az eredeti kérést és a kérdéses rekordokat, feltéve, hogy a DNS kiszolgáló hozzá tud jutni a szükséges erőforrásrekordokhoz.

A DNS lekérdezés, vagy az 1034-es RFC szóhasználatával élve normál lekérdezés a céltartomány nevét, a lekérdezés típusát és osztályát foglalja magában. A lekérdezés azokra a rekordokra *(akár az összesre)* vonatkozik, amelyeket az lekérdező szeretne megtudni.

DNS frissítés

A DNS frissítést egy DNS ügyfél akkor kéri a DNS kiszolgálótól, ha egy adott tartományban új erőforrásrekordot akar létrehozni, vagy meglévőt módosítani, illetve törölni. Megváltoztathatjuk például a gep.falatrax.hu nevet úgy, hogy a 10.10.1.100 címre mutasson. Ezt a műveletet dinamikus frissítésnek is hívják.

DNS zónák

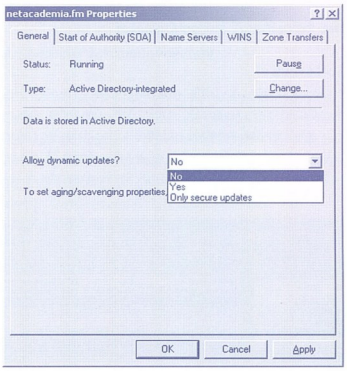
A DNS névtér bizonyos részét felépítő és karbantartó kiszolgáló felelős a kérdéses névtérrészért, a zónáért, amely a DNS replikáció alapegysége is. A zóna tulajdonképpen egy fájl, amely egy vagy több DNS tartomány egy vagy több erőforrás-rekordját tartalmazza. *(A zóna minden „rendes" DNS Serveren egyszerű szövegfájl, míg a Windows 2000 esetében a zóna tartalma az Active Directoryban is tárolódhat. Mivel a zóna fogalma eléggé ismeretlen a kezdő DNS-ezők számára, ezért a cikk hátralévő részébe néha beszúrtam hogy „...azaz fájl...", amit értsünk úgy, ahogy kell: a zónafájl mindaddig fájl valahol a merevlemezen, amíg be nem küldjük az Active Directoryba.)*

A Windows 2000 háromfajta zónát támogat:

- A standard elsődleges *(standard primary)* a zóna eredeti példányát tárolja, amit másodlagos zónákba replikálhat. A zónában történő változások vezetése mindig a normál elsődlegesben történik. Ez a zónatípus tehát egyszerű szövegfájlokban tárolja a DNS rekordokat.
- A standard másodlagos *(standard secondary)* a zónaadatok csak olvasható példányát tárolja, ami növelheti a teljesítményt és a rugalmasságot. A Windows 2000 DNS Szerveren a secondary zónafájl is írhatónak látszik, mert az írási kéréseket ez a kiszolgáló automatikusan átirányítja az írható példányra *(primary)*.
- Az Active Directoryval integrált zóna a Microsoft saját zónatípusa, melynek adatai a Windows 2000 Active Directoryban *(AD)* találhatók, ennek megfelelően a replikációja AD replikációval valósul meg.

A zóna eredeti példányát az elsődleges zóna DNS kiszolgálója tárolja. A zónának itt van egy úgynevezett SOA bejegyzése, amellyel önmagát elsődlegesnek nevezi ki. A teljesítmény és a redundancia fokozására az elsődleges zónaállomány automatikusan átmásolható másodlagos zónákban lévő DNS kiszolgálókra. Ha a zónában változás következik be, például új **A** rekord jön létre, akkor az elsődleges zónaállomány módosul, majd átmásolódik a másodlagos zónákba. A zónaállomány átvitele zónareplikációval valósul meg.

A Windows 2000-ben az első zóna a létrehozásakor csak egyetlen DNS tartománynévre *(például falatrax.hu)* vonatkozó adatokat tartalmaz. Ezután a zónában készíthetünk erőforrás-rekordokat kézzel, vagy engedélyezhetjük a tartomány dinamikus frissítését.



Ha a zóna dinamikus frissítése engedélyezett, akkor a Windows 2000 munkaállomások közvetlenül képesek módosí-

tani a DNS kiszolgáló **A** és PTR bejegyzéseit. Ha a gép egyben DHCP ügyfél is, akkor a DHCP kiszolgáló beállítható úgy, hogy az frissítsen.

Ha a zóna elkészült, további altartományokat kapcsolhatunk hozzá *(például marketing.falatrax.hu)*. Ennek oka lehet például az, hogy egy új épület számára akarunk DNS szolgáltatást biztosítani, amit a szülőtartománytól elkülönülten kell kezelni. Az így létrejött altartományban, amely akár egy külön zónában is lehet, erőforrás-rekordokat kell készíteni *(például címrekordot a alma.marketing.falatrax.hu számítógépnek)*.

Ha a zónát alkotó tartományok valamelyikéhez további tartományokat kapcsolunk, ezek tartozhatnak az eredeti vagy másik zónába *(...azaz fájlba...)*, ahogy ez az alábbi ábrán is látható. A falatrax.hu alatti marketing.falatrax.hu altartomány például lehet ugyanabban, de külön zónában is. Az altartomány így kezelhető az eredeti zóna részeként, vagy delegálható egy másik, az altartomány kezelésére létrehozott zónába.

Zónák és tartományok

A fenti példában a falatrax.hu tartománynak van egy hr.falatrax.hu és egy marketing.falatrax.hu altartománya is. Mindkettő egy-egy cím-rekorddal rendelkezik. Külön zónában találhatók, más-más DNS kiszolgálók kezelésében. A gep.falatrax.hu címrekord a falatrax.hu, a masina.hr.falatrax.hu pedig a hr.falatrax.hu zónába *(...azaz fájlba...)* tartozik.

Active Directory-val integrált zónák

A Windows 2000 DNS szolgáltatásának egyik legfőbb újdonsága, hogy a zónákat képes az AD-ben tárolni. Az AD-val integrált zóna elsődleges DNS zóna, amely az AD replikáció segítségével másolódik át más elsődleges AD zónákba *(nem a hagyományos zónaátvitellel)*. A zónák tárolásának ez a módja a Microsoft egyedi megoldása, de több előnye is van: az ilyen zónák valódi többforrásúak, így a módosítások bármelyik DNS kiszolgálón végbemehetnek. Ez növelheti a DNS szolgáltatás hibatűrését. Az AD replikációnak köszönhetően a zónaállományok átvitele a lassú kapcsolatokon keresztül hatékonyabb lehet, mivel az adatok a telephelyek között tömörítve utaznak.

Figyelem! Az AD integráció és a globális katalógus *(GC)* szerep üti egymást! Erről szól a „DNS Server Generates Event 4011 [Q252695]" című Knowledge Base cikk, mely leírja, hogy ha egy tartományvezérlő egyben globális katalógus kiszolgáló is, akkor bizonyos, DDNS regisztrációt igénylő komponensek előbb indulnak el, mint maga a DNS kiszolgáló, emiatt ronda hibaüzenetek jelennek meg az Eseménykezelőben rendszerindításkor. Megoldási javaslatok:

1. **A zóna ne legyen AD integrált!**
2. **A zóna ne ugyanazon a gépen legyen, és akkor lehet AD integrált!**
3. **A gép ne legyen GC, és akkor maradhat a zóna AD integrált!**

Fordított lekérdezési zónák

A DNS kiszolgálókhoz érkező kérések legtöbbjében a keresés az **A** címrekordban szereplő DNS néven alapul. Ez a lekérdezéstípus a válaszban IP címet vár, és normál lekérdezésnek hívjuk. A DNS fordított lekérdezést is lehetővé tesz, amellyel egy IP címhez tartozó nevet kereshetünk meg – például, mi a DNS neve a 101.10.1.100 IP című számítógépnek?

A fordított lekérdezések támogatására egy különleges, in-addr.arpa nevű tartományt foglaltak le az Internet DNS névteréből. Az altartományok neve az IP címek *(tizes számrendszerbeli)* számjegyeiből képződik, fordított sorrendben, egymástól ponttal elválasztva. A fordított sorrend szükséges, mert bár az IP címeket is balról jobbra olvassuk, de a DNS nevekkel ellenkező irányban értelmezzük *(balról jobbra haladva a cím egyre meghatározottabbá válik)*. Emiatt az in-addr.arpa tartomány építésekor az IP cím oktettjei fordított sorrendben követik egymást. A 192.168.100.0 alhálózat fordított lekérdezési zónája például 100.168.192.in-addr.arpa. Ezzel a megoldással az in-addr.arpa DNS fa alsóbb ágainak kezelése delegálható ahhoz a szervezethez, amelyik a megfelelő IP cím tartományokat megkapja.

Az in-addr.arpa PTR rekordokat használ, amelyek IP címekhez rendelnek tartományneveket. A normál keresés zónájában ezek megfelelői az **A** címrekordok. A fordított lekérdezés akkor eredményes, ha a mutató érvényes, tehát létezik egy hozzárendelt címrekord is.

Az in-addr.arpa tartományt csak az IPv4 *(Internet Protocol version 4)* protokollal működő hálózatok használják. A Windows 2000 DNS MMC *(Microsoft Management Console)* moduljának New Zone *(új zóna)* varázslója ezt használja az új, fordított keresési zónák elkészítésekor. Az IPv6 *(Internet Protocol version 6)* protokollon alapuló hálózatokban a fordított keresési zónák az ip6.int tartományban találhatók.

Fordított lekérdezések

Fordított lekérdezésnél a DNS kiszolgálónak a megadott IP címhez tartozó DNS tartománynevet kell válaszként visszaküldenie. A fordított lekérdezések valójában olyan normál lekérdezések, amelyek a fordított keresési zónára vonatkoznak. A DNS szabvány szerint a fordított lekérdezési zónák és a PTR bejegyzések elkészítése nem kötelező. Ennek megfele-

lően a Windows 2000-ben sem szükségesek, bár egyes alkalmazások kihasználják a biztonság növelésére.

Inverz lekérdezések

Az inverz lekérdezések eredeti leírása az 1032-es RFC-ben olvasható, de ez mára elavult. Célja egy IP címhez tartozó név megtalálása volt, nem szabványos DNS lekérdezéssel. Használata a DNS szolgáltatás ellenőrzését és javítását segítő NSLOOKUP.EXE korai változataira korlátozódik. A Windows 2000 DNS kiszolgáló felismeri és elfogadja az inverz lekérdezéseket, amelyekre „hamis" inverz választ küld.

A DNS lekérdezések típusai

A DNS lekérdezések két osztályba sorolhatók: rekurzív és iteratív lekérdezések.

A rekurzív lekérdezésre a DNS kiszolgálóknak teljes választ kell adniuk, még akkor is, ha ehhez más DNS kiszolgálók segítségét is igénybe kell venniük. A teljes válasz elkészítéséhez a kiszolgáló egymást követő iteratív lekérdezéseket küld más DNS kiszolgálóknak a lekérdezést végrehajtó számítógép nevében.

Iteratív lekérdezésnél a kérést küldő számítógép a további kiszolgálók igénybevétele nélkül adható legjobb választ várja a DNS kiszolgálótól.

A munkaállomások általában rekurzív lekérdezéseket küldenek; feltételezik, hogy a DNS kiszolgáló vagy tudja a választ, vagy képes megkeresni. A DNS kiszolgálók ennek megfelelően többnyire iteratív lekérdezéseket küldenek más DNS kiszolgálóknak, ha a rekurzív lekérdezésekre nem tudják a választ.

DNS lekérdező

A DNS lekérdező a Windows 2000 egyik rendszerösszetevője, amely DNS kiszolgáló*(k)*nak küld lekérdezéseket. A Windows 2000 TCP/IP protokollverem beállításakor általában megadjuk legalább egy DNS kiszolgáló IP címét, amely*(ek)*hez a lekérdező a lekérdezéseket elküldi.

A Windows 2000 lekérdezője a DNS ügyfél eleme, ami a TCP/IP protokollal együtt automatikusan települ, és a services.exe folyamat részeként fut. Más Windows 2000 szolgáltatásokhoz hasonlóan a DNS ügyfél is a System fiókkal jelentkezik be.

A DNS lekérdező gyorsítótára

Gyakori eset, hogy egy számítógépnek rendszeresen kapcsolatba kell lépnie más számítógépekkel, ezért ugyanannak a DNS névnek *(például a levelezés-kiszolgáló nevének)* az átalakítását sokszor kellene elvégeznie. Ennek elkerülésére a Windows 2000 egy különleges gyorsítótárat használ a DNS adatok ideiglenes tárolására.

A DNS ügyfélszolgáltatás a lekérdezésekre kapott válaszok erőforrásrekordjait ideiglenesen, a TTL *(Time-To-Live)* beállításnak megfelelő ideig megőrzi. A gyorsítótár adatai felhasználhatók a beérkező kérdések megválaszolására. Alapértelmezés szerint a gyorsítótár TTL-je a lekérdezésre kapott válaszban szereplő TTL értéken alapul. A lekérdezés tárgyát képező tartománynévért felelős DNS kiszolgáló határozza meg az egyes erőforrásrekordok TTL-jét a válasz elküldése előtt.

A gyorsítótár tartalma a **IPCONFIG /DISPLAYDNS** paranccsal jeleníthető meg.

Negatív gyorsítótár

A DNS ügyfélszolgáltatás a hagyományos mellett negatív gyorstárazást is végez, ha a lekérdezésben szereplő tartománynévhez nem tartozik erőforrásrekord, vagy egyáltalán nem létezik a tartomány. Ekkor a lekérdezés sikertelen volta tárolódik el ideiglenesen, ami megakadályozza a nem létező nevekre vonatkozó ismételt lekérdezéseket.

Ha egy DNS kiszolgálónak továbbított kérésre negatív válasz érkezik, akkor alapértelmezett esetben 5 percig negatív válasz érkezik minden olyan lekérdezésre, amely ugyanazt a tartománynevet kérdezi. A negatív válaszok a sikereseknél rövidebb ideig maradnak a gyorsítótárban, hogy az ott lévő adatok minél kevésbé legyenek elavultak. A negatív válaszok tárolási ideje szabályozható az alábbi rendszerleíró azonosítóval:

NegativeCacheTime
Az azonosító helye: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dnscache\Parameters
Adattípus: REG_DWORD–Time, másodpercben
Alapértelmezett érték: 0x12c (decimális értéke 300 másodpercnek, vagyis 5 percnek felel meg)
Értelmezési tartomány: 0–0xFFFFFFFF (a túlságosan elavult bejegyzések elkerülésére egy napnál kisebb érték javasolt)
Alapértelmezésben létező azonosító

A negatív gyorsítótár csökkenti a DNS kiszolgálók terhelését, de ha a szóban forgó erőforrásrekordokra később szükség lesz, újabb lekérdezésekkel megszerezhetők.

Ha – alapértelmezett esetben 30 másodpercig – egyik DNS kiszolgáló sem elérhető, az időkorlátozás helyett a további lekérdezések azonnal hibát eredményeznek. Ezzel időt takarítanak meg azok a szolgáltatások, amelyek DNS lekérdezéseket végeznek a rendszertöltés *(boot)* során.

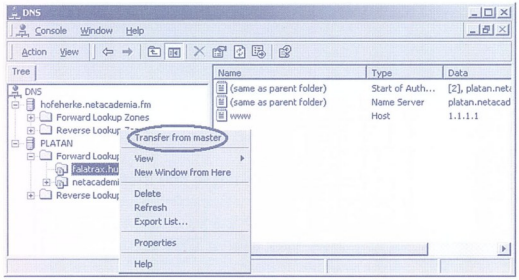
A DNS gyorsítótára új szolgáltatás a Windows 2000-ben. Sok rendszergazdát az őrületbe is kerget, mivel bizonyos esetekben megnehezíti a hibakeresést, hiszen látszólag hiába javítgatjuk ki gondosan a felfedezett DNS hibákat, a javítások mégsem jutnak érvényre. Ilyenkor mindig gondoljunk arra, hogy a DNS gyorsítótár még a régi hibás adatokat tartalmazza és sürgősen adjuk ki a következő parancsot:
IPCONFIG /FLUSHDNS
Na ugye, hogy megy?!

Zónaátvitel

A DNS rugalmasságának és teljesítményének fokozására minden standard elsődleges zónához legalább egy standard másodlagos zónát is érdemes telepíteni, amelynek az adatai egy másik DNS kiszolgálón tárolódnak. Az elsődle-

ges zónában végbemenő változásokat követően a zónaadatoknak replikálódniuk kell a másodlagos zónákba: ez a folyamat a zónaátvitel.

A zónaátvitel többnyire automatikusan, a SOA rekordban előírt időközönként történik. Kézi vezérléssel is végrehajtható a DNS MMC modulban, ha gyanítható, hogy a zóna frissítése nem megfelelő.



Ha egy Windows 2000 DNS kiszolgálón standard másodlagos zónát létesítünk, akkor oda a standard elsődleges zóna összes erőforrásrekordja átmásolódik. A DNS kiszolgálók korai változataiban akkor is végbemegy a teljes zónaátvitel, ha a másodlagos zóna csak az adatok összehangolását kéri az elsődlegestől, ami nagy zónáknál meglehetősen időigényes lehet, és a hálózati erőforrások pazarlását is magával vonja. Ez fontos kérdés, mert a zónaátvitelre minden olyan alkalommal szükség van, amikor az elsődleges zóna megváltozik, például a tartományban egy új címrekordot készítünk, vagy egy meglévőt megváltoztatunk.

Az erőforrásrekordok replikálását követően az új, standard másodlagos zóna DNS kiszolgálója rendszeresen megkérdezi az elsődleges zóna DNS kiszolgálóját, hogy történt-e változás. A SOA rekordban meghatározott időközönként ellenőrzi, hogy az elsődleges zóna verziószáma megváltozott-e. Ha nőtt, akkor zónaátvitelt kell végrehajtani. Ezt a folyamatot követhetjük nyomon az alábbi Network Monitor naplórészletből, ahol a két kommunikáló DNS kiszolgáló a már jól ismert GEP és MASINA:

```
1 563.890835 Gep Masina DNS 0x6000:Std Qry
for falatrax.hu. of type SOA on class INET
addr. 10.10.2.200 10.10.1.200
2 563.890835 Masina Gep DNS 0x6000:Std Qry
Resp. for falatrax.hu of type SOA on class
INET addr. 10.10.1.200 10.10.2.200
3 563.890835 Gep Masina DNS 0x4000:Std Qry
for falatrax.hu of type Req for incrmntl
zn Xfer on class INET 10.10.2.200
10.10.1.200
4 563.890835 Masina Gep DNS 0x4000:Std Qry
Resp. for falatrax.hu of type SOA on class
INET addr. 10.10.1.200 10.10.2.200
```

A fenti naplózó szerint a másodlagos zóna DNS kiszolgálója kérdezi az elsődleges zónát, amelyre válaszként a SOA bejegyzés érkezik. A másodlagos zóna megállapítja, hogy a verziószám megnőtt az elsődleges DNS kiszolgálón, ezért *(inkrementális)* zónaátvitelt kezdeményez.

A kézi vezérlésű DNS kiszolgálóknál ez jellemző hibaforrás: az elsődleges zónát megváltoztatják, de a verziószámot nem, így a replikáció elmarad. A Windows 2000-ben a zóna megváltoztatása – történjék akár a DNS MMC modullal, vagy dinamikus bejegyzéssel – automatikusan frissíti a verziószámot, így biztosítja, hogy a verziószám következő ellenőrzésekor a zónaátvitel bekövetkezzen.

Növekményes *(inkrementális)* zónaátvitel

A nagy zónák átvitele a sávszélesség jelentős részét lefoglalhatja, főként lassú, WAN kapcsolatok esetén. A zónaátvitel hatékonyságának növelésére a Windows 2000 új megoldást vezetett be a DNS zónák replikálásához: a növekményes zónaátvitelt, ami nem a teljes zónát továbbítja, csak a változásokat. Ezzel jelentősen csökken a másodlagos zónák aktualitásának megőrzéséhez szükséges hálózati forgalom. Leírása az 1995-ös RFC-ben olvasható.

A növekményes zónaátvitelnél először meg kell határozni a zóna forrása és replikált változata közötti különbségeket. Ha a zónák SOA bejegyzésében található sorszámok azonosak, nincs szükség zónaátvitelre.

Ha a forrás verziószáma nagyobb a kérést végrehajtó másodlagos kiszolgálóénál, akkor a megváltozott erőforrásrekordokból álló zónaátvitel zajlik le. A zónát kezelő DNS kiszolgálónak nyilvántartást kell vezetnie a zóna változásairól, hogy a növekményes zónaátvitelre vonatkozó kérésekre a változásokat el tudja küldeni. A Windows 2000 a növekményes változásokat a \Winnt\system32\dns mappában lévő szövegfájlokban tárolja, melyek neve a megfelelő zóna adatait tartalmazó fájl nevéből képződik (ez utóbbi a zóna készítésekor határozható meg). Ha a falatrax.hu zóna adatait a falatrax.hu.dns fájlban tároljuk, akkor a növekményes frissítés a falatrax.hu.dns.log fájlban lesz naplózva.

Active Directoryval integrált zóna replikációja

A standard zónák a hagyományos zónaátvitellel replikálódnak. Az AD-vel integrált zónák ellenben az AD replikációt használják a frissítéshez. Ennek előnyei:

- A DNS kiszolgálók többforrásúak. A standard DNS zónáknál az összes változtatást az elsődleges DNS kiszolgálón kellett végrehajtani. Az AD-vel való integráció következményeként a változtatások bármelyik DNS kiszolgálón végrehajthatók, ami javítja a teljesítményt, a méretezhetőséget és a hibatűrést.
- Az AD replikáció hatékonyabb és gyorsabb: nem a teljes zónát továbbítja, csak a megváltozott adatokat, így a hálózatot is csak ezek terhelik. A telephelyek közti, többnyire lassú kapcsolatokon történő replikáció jelentősen tömöríti az átvitt adatokat.

- A replikációs topológiát csak egyszer kell megtervezni és kivitelezni, ugyanazt használjuk az AD és a DNS változások replikálásakor is.

Az AD-t használó vállalatoknál javasolt az AD-vel integrált zónák alkalmazása. Ha azonban valamelyik külső cég DNS kiszolgálóját használják, az nem fogja támogatni ezt a zónatípust.

Zónadelegáció

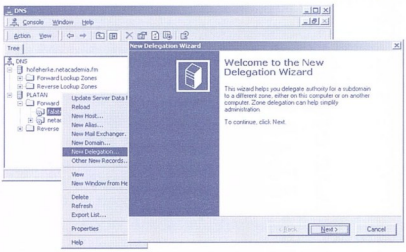
A DNS egy elosztott adatbázis, amit kifejezetten a HOSTS.TXT-vel történő névátalakítás korlátainak áthidalására hoztak létre. Miért méretezhető a DNS olyan nagy névterekre vagy hálózatokra, mint például az Internet? A tartománykezelés delegálhatósága miatt. Zónadelegációról akkor beszélünk, ha a szülőtartomány tulajdonosa az altartomány erőforrásrekordjainak kezelését az altartomány tulajdonosára bízza.

Az Internet első szintű, hárombetűs *(például .com, .org, .net stb.)* és földrajzi *(például .uk, .jp stb.)* tartományneveinek elosztott átalakítását 13 gyökérkiszolgáló *(A.ROOT-SERVERS.NET – M.ROOT-SERVERS.NET)* végzi. Segítségükkel az Internet számítógépei a teljes DNS adatbázishoz hozzáférnek. A gyökér és az elsőszintű tartományok alatt helyezkednek el a független szervezetek hatáskörébe tartozó tartományok és altartományok. Az elsőszintű tartományok közül néhány további hierarchiát mutat. A .hu tartománynak például van co.hu, info.hu, org.hu, erotika.hu altartománya, a teljes lista itt tekinthető meg:

http://www.nic.hu/regszab/sld.shtml

A delegáció a DNS fa elmetszésével szemléltethető: a metszésvonal alatti tartományért való felelősséget a vonal feletti tartomány átadja a vonal alattinak. A hr.falatrax.hu altartomány a falatrax.hu tartomány alatt helyezkedik el. A delegáció eredményeként az alárendelt tartományért másik kiszolgáló lesz felelős.

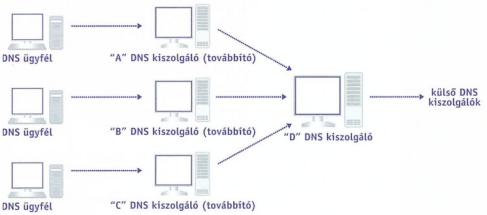
A delegációhoz a szülőzónában lennie kell **A** és **NS** bejegyzésnek, melyek mindegyike a delegált tartomány gyökerére mutat. A falatrax.hu zónában például lennie kell a hr.falatrax.hu-ra mutató A és NS bejegyzésnek. A Windows 2000-ben varázsló könnyíti meg a delegáció létrehozását.



Továbbító *(forwarder)* és szolga DNS kiszolgáló

Ha egy lekérdező megszólít egy DNS kiszolgálót, akkor az először a saját gyorstárolóját felhasználva próbálja megtalálni a tartománynevet, majd visszaküldeni a megfelelő erőforrásrekordokat. Ha ez nem sikerül, a kiszolgáló iteratív lekérdezésekkel próbálja megoldani a feladatot. Az első lekérdezést egy gyökérkiszolgálónak küldi el. Ez a módszer azonban nem biztos, hogy megfelelő, ha a kiszolgáló egyike egy olyan telephely DNS kiszolgálóinak, amely lassú kapcsolaton keresztül kommunikál a világgal.

Az alábbi ábrán látható, hogy a továbbító egy DNS kiszolgáló, amelyhez más DNS kiszolgálók fordulnak, mielőtt a szükséges névátalakítást megkísérelnék végrehajtani.

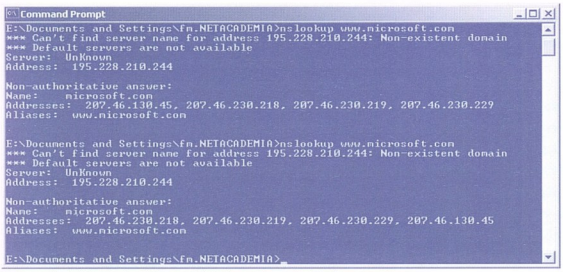


Ha a példában szereplő A, B és C kiszolgálók bármelyike rekurzív lekérdezést hajt végre, először a helyben tárolt zónák vagy a gyorsítótár alapján próbálnak választ adni. Ha ez nem sikerül, akkor nem külső DNS kiszolgálókhoz fordulnak, hanem a D kiszolgálónak küldik el a kérést, amely nagyobb eséllyel tud válaszolni a saját gyorsítótárjából. Ez az elrendezés csökkenti a lekérdezések megválaszolásához szükséges külső forgalmat.

Ha a továbbító *(példánkban a D kiszolgáló)* nem tud az A, B vagy C kiszolgálók kérdésére válaszolni, ez utóbbiak ismét maguk próbálják megszerezni a választ, most már iteratív lekérdezésekkel. Ennek a nemkívánatos jelenségnek a kiküszöbölésére találták ki a szolga DNS kiszolgálókat. Ezek olyan továbbítók, amelyek a lekérdezéseket csak továbbíthatják. A DNS kiszolgálót így arra kényszeríthetjük, hogy minden lekérdezési feladathoz a beállított továbbítókat használják.

Round robin terheléselosztás

A round robin módszert a hálózati erőforrások terhelésének elosztására használják. Ha egy lekérdezéshez több erőforrásrekord is tartozik, akkor a round robin módszer alkalmazásánál a válasz mindig a soron következőt tartalmazza – az utolsó bejegyzést ismét az első követi. Ez az ügyfélprogramok által gyakran használt webkiszolgálók és más, többkapcsolatos, azaz több hálózati kártyával, illetve IP címmel rendelkező *(multihomed)* számítógépek terheléselosztásának nagyon egyszerű formája. Az alábbi ábrán annak eredménye látható, hogy NSLOOKUP paranccsal gyors egymásutánban kétszer lekérdeztem a www.microsoft.com IP címét. Megfigyelhető, hogy noha mindkét esetben négy IP címet kaptam vissza, a második lekérdezésnél más a sorrend, így az ügyfélalkalmazások másik kiszolgálóra jutottak volna.



A round robin csak akkor működhet, ha a lekérdezett névhez a zónában több **A** címrekord tartozik.

Dinamikus frissítésű DNS ügyfél

Nagy hálózatokban az összes erőforrásrekord begyűjtése és érvényességük megőrzése komoly feladat lehet. A címrekordok karbantartása akár egy vagy több személy teljes munkaidejét is igényelheti. Ennek megkönnyítésére a Windows 2000 támogatja a DNS dinamikus frissítését, melynek részletes leírása a 2136-os RFC-ben olvasható.

Dinamikus DNS-nél az ügyfél DNS bejegyzési kérelmet küld a DNS kiszolgálónak, hogy frissítse az ügyfél „A" címrekordját. Ha az ügyfél ezen kívül DHCP ügyfél is, akkor a DHCP bérlések kezelésekor a címmel kapcsolatos események *(például új cím kiadása vagy cím megújítása)* bekövetkezésekor a DHCP ügyfél a DHCP kiszolgálónak 81-es DHCP opciót küld a teljes nevével együtt. A DHCP kiszolgáló ennek hatására PTR bejegyzést hajt végre az ügyfél nevében. A statikusan beállított Windows 2000 operációs rendszerek saját maguk elvégzik az A és PTR bejegyzést is.

Ha egy Windows 2000 DHCP ügyfél olyan, alacsonyabb szintű DHCP kiszolgálót szólít meg, amely nem ismeri a 81-es opciót, akkor az ügyfél saját maga gondoskodik a PTR bejegyzésről. A Windows 2000 DNS kiszolgálót felkészítették a dinamikus frissítések kezelésére.

Azért kell ezt a módszert használni *(az ügyfél frissíti az **A**, a DHCP kiszolgáló pedig a PTR bejegyzést)*, mert csak a munkaállomás tudja, hogy a számítógép mely IP címei tartoznak egy adott névhez. Mivel a DHCP kiszolgáló hiányosan informált, nem tudja hiánytalanul végrehajtani az **A** erőforrás bejegyzését. Szükség esetén a DHCP kiszolgáló is beállítható úgy, hogy mindkét rekordtípus bejegyzését elvégezze.

A következő részben a DNS működését és az erőforrásrekordok felépítését elemezzük.

folytatjuk...

Fóti Marcell
marcellf@netacademia.net

Encrypting File System

Személyes adataink védelme

Az EFS biztosítja azt az alapvető fájltitkosítási technológiát, amellyel az NTFS fájlok titkosítva tárolhatók a lemezmeghajtón. Az EFS részben megoldja azokat az adatvédelmi problémákat, amelyeket más operációs rendszerek eszközei okoznak, amelyekkel a felhasználók a hozzáférési jogok ellenőrzése nélkül férhetnek hozzá az NTFS köteten tárolt fájlokhoz. Az EFS-sel az NTFS fájlok adatai titkosítva tárolódnak a lemezen. Az EFS nyílt kulcsú titkosítási technológiát használ, amely integrált rendszerszolgáltatásként fut, így egyszerűen kezelhető, nehezen feltörhető és a felhasználó számára láthatatlanul működik. Ha a titkosított NTFS fájlhoz olyan felhasználó próbál meg hozzáférni, aki privát kulccsal rendelkezik, képes azt megnyitni, és úgy dolgozhat vele, mint egy normál dokumentummal. Ha a felhasználó nem rendelkezik privát kulccsal a fájlhoz, nem is férhet hozzá a dokumentumhoz.

Az értékes információkat gyakran védelem nélküli fájlokban tároljuk a merevlemezen. Az NTFS partíción tárolt információkhoz való hozzáférés korlátozható, ha a Windows 2000 az egyetlen futtatható operációs rendszer, és a merevlemez meghajtó nem vehető ki a gépből. Ha azonban valaki nagyon hozzá akar férni az információhoz, ez nem nehéz, ha fizikailag hozzá tud férni a számítógéphez vagy a merevlemezhez. Az illetéktelen hozzáférés sok esetben problémát jelenthet:

- **Ellopott laptop** – Az őrizetlenül hagyott laptop pillanatok alatt eltűnhet. Mi van akkor, ha a tolvaj nem a számítógépet akarja eladni, hanem a merevlemezen tárolt értékes információk érdeklik?
- **Engedély nélküli hozzáférés** – Ha az irodai számítógépeket őrizetlenül hagyjuk, bárki bemehet a helyiségbe, és információkat lophat el azokról.

A személyi számítógépek biztonsága általában jól mérhető úgy, ha a merevlemezről történő – szokásos – rendszerindítás helyett megpróbáljuk a gépet idegen operációs rendszerrel, akár flopiról indítani. Amellett, hogy ezzel a módszerrel el tudunk kerülni bizonyos *(merevlemezhibákból, és hibás rendszerindító partícióból eredő)* rendszertöltési problémákat, sajnos ez egyúttal lehetővé teszi azt is, hogy a számítógépen más operációs rendszert indítsunk el. Ez azt jelenti, hogy ha valaki fizikailag hozzáfér a rendszerhez, az az NTFS fájlrendszer olvasására képes eszközzel esetleg olyan adatokat is elolvashat a lemezen, melyhez nyilvánvalóan nincs joga. Ha végiggondoljuk, mi is történik ilyenkor, kiderül, hogy csodáról szó sincsen. Az NTFS tárolja ugyan a fájlok hozzáférését meghatározó jogosultságlistákat *(ACL)*, ám az ellen nemigen tudunk mit tenni, ha az idegen operációs rendszer rá sem hederít e listákra – megkerülheti a hozzáférés-szabályozás beépített biztonsági funkcióit. Hiába no, igaz a régi mondás: a halott indián a legjobb indián. A halott *(el nem indított)* Windows 2000 nyilván nem látja el biztonsági funkcióit. Igazán éles környezetben már csak emiatt is érdemes gondoskodni a számítógépek fizikai biztonságáról *(magyarul elzárásáról)*, amely persze egyúttal jelentősen csökkenti a havonta eltűnő egerek számát is. A másik lehetséges módszer a BIOS jelszóvédelmének használata – lenne. Ez a lehetőség azonban szóba sem jöhet ott, ahol egy számítógépet több felhasználó használ. De még ha használnák is – a mindenki által ismert jelszó nem nyújt nagy biztonságot.

E problémára csak az adattitkosítás jelent megoldást. Létezik néhány olyan termék, amely jelszóból származó kulcsok használatával alkalmazásszintű fájltitkosítást tesz lehetővé. E módszerek legtöbbje azonban csak korlátozottan használható, mindegyik hiányos valahol:

- **Manuális titkosítás és dekódolás minden használat alkalmával.** A titkosítási szolgáltatás a legtöbb termékben nem a háttérben történik. A fájlt minden használat előtt dekódolni, majd a használat után újra titkosítani kell. Ha elfelejtjük a titkosítást, a fájl védelem nélkül marad. Mivel a fájl titkosítását *(és dekódolását)* a felhasználó kezdeményezi, ez igen gyakran elmarad.
- **Az ideiglenes- és lapozófájlok használatából eredő problémák.** Sok alkalmazás *(például a Microsoft Word)* ideiglenes fájlokat hoz létre egy dokumentum szerkesztésekor. Ezek az ideiglenes fájlok akkor is titkosítás nélkül maradnak a lemezen, ha az eredeti dokumentum titkosítva van, így egyszerűvé válik az adatok lopása.
- **Az alkalmazásszintű titkosítás Windows felhasználói módban történik.** Ez azt jelenti, hogy a felhasználó titkosítási kulcsa tárolódhat a lapozófájlban. A lapozófájl adatainak felhasználásával igen egyszerűvé válik az egy kulccsal titkosított dokumentumokhoz való hozzáférés.
- **Gyenge adatvédelem.** A kulcsok jelszavakból származnak. Egyszerű, könnyen megjegyezhető jelszavak használata esetén ez a védelem könnyen feltörhető.
- **Nincs adat helyreállítás.** Sok termék nem biztosít helyreállítási szolgáltatást. Ez még jobban elbátortalaníthatja a felhasználókat, különösen azokat, akik nem akarnak egy új jelszót megjegyezni. A jelszóalapú adathelyreállítás használata újabb hozzáférési problémát vet fel. Az adattolvajnak csak a jelszóra van szüksége ahhoz, hogy helyreállítsa a titkosított fájlokat.

Az Encrypting File System *(EFS)* megoldást nyújt a fenti problémákra. A továbbiakban az EFS titkosítási technológiáját, a titkosítás integráltságát, a felhasználói tennivalókat és az adat helyreállítást elemezzük.

Az EFS titkosítási technológia

Az EFS nyílt kulcsú titkosításon alapul, és kihasználja a Windows CryptoAPI architektúrájának előnyeit. A fájlok egy véletlenszerűen generált kulccsal kerülnek titkosításra, amely független a felhasználó nyílt/privát kulcspárjától; így a szokásos feltörési módszerek használhatatlanok.

A fájltitkosítás bármilyen szimmetrikus titkosítási algoritmust képes használni. Az EFS első kiadása a DES-t használja titkosítási algoritmusként. A következő kiadásokban más titkosítási algoritmusok is elérhetőek lesznek.

nehézségi fok: ●●●○

Windows 2000

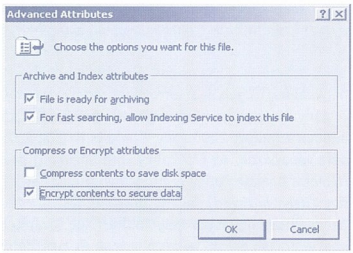
Az EFS távoli fájlkiszolgálókon tárolt fájlok titkosítását és dekódolását is lehetővé teszi. Ilyenkor az EFS az adatokat csak a lemezen titkosítja. A hálózaton továbbra is titkosítatlanul közlekednek az adatok. A hálózati forgalom titkosítására használjunk SSL vagy IPSec protokollt!

Az EFS szorosan integrált az NTFS-sel. NTFS köteten még az ideiglenes fájlok is öröklik a titkosítást, ilyenek létrehozásakor az eredeti fájl minden tulajdonsága átmásolódik az ideiglenes fájlra. Az EFS az operációs rendszer kernelben található, és a nem lapozható memóriakészletet használja a fájltitkosítási kulcsok tárolására, így azok sohasem kerülnek a lapozófájlba.

A felhasználó tennivalói

Az EFS alapértelmezett beállításaival a felhasználók azonnal, rendszergazdai beavatkozás nélkül titkosíthatnak fájlokat. Ha a felhasználó a fájltitkosításhoz nem rendelkezik nyílt kulcspárral, az EFS automatikusan létrehoz neki egyet. A fájltitkosítás és -dekódolás fájlonként vagy könyvtáranként is elvégezhető. A titkosításra kijelölt könyvtárban minden létrehozott fájl *(és alkönyvtár)* automatikusan titkosításra kerül, s minden fájl egyedi titkosítási kulccsal rendelkezik. Ha egy fájlt egy titkosított könyvtárból ugyanazon a köteten belül egy titkosítatlanba helyezünk át, a fájl megtartja eredeti jellemzőit, azaz titkosított marad. A titkosítási és dekódolási szolgáltatások a Windows Intézőből *(Explorer)* érhetők el, míg gyakorlott felhasználók és helyreállítási ügynökök parancssori eszközöket is használhatnak.

Az Explorerben nyissuk meg a titkosítandó fájl vagy mappa tulajdonságlapját, majd kattintsunk az Advanced nyomógombra, és megjelennek a titkosítási és tömörítési lehetőségek. *(Egy fájl nem lehet egyszerre tömörített és titkosított!)*



A fájlt a használat előtt dekódolni kell, ám ezzel nincs teendőnk, a titkosítás és a dekódolás ugyanúgy a lemez adatforgalma közben, láthatatlanul megy végbe, mint a ki- és betömörítés. Az EFS automatikusan felismeri a titkosított fájlokat és megkeresi a felhasználó kulcsát a rendszer kulcstárjában. Mivel a kulcstárolás a CryptoAPI-ra épül, a felhasználók a kulcsokat biztonságos eszközökön, például intelligens kártyákon is tárolhatják.

Az EFS első kiadása nem támogatja a fájlok osztott használatát. Ez azt jelenti, hogy hiába van sok felhasználónak jogosultsága egy titkosított fájlra, kizárólag a titkosítást végző személy *(no meg a helyreállítási ügynök)* képes az megnyitni. Azt is mondhatnánk, hogy „aki kapja, marja", vagyis akinek legelőször jut eszébe a fájl titkosítása, azé a fájl a továbbiakban.

Az EFS architektúra azonban úgy készült, hogy nyílt kulcsok használatával a fájlmegosztást bármennyi felhasználó között lehetővé tegye. Így elvileg bárki, aki a titkosításban rész vett, saját privát kulcsa használatával dekódolhatja a fájlokat. A Windows 2000 következő kiadásában a felhasználók egyszerűen hozzáadhatók *(és eltávolíthatók)* lesznek a titkosított dokumentumokon engedélyezett felhasználók csoportjához.

Adat helyreállítás

Az EFS beépítetten támogatja az adat-helyreállítást. A Windows 2000 biztonsági infrastruktúra kötelezővé teszi adat-helyreállítási kulcsok beállítását. A fájltitkosítás csak akkor használható, ha a rendszerhez be van állítva egy vagy több helyreállítási kulcs. A helyreállítási kulccsal csak a fájl véletlenszerűen generált titkosítási kulcsa válik elérhetővé, az eredeti felhasználó privát kulcsa nem. Ez biztosítja, hogy a helyreállítási ügynök még véletlenül sem juthat mások személyes információihoz.

A helyreállítás olyan vállalati környezetek számára készült, amelyekben a vállalatnak szüksége lehet arra, hogy olyan adatokat állítson helyre, amelyeket egy azóta kilépett alkalmazott titkosított, és az is fontos, hogy a helyreállítás akkor is elvégezhető legyen, ha a titkosítási kulcsok elvesznek.

A helyreállítási házirend a Windows 2000 tartomány tartományvezérlőjén állítható be. Ez a házirend a tartomány minden számítógépére vonatkozik. A helyreállítási házirendet a tartományi rendszergazdák állíthatják be, vagy a feladatot kijelölt adatvédelmi felhasználóknak is kiadhatják. Így jobban és rugalmasabban szabályozható, hogy ki állíthat helyre titkosított adatokat. Az EFS több helyreállítási ügynököt is használhat, több helyreállítási kulcsot is beállíthatunk.

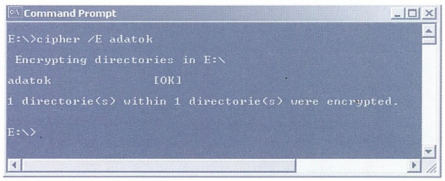
Az EFS otthoni környezetben is alkalmazható. Ha nincs Windows tartomány, az EFS automatikusan generál helyreállítási kulcsokat és azokat számítógépkulcsként menti el. Az otthoni felhasználók a rendszergazda fiók használatával, parancssorból állíthatják vissza adataikat.

Az Encrypting File System használata parancssorból

A grafikus felület mellett a Windows 2000 parancssori eszközt *(Cipher.exe)* is tartalmaz a leggyakoribb műveletek elvégzéséhez.

A C:\adatok könyvtár titkosítása:

```
C:\>cipher /e adatok
```



Az összes, „bla" karaktersort tartalmazó nevű fájl titkosítása:

```
C:\>cipher /e /s *bla*
```

A parancs teljes kapcsolókészlete a következő:

```
CIPHER  [/E|/D]  [/S[:dir]]  [/A]  [/I]
[/F]  [/Q]  [fájlnév [...]]
```

- **/E** *(Encrypt)* A megadott fájlok és könyvtárak titkosítása. A könyvtárak ilyenkor megjelölődnek, hogy a később hozzáadott fájlok is titkosításra kerüljenek.
- **/D** *(Derypt)* A megadott fájlok és könyvtárak dekódolása. A könyvtárak ilyenkor megjelölődnek, hogy a később hozzáadott fájlok se kerüljenek titkosításra.
- **/S** *(Subdir)* A megadott műveletet az adott könyvtárban és annak alkönyvtáraiban található fájlokon hajtja végre. Az alapértelmezett hely az aktuális könyvtár.
- **/I** *(Ignore)* A megadott műveletet akkor is folytatja, ha közben hiba lép fel. Alapértelmezésben a CIPHER hiba esetén leáll.
- **/F** *(Force)* A titkosítási műveletet minden kijelölt fájlon végrehajtja, még azokon is, amelyek már titkosítva vannak. Alapértelmezésben a már titkosított fájlok kimaradnak az új titkosítás során.
- **/Q** *(Quiet)* Csak a legszükségesebb információkat jeleníti meg.

Paraméterek nélkül a CIPHER az aktuális könyvtár és az abban található összes fájl titkosítási állapotát jeleníti meg. A fájl minden olvasási művelet során láthatatlanul dekódolódik, íráskor pedig titkosítódik. Annak megállapításához, hogy a fájl titkosítva van-e, a felhasználó a fájl tulajdonságai között ellenőrizheti, hogy a titkosítás be van-e kapcsolva. Mivel a titkosítás a háttérben zajlik, a felhasználó ugyanúgy használhatja a fájlt, mint a titkosítás előtt, például ugyanúgy megnyithatja és szerkesztheti a Word dokumentumot vagy Notepad-del a szöveges fájlt. Ha a titkosított fájlt más felhasználó próbálja megnyitni, nem férhet ahhoz hozzá, mivel nem rendelkezik kulccsal a fájl dekódolásához.

A rendszergazdáknak nem szabad a rendszerkönyvtár fájljait titkosítani, mivel ezekre szükség van a rendszer indításához. A rendszerindítás során a felhasználók kulcsa nem hozzáférhető a fájlok dekódolásához. Egy titkosított rendszerfájl könnyen használhatatlanná teheti a rendszert. *(Halálos hibát okoz sajnos az autoexec.bat titkosítása is. A gép többet nem indul el :-( )* A Windows Intéző védekezik is ez ellen, mivel nem titkosítja a rendszerfájlokat. A Windows későbbi kiadásai lehetővé fogják tenni a rendszerfájlok titkosítását.

Az EFS béta változatai lehetővé tették a titkosított fájlok továbbítását a gépek között. A COPY parancs korai változatainak volt két csodálatos kapcsolója titkosított fájlok másolására, azonban a végleges termékben – sajnos-sajnos – ezek már nincsenek benn.

Az Export Encrypted File *(titkosított fájl exportálása)* és az Import Encrypted File *(titkosított fájl importálása)* lehetővé tette *(volna)*, hogy akár flopilemezre *(tehát FAT fájlrendszerre)* is titkosítottan tegyük ki a fájlokat. Ezután ez az exportfájlt más fájlrendszerre, vagy akár szalagos meghajtóra is átmásolható, illetve e-mail mellékletként is elküldhető, ugyanúgy mint egy normál fájl. Hogy használható legyen azon a rendszeren, ahová kerül, a fájlt NTFS kötetre kell importálni, így az titkosítottként jön létre.

Fájlok egyszerű másolásakor a másolat titkosítatlan lesz, ha az a könyvtár, amelybe másolunk, nincs titkosítottként megjelölve. Ez azért van, mert a normál másolási parancs fájlolvasási műveletet használ, amelyhez az EFS a háttérben dekódolja a fájlt. Így egy titkosított fájlról titkosítatlan másolatot hozunk létre.

Könyvtártitkosítás

Könyvtárakat is megjelölhetünk titkosítottként. A könyvtár titkosítottá tétele azt jelenti, hogy a könyvtárban létrehozott minden fájl alap-értelmezetten titkosított, és minden létrehozott alkönyvtár alap-értelmezetten titkosítottként megjelölt lesz. A könyvtár fájllistája nem titkosított, így a fájlok a szokott módon jelennek meg, ha a könyvtárhoz megfelelő hozzáféréssel rendelkezünk.

A könyvtárak titkosítottként történő megjelölése a fájltitkosításhoz hasonló. Kijelöljük a könyvtárat, és a titkosítást választjuk a Windows Intézőben. Ilyenkor a felhasználó választhat, hogy csak a könyvtárat jelöli meg titkosítottként, vagy a könyvtár minden fájlját és alkönyvtárát is titkosítja. A könyvtártitkosítás lehetővé teszi, hogy a felhasználók az értékes fájlokat egyszerűen a titkosított könyvtárba másolva biztosak lehessenek abban, hogy az titkosítottan tárolódik.

Helyreállítás

Az EFS helyreállítási házirendje a rendszer átfogó biztonsági házirendjének része *(a tartomány biztonsági házirendjének része a Windows tartományban, vagy a helyi biztonsági házirend része önálló számítógépek és kiszolgálók esetében)*. Windows 2000 telepítése után az alap-értelmezett helyreállítási ügynök a tartomány rendszergazdája. A tartomány biztonsági házirendjének részeként a tartomány minden Windows 2000 vagy Windows NT alapú számítógépére vonatkozik. Az EFS Policy *(EFS házirend)* felhasználói felület a Domain Policy *(Tartományházirend)* és a Local Policy *(Helyi házirend)* felület integrált része. E felülettel a helyreállítási ügynökök helyreállítási kulcsokat generálhatnak, exportálhatnak, importálhatnak és menthetnek el.

A helyreállítási házirend és a rendszer biztonsági házirendjének integrálása következetesen alkalmazható biztonsági modellt hoz létre. A Windows biztonsági alrendszere végzi az EFS házirend kezelését, replikációját és tárolását. Így a felhasználók a fájltitkosítást olyan rendszeren is elvégezhetik, amely ideiglenesen offline módban van *(például egy laptopon)*, ugyanúgy, mint ahogy a tárolt bizonyítványokkal a tartományi fiókjukra is bejelentkezhetnek.

Az EFS csak akkor használható, ha tartományszinten *(vagy helyileg, ha a számítógép nem egy tartomány tagja)* be van állítva helyreállítási házirend. A helyreállítási házirendet a

Windows 2000

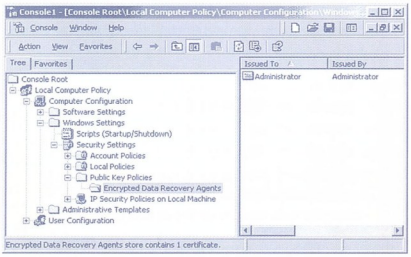
tartományi rendszergazdák *(vagy erre kijelölt alkalmazottak, a helyreállítási ügynökök)* állíthatják be. A helyreállítási házirend kezeli a tartomány minden számítógépéhez a helyreállítási kulcsokat.
Ha egy felhasználó elveszíti a privát kulcsát, a kulccsal védett fájl visszaállítható a helyreállítási ügynökök saját privát helyreállítási kulcsával.

Az alap értelmezett helyreállítási kulcs adatvédelme egy önálló számítógépen

A helyi rendszergazda első bejelentkezésekor minden önálló számítógépre telepítődik az alap értelmezett helyreállítási házirend. Ez a helyi rendszergazdát teszi a számítógép alap-értelmezett helyreállítási ügynökévé.

Ennek megváltoztatása létkérdés, mert sajnos elég könnyen idegenek is Administrator-rá válhatnak egy ellopott számítógépen a SAM törlésével – s ezzel egy csapásra az összes titkosított fájl helyreállítási ügynökévé is válnak!

1. Kattintsunk a Start-ra, kattintsunk a Run-ra, majd az Open mezőbe írjuk be, hogy MMC. Kattintsunk az OK-ra.
2. Kattintsunk a Console-ra, majd az Add/Remove Snap-In-re. Kattintsunk az Add-re.
3. Kattintsunk a Group Policy-re, majd az Add-re. Hagyjuk meg az alap-értelmezett Local Computer beállítást, majd kattintsunk a Finish-re. Kattintsunk a Close-ra, majd az OK-ra.
4. A Local Computer Policy mellett kattintsunk a +-ra. Ugyanígy nyissuk ki a Computer Configuration-t, a Windows Settings-t, a Security Settings-t, majd a Public Key Policies-t, majd kattintsunk az Encrypted Data Recovery Agents-re. A képernyőn az alábbi ábrához hasonlónak kell megjelennie.



A házirendben láthatunk egy Administrator bizonyítványt is. Ez a helyi rendszergazdát teszi az alap értelmezett helyreállítási ügynökké. Ha ezt a bizonyítványt letöröljük, üres helyreállítási házirendet hozunk létre, amely kikapcsolja az EFS-t. Az EFS nem engedélyezi az adattitkosítást, ha nincs helyreállítási ügynök. A bizonyítványhoz kapcsolódó helyreállítási kulcs védelméhez...

1. Kattintsunk a Console-ra, majd kattintsunk az Add/Remove snap-ins-re. Kattintsunk az Add-re.
2. Kattintsunk a Certificates-re, majd az Add-re. Kattintsunk a Current User-re, a Finish-re, a Close-ra, majd az OK-ra.
3. A Certificates–Current User mellett kattintsunk a +-ra. Ugyanígy nyissuk meg a Personal mappát. A bal oldali panelen kattintsunk a Certificates-re.
4. A jobb oldali panelen kattintsunk az Administrator-re és menjünk az Intended Purposes-re. Állítsuk be File Recovery-re. Exportáljuk a bizonyítványt és a privát kulcsot egy .pfx fájlba.
5. A .pfx fájl létrehozása után töröljük le a bizonyítványt és a hozzá tartozó privát kulcsot a Personal store-ból. Így a kulcs egyetlen példánya a .pfx fájlban található. Ehhez a jobb oldali panelen kattintsunk az Administrator-re, majd az eszköztárban kattintsunk a piros X-re. Egy üzenet figyelmeztet arra, hogy ezután nem tudjuk dekódolni az e bizonyítvánnyal titkosított adatokat. A folytatáshoz kattintsunk a Yes-re.
6. Helyezzük el a .pfx fájlt egy széfben vagy zárható szekrényben. Ezt a fájlt csak akkor kell használnunk, ha fájlt kell visszaállítanunk.

Helyreállítási bizonyítvány kérése

Ha az alap-értelmezett helyreállítási házirend használata mellett döntünk, nem kell helyreállítási bizonyítványt kérnünk. Ha azonban a tartományban több helyreállítási ügynökre van szükség, vagy ha a helyreállítási ügynök jogi vagy vállalati okokból nem a tartományi rendszergazda, bizonyos felhasználókat helyreállítási ügynökké kell tenni, és e felhasználók számára fájl-helyreállítási bizonyítványt kell adni.
Ehhez a következőt kell tenni:

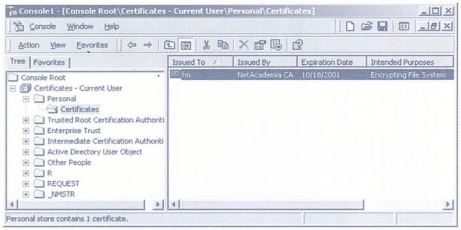
- Ha még nincs, létre kell hozni egy Enterprise Certificate Authority-t *(CA). (Erről egy későbbi számban írunk.)*
- Az Enterprise CA-nak lehetővé kell tennie a kijelölt felhasználóknak vagy ügynököknek, hogy helyreállítási bizonyítványt kérjenek és kapjanak.
- Minden kijelölt felhasználónak kérnie kell egy helyreállítási bizonyítványt.

A kulcsok helye

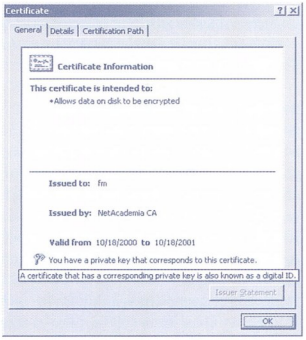
Felmerülhet a kérdés, hogy vajon hol tárolja az operációs rendszer a felhasználó kulcspárjait. Nos, amíg ki nem exportáljuk, addig nem a lehető legbiztonságosabb helyen vannak - a felhasználó profiljában. Meg is lehet tekinteni a kulcspárokat a következő módon:

1. A Microsoft Management Console *(MMC)* indításához kattintsunk a Start-ra, kattintsunk a Run-ra, az Open mezőbe írjuk be, hogy mmc, majd kattintsunk az OK-ra.
2. A Console menüben kattintsunk az Add/Remove snap-ins-re, majd kattintsunk az Add-re.
3. Keressük meg a Certificates snap-in-t, majd kattintsunk az Add-re. Válasszuk a My user account-ot, és kattintsunk a Finish-re. Kattintsunk a Close-ra, majd az OK-ra.
4. Keressük meg az Encrypting File System bizonyítványokat a Personal certificate store-ban. A Certificates–Cur-

rent User mellett kattintsunk +-ra. Nyissuk meg a Personal mappát. Kattintsunk a Certificates-re.



Már az Intended Purposes mezőben is olvasható, hogy a bizonyítvány EFS titkosításra használható. Ha pedig meg is nyitjuk, az is láthatóvá válik, hogy itt egy privát kulccsal is rendelkező, úgynevezett Digital ID-ről van szó.



Fájlok visszaállítása egy másik számítógépre

Ha egyes titkosított fájlokat nem csak azon a számítógépen akarunk használni, amelyen azok titkosításra kerültek, biztosítanunk kell, hogy a titkosítási bizonyítvány és a hozzá tartozó privát kulcs hozzáférhető legyen a másik rendszeren is. Ezt vándorló *(központi)* profil *(Roaming Profile)* használatával vagy a kulcsok manuális áthelyezésével érhetjük el.

- **A Roaming Profile használata.** Ha még nincs központi profilunk, kérjük meg a rendszergazdát, hogy hozzon létre egyet. Ha rendelkezünk központi profillal, az általunk használt titkosítási kulcsok minden számítógépen azonosak, ha ugyanazzal a felhasználói fiókkal jelentkezünk be. Még ha központi profilt használunk is, akkor is szükséges lehet a titkosítási bizonyítvány és a privát kulcs biztonsági mentése. Ha elveszítjük azokat a kulcsokat, amelyek lehetővé teszik, hogy dekódoljunk egy fájlt, a kijelölt helyreállítási ügynök *(alapértelmezésben a helyi vagy a tartomány rendszergazda)* helyreállíthatja a titkosított fájlt.
- **A kulcsok manuális áthelyezése.** A kulcsok manuális áthelyezése előtt készítsünk biztonsági másolatot a titkosítási bizonyítványról és a privát kulcsról. Így a bizonyítvány és a kulcs egy másik rendszeren is visszaállítható.

A titkosítási bizonyítvány és a privát kulcs biztonsági mentése

1. Az egér jobb gombjával kattintsunk a bizonyítványunkra, kattintsunk az All Tasks-ra, majd kattintsunk az Export-ra. Ezzel elindítjuk a Certificate Manager Export Varázslót. Kattintsunk a Next-re.
2. Kattintsunk a Yes, export the private key-re. Kattintsunk a Next-re.
3. A használható exportformátum a Personal Information Exchange-PKCS#12, vagy a .pfx *(personal exchange format)*. Kattintsunk a Next-re.
4. A.pfx adatok védelmére adjunk meg jelszót. Kattintsunk a Next-re.
5. A .pfx adatok tárolásához adjuk meg az elérési útvonalat és a fájlnevet. Esetünkben írjuk be, hogy c:\mykey. Kattintsunk a Next-re.
6. Az exportálandó bizonyítványok és kulcsok listája jelenik meg. A megerősítéshez kattintsunk a Finish-re.
7. A Varázsló és a snap-in bezárásához kattintsunk az OK-ra.

Így a titkosítási bizonyítványt és a privát kulcsot egy .pfx fájlba exportáltuk, amelyről biztonsági másolatot kell készítenünk.

A titkosítási bizonyítvány és a privát kulcs visszaállítása egy másik rendszerre

1. Másoljuk a .pfx fájlt egy flopira, és vigyük át arra a számítógépre, amelyre a titkosítási bizonyítványt és a privát kulcsot importálni akarjuk.
2. A célszámítógépen is töltsük be az előbb használt MMC Certificates snap-int.
3. A.pfx fájl importálásához az egér jobb gombjával kattintsunk a Personal store-ra, kattintsunk az All Tasks-ra, majd kattintsunk az Import-ra. Így elindítjuk a Certificate Manager Import Varázslót. A bizonyítvány és a privát kulcs importálásához végezzük el a Varázsló lépéseit.
4. Adjuk meg a .pfx fájl elérési útvonalát. Példánkban ez c:\mykey.pfx.
5. A .pfx adatok kicsomagolásához adjuk meg a jelszót.
6. Kattintsunk a Place all certificates in the following store-ra, majd fogadjuk el a Personal certificate store-t. Kattintsunk a Next-re.
7. Az importálás elindításához kattintsunk a Finish-re, majd az OK-ra. Az importálás befejezése után a Varázsló bezárásához kattintsunk az OK-ra.

Ezután rendelkezünk az ahhoz szükséges kulcsokkal, hogy a titkosított fájlokat ugyanúgy használhassuk, mint azon a számítógépen, amelyen a biztonsági mentés történt.

Mappák és fájlok távoli kiszolgálón

Távoli kiszolgálókon tárolt fájlokat is titkosíthatunk és dekódolhatunk, valamint használhatjuk is a titkosított fájlokat. Ez távoli hozzáféréssel is lehetséges, és úgy is, ha a másik számítógépre helyileg lépünk be. Ne feledjük azon-

ban, hogy ha a titkosított fájlokat a biztonsági mentéssel és visszaállítással helyezzük át, a megfelelő titkosítási bizonyítványt és a privát kulcsokat is át kell helyeznünk ahhoz, hogy a titkosított fájlokat az új helyen is használhassuk. Megfelelő privát kulcsok nélkül nem nyithatjuk meg vagy dekódolhatjuk a fájlokat.

Az EFS felépítése

Titkosítás

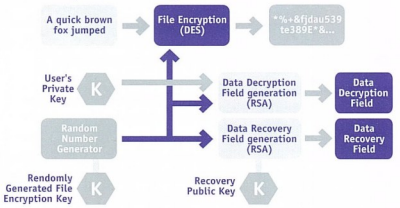
Az EFS az adattitkosítást és -dekódolást az eddig taglaltaknak NEM megfelelően, a közhiedelemmel ellentétben nem nyílt kulcsú algoritmussal végzi. Ennek több oka is van, az egyik ezek közül a teljesítmény: a szimmetrikus titkosítási algoritmusok *(például a DES)* egy nagyságrenddel gyorsabbak, mint a kulcspárral működők. A másik ok pedig a fájl többfelhasználóssá tétele. A nyílt kulcsú kulcspárral NEM a fájl tartalmát titkosítjuk, hanem azt a DES kulcsot, ami a fájl egyedi titkosító kulcsa. Ezt úgy lehet elképzelni, mintha minden jogosult felhasználó, valamint a helyreállítási ügynök rendelkezne a KULCSDOBOZ KULCSÁVAL. Privát kulcsával nem magát a fájlt nyitja, hanem a fájl kulcsát rejtő ládikát.

A fájlok egy véletlenszerűen generált fájltitkosítási kulcs *(file encryption key, FEK)* használatával kerülnek titkosításra, amely független a felhasználó nyilvános vagy privát kulcspárjától; így kizárja a kódanalizáláson alapuló hozzáférési kísérleteket. Ezután a FEK külön *(egy vagy több kulcstitkosítási nyílt kulcs használatával)* titkosításra kerül, így létrejön a titkosított FEK-ek listája. A FEK titkosítása a felhasználók kulcspárjának nyílt részével történik. A titkosított FEK-ek listája ezzel a titkosított fájllal együtt tárolódik a speciális adatdekódolási mező *(Data Decryption Field, DDF)* nevű EFS attribútumban. A fájltitkosítási információ szorosan kapcsolódik a fájlhoz. A dekódoláshoz a felhasználók kulcspárjának privát része kerül felhasználásra. A FEK dekódolásához szintén a kulcspár privát része kell. A felhasználók kulcspárjának privát részét máshol kell biztonságosan tárolni - intelligens kártyákon vagy egyéb biztonságos tárolóeszközökön.

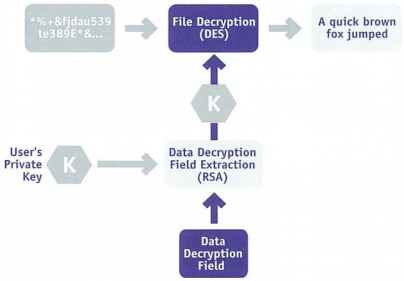
Elvileg a FEK titkosítható lenne szimmetrikus algoritmussal, például jelszóból származó kulccsal is. Az EFS ezt nem támogatja, mivel a jelszó alapú sémák mindenképpen gyenge védelmet nyújtanak a feltörési kísérletekkel szemben.

A titkosított helyreállítási FEK-k e listája szintén a fájllal együtt tárolódik az adat-helyreállítási mező *(Data Recovery Field, DRF)* nevű speciális EFS attribútumban. A DRF-ben csak a helyreállítási kulcspárok nyílt részére van szükség a FEK titkosításához. A normális fájlrendszerműveletekhez ezeknek a nyílt helyreállítási kulcsoknak mindig megtalálhatóknak kell lenniük az EFS rendszerben. A helyreállítás ritkán végzett művelet, amelyre csak akkor van szükség, ha egy felhasználó kilép a vállalattól vagy elveszíti a kulcsát. A helyreállítási ügynökök ezért a kulcsok privát részét máshol *(intelligens kártyán vagy más biztonságos tárolóeszközön)* is tárolhatják.

A következő ábrák a titkosítási, a dekódolási és a helyreállítási folyamatot mutatják be.



A fenti ábra a titkosítási folyamatot mutatja be. A fájl egy véletlenszerűen generált FEK használatával kerül titkosításra. Ez a fájltitkosítási kulcs *(FEK)* a fájllal együtt tárolódik a felhasználó nyílt kulcsával titkosítva a DDF-ben és a helyreállítási ügynök nyílt kulcsával titkosítva a DRF-ben. Ez az ábra csak egy felhasználót és egy helyreállítási ügynököt jelenít meg — a valóságban több, független kulcscsal rendelkező felhasználó és helyreállítási ügynök is lehet *(bár az EFS első kiadása csak egy felhasználót támogat)*. Az alábbi ábra a dekódolási folyamatot mutatja be:



A FEK a felhasználó privát kulcsával és a DDF titkosított FEK elemének használatával dekódolódik. A FEK használatával a fájl adatai blokkonként dekódolódnak. Nagy fájl esetén csak adott blokkok dekódolására van szükség, nem kell az egész fájlt dekódolni.

Felhívom kedves olvasóim figyelmét, hogy e havi Dupla KV rovatunkban egy, az EFS-hez kapcsolódó ügyes trükkről esik szó, melynek segítségével Encrypt/Decrypt menüpontot varázsolhatunk a fájlok gyorsmenüjébe!

Fóti Marcell
marcellf@netacademia.net

Biztonságos Exchange levelezés 1. rész

Biztonságos levelezés Microsoft Exchange 5.5 és Exchange 2000 segítségével

Az elektronikus levelezés nemcsak mennyiségében nőtt az utóbbi években, hanem tartalmában is változik. A multinacionális, és több telephellyel rendelkező cégek, valamint az Interneten folytatott kereskedelem térnyerésével az elektronikus levelek egyre nagyobb hányada tartalmaz bizalmas üzleti információkat, ellentétben az Internetes hőskorban leginkább elterjedt baráti levelezéssel.

Mindezek következtében az elektronikus levelezés biztonsága egyre fontosabbá válik. Gondoljunk csak a nemrégiben a „szerelemvírus" által okozott közvetett és közvetlen károkra, láthatjuk, hogy már ma is mennyire függenek a cégek az elektronikus levelezéstől.

Ebben a cikkben áttekintést adunk az elektronikus levelezés biztonsági kérdéseiről és a biztonság növelésének lehetőségeiről Microsoft Exchange Server 5.5 és Exchange 2000 Server környezetben.

Mi ellen védekezzünk?

Nézzük meg, hogy milyen veszélyek is leselkednek egy levelezőrendszerre?

A legnyilvánvalóbb veszély, hogy az elektronikus levelekhez illetéktelenek férnek hozzá. Ezek az „illetéktelenek" lehetnek cégünkön belül *(ez a gyakoribb!)*, vagy az Interneten. A hozzáférés lehet csak a levél elolvasása, vagy a levél módosítása.

A második fő veszélyforrás az, ha valaki úgy küld valakinek levelet, mintha azt mi küldtük volna, azaz csal.

A harmadik problémakör, hogy a levelezőrendszerünket olyan nagy terhelésnek teszik ki, ami a „hasznos" terhelés kárára megy. Ennek szélsőséges esete az, hogy a levelező kiszolgálónkat, vagy annak bizonyos komponenseit teljesen „lehalasztják" *(Denial of Service, DoS támadás)*.

A negyedik nemkívánatos jelenség a vírusok és trójai falovak bejuttatása e-mailen keresztül. Ezek aztán a legkülönfélébb károkat tudják okozni *(munkaállomások tönkretétele, vagy az előbb említett DoS támadás a levelező kiszolgáló ellen, immár a saját hálózatunkról)*.

Az utolsó, de legáltalánosabb dolog amire fel kell készülnünk, a rendszerünk bármilyen okból történő meghibásodása. Ez ellen nem tudunk tökéletes védelmet kialakítani, hiszen „ami elromolhat az el is romlik", úgyhogy a leghibatűrőbb, legdrágább rendszer esetében is fel kell készülnünk a rendszerünk újraépítésére, ami lehet akár csak a leváladatbázis visszaállítása, de akár lehet komplett levelezőrendszerünk új hardverre történő újratelepítése is.

Helyezzük biztonságba a kiszolgálót

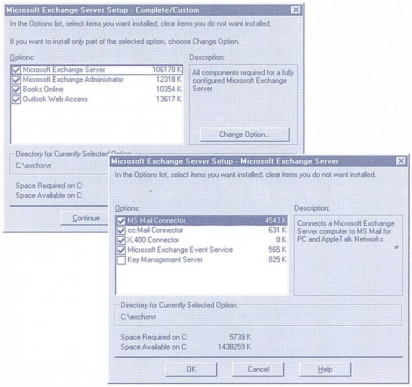
A legfontosabb a védekezési stratégia kialakításában, hogy magát a levelező kiszolgálónkat biztonságossá, illetéktelenek számára hozzáférhetetlenné tegyük. Ez egyrészt azt jelenti, hogy a kiszolgálónk biztonságos módon legyen elhelyezve, például legyen zárva az a helyiség, ahol tartjuk. A hálózatot nem tudjuk elzárni, legalábbis a belső felhasználók elől nem, de a hálózat biztonságát fokozhatjuk. Ezeken kívül az operációs rendszert és a levelező kiszolgálót olyan beállításokkal lássuk el, ami a lehető legnagyobb biztonságot nyújtja.

Most itt következhetne egy általános ismertetés a hálózatok és a Microsoft Windows NT 4.0 és Windows 2000 biztonsági lehetőségeiről, de ez nem tárgya ennek a cikknek. Csak pár „hívószó" álljon itt:

- hub helyett használjunk kapcsolót,
- használjunk tűzfalat, de legalább két hálókártyás kiszolgálót,
- a hálózati kártyákról szedjük le a nem használatos szolgáltatásokat,
- használjunk NTFS fájlrendszert FAT helyett,
- vigyázzunk az „everyone full controll" jellegű alapbeállításokra mind a megosztásoknál, mind az erőforrás-hozzáféréseknél!

Telepítsünk biztonságosra

A cikk következő részeiben az egyszerűség kedvéért feltételezzük, hogy az Exchange 5.5-öt Windows NT 4.0-án használjuk, míg az Exchange 2000-et természetesen Windows 2000 Advanced Serveren futtatjuk Active Directory-val a háttérben. Az Exchange kiszolgálónk telepítésére is érvényes az előbb említett általános tanácsok közül jó néhány. Az Exchange 5.5 esetében az telepítésnél csak azokat a szolgáltatásokat rakjuk fel, amit ténylegesen használunk. Először is ne használjuk „Typical" meg a „Minimal" lehetőséget a telepítésnél, mivel úgysem jegyezzük meg fejből, hogy például a Microsoft által tipikusnak vélt komponensek mik is. Ezért a „Complete/Custom" lehetőség irányába haladjunk. Ez is egy kicsit „becsapós", mert a választható komponensek az „Exchange Server" „mögött" vannak elrejtve, a „Change Option..." gomb megnyomása után megjelenő dialógusablakban választhatók ki, így gyakran ottfelejthetjük például a biztonsági veszélyt rejtő MSMail Connectort, vagy épen a biztonságot szolgáló Key Manager Server marad ki, mert az alapértelmezésben például nincs kiválasztva.



Az exchange 5.5 telepítő alapbeállításai

A következő biztonsági szempontból kritikus lépés az amúgy egyszerű folyamatban a szolgáltatás által használt felhasználói fiók, azaz „service account" kiválasztása. A telepítő párbeszédpanelje alaphelyzetben azt a fiókot ajánlja fel, akinek a nevében a telepítés éppen folyik. Ezt általában nem jó választani több okból sem. Egyrészt a telepítést végző személlyel ellentétben a szolgáltatás-fióknak nem kell rendszergazdának lennie, azaz sérül a „szükséges minimális jogkörök kiosztása" szabály, másrészt, ha figyelmetlenségből a különösen védendő adminisztrátori fiók jelszavát úgy változtatjuk meg, hogy ezt nem jelezzük az Exchange számára, akkor a kiszolgáló egy esetleges újraindulásnál nem fog rajta az Ex-

nehézségi fok:

Microsoft BackOffice

change elindulni. Azaz a telepítés előtt hozzunk létre egy külön felhasználói fiókot, jó nehezen kitalálható jelszóval, amit csak az Exchange szolgáltatás futtatására használunk, és ezt válasszuk ki a telepítéskor.
Ennek a fióknak alaphelyzetben nem kell semmi különös jogot adni, illetve nem kell semmilyen speciális csoportba tenni, csak az alaphelyzet szerinti „Domain Users" csoportban kell bennhagyni, a szükséges egyéb jogokat a telepítő automatikusan megadja. Néhány egyéb komponens, mint például egyes vírusvédelmi megoldások, vagy a korábban említett Key Management Server azonban egyéb igényeket is támaszthatnak, ezt természetesen vegyük figyelembe. Ilyenek lehetnek bizonyos könyvtárak olvasásának joga, lokális bejelentkezési jog, esetleg helyi adminisztrátori jog.
Az Exchange 2000 telepítése ilyen szempontból egyértelműbb, a komponensek a telepítő-varázsló megfelelő lapján kiválaszthatók. A szolgáltatási fiók pedig maga a System Account, hiszen a Windows 2000-ben már ez a fiók is alkalmas hálózaton keresztül más gépeken szolgáltatások igénybevételére, ami az Exchange esetében például az Active Directory eléréséhez kell. Ez a fiók biztonságosabb, mint bármilyen más, általunk felvett felhasználói fiók, hiszen ennek a jelszavát csak a gép tudja, és a Windows 2000 automatikusan változtatja 7 naponként.
A telepítés utolsó lépéseként tegyük föl az aktuális javítócsomagokat, ami jelenleg az Exchange 5.5-nél az SP3 *(4-es hamarosan megjelenik)*, az Exchange 2000-nél még nincsen, de már készül. A javítócsomagok nemcsak szigorúan vett javításokat tartalmaznak, hanem éppen a biztonsági lehetőségek terén újabb funkciókat is adnak.

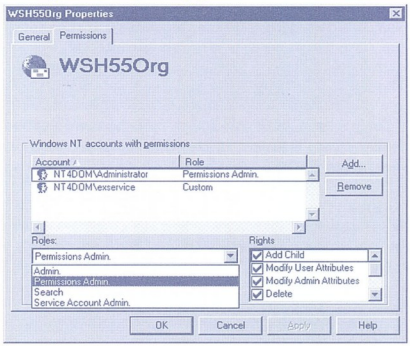
Telepítés utáni első teendők

A telepítés után a „friss" Exchange-nél van néhány olyan biztonsági hiányosság, amit rossz szándékú emberek kihasználhatnak, ezért érdemes ezeket először befoltozni, mert ha már lehetővé tesszük az emberek számára, hogy bejelentkezzenek az Exchange rendszerbe, akkor már lehet, hogy késő, és néhányan „szórakozásból" tönkretehetik azt.
Mindkét Exchange verzió esetében telepítés után szabad a vásár a nyilvános mappák létrehozására és ezekbe korlátlanul tölthetnek fel dokumentumokat tetszőleges méretben. Ugyanilyen szabadsággal rendelkeznek a saját levelesládáikban is. Mindezekkel rövid idő alatt annyi dokumentumot tölthetnek a kiszolgálóra, hogy az „meghal".
Amennyiben használjuk a levelek nyomkövetése *(Message Tracking)* funkcióit, akkor a nyomkövetési információkat tartalmazó mappának *(Exchange 5.5: Tracking.log, Exchange 2000 esetében szervernév.log)* megosztási jogaira legyünk figyelemmel, hiszen alaphelyzetben ez a mappa mindenki számára olvasható a hálózaton keresztül, így akár bizalmas levelekről is kinyerhetnek fontos információkat.

Jogosultságok

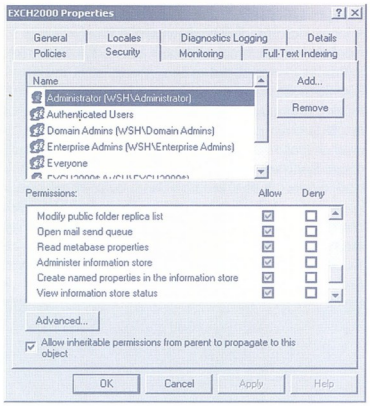
A legtöbb biztonsági rést azáltal tehetjük a rendszerünkbe, ha nem megfelelően állítjuk be a jogosultságokat. Ugye gyakori eset, hogy a levelező kiszolgálónkat nemcsak egy embernek kellene rendszer-felügyelni, hanem több rendszergazdát szeretnénk különböző szereppel felruházni. Itt mindig ügyeljünk a „minimális szükséges jog" kiadásának szabályára.
Az Exchange 5.5 rendszernél kicsit hibrid a megoldásunk, hiszen külön van az NT biztonsági rendszere és adatbázisa *(SAM)*, és az Exchange 5.5 címtára *(Directory)*. A kettő közötti kapcsolatot az NT felhasználói fiókok részére az Exchange címtár objektumaira kiadott jogosultságok jelentik. A jogok kiosztásának megkönnyítésére az Exchange 5.5 szerepeket definiál, amelyek beállításával egy előre elkészített jogosultsághalmazt tudunk egy mozdulattal beállítani.
Bővebben itt nem kívánunk kitérni a jogosultságok és szerepek rejtelmeire, de annyit biztonság szempontjából fontos megjegyezni, hogy ha az Exchange 5.5 címtárból minden olyan felhasználót leszedünk, akinek van joga a jogosultságokat átállítani *(Permissions Admin)*, akkor címtárunk meglepő módon éppen védtelen lesz. Azaz annak érdekében, hogy mindenki kizárásával nehogy a rendszer felügyelhetetlen legyen, inkább azt választották a fejlesztők, hogy ilyen esetben inkább legyen szabad a gazda, azaz bárki, aki eléri az Exchange 5.5 Servert tudja azt konfigurálni :-0

Exchange 5.5 szerepek és jogok



Az Exchange 5.5 esetében az is nehézkessé teszi a biztonságos jogosultsági rendszer kiépítését, hogy az Exchange üzenet-adatbázisaiban *(Mailbox Store és Public Store)* és bizonyos korlátozásoknál már nem NT fiókoknak osztunk ki és veszünk el jogokat, hanem az Exchange címtárban definiált postaládáknak és csoportoknak *(Distribution Lists)*.
Az Exchange 2000 esetében az Active Directory integráció miatt itt egyben, az NTFS jogosultsági rendszeréhez teljesen hasonló módon kezelhetjük a jogosultságokat, azaz akinek megfelelő joga van az Active Directory Exchange objektumaihoz hozzáférni, akkor az tudja konfigurálni az Exchange rendszerünket. Ezen még az is újdonság az 5.5-höz képest, hogy az objektumhierarchia magasabb szintjén kiadott jogokat alsóbb szinten korlátozni tudjuk, ezzel sokkal hatékonyabban és finomabban tudjuk azokat kiadni.

Exchange 2000 jogok az Exchange System Managerben



Természetesen bizonyos műveletekhez szükséges magán az Exchange szolgáltatásokat futtató gépen is hozzáféréssel rendelkeznie a rendszergazdának, de például postaládát tud létrehozni egy olyan „Domain Admin" rendszergazda is, akinek az Exchange 2000 rendszerben csak „View Only" szerepe van. Azaz az Exchange 2000 esetében ügyelni kell arra, hogy bizonyos funkciók feletti ellenőrzés kikerült az Exchange Server felügyelete alól és az Active Directory-hoz került. Az Exchange rendszerben jogosultságokat egy ú.n. „Delegation Wizard" segítségével tudunk legegyszerűbben kiosztani, de természetesen módunk van arra is, hogy minden egyes objektumra a varázsló nélkül osszunk ki jogokat.

Hálózati és tartalom-titkosítás, elektronikus aláírás

Eddig azzal foglalkoztunk, hogy magát a levelező kiszolgálónkat megvédjük, de mi van magukkal a levelekkel, amik elhagyják a kiszolgálót és a belső hálózaton, vagy az interneten elkezdenek vándorolni?

A belső hálózatunkban viszonylag biztonságos a levelek vándorlása Exchange Server és Outlook között abból a szempontból, hogy ha valaki figyeli a hálózatunkat, akkor nem fogja a levelek tartalmát egy az egyben látni, mert az RPC alapú kommunikáció eleve nem „clear text", de nem visszafejthetetlen! Ha ennél biztosabbat akarunk, akkor használhatjuk az Outlookban az RPC kommunikáció titkosítását; a régebbi verzióknál 40 bites titkosításunk volt, az Outlook 2000 SR1-nél már 128 bites titkosításra van lehetőségünk.

Az előbb említett védelem csak arra az időre védi a levelünket, amíg a hálózaton utazik. Mi van, ha illetéktelen helyre érkezik? Ott minden további nélkül ismét olvashatóvá válik. Ez úgy képzelhető el, hogy a fürkésztekintetű rendszergazdánk „véletlenül" jogot oszt ki a mi postafiókunkra magának és ezek után gond nélkül nézegeti a levelezésünket.

Hasonló problémánk lehet az internetes levelezéskor, azzal kiegészítve, hogy alaphelyzetben a levélküldés protokollja *(SMTP)* és az internetes kliensek *(pl.: Outlook Express)* levélletöltő protokollja *(POP3, IMAP4)* is olvasható szöveges formátumban továbbítja a leveleket, azaz mind a hálózati forgalmat figyelők, mind az esetlegesen levelünket továbbító kiszolgálókhoz hozzáférők olvashatják a levelezésünket.

Ezen problémák kiküszöbölésében segít a levél titkosítása, melynek során különböző titkosítási kulcsokkal olvashatatlanná tesszük azt.

Az internetes levelezés esetén is van mód magának a kommunikációnak a titkosítására, azaz hogy nemcsak a levelet titkosítjuk, hanem magát azt a protokollt is, amelyen keresztül a levél eljut egyik helyről a másikra. Ezt a Secure Socets Layer és a Transport Layer Security protokollok szolgálják, amelyek a felsőbb szintű protokollok *(pl.: IMAP, http, SMTP, POP)* alatt képeznek titkos csatornát. Hasonló célt szolgál a Virual Private Network kialakítása is *(PPTP vagy L2TP)*.

Amellett, hogy garantálni kell, hogy levelünket tényleg csak a címzett tudja elolvasni, őt is meg kell nyugtatni, hogy azt a nagyon titkos levelet tényleg mi küldtük, méghozzá úgy, hogy abba útközben nem írt bele senki. Ez nem feltétlenül ugyanaz, mint titkosítani valamit, hiszen lehet, hogy nem bánom, hogyha a levelemet bárki elolvassa, csak az a fontos számomra, hogy aki elolvassa az egyértelműen meg tudjon győződni arról, hogy azt tényleg én írtam. Erre való az elektronikus aláírás.

Titkosítás kulcsokkal, bizonyítványokkal

Hogyan tudjuk elérni mindazt, amit az előzőekben leírtam? Okos emberek kitaláltak különböző matematikai algoritmusokat arra, hogy hogyan lehet speciális számokkal *(kulcsokkal)* az adatokat jelentő másik számokat úgy megváltoztatni, hogy a visszafejtéshez csak az eredeti kulcs, vagy egy bizonyos másik kulcs alkalmas, azaz csak úgy kitalálni nem lehet a kódoló kulcsot, csak próbálgatással. Ha a kulcs elég nagy lehet, akkor kitalálni azt olyan sok próbálgatással jár, amit a jelenlegi technikákkal nagyon sok időbe telne, akár több száz évbe is. Gondoljunk egy olyan aktatáska kinyitására, amelyen nem 4 számjegy a számzár, hanem pl. 100!

A titkosításra két fő módszer van. Az egyik az, hogy a címzett és a feladó megegyezik egy közös kulcsban, amivel a feladó titkosítja a levelet, majd ugyanezzel a kulccsal a címzett visszafejti. A másik módszer a kétkulcsú vagy más szóval nyilvános kulcsú titkosítás, amelyben az adatot a feladó a címzett nyilvános, vagy más szóval titkosító kulcsával titkosítja, elküldi, és ezt csak a nyilvános kulcs privát párjával rendelkező címzett tudja visszafejteni. Az első előnye, hogy a rendelkezésünkre álló algoritmusok nagyon gyorsak, azaz nagy mennyiségű adatot, vagy akár adatfolyamokat is lehet vele titkosítani. A második előnye, hogy nem kell azzal törődni, hogy vajon azt az egy kulcsot, ami az egykulcsú titkosításnál van, vajon hogyan juttatjuk el biztonságos módon a feladónak, hanem mivel itt kulcspárok vannak, az egyik felét, a nyilvános kulcsot bárkinek, bármilyen módon elküldhetjük.

De vajon mi biztosítja, hogy ha én Vegetári János kollégámnak akarok egy titkos levelet küldeni, és megkérem, hogy küldje el a titkosításra használt nyilvános kulcsát *(és kapok is egy kulcsot)*, hogy az tényleg tőle van? Hiszen ezt a kérést, amíg nem tudok vele titkosan levelezni, bárki elcsípheti, és gyorsan elküldi a saját kulcsát. Az ilyen jellegű problémákra találták ki a bizonyítványt kiadó hatóságokat *(Certificate Authority)*. Ezek akár olyan ténylegesen működő szervezetek is lehetnek, akik számára valamilyen módszerrel hitelesítem magam, amit ők elismernek, és az én nyilvános kulcsomat hajlandóak minősíteni, hogy az tényleg az enyém. Ha Vegetári János kollégámtól kapok egy bizonyítvánnyal ellátott kulcsot, akkor megbizonyosodhatok arról, hogy ez tényleg az övé, például úgy, hogy lekérem az interneten keresztül a hitelesítő cégtől, hogy az ilyen és ilyen sorszámú bizonyítvány tényleg hiteles-e.

Mit tudunk mindezekkel a technológiákkal kezdeni?

- **Titkosítás.** Generálunk egy titkos kulcsot, és egykulcsú titkosítással titkosítjuk a levelet, majd a végére biggyesztjük a címzett nyilvános kulcsával titkosított titkos kulcsot. Ennek az az egyik előnye, hogy gyors *(a nagy adattömeg egykulcsú titkosítással van kódolva)*, és mégis titkosan megy át a címzetthez a titkosító kulcs. Másrészt, ha több címzettem van, akkor nem az egész üzenetet kell minden egyes címzett nyilvános kulcsával kódolni, hanem csak a titkosító kulcsot, ami általában jóval kisebb adattömeget jelent, mint maga a levél. A címzett*(ek)* a privát kulcsá*(ai)*val kibontja*(ák)* a titkosító kulcsot, és azzal dekódolja*(ák)* a levelet.
- **Elektronikus aláírás.** Készítek egy hash-t *(hash = az adatból létrehozott, annál jóval kisebb szám, amiből nem lehet visszafejteni az eredeti adatot, és ha az adat akár csak 1 számjegyét is megváltoztatom, akkor a hozzá tartozó hash lényegesen megváltozik)* a levélből, és azt kódolom a privát kulcsommal, és ezt a kódolt hash-t, a nyilvános kulcsomat és annak bizonyítványát a levél mellé rakom. Ezzel maga a levél olvasható marad *(átlátszó aláírás)*, de a címzett tudja ellenőrizni a mellékelt bizonyítványt, és meg tud győződni, hogy a levelet tényleg én küldtem. Sőt, ha ő is elvégzi a hash képzést, és összehasonlítja azzal, amit én kódolt formában a levélhez raktam, akkor el tudja dönteni, hogy a levelet útközben módosították-e. Ha módosították volna, akkor eltérne egymástól a két hash. *(Csinálhatunk ú.n. nem átlátszó aláírást is, ami az egész üzenetet beteszi magába az aláírásba, bináris formában, hogy ha a levél különböző levelező kiszolgálókon megy át, nehogy azok véletlenül megváltoztassák, pl. a szöveges részt áttördeljék, ezzel megsértve a levél módosítatlanságát.)*

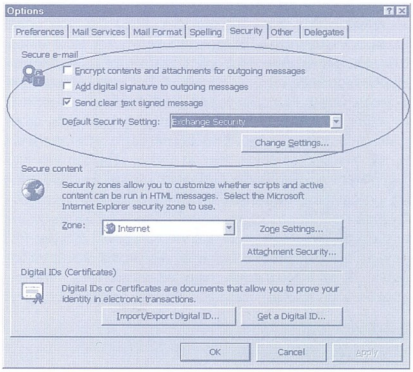
Térjünk csak vissza egy pillanatra a bizonyítványok kiadására! Honnan tudom, hogy egy bizonyítványt tényleg az a szervezet küldte, akitől én kértem? Hát onnan, hogy a bizonyítványt a kiállító szervezet is aláírja elektronikusan. Sőt, miért higgyek én mindenféle kis hitelesítő szervezet aláírásában? Általában a hitelesítő szervezet aláírásában benne van az is, hogy őt milyen felettes hitelesítő szervezet minősítette. Ha bizalmatlan vagyok, végig tudom ellenőrizni az egész minősítési hierarchiát. A tetején sajnos lesz egy olyan CA, amit muszáj elfogadnom hitelesnek, mert azt már nem fogja senki hitelesíteni, csak saját maga.

Certificate Server és Key Management Server

Az Exchange 4.0 verziójától kezdődően a telepítő CD-n megtalálható az emelt szintű biztonságot megvalósító opcionális komponens, a Key Management Server *(KMS)*. Ez teszi az Outlook kliensprogram felhasználásával, lehetővé teszi hogy a felhasználók elektronikusan titkosított és/vagy aláírt leveleket küldjenek egymásnak. A KMS két kulcspárt, egy aláíró és egy titkosító kulcspárt kezel a felhasználóknak, és egy külön adatbázisban tárolja annak érdekében, hogy ha a felhasználók elveszítik a privát kulcsaikat, akkor vissza lehessen állítani azokat ebből az adatbázisból. Az Exchange 5.5 SP1 óta ezeket a kulcsokat olyan bizonyítvánnyal láthatjuk el, amelyek megfelelnek a Secure MIME előírásoknak, így szabványosak, tehát a nem Exchange és nem Outlook felhasználóknak is tudunk ezzel titkosított és aláírt leveleket küldeni. Az ehhez szükséges bizonyítványokat nem maga a KMS állítja elő, hanem szükséges egy saját Certificate Servert is igénybe venni ebből a célból *(NT 4-ben az Option Pack-ben van, a Windows 2000-ben már „gyárilag")*.

A KMS használatához tehát először szükségünk van egy Certificate Serverre. Ez az Exchange 5.5 esetében egy dedikált Certificate Server lesz, mivel rá kell installálnunk egy olyan speciális biztonsági modult, ami alkalmassá teszi arra, hogy megértse a KMS bizonyítvány-igénylését, és ezzel a „normál" felhasználói bizonyítványkérésekre alkalmatlanná tesszük. Az Exchange 2000 esetében már nem kell ilyen durva beavatkozást véghezvinnünk a Certificate Serveren, hiszen ez már alaphelyzetben megérti a KMS kéréseit. Amiről gondoskodni kell viszont Exchange 2000 esetében, az a Certificate Server Active Directory integrációja, azaz ú.n. Enterprise CA-t kell létrehoznunk. Erre azért van szükség, mert a bizonyítványokat és az egyes felhasználók titkosítási képességét a KMS az Active Directory-ban fogja tárolni, illetve Active Directory objektumok, az Exchange Servert futtató gépek fognak bizonyítványokat igényelni a felhasználók részére. Ha elkészítettük és felkonfiguráltuk a KMS komponensünket, akkor már csak a felhasználóinkat kell bevonni az emelt szintű biztonság lehetőségébe. *(Ez azért nem ilyen egyszerű, de ezt inkább gyakorlatban kellene megmutatni.)*

Outlook kliens az emelt szintű biztonsági lehetőséggel

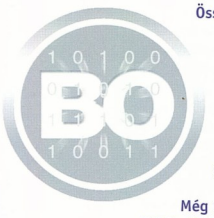


A cikk következő részében az Outlook Web Access, az internetes kliens protokollok, az SMTP relay lehetőség, kerül terítékre a mentés helyreállítás problémáiról, vírusvédelem lehetőségeiről és az Outlook 2000 SR1 biztonsági újdonságairól is.

A téma iránt érdeklődőknek ajánljuk a WSH és a Netacadémia Exchange Serverek biztonságáról szóló tanfolyamát, érdeklődjön e-mailben!

Soós Tibor (MCSE+I, MCT)
WSH Oktatóközpont
t.soos@wsh.hu

Transact SQL 2. rész



Összetett lekérdezések

Cikkünk előző részében elindultunk a Microsoft SQL 2000 programozásának izgalmas útján. Megnéztük, hogyan írhatunk egyszerű lekérdezéseket, amelyekkel apróbb feladatokat adhatunk az adatbázisnak. Ebben a részben jobban belemélyedünk a lekérdezések lelkivilágába, és megnézzük, hogy komolyabb feladatokat hogyan oldhatunk meg a Transact SQL segítségével.

Még mindig együtt!

Előző cikkünk végén az illesztésekkel *(JOIN)* foglalkoztunk, és eljutottunk odáig, hogy illesztés segítségével logikailag összetartozó, de fizikailag több táblára szétdarabolt adatokat újra egyesíthetünk. Megbeszéltük, hogy az INNER JOIN segítségével meg lehet találni az összetartozó sorokat. Megnéztük, hogy vannak olyan esetek, amikor nemcsak a párok érdekesek, hanem szükség van azokra a sorokra is, amelyekhez nincs kapcsolódó sor más táblákban, ilyenkor használtuk az OUTER JOIN-t. A LEFT és a RIGHT OUTER JOIN segítségével kilistáztathattuk azokat a sorokat is, amelyeknek nem volt párja a másikban.

Az OUTER JOIN eddig még nem említett válfaja a FULL OUTER JOIN. Ez mindkét tábla tartalmát kilistázza, függetlenül attól, hogy talált-e egyezést a másik táblában, vagy sem. Ennek felhasználása már elég speciális. Például a Northwind adatbázis Customers és Orders táblái között egy LEFT OUTER JOIN-nak van értelme, hisz kilistázza azokat a vásárlókat, akiknek nincsenek megrendeléseik. A fordított helyzet *(egy jól megtervezett és implementált adatbázisban)* elvileg elő sem állhat, azaz, hogy vannak olyan megrendelések, amelyekhez nincs megrendelő. Ez a hivatkozási *(referential)* integritás megsértése volna, hisz az Orders táblában van egy idegen kulcs *(foreign key)* a Customers táblára. Ennek ellenére a gyakorlatban sokszor előfordul, főleg, amikor egy régebbi rendszerből költöztetünk adatokat egy újabba, hogy bizony sok helyen baj van az adatok épségével. Tegyük fel, hogy az előbb említett két táblát egy másik adatbázisból kaptuk, és az a feladatunk, hogy állapítsuk meg, rendben vannak-e a hivatkozási szabályok. Mi sem egyszerűbb:

```
SELECT
   Customers.CustomerID,
   Customers.CompanyName,
   Orders.CustomerID
FROM
   Customers
FULL OUTER JOIN
   Orders
ON
  Customers.CustomerID = Orders.CustomerID
WHERE
   Orders.CustomerID IS NULL OR
   Customers.CustomerID IS NULL
```

Mit várunk a lekérdezéstől? Azt, hogy kilistázza az összes megrendelést, amelynek nincs gazdája *(Customers.CustomerID IS NULL)*, és kilistázza azokat a vásárlókat, akiknek nincs megrendelése *(Orders.CustomerID IS NULL)*. Az adatok értelmezését figyelembe véve csak az előbbi valódi probléma, az utóbbi nem. Ez azért van, mert logikailag a Customers és az Orders tábla között egy vagy több kapcsolat van, azaz minden tételhez a Customers táblában tartozhat nulla vagy több tétel a másikban. Ennek megfordításaként viszont minden egyes tételhez az Orders táblában kell lennie egy megfelelő tételnek a Customers táblában. A teljesség kedvéért íme a lekérdezés kimenete, melyből látszik, hogy nincs árva megrendelés *(ahol az első CustomerID NULL értékű lenne)*:

```
CustomerID CompanyName          CustomerID
FISSA      FISSA Fabrica S.A.   NULL
PARIS      Paris spécialités    NULL
```

Az illesztések közül már csak egy maradt hátra, amelyet csak nagyon ritkán, elsősorban tesztadatok generálására használunk. Ennek neve CROSS JOIN, és a matematikából ismert Descartes szorzatot valósítja meg. Adatbázisra lefordítva ez azt jelenti, hogy az első tábla minden sorát összepárosítja a másik tábla minden sorával, azaz tulajdonképpen egy speciális INNER JOIN, amelynek feltétel része *(ON ...)* mindig igaz.

Nézzük meg, hogy a CROSS JOIN segítségével hogyan lehet kevés kiinduló adatból nagyszámú tesztadatot generálni! Tegyük fel, hogy teszt felhasználókra van szükségünk. Kiindulásként felvittük kilenc személy vezeték- és keresztnevét egy táblába *(Employees)*, azt szeretnénk, hogy a vezeték- és keresztnevek kombinálásával előállítsunk felhasználói neveket. Ha minden vezetéknevet összepárosítunk minden keresztnévvel, akkor 9x9=81 nevet fogunk kapni. Hogyan néz hát ki a generáló script?

```
SELECT
   E1.FirstName, E2.LastName
FROM
   Employees E1
CROSS JOIN
   Employees E2

...
Robert      Buchanan
Laura       Buchanan
Anne        Buchanan
Nancy       Callahan
Andrew      Callahan
...
```

Két csel is el van rejtve ebben a rövid lekérdezésben. Lehet egy táblát önmagával illeszteni? Igen! Ezt hívják self joinnak. De honnan tudja a SELECT, hogy mikor melyik példányra hivatkozunk? Onnan, hogy átnevezzük őket, álnevet adunk nekik *(alias)*. Így bárhol a lekérdezésben az első Employees táblára E1 néven lehet hivatkozni, míg a másikra E2 néven. Így a fordító nem fog kétségek között vergődni, hogy éppen mire gondoltunk. Tábla álneveket bármikor használhatunk, nem csak illesztések esetén. Sokszor hosszú táblaneveket rövidítünk velük, például az [Orders Details] táblát od-re.

Maradjunk még az illesztéseknél, mert sok olyan finomság van bennük, amelyeket ha nem tud valaki előre, csak több napos bosszankodás után fogja felfedezni.

A fránya *=

Az illesztések formális leírásának van egy másik formája is, amelyet szándékosan elhanyagoltam eddig, mert elavult, és nem lehet vele minden feladatot egzaktul megfogalmazni. Mivel azonban nagyon sokan használják, nem hagyhatom ki a tárgyalásból, hisz ha már együtt kell élnünk vele, legalább ismerjük az árnyoldalait. Összehasonlításként leírok egy lekérdezést az INNER JOIN felhasználásával, majd a régi módon:

nehézségi fok: ●●○○

Microsoft BackOffice

```
SELECT
   Customers.CustomerID,
   Customers.CompanyName,
   Orders.OrderDate
FROM
   Customers
INNER JOIN
   Orders
ON
   Customers.CustomerID =
Orders.CustomerID
```

Régi módon:

```
SELECT
   Customers.CustomerID,
   Customers.CompanyName,
   Orders.OrderDate
FROM
   Customers, Orders
WHERE
   Customers.CustomerID =
Orders.CustomerID
```

Azaz válogassa ki azokat a sorokat a két táblából, ahol *(WHERE)* a Customers.CustomerID = Orders.CustomerID. Teljesen logikus, és nincs is vele baj. Ha keresni akarjuk a kákán a csomót *(és miért ne tennénk)*, akkor hogy van az, hogy a WHERE után lehetnek olyan kifejezések, amelyek sorok szűrését végző feltételeket tartalmaznak, és olyanok is, amelyek táblák logikai összekapcsolását tartalmazzák? Nem két, teljesen különböző funkcióról van itt szó? De! És ez visszsza is fog ütni mindjárt *(megvan az első csomónk)*!
A probléma az OUTER JOIN-oknál kezdődik. A régebbi OUTER JOIN-t megvalósító kifejezés a WHERE a *= b volt, és attól függően, hogy a csillag melyik oldalán van az egyenlőségjelnek, lehet jobb vagy bal oldali illesztést kifejezni. Ez ugyanazt jelenti, mint az OUTER JOIN? *(És most mindenki tegye fel és válaszolja meg magának ezt a kérdést, mielőtt tovább olvasna.)* Nem! Nézzük meg egy példán keresztül a csalást!
Új formátum:

```
SELECT
   Customers.CustomerID,
   Customers.CompanyName,
   Orders.OrderDate
FROM
   Customers
LEFT JOIN
   Orders
ON
   Customers.CustomerID =
   Orders.CustomerID
```

Régebbi formátum:

```
SELECT
   Customers.CustomerID,
   Customers.CompanyName,
   Orders.OrderDate
FROM
   Customers, Orders
WHERE
   Customers.CustomerID *= Orders.CustomerID
```

A két lekérdezés kimenete azonos:

```
BONAP       Bon app'              1998-05-06
RATTC       Rattlesnake Canyon    1998-05-06
PARIS       Paris spécialités     NULL
FISSA       FISSA Fabrica S.A.    NULL
```

Akkor miért kritizálom a *= formátumot? Mindjárt kiderül. Próbáljuk meg kiszűrni például a Bon app' céget a listából, ügyelve arra, hogy a LEFT JOIN által behozott NULL-okat ne szűrjük ki. Az új formához csak egy feltételt kell adni:

```
WHERE
   Orders.CustomerID <> 'BONAP' OR
   Orders.CustomerID IS NULL
```

```
RATTC       Rattlesnake Canyon 1998-05-06
PARIS       Paris spécialités  NULL
FISSA       FISSA Fabrica S.A. NULL
```

És a kimenetből tényleg eltűnt a kérdéses sor, míg a NULL-osok megmaradtak. Egy kis magyarázatot azért megér, hogy miért kell a második feltétel is, miért nem elég csak az első. Az előző részben részletesen foglalkoztunk vele, hogy a NULL azt jelenti, hogy nincs adtat, így az Orders.CustomerID <> 'BONAP' feltételnél kiesnének a NULL-okat tartalmazó sorok, mert egy NULL-al végzett összehasonlításnak nem lehet eldönteni az igazságtartalmát. Ezért kellett bevetni az IS NULL-t.
Itt az ideje, hogy kiugrasszuk a nyulat a bokorból! Írjuk át a lekérdezést a régi szintaxisra:

```
WHERE
   (Customers.CustomerID *=
   Orders.CustomerID) AND
   (Orders.CustomerID <> 'BONAP' OR
   Orders.CustomerID IS NULL)
```

És mit látunk kimenetnek?

```
RATTC       Rattlesnake           1998-05-06
PARIS       Paris spécialités     NULL
FISSA       FISSA Fabrica S.A.    NULL
BONAP       Bon app'              NULL
```

Ott virít a Bon app', pedig kiszűrtük *(ráadásul lemaradt a megrendelése is)*! Miért? Azért, mert a NULL ellenőrzése előbb történik meg a szerverben, mint az illesztés, így a bal illesztés behozza újra a kiszűrni kívánt sort. De hogy lehettünk ennyire figyelmetlenek, miért nem a Customers.CustomerID-ra szűrünk, miért az Orders.CustomerID-ra. Ez biztos bejön! Nézzük csak:

```
WHERE
   (Customers.CustomerID *=
   Orders.CustomerID) AND
   (Customers.CustomerID <> 'BONAP' OR
   Orders.CustomerID IS NULL)
```

Nem mutatom meg a kimenetet, de még mindig benne van a BONAP! Hogy lehet ez? Úgy, hogy a Orders.CustomerID IS NULL most az illesztés után hajtódott végre, így behozta a BONAP-ot. Szeszélyesebb az adatbázis motor, mint az időjárás! Vagy mégsem?

A megoldás

Ne fokozzunk tovább a feszültséget! Miután rájöttünk, hogy az IS NULL vizsgálat hozza be a nemkívánatos sorokat, vegyük ki a lekérdezésből.

```
WHERE
   (Customers.CustomerID *=
Orders.CustomerID) AND
   (Customers.CustomerID <> 'BONAP')
```

És eltűnt a Bon app'! Lehet, hogy ez sok hűhó semmiért, és hogy ez egy olyan nyilvánvaló dolog volt, amit egy tapasztalt SQL programozó azonnal kiszúr. Lehet, bár azért még nekik is lehet fejtörést okozni. Mert rakjuk csak be a régi stílusú külső illesztésünket egy SQL nézetbe *(View)*. Azoknak a kedves olvasóknak, akik még nem használtak nézeteket, néhány szó róla. Nézetekbe olyan lekérdezéseket szoktunk „becsomagolni", amelyeket több helyen is fel fogunk használni, így nem kell mindig leírni őket. A nézetek úgy viselkednek, mintha ők valamiféle virtuális táblák lennének, amelyek a bennük található lekérdezéseket táblaként adják vissza. Na már most, tegyük fel, hogy több ember dolgozik egy projekten. Az egyik megírja a már sokszor emlegetett lekérdezést egy nézetbe, természetesen a régi szintaxissal. Legyen a nézet definíciója:

```
CREATE VIEW
   CustOrders
AS
SELECT
   Orders.CustomerID,
   Customers.CompanyName,
   Orders.OrderDate
FROM
   Customers, Orders
WHERE
   Customers.CustomerID *=
   Orders.CustomerID
```

Ezután a tudatlanság boldogságában leledző társprogramozó ki szeretné szűrni a BONAP-ot:

```
SELECT
   CustOrders.CustomerID,
   CustOrders.CompanyName,
   CustOrders.OrderDate
FROM
   CustOrders
WHERE
   CustOrders.CustomerID <> 'BONAP' OR
   CustOrders.CustomerID IS NOT NULL
```

És itt áll égnek a haja, mert a szerver látszólag nem működik normálisan, hisz nem szűrte ki a megfelelő sort!
Összegezve: ne használjuk a régi formátumú illesztéseket, mert félreértésekhez vezethet. Emellett a későbbi verziójú SQL Serverek nem fogják támogatni. Ha más nem, ez elégséges érv lehet.

Egymásba ágyazva

Az SQL nyelv egyik leghatékonyabb eszköze, hogy a WHERE feltételbe nemcsak egyszerű logikai kifejezéseket írhatunk, hanem további lekérdezéseket is. Ráadásul a két lekérdezés között lehet kapcsolatot is teremteni. Például listázzuk ki azokat az alkalmazottakat, akik már teljesítettek megrendeléseket, azaz az Orders táblában van olyan sor, ami az Employee táblában található alkalmazottra mutat:

```
SELECT
   LastName, FirstName
FROM
   Employees
WHERE
   Employees.EmployeeID IN
   (SELECT
      EmployeeID
    FROM
      Orders)
```

Másképpen fogalmazva listázzuk ki az EmployeeID-kat az Orders táblából, majd az Employees táblából válogassuk le azokat a sorokat, amelyeknek az EmloyeeID-ja megegyezik valamelyik Orders táblából származó EmployeeID-val (IN). Ezt a lekérdezést át lehetne írni JOIN-ra is:

```
SELECT
   LastName, FirstName
FROM
   Employees
INNER JOIN
   Orders
ON
   Employees.EmployeeID =
Orders.EmployeeID
```

Általánosságban igaz, hogy minden JOIN-t át lehet írni egymásba ágyazott lekérdezésre, de visszafelé ez nem feltétlenül igaz. Azaz vannak olyan egymásba ágyazott lekérdezések, amelyek egy egyszerű illesztésnél bonyolultabb dolgot valósítanak meg. Például listázzuk ki azokat a termékeket *(Products)*, amelyek ára 10 dollár alatt van, de volt olyan alkalom, amikor egyszerre eladtak valamelyik termékből több mint 800 dollárnyit.

```
SELECT
   ProductID, ProductName
FROM
   Products AS p
WHERE
   UnitPrice < 10 AND
   ProductID IN
   (SELECT ProductID
    FROM
      [Order Details] od
    WHERE
      od.Quantity * p.UnitPrice > 800)

ProductID   ProductName
41           Jack's New England Clam
45           Rogede sild
75           Rhönbräu Klosterbier
```

Microsoft BackOffice

Hogyan képzelhetjük el a lekérdezés működését? Az első SELECT végiglépked a Products tábla azon sorain, melyekben a UnitPrice mező értéke kisebb, mint 10. Minden kiválasztott sornál elindít egy belső ciklust *(a belső SELECT)* az Order Details táblára, és keres olyan sorokat, amelyekre teljesül a WHERE-ben megadott feltétel. Ennek specialitása, hogy a külső SELECT által pillanatnyilag kiválasztott sorból származó adatot is felhasználja *(pl.UnitPrice)*. Az ilyen típusú egymásba ágyazott lekérdezéseket Correlated Subquery-nek nevezzük.
Mikor érdemes használni egymásba ágyazott lekérdezéseket, és mikor illesztést? Az attól függ. Ha nincs különösebb követelmény a lekérdezés teljesítményére, akkor használjuk azt, ami a probléma természetes nyelvi megfogalmazásához legközelebb áll, így később könnyebben érthető és karbantartható lesz a script. Ha fontos az optimális teljesítmény, akkor inkább használjunk illesztést. Miért? Illesztések esetén az optimalizáló meg tudja választani, hogy milyen sorrendben hajtsa végre az utasítást, így minimalizálni tudja a végrehajtáshoz szükséges költséget. Egymásba ágyazott lekérdezéssel beledrótozzuk a végrehajtás sorrendjét a lekérdezésbe, így nem sok teret hagyunk az optimalizálónak a gondolkodásra. Ennek ellenére majd mutatok eseteket, amikor lassabb lesz az illesztés. A végső ítéletet csak a lekérdezés költségének vizsgálatával lehet kimondani *(Query Analyzer, Show Execution Plan)*.

A duplikált adatok problémája

Van egy elég gyakori feladat, amelynek megoldása első nekifutásból nem kézenfekvő. Nevezetesen, hogy keressük meg egy táblában az ismétlődő sorokat. Leginkább ez is adatmigrációnál jön elő, amikor az SQL Serverbe bemásolt adathalmaz egy oszlopára szeretnénk ráadni egy UNIQE vagy PRIMARY KEY-t, de van benne egy-két ismétlődő sor, ami megakadályozza ezt. Ilyenkor meg kell keresni azokat a sorokat, amelyekben azonosak a kérdéses oszlopok értékei. Ezzel ugye az a baj, hogy a WHERE egyszerre mindig csak egy sorral foglalkozik. Hogyan lehetne rávenni, hogy ugyanannak a táblának a sorait hasonlítgassa össze egymással? A megoldás egy self-join. „Sajnos" a Northwind egy konzisztens adatbázis, így abban nem fogunk találni olyan duplikált sorokat, amelyeket ki lehetne szűrni, mint logikailag hibásakat. Ezért a példa kedvéért nézzük a következő táblát *(Users)*:

nID	FirstName	LastName	BirthDay
1	István	Király	1954-05-22
2	László	Lőrincz	1961-11-14
3	Jenő	Rejtő	1910-01-14
4	Jenő	Rejtő	1910-01-14

Tegyük fel, hogy ezt a táblát úgy kaptuk, hogy kiindulásként volt egy HR-től kapott konszolidálatlan felhasználói adatbázis Excel-ben, amit például a Data Transformation Services segítségével beimportáltunk egy Users nevűtáblába. Azért, hogy tudjunk hivatkozni a sorokra, adtunk mindegyiknek egyedi azonosítót egy szám formájában. Szeretnénk megtalálni azokat az embereket, akik többször is szerepelnek a táblában. Tegyük fel, hogy két sort *(és embert)* akkor mondunk azonosnak, ha a vezeték- és keresztnevük azonos, valamint egy napon születtek. Keressük hát meg, ki a kakukktojás!

```
SELECT
   u1.nID, u1.FirstName,
   u1.LastName, u1.BirthDay
FROM
   Users u1
INNER JOIN
   Users u2
ON
   u1.FirstName = u2.FirstName AND
   u1.LastName = u2.LastName AND
   u1.BirthDay = u2.BirthDay
WHERE
-- Máskülönben minden sornál megtalálná
-- önmagát, mint a sajt párját!
   u1.nID <> u2.nID
```

Az eredményen remélem senki nem lepődik meg:

nID	FirstName	LastName	BirthDay
3	Jenő	Rejtő	1910-01-14
4	Jenő	Rejtő	1910-01-14

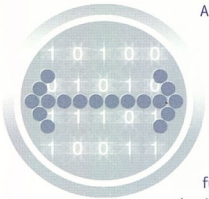
Mi itt a szokatlan? Az, hogy illeszteni lehet több mezőre is, sőt nem csak numerikus mezőkre! Általában az él a legtöbb fejlesztőben, hogy biztos, ami biztos, minden táblához generálok egy mesterséges kulcsot *(Surrogate Primary Key)*, és azon keresztül illesztem a táblákat. Ez helyes, ha gyors adatbázist akarunk építeni, de azért nem csak egy egész szám lehet gyors. Mi van például a 2-3-4 betűs azonosítókkal? Azokat talán lassabb összehasonlítani, mint az egészeket? Nem sokkal, viszont sokkal könnyebben átlátható adatbázist kapunk.
Például egy felhasználói adatbázisban az osztály, ahol az alkalmazott dolgozik valószínűleg egy külön táblában lesz eltárolva. Numerikus kulcsokat használva az IT 1 lesz, a Finance 2 satöbbi. Ezzel szemben egy 3 karakteres azonosítóval az IT lehet 'IT', a Finance 'Fin', a Human Resources 'HR' és így tovább. Így a felhasználói táblát böngészve nem 'Kiss József', '2'-t fogunk látni, hanem 'Kiss József', 'Fin'-t, ami azért könnyebben dekódolható, nem?

Zárszó

A következő számban még további lekérdezéseket írunk a GROUP BY, a HAVING és társaik segítségével, hogy csodaszép statisztikákat tudjunk generálni. Mindenkit visszavárok!

Soczó Zsolt MCSE, MCSD
Protomix Rt.

BUSINESS INTERNET
hálózati terheléselosztás



A Network Load Balancing *(továbbiakban NLBS)* a Windows 2000 Advanced Server és a Windows 2000 Datacenter Server operációs rendszerek beépített szolgáltatása. Az NLBS nagyobb rendelkezésreállást és stabilitást biztosít az Internet-kiszolgáló alkalmazások *(például web- vagy FTP-kiszolgálók, és más, nagy fontosságú kiszolgálók)* számára. A Windows 2000-t futtató számítógép önállóan a nagy rendelkezésreállási elvárásoknak csak korlátozottan felel meg, de az NLBS segítségével fürtbe rendezett Windows 2000 Advanced Server-ek kielégítik a nagy fontosságú és –rendelkezésreállású rendszerekkel szemben támasztott biztonsági és teljesítmény-elvárásokat.

A szükséges kiszolgáló-alkalmazások *(web-, FTP-, telnet-, vagy levelező-kiszolgálók)* a fürt minden tagkiszolgálóján futhatnak. A szolgáltatások egy része *(például a webkiszolgáló)* az összes tagkiszolgálón egyidejűleg működik, az NLBS pedig a beérkező hálózati terhelést elosztja a tagok között, míg más szolgáltatásokat *(például a levelezést)* egyidejűleg csak a fürt egy tagkiszolgálója látja el. Minden esetben az NLBS feladata, hogy hiba esetén a kiesett kiszolgáló forgalmát egy másik, működő taghoz irányítsa át.

Az NLBS működéséhez szükséges konfiguráció

A Windows 2000 NLBS szolgáltatása hálózati meghajtóként működik, méghozzá olyan alacsony szinten, hogy a TCP/IP hálózati összetevők az NLBS jelenlétét nem is észlelik.

A legjobb teljesítmény elérése érdekében az NLBS általában számítógépenként két hálózati csatolót használ: az egyik csatoló feladata az ügyfelek kiszolgálása *(mint rendesen)*, a másik pedig az NLBS fürt „adminisztratív" hálózati forgalmát kezeli. Természetesen nincs akadálya annak sem, hogy minden forgalom gépenként egyetlen hálózati csatolón keresztül bonyolódjon.

Az NLBS fürt-alkalmazások adatbázis-elérése

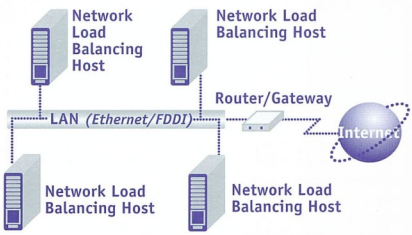
Sok kiszolgáló-alkalmazás működése közben adatbázisban dolgozik. Ha egy fürtben több ilyen alkalmazás működik egyidejűleg, akkor különös gondot kell fordítani az adatbázis-műveletek szinkronizálására. Egyszerűbb esetben minden tagkiszolgáló dolgozhat a saját, offline adatbázis-másolatán, majd szükség esetén a változásokat frissítik *(migrálják)* egymás között, vagy a központi, „valódi" adatbázisban. Máskor egyidejűleg, hálózaton keresztül érik el az adatbázis-kiszolgálót, de előfordulhat a fenti két eset kombinációja is. Például: a weblapok forrásállományait nyugodtan felmásolhatjuk az összes tagkiszolgálóra, így rövidebb válaszidőt és nagyobb hibatűrést kapunk. A weblapokon keresztül érkezett adatbázis-elérést igénylő műveleteket ugyanakkor kezelheti különálló, megbízható adatbázis-kiszolgáló, amely egyszerre az összes tagkiszolgálót ellátja.

A fontos vállalati alkalmazások a hibatűrő szolgáltatás biztosítása érdekében nagy rendelkezésreállású adatbázis-konfigurációt igényelnek. Ilyenkor maga az adatbázis kiszolgáló is fürtbe rendezett számítógépeken fut, így biztosítja a nagy rendelkezésreállást és a méretezhetőséget. Ilyen adatbázis-kiszolgáló lehet például a Cluster szolgáltatás segítségével megvalósított kétpontos fürtre telepített Microsoft SQL Server. A Cluster szolgáltatás biztosítja, hogy ha a két csomópont közül az egyik kiesik *(a csomópontot megvalósító számítógép meghibásodik)*, a másik csomópont átveszi a hibás egység feladatait. Teheti mindezt azért, mert a fürt két csomópontja közös merevlemez-alrendszert használ. Az SQL Server ügyfelei pedig az átállásból csaknem semmit sem vesznek észre.

Fontos, hogy a két fürtözési módszert megkülönböztessük egymástól. Az első, az NLBS fürt feladata elsősorban a beérkező TCP/IP hálózati forgalom elosztása a rendelkezésre álló tagkiszolgálók között. Ezek a független tagkiszolgálók egy NLBS fürtöt képeznek. A második módszer a Cluster szolgáltatás segítségével megvalósított fürt, amelyet két vagy több, fizikailag szorosan összekapcsolt, közös lemez-alrendszert használó számítógép valósít meg. A Cluster szolgáltatás segítségével leggyakrabban nagy rendelkezésreállású adatbázis-kiszolgálókat építenek *(amit azután természetesen akár NLBS fürtök háttérkiszolgálójaként is lehet használni)*. A két fürttípus együttes használatával teljeskörű, hibatűrő és nagy rendelkezésreállású fürtkörnyezetet teremthetünk.

A hálózati terheléselosztás működése

A hálózati terheléselosztás segítségével megbízható és méretezhető hálózati kiszolgálót hozhatunk létre két vagy több tagkiszolgáló számítógép fürtbe rendezésével. Az internetes ügyfelek a fürtöt a fürt saját IP címén *(címein)* keresztül érik el. Az ügyfelek számára a fürthöz való csatlakozás ugyanolyan, mintha közvetlenül a fürt egy tagkiszolgálójához kapcsolódtak volna; a kiszolgálón futó alkalmazások sem észlelik, hogy egy fürtbe rendezett számítógépen futnak. *(A Cluster fürt előnyeit csak speciális, erre tervezett alkalmazások képesek kihasználni.)* Az NLBS fürt azonban mégsem ugyanolyan, mintha több különálló számítógépen futtatnánk az alkalmazásokat, hiszen a tagkiszolgáló hibája esetén a szolgáltatás nem szakad meg, csak a bejövő hálózati kapcsolatok átkerülnek a fürt még működő tagkiszolgálóihoz. A fürt ezen kívül – a terheléselosztással működő portokon – nagyobb hálózati terhelést képes elviselni, miközben gyorsabb válaszidőt biztosít az ügyfelek számára.



Ha egy NLBS fürt tagkiszolgálója meghibásodik, a Network Load Balancing a beérkező hálózati forgalmat automatikusan a még működő tagkiszolgálók között osztja el. A leálló tagkiszolgálóval létesített hálózati kapcsolatok elvesznek, de a szolgáltatás a fürtben továbbra is elérhető marad. A legtöbb esetben *(például webkiszolgálók esetén)* az ügyfélszoftver au-

nehézségi fok:

tomatikusan újra megpróbálkozik a megszakadt kapcsolatok felépítésével *(tudtán kívül, de ezúttal már egy másik, még működő tagkiszolgálóval)*, az ügyfél pedig ebből mindössze csak a néhány másodperces várakozást veszi észre.

Az NLBS a fürt egy vagy több saját IP címére érkező hálózati forgalmat elosztja a rendelkezésre álló tagkiszolgálók között. A tagkiszolgálók egyidejűleg válaszolnak a különböző ügyfelek kéréseire, akár egy ügyfél több kérésére is. Például: az ügyfél által megtekintett weboldal forráskódja a fürt egyik, míg az oldalon található képek a fürt egy másik tagkiszolgálójáról érkezik. Ez meggyorsítja a kiszolgáló-műveleteket és csökkenti a válaszidőt.

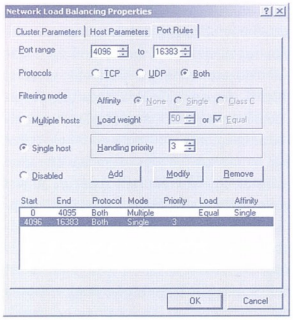
Az NLBS fürt minden – közös alhálózaton található – tagkiszolgálója észleli a teljes bejövő hálózati forgalmat. A tagkiszolgálókon található NLBS eszközmeghajtó szűrőként működik a fürtre csatlakozó hálózati kártya és a számítógép TCP/IP komponensei között: a bejövő forgalom egy részét továbbengedi, más részét elnyeli *(a TCP/IP nem is tud róla, hogy fürtben működik – hiszen hozzá már csak az a hálózati forgalom jut el, amit az NLBS eszközmeghajtó jónak lát)*.

A Network Load Balancing teljesen elosztott algoritmus segítségével az ügyfél IP címe, a port és egyéb információk alapján rendeli a beérkező kéréseket a tagkiszolgálókhoz. A beérkező csomagokat minden tagkiszolgáló megkapja és kielemzi, majd az algoritmus segítségével eldönti, melyik tagkiszolgáló dolgozza fel a kérést. Ez a hozzárendelés egészen addig érvényben marad, míg a fürtben található tagkiszolgálók száma meg nem változik. Az NLBS szűrő algoritmus sokkal hatékonyabb és nagyobb sávszélesség kiszolgálására képes, mint a központosított terheléselosztó megoldások, amelyekben a központi egység a hálózati csomagokat módosítja, majd továbbküldi a megfelelő tagkiszolgálóhoz. Az NLBS a tagkiszolgálókon fut, így működése nem korlátozódik adott processzortípusokra vagy hálózati megoldásokra.

A hálózati forgalom elosztása

Az NLBS a következő módon osztja el a beérkező Transmission Control Protocol *(TCP)* és User Datagram Protocol *(UDB)* csomagokat a tagkiszolgálók között: a fürt saját IP címére érkező csomagokat a fürt minden tagkiszolgálója megkapja, majd az NLBS szűrők a csomagokban használt TCP és UDP portok alapján megszűrik ezt a forgalmat, mielőtt azok még a TCP/IP meghajtót elérnék.

Az NLBS csak a TCP és UDP csomagokat dolgozza fel, nem szűri a beérkező Internet Control Message Protocol *(ICMP)*, Internet Group Membership Protocol *(IGMP)*, az IP- és hálózati címek *(MAC address)* összerendelésére használt Address Resolution Protocol *(ARP)* és más IP protokollokat sem. Minden ilyen forgalom akadály nélkül továbbítódik a tagkiszolgálók TCP/IP meghajtói felé. A TCP/IP felépítéséből következik, hogy nem okoz gondot a fürt IP címéről, több tagkiszolgálótól egyidejűleg érkező többszörös válasz, bár ez a pont-pont alapú TCP/IP alkalmazások *(például a ping)* esetén probléma lehet. Ilyenkor a tagkiszolgálók saját, közvetlen IP-címét kell használni.



A hálózati terheléselosztás szabályainak beállítása

Konvergencia

Az NLBS fürt tagkiszolgálói szabályos időközönként broadcast vagy multicast üzenetekkel hangolják össze a működésüket, és felügyelik a fürt működését. Amikor a fürt állapota megváltozik *(ez történhet egy tagkiszolgáló leállásával, kilépésével vagy éppen megjelenésével)*, a fürtben elindul egy folyamat, melyben a tagkiszolgálók feltérképezik a fürt új állapotát, és alap értelmezett tagot választanak *(ez a legkisebb prioritású tagkiszolgáló lesz)*. Ezt a folyamatot nevezzük konvergenciának. Miután ez megtörtént, a Windows 2000 eseménynaplójában megjelennek a konvergencia befejezésére utaló bejegyzések.

A konvergencia alatt a kiesett tag kivételével minden kiszolgáló folytatja a működést, a változás a még működő tagokkal létesített hálózati kapcsolatokat nem érinti. A konvergencia befejeztével a kiesett tagkiszolgáló felé létesített hálózati kapcsolatokat a még működő tagkiszolgálók átveszik. A hálózati forgalom úgy oszlik el a tagkiszolgálók között, hogy a hálózati terhelés egyenletes maradjon. Ha a fürtben új tagkiszolgáló jön létre, a konvergencia során belép a fürtben működő tagkiszolgálók sorába. A fürt bővítése nem zavarja a meglévő hálózati kapcsolatokat, sem az ügyfél-, sem a kiszolgáló-alkalmazásokat. A több TCP kapcsolatot egyidejűleg használó alkalmazásoknál azonban előfordulhat, hogy a konvergencia során – az affinitás ellenére – a két TCP kapcsolat különálló tagkiszolgálóra kerül. *(Hiszen az affinitás csak a fürt felépítésének megváltozásáig érvényes.)*

A Network Load Balancing fürt tagkiszolgálója addig számít „élőnek", míg az folyamatosan részt vesz a fürt tagkiszolgálói között folyó kommunikációban. Ha a tagkiszolgálók adott ideig nem kapnak üzenetet egy másik tagkiszolgálótól, azt hibásnak veszik, és megkezdődik a konvergencia a kieső tagkiszolgáló terhelésének elosztására. Az üzenetek közötti időtartam és a konvergencia indításáig kimaradó üzenetek száma beállítható, az alapértelmezés 1000 ms *(egy másodperc)*, és 5 kihagyott üzenet. Mivel ezek a paraméterek ritkán változnak, a Network Load Balancing dialógusablakából nem módosíthatók: értékük szükség esetén a regisztrációs adatbázisban változtatható meg.

BIZTONSÁG
NTLMv2

Hitelesítés

A Windows 2000 biztonságát bemutató sorozatunkban foglalkozzunk a hálózat biztonságával, a hálózaton átmenő jelszavak életével egy kicsit! Megismerjük az azonosítás alapjait, s példákon keresztül megnézzük, hogy milyen lehetőségünk van vegyes környezetben üzemelő hálózatok biztonságosabbá tételére. Végül készítünk egy házirendfájlt *(policy)*, amivel a hálózat összes gépén egyidejűleg ki- vagy bekapcsolhatjuk, illetve szabályozhatjuk az NTLMv2 – t, mely óriási segítséget nyújt a hálózat biztonságának megőrzésében azzal, hogy a jelszavak titkosítására sokkal jobb algoritmust használ, mint elődei. Gondoljuk csak végig, hogy naponta a hálózatukon hányan és hányszor jelentkeznek be, mindannyiszor elküldve jelszavukat! Az olvasótábor egyik fele nyilván tudja ezt a tényt, s biztosra veszem, hogy olyanok is vannak szép számmal, akik a veszélyeket is átlátják, mások azonban talán nem foglalkoztak részletesen a bejelentkezéssel – nekik új információval szolgálhatok. Hát nézzük meg közösen, hogy miről is van szó!

A hálózat biztonsága

Tulajdonképp mit is értünk a hálózat biztonságán? Egy biztonságos hálózatban egyaránt helyet kap a felhasználói adatok védelme, a kiszolgálók fizikai védelme, a szalagos mentések megfelelő tárolása, az azonosítási folyamat adatainak biztonságos csatornákon történő továbbítása és egyéb óvintézkedések. Különböző mélységű azonosítási szinteket állapíthatunk meg, ahol a nagyobb mélység a biztonságosabb hálózatot jelenti.

Egymélységű azonosítás

A „Tudok valamit" elve érvényesül. Tudok egy jelszót és a hozzátartozó felhasználónevet és az azonosításhoz ez elegendő. Könnyen és gyorsan beláthatjuk, hogy ez nem nyújt elegendő biztonságot minden hálózaton, mert Sherlock Holmes titkos adataihoz akár Dr. Watson is könnyedén hozzáférhet, ha Mr. Holmes felhasználónevét és jelszavát használja. Ha megfelelő szabályokat figyelembe véve használja jelszavát és felhasználónevét Mr. Holmes, akkor már megfelelő védelmet nyújthat ez is. Természetesen hálózati analizátort használva és a HASH–t elkapva a visszafejtés nem lehetetlen. *(Erre nyújt megfelelő védelmet a KERBEROS és az NTLMv2 együttes használata.)* A megfelelő jelszóhasználati szabályokat ilyenkor is figyelembe kell venni, ilyen lehet a jelszavak rendszeres cseréje és megfelelő tárolása *(az a jelszó nem jelszó, ami le van írva)*.

Kétmélységű azonosítás

Itt a „Tudok valamit és a birtokolok valamit" elve érvényesül. Tudok egy jelszót, a hozzátartozó felhasználónevet és birtokomba van valami. Ebben az esetben a különbség az előző módszerhez képest, csak a valami, :) birtokomban van egy tárgy, ami mondjuk egy SmartCard. A kétmélységű azonosítás az előző példa egyetlen hibáját küszöböli ki, az pedig a jelszó és a felhasználónév illetéktelen kezekbe kerülése. Semmit sem ér az érvényes jelszó, ha nincs a birtokunkban – például – egy SmartCard. 100 %-os biztonságot ez sem tud nyújtani, mert a SmartCard is hamisítható, de körülményesebb és a költségei is lényegesen magasabbak, mint egy jelszó megszerzésének. Jó, megfelelő biztonságot tud nyújtani a tipikus vállalati hálózatoknak. További nagy előnye, hogy a Microsoft Windows 2000 tartalmazza a SmartCard azonosítás használatát.

Hárommélységű azonosítás

Ilyenkor a „Tudok valamit, birtokolok valamit és tudom magam azonosítani" elve érvényesül. Tudok egy jelszót, a hozzátartozó felhasználónevet, birtokomba van valami, és kétséget kizáróan tudom magam azonosítani. A kétmélységű azonosítást fejleszti tovább ez a módszer, ami azáltal nyújt többet, hogy kétséget kizáróan tudom magam azonosítani *(például egy DNS molekulával, egy ujjlenyomattal, vagy írisz-vizsgálattal)*. Ilyen szintű biztonságot egy – két katonai bázison tapasztalhatnánk – de oda avatatlan szemek nem juthatnak be. Azt hiszem említenem sem kell, hogy ez sem nyújt megfelelő biztonságot, mert nincsenek olyan az ellenőrző eszközök, melyeket nem lehet becsapni. Engedjék el a fantáziájukat és gondolják végig, hogy hogyan lehet ezt a módszert hamisítani és feltörni!

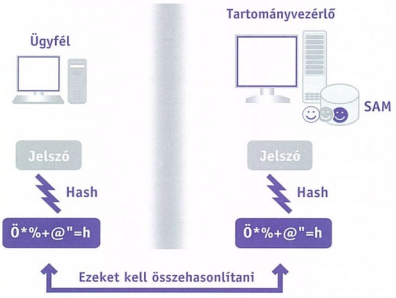
Az elmélet csak elmélet, de mit mutat a gyakorlat?

A gyakorlat azt mutatja, hogy a biztonság eddig csak néhány küldetéskritikus alkalmazás esetén tűnt fontosnak, a cégek azok fejlesztésére fektettek hangsúlyt. Ilyen alkalmazások a webkiszolgálók, az online áruházak, a banki alkalmazások. Kevesebbet hallottunk eddig a LAN–ok és WAN–ok megfelelő biztonságának bevezetéséről. A Microsoft operációs rendszerei ennek lehetőségét a kezdektől magában hordozzák. A megfelelő biztonság napjaink fontos kérdésévé válhat az intranetek mind komolyabb terjedésével. Az olyan hálózatok esetén, ahol egymélységű azonosítást használunk *(azaz a magyarországi hálózatok közel 100%-ánál)* nem elegendő a fájl- vagy megosztásszintű védelem, védenünk kell a felhasználó személyes azonosító adatait is. Saját tapasztalataim sajnos azt mutatják, hogy – akár nagyvállalati szinten – a felhasználók nem megfelelően képzettek, a jelszó használatának legelemibb szabályait sem tartják be – gyakran a monitorra ragasztják a jelszavas cetlit. Napjainkban, amikor a telefonnak is PIN kódja van, a jelszavak megfelelő használata elengedhetetlenül fontos *(lenne)*. A „jelszó olyan mint a fogkefe: mindennap használjuk és sűrűn kell cserélni" idézetet ne felejtsék el! Mit tehet a rendszergazda, ha munkáját a felette álló vezetés határozza meg, emiatt nem állíthatja be, hogy a jelszavak 21 nap múlva lejárjanak? Nem beszélhetünk biztonságos hálózatról, ha a felhasználónak a jelszava nem jár le és nem kell megváltoztatni az alap értelmezett jelszót! Persze a rendszerek gazdája – ez a leleményes ember – erre is kitalál valamit és az alap értelmezett jelszót felhasználófüggővé teszi, ezzel is csökkentve a biztonsági rések számát! Másik lehetőség, ha a hálózaton utazó csomagok útját teszi biztonságossá. Hogy lehet ezt megtenni?

nehézségi fok:

Az azonosítás alapja *(tördelő algoritmus)*

A jelszótitkosítás egyik elterjedt módja a tördelő algoritmus használata. Míg a „hagyományos" titkosítás kétirányú folyamat, ami azt jelenti, hogy a titkosított adatból az eredeti előállítható, addig a tördelés egyirányú folyamat *(amit egyszer titkosítottam, abból többé nem állítható elő az eredeti)*. Egy másik hasonlattal élve a tördelő algoritmus olyan, mint a húsdaráló: felül bemegy az ínycsiklandozó karaj, alul kijön a darált hús, amiből már soha az életben nem lesz többé karaj, maximum fasírt. Az egyetlen lényeges különbség a húsdaráló és tördelő algoritmus között az, hogy a HASH-ek jellemzőek az eredeti adatra, azaz helyettük felhasználhatók azonosításra, míg a darált hús nemigen lenne egyértelmű helyettesítője a karajnak... Nem tűnik túl praktikus eljárásnak, ugye? De gondoljuk végig, ha én nem tudom visszafejteni a HASH-ből az eredeti jelszót, akkor senki más sem, ami azt jelenti: az adat biztonságban utazik a hálózaton! A bejelentkezés tördelő algoritmust használó hálózat esetén nagyon egyszerű, ilyet használ a Windows NT is. A bejelentkezni kívánó gép elküldi a tartományvezérlő számára a felhasználó jelszavából képzett HASH-t. Ezek után a kiszolgáló a nála tárolt jelszó adataiból szintén előállítja a HASH-t és ha a két HASH megegyezik, akkor a jelszó helyes, a felhasználó azonosítva. A Microsoft kétféle megvalósítást kínál a tördelő algoritmus használatához. Az egyik a LanMan – ami elég gyenge megoldás, és általában csak kompatibilitási okokból használjuk *(ha a hálózatban Win9x-ek is vannak)*, – a másik pedig a Challange/Response – ami már komolyabb, nehezebben megfejthető, a húsdarálóhoz közelebb álló tördelést biztosít Windows NT és Windows 2000 munkaállomások számára.

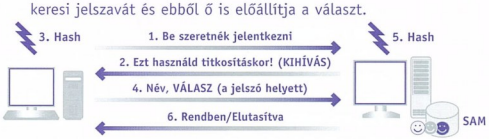


Az azonosítás folyamata

Challenge/Response (kihívás/válasz):

Az előzőek alapján tudhatjuk, hogy a Challenge/Response azonosítás tördelő algoritmust használ. Úgynevezett fűszerezett *(salted, sózott)* algoritmusról van szó, amely azt jelenti, hogy az algoritmus kimenete nemcsak a tördelendő bemenettől, hanem egy csipetnyi módosítókódtól *(ez a fűszer)* is függ. *(Automatikusan fűszerező húsdaráló.)* A fűszer lehetővé teszi, hogy a HASH mindig egy picit másmilyen legyen, ezzel is nehezítve a jelszótolvaj dolgát. Felmerülhet a kérdés, hogyan lehetséges, hogy mind a kiszolgáló mind az ügyfél a jelszóból ugyanazt a HASH-t állítja elő? Ez csak úgy lehetséges, ha mindketten azonos módosítókódot használnak a HASH előállításához – azt pedig nyilván meg kell beszélniük, azaz át kell küldeni a hálózaton. Jogos lehet a kérdés, ha a kód átmegy a hálózaton azt elkapva a csomagok nem nyithatók-e ki? Elvileg nem, mert ezek a kulcsok egyediek és egyszer használatosak. Nem lenne értelme egy egyszer használatos kulcsot elkapni, hacsak a kulcshoz tartozó HASH-t nem tudom elkapni. Az azonosítást tehát megelőzi egy egyeztetés, ami a Challenge /Response esetén az alábbiak szerint zajlik :

1. A felhasználó számítógépe jelzi a kiszolgálónak, hogy be szeretne jelentkezni. Kiss Pista begépeli a jelszavát, ami a tartományvezérlő SAM adatbázisban is megtalálható, így alapot ad az összehasonlításra.
2. A kiszolgáló visszaküld egy 8 bájtos KIHÍVÁST *(salt)*, ami a következő pontban történő titkosítás alapját képezi.
3. A Kiss Pista által begépelt jelszót a felhasználó számítógépe a salt felhasználásával titkosítja, és előáll a HASH, a válasz.
4. A felhasználó gépe visszaküldi a HASH-t és Kiss Pista felhasználónevét.
5. A kiszolgáló a visszaküldött „csomagból" kiolvassa a felhasználó nevét. Ez alapján a SAM adatbázisból előkeresi jelszavát és ebből ő is előállítja a választ.



Két lehetőség marad, a tartományvezérlő, és a felhasználó gépe által előállított HASH vagy megegyezik, vagy nem. Ennek megfelelően a bejelentkezés vagy elutasítva, vagy elfogadva. Ez képezi a Challenge/Response hitelesítés alapját, ami két gép között még elegendő biztonságot nyújt. Viszont ha a hálózatunkban levő munkaállomások számát növeljük, és abból kiragadunk két felhasználói gépet és egy tartományvezérlőt, akkor máris láthatjuk, hogy a csomagok a hálózaton indokolatlanul többször utaznak. Ha Kiss Pista el szeretné érni Nagy József gépének egy erőforrását, akkor eljárás az első 4 pontja azonos lesz a fentivel. Viszont az 5. pontban Nagy József gépe észreveszi, hogy az ő SAM adatbázisában nincs bent Kiss Pista felhasználó. Ekkor az egész csomagot, *(KIHÍVÁS, FELHASZNÁLÓNÉV, VÁLASZ)* átküldi a tartományvezérlőnek, és ő dönti el a hitelességet. Ekkor a hálózaton már kétszer átment a csomag és ez a szám csak növekedik, ha tartományok között történik a hitelesítés. Sajnos csak elmélet az, hogy a Challenge/Response módszer nem törhető fel. Itt nem említ*(het)*jük meg annak az Internetről letölthető alkalmazásnak a nevét, mellyel hálózatról lelopott LanMan és Challenge/Response HASH-ek alapján a jelszavak tökéletesen visszaállíthatók.

Ez nem titok: a LanMan és az NTLM sebezhető, támadható, hisz a tördelő algoritmus publikussá válásával annak gyengéi is napvilágra kerültek. Az alábbi ábra az LanMan HASH minden hacker által ismert előállítási algoritmusa:

- A jelszó nagybetűsítése *(első fatális hiba: a változatok száma felére esik)*.
- 14 bájtra feltöltés szóközökkel (*második fatális hiba: a jelszó után mindig szóköz lesz)*, hogy meglegyen a 112 bit hossz.
- Ez 2 darab 56 bites DES kulcs *(harmadik fatális hiba: a DES nem is HASH algoritmus!)*.
- Ezzel a két kulccsal titkosítják a 0x4B47532140232425 mágikus, ám konstans számot *(harmadik fatális hiba: nincs salt)*, ebből áll elő az eredmény, a 16 bájtos „HASH".

Ráadásul, ha a jelszó rövidebb, mint 8 karakter, akkor a második 8 bájt mindig 0xAAD3B435B51404EE: ránézésre látszik, hogy rövidebb, vagy hosszabb mint 8!

A Microsoft éppen ezért kifejlesztett egy új azonosítást, melynek neve NTLMv2 és lényegesen erősebb HASH algoritmust használ, mint elődei, s melynek a mai napig nincs „ellenszere", azaz még senkinek sem sikerült az NTLMv2 darált húsból karajt csinálnia :)

NTLMv2

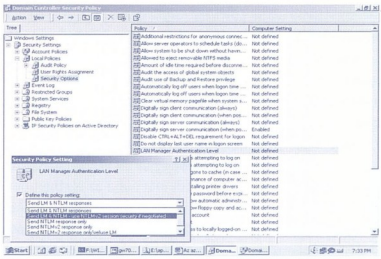
A Microsoft az NTLM és a LanMan könnyű sebezhetősége miatt alkotta meg az NTLMv2-t, amit a Windows NT 4.0 Service Pack 4 óta használhatunk *(így a Windows2000-ben is elérhető)*. A Windows 95 és a Windows 98-as munkaállomások az NTLMv2 támogatást akkor használhatják, ha a Directory Services Client telepítésre kerül *(a Windows2000 CD-ROM-ról Clients\Win9x\Dsclient.exe)*. Ezután a Windows9x-ben megjelennek azok a fájlok, amelyek lehetővé teszik az NTLMv2 használatát. *(Secur32.dll; Msnp32.dll; Vredir.vxd; Vnetsup.vxd)* A telepítés után a Windows9x kompatibilitási okok miatt még az eredeti LanMan azonosítást használja, de a következő registry értékek módosításával le lehet erről beszélni:

```
HKLM\System\CurrentControlSet\control\LSA
Value Name: LMCompatibilityLevel
Data Type:    REG_DWORD
Value: 3
Valid Range: 0-3
```

Az említett Registry érték Windows NT Server és Workstation esetében is megegyezik de a Windows9X-től eltér:

```
HKLM\System\CurrentControlSet\control\LSA
Value Name: LMCompatibilityLevel
Data Type:    REG_DWORD
Value: 3
Valid Range: 0-5
```

Windows2000-ben a Start menü\Programok\Ad-ministrative Tools menü alatt érhetjük el a beállításhoz szükséges eszközöket. *(Local Security Policy; Domain Controller Security Policy és Doman Security Policy)* GrouPolicy segítségével is szerkeszthető az autentikáció, ez nagyban megkönnyíti a munkaállomások konfigurálását.



NTLMv2 állítása Windows2000 – ben

Kerberos v5

Windows2000-es tartományokban elérhetővé válik az azonosítás egy új formája a Kerberos. A Unix-világban már régóta jól megszokott azonosítási forma most már RFC-t is kapott, ennek köszönhetően a Windows2000-ben ez is elérhető! Ugyan kompatibilitási problémák adódhatnak a Unix és Windows 2000 között mert a Windows 2000 a Kerberos V5-t használja *(és azért is, mert a Windows 2000 implementáció nem hagy üresen egy olyan mezőt a Kerberos jegyben* ***(AuthData)****, amit nem szokás kitölteni)*. A Kerberos azonosítás során nem az azonosításban résztvevő gépek küldik egymásnak a csomagokat, mint a kihívás/válasz azonosítás esetén. Itt az azonosítás folyamatában három gépről kell beszélnünk, ahol részt vesz egy jegykibocsátó, egy jegyhitelesítő és egy jegyigénylő gép. Míg a kihívás/válasz azonosítás esetén a felhasználónév és a jelszó *(HASH)* utazik a hálózaton, addig a Kerberos esetén csak jegyek utaznak – Triple DES kulcsokkal titkosítva. A Kerberos talán egyik legnagyobb előnye az NTLM akármelyik verziójával szemben, hogy a kezdeti azonosítás után a felhasználó adataiból képzett HASH-ek NEM közlekednek állandóan a hálózaton, így teljesen értelmetlen a napközbeni hálózati forgalom elkapásával és megtörésével kísérletezni, hisz abban még darált hús sincs.

Kerberos jegyek? Miért beszélek jegyekről? Ez talán még korai! Előtte még nagyon fontos fogalmat kell megvizsgálnunk, ami a Windows NT 4.0 – s és Windows2000 – is tartományban is fontos szerepet játszik, ez az...

Acces Token

Windows NT 4.0-s tartományba bejelentkezve a felhasználói felület megjelenése előtt kiértékelődik és letárolódik a felhasználó úgynevezett Access Tokenje. Az Access Token tartalmazza a felhasználó jogosultságainak kiértékeléséhez szükséges dolgokat, többek között minden csoportot, melynek a felhasználó tagja. Az Access Token egy igazolvány, aminek felmutatásával bizonyos szolgáltatásokat meghatározott módon vehetek igénybe. Nagyon fontos megjegyezni, hogy az Access Token nem frissül azonnal, ha a felhasználó tulajdonságai, például csoporttagsága megváltozik – csak ismételt bejelentkezés esetén értékelődik ki újra. Minden erőforrás rendelkezik egy ACL-el *(Ac-*

cess Controll List) ami tartalmazza, hogy mely NT-beli felhasználók vagy csoportok milyen jogosultsággal érhetik el. Egyszerűbb a dolgunk, ha azt próbáljuk elképzelni, hogy az Access Token tartalmazza a kulcsokat, az ACL pedig zárakat. A kulcsok a zárakat kinyithatják *(teljesen, vagy csak fokozatosan)*, de lehet olyan kulcsom is, ami nem nyithatja ki semmilyen körülmények közt a zárat, ilyen a NO ACCESS jogosultság, Windows 2000-ben a DENY. Windows NT 4.0 – ban az Access Tokennek van egy nagy hátránya: az idők végezetéig érvényes. Ha egy felhasználói gépen Dr. Watson bejelentkezik, Access Tokenje tartalmazza a csoporttagsági információkat. Ha Sherlock időközben módosítja Dr. Watson csoporttagságát, az nem lesz hatással a munkaállomáson kiértékelődött Access Tokenre. A Windows 2000 Kerberosa természetesen erre is nyújt megoldást. A kiadott Kerberos jegyek *(melyekben az Access Token megtalálható)* alapértelmezésben 8 óra alatt lejárnak, emiatt óránként automatikusan és biztosan kiértékelődik a csoporttagság és mi egyéb. A kiadott jegyek nyomon követésére alkalmas eszköz a Kerberos Tray, ami a Windows 2000 ResourceKit-ben található meg.

Most, hogy megismertük az Access Token lényegét, és megértettük, hogy ez nem azonos a jegyekkel, nézzük meg a Kerberos V5 azonosítás lépéseit.

A Kerberos V5 azonosításához szükséges szolgáltatások, az azonosítás folyamata:

- **KDC *(Key Distribution Center)***: a Triple DES kulcsok generálását végzi, amely a jegyek titkosításához szükséges
- **AS *(Authentication Service)***: a fő-fő azonosító szolgáltatás ez, ahol a Kerberos megbizonyosodik arról, hogy kik vagyunk. Ekkor kapjuk meg a további jegyek váltására feljogosító szuperjegyet, a TGT – t *(Ticket Granting Ticket)*, és ez az egyetlen eset, amikor a felhasználó jelszavából képzett – valójában a jelszóval, mint kulccsal titkosított – adatok jelennek meg a hálózaton.
- **TGS *(Ticket Granting Service)***: A szogáltatásjegyeket ő adja ki. A kiadott jegyek a felhasználó azonosítására szolgálnak. A jogosultság kiértékeléséhez továbbra is Access Token szükséges!

1. Kiss Pista be szeretne jelentkezni a tartományba. Először a saját gépéhez szükséges jegyet igényelnie. A felhasználó bejelentkezési igénye eljut az AS-hez, mely a felhasználó jelszavával van titkosítva. Ez visszafejthető titkosítás, tehát nem HASH. Az AS elkérheti a címtártól a felhasználó jelszavát, s ezzel kibontja a csomagot, így egyértelműen megbizonyosodhat arról, hogy Kiss Pista valóban az akinek mondja magát *(vagy csak tudja valaki a jelszavát)*.

2. Az AS jelzi a KDC-nek, hogy újabb kulcsra lesz szükség. A KDC az AS-nek visszaküld egy speciális kulcsot. Az AS egy olyan jegyet küld vissza a felhasználónak, ami tartalmazza a KDC speciális kulcsát, és a kulcs lejárati idejét, ami a KERBEROS belső kulcsával van titkosítva! Ezt TGT-nek nevezzük, amit az AS Kiss Pista jelszavával titkosít így a TGT biztonságosan utazik a hálózaton.

3. A felhasználó a TGT-t tárolja, és a benne lévő kulccsal kér jegyet a TGS-től, amíg az le nem jár! Lejárata után ismét TGT-t kell igényelnie.

A Kerberos azonosítás lépései roppant érdekes és logikus lépések láncolatából állnak, amelynek tökéletes megértéséhez egy külön cikk terjedelme is kevés lenne. További érdekesség, hogy a Kerberos mindenképpen Windows 2000 Active Directoryt igényel, s ahol ez nincs, ott Kerberos sincs *(NT4 tartomány nem elég!)*, emiatt egy nem várt fordulattal ismét az NTLMv2-vel folytatom a történetet.

Az NTLMv2 bevezetésének lépései

Az NTLMv2 segítséget nyújt abban, hogy az azonosítási folyamat biztonságosa*(bba)*n történjen meg, hogy harmadik fél jelszólopási esélyét minimálisra csökkentsük. Gyakorlatilag többletköltségtől mentesen *(leszámítva a munkaerő árát)* bevezethető, melynek elvi lépései a következők:

1. Meg kell határozni a tartomány típusát, amiből is kiderül, hogy hány tartományunk van. Egy tartomány esetén nem kell tartanunk a TRUST-ok megszakadásától, míg Multi Master típusú tartomány esetén erre figyelnünk kell!
2. Meg kell vizsgálnunk a felhasználói géppark összetételét. Windows9x-es felhasználói gépeken számolni kell a DSCLient telepítésével, míg Windows NT 4.0 esetén legalább négyes javítócsomagnak kell lennie a gépeken. Ha ezek a feltételek teljesülnek, jöhet az NTLMv2 beállítása *(WindowsMe-n az NTLMv2-t a cikk írásáig nem sikerült életre kelteni)*.
3. Először a tartományvezérlőket kell átállítani olyan módon, hogy az még a LanMan-t és az NTLM – t is egyaránt elfogadja.
4. Most következnek a munkaállomások! Először az alapfeltételnek kell teljesülnie *(DSClient; SP4)* utána lehet az NTLMv2-t beállítani. Konfigurálási hibától óvhatjuk meg magunkat, ha az NTLMv2 – t nem kézzel, hanem házirend *(policy)* segítségével konfiguráljuk.
5. Ha az összes felhasználói számítógép is átállításra került, a tartományvezérlőket is átállíthatjuk, hogy innen csak az NTLMv2-t fogadják el és utasítsák el a LanMan és NTLM azonosítási kísérleteket.

Hogy készíthetek egy saját Policy-t amivel szabályozhatom a felhasználói gépek NTLMv2 azonosítását?

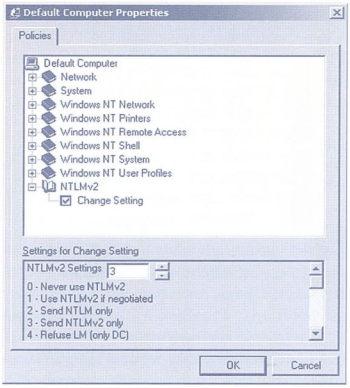
Minden Microsoft operációs rendszer a tartományba való bejelentkezéskor házirendfájlt keres. Ennek neve és helye operációs rendszertől függ, Windows9x-ek Config.pol fájlt, Windows NT alapú rendszerek pedig NTConfig.pol fájlt keresnek. Egy-egy házirendfájl tartalmának elemeit az úgynevezett policy template fájlok *(ADM fájlok)* határozzák meg. Az ADM fájlok azt a folyamatot vezérlik, ahogyan a munkaállomás módosításokat végez saját regisztrációs adatbázisán a bejelentkezési folyamat során *(programozáshoz hasonló kódokat tartalmaz)*. Nagy előnyük a Policy fájloknak a Logon Scriptel szemben, hogy a rendszer nevében futnak, így gyakorlatilag bármilyen rendszerszintű módosítás elvégezhető velük. *(A Logon Scrpit a bejelentkező felhasználó jogosultságait használja. Persze ez megkerülhető, de a lehetséges megoldások nem nyújtanak olyan biztonságot mint a Policy fájlok.)*

Házi feladat

Készítsünk NTLMv2 Policy-t Windows NT 4.0-ra *(és nem Windows 9x-re! Ahhoz másik Policy Editor kell, más ADM-et kell betölteni és más lesz a fájl neve. Lásd később.)* Ennek lépései a következők:

1. Lépjünk be egy Windows NT 4.0 kiszolgálóra, vagy munkaállomásra. Indítsuk el a Poledit-et *(fent van a tartományvezérlőn)*, vagy készítsünk egy másolatot a Windows NT 4.0 Resource Kit-ben találhatóról. Az elindítás után egy üres ablakot látunk magunk előtt.
2. Töltsük le a példa ADM-t a Netacademia kiszolgálójáról. *(A http://technet.netacademia.net/feladatok címről indulva könnyen megtalálhatók a Windows 9x-hez és Windows NT-hez készített két különböző ADM fájl)*. Ha ezzel megvagyunk, akkor az üres Poleditben az Options menü alatt válasszuk ki a Policy Template...-t. Ekkor feljön egy ablak, ahol az ADD gomb megnyomásával kiválaszthatjuk, hogy melyik ADM-t szeretnénk felhasználni a Policy készítéséhez. Ha ezzel elkészültünk, térjünk vissza az előző szürke ablakhoz.
3. Már meglevő Policy módosításához a File menüből az Open Policy-t, új Policy készítéséhez a New Policy-t válasszuk. Az adott gép hirtelen konfigurálásához az Open Regitry-t kell választanunk.
4. A kiválasztás után egy Default Computer és egy Default User-t láthatunk. Kettőt kattintva valamelyikre, annak tulajdonságait állíthatjuk. Az NTLMv2 konfigurálásához a registry-t kell módosítanunk a HKLM alatt, ez a Default Computer-ben végezhető el. Kettőt kattintva a Default Computer-re, megjelenik egy új ablak, amiből keressük ki az NTLMv2-t. Lenyitva a Change Setting-et aktívvá kell tenni *(egy pipa jelzi)* és ki kell választani a megfelelő szintet. A módosítás elvégzése után a Policy-t mentsük el!
5. Az elkészült Policy-t mentsük vissza a NETLOGON könyvtárba, majd egy felhasználói munkaállomásról lépjünk be a tartományba. A módosításoknak életbe kell lépniük. Ezt legegyszerűbben – micsoda véletlen – a Policy Editorral ellenőrizhetjük le. Az 1-es és a 2-es pontban foglaltakat ismét el kell végeznünk, majd a File menü Open Registry-t kell kiválasztanunk. Ezek után az NTLMv2-t kell lenyitnunk és ellenőriznünk, hogy a módosítások életbe léptek-e. Amennyiben nem, el kell kezdenünk a hibakeresést.

Windows 9x-re alkalmas Policy fájlt is készíthetünk a fenti lépéseket követve, annyi különbséggel, hogy a lépéseket Windows9x-es gépen a hozzá tartozó Poledit-tel kell elvégeznünk és a Policy fájl neve Config.pol kell, hogy legyen.



NTLMv2 konfigurálása Policy-ból

És végül

El szeretném mondani, hogy a hálózatok az alapvető szabályokat betartva, biztonságosak. A Windows NT 4.0 és a Windows 2000-es hálózatok biztonságához kétség sem férhet, ha az jól karban van tartva, megfelelő személyzet üzemelteti! Egy hálózat biztonságához sokat ad a jó tervezés, a hibátlan üzemeltetés és a megfelelő eszközök. Ezek a Microsoft operációs rendszereiben megtalálhatók, használhatók és elérhetők!

Harmath Zoltán, MCSE MCP+i
zolee@geniusgroup.hu

Utószó

Ez a cikk nem jöhetett volna létre, ha Zoltán nem szán néhány emberévet életéből az NTLMv2 megismerésére és beindítására, ugyanis míg az elvi lehetőség régóta adott volt, az NTLMv2 nem működött Windows 9x-en a gyári leírásban és egyéb helyeken fellelhető beállításokkal. Ebből két következtetés vonható le:

1. *Vagy senki nem használta, mi több, ki sem próbálta e lehetőséget előtte e földön.*
2. *Vagy senki az égvilágon nem ellenőrizte le, és nem vette észre, hogy AZ és ÚGY, ahogyan meg van írva, nem működik.*

Bármelyik állítás az igaz, „kollektíven" szégyelljük magunkat, és használjuk az általa összebarkácsolt ADM-eket.

(Lapzárta után érkezett a hír, hogy végre Redmond is kapcsolt, és új Knowledge Base cikket eresztett meg, mely már helyesen tartalmazza a beállítás menetét.)

fm

Microsoft

DNS checklist

Kicsit rendhagyó módon indul e havi dupla KV-nk. E hónapban egy olyan témával kezdjük, mely havonta négyszer-ötször fel szokott bukkanni a NetAcademia levelezési listán. Eleinte lelkesen válaszolgattunk, a múlt héten azonban már csak ennyit írtam lustán a kérdezőnek: DNS.

Az alább felsorolt kérdések mindegyike így kezdődik: Windows 2000 Active Directory telepítése után...

K1: ... a Windows 2000 Professional számítógépek egyáltalán nem, vagy csiga lassan jelentkeznek be a tartományba.
K2: ... Az Active Directory Users and Computers néha azt üzeni, hogy nincs meg a domain, miközben nyílván megvan, mert éppen azon dolgozom.
K3: ... a tartományvezérlő nem látja az Internetet. Ja, a többi gép sem.
K4: ... szeretnék még egy DC-t telepíteni, de a telepítési varázsló *(DCPORMO.EXE)* azt írja, hogy a tartomány nem található. Ez nem igaz, itt van fél méterre tőlem.
K5: ... szeretnék még egy DC-t telepíteni, de állandóan Acces Denied üzenetet kapok, pedig már szinte az Atyaúristen csoportba is behelyeztem magam.
K6: ... a két tartományvezérlő több napja nem látja egymást, és nem történik meg a replikáció. Az egyik DC-n felvett felhasználó egyszerűen nem jelenik meg a másikon.

V: DNS

Ennyit írtam, ez azóta is a lista legnagyobb információsűrűségű válasza, s ezzel elintézettnek is vettem az ügyet, de L. Zs. barátom, aki éppen talán K2-vel vívott több napos küzdelmet, további kérdésekkel bombázott:
Mondjam meg, mikor indul a legközelebbi DNS tanfolyam. *(Nincs ilyen tanfolyam)*
Vagy mutassak neki egy jó magyar nyelvű könyvet a DNS-ről: *(Nincs)*
Akkor menjek a fenébe :-)
A teljes szomorú igazság az, hogy ez utóbbi levélváltás már nem a lista színe előtt zajlott, s a probléma megoldása sem került fel oda. Az alábbiakban felsorolok néhány KV-t, melyek remélem segítenek a mindenütt felbukkanó DNS problémák gyors megoldásában, bár nem pótolják a DNS alapos megtanulását Ez utóbbiban segít az ezen lapszámban elindított DNS cikksorozat.

K: Megy-e az Active Directory DNS kiszolgáló nélkül?

V: Alig. Menni megy, de telesírja bánatában az Eseménynaplót.

K: Miért?

V: Mert a Windows 2000 tartományok tökéletesen egybeesnek a DNS tartományokkal, és az elavult WINS helyett mind a tartományvezérlők, mind pedig a 2000-es munkaállomások DNS lekérdezéssel keresik meg a hálózat erőforrásait. Ha nincs DNS, akkor bizonyos szolgáltatásoknak lőttek. Nincs például Kerberos bejelentkezés, hogy csak egyet említsek a sok közül.

K: Muszáj-e Microsoft DNS Servert használni?

V: Egyáltalán nem. Minden olyan DNS Server megfelel, amely lehetővé teszi SRV rekordok felvételét. Ez az egyetlen követelmény. A közhiedelemmel ellentétben nem szükségszerű dinamikus DNS-t használni. Ha nem DDNS kiszolgálónk van, az sem tragédia. Az AD telepítő varázsló ugyanis az összes szükséges DNS rekordkupacot beleírja egy NETLOGON.DNS nevű, Bind kompatibilis zónafájlba, amit akár kézzel is betermelhetünk – bár a DDNS sokkal kényelmesebb.

K: Ha feltelepítem a DC-re a DNS Server-t, akkor már mehet is a DCPROMO?

V: Nem. Attól még, hogy felkerült, nem változik meg automatikusan a gép TCP/IP beállítása, így hiába van ott, ugyanazon a gépen a DNS Server, a masina továbbra is a korábban beállított DNS Server-t fogja használni. Ha ez az Internetszolgáltatód, akkor a DCPROMO dinamikus DNS bejegyzési kérelmei OTT KINN fognak megjelenni *(és eldobódni)*, a telepített DNS szervered meg üres marad!!! Erről a szomorú tényről Network Monitorral lehet meggyőződni.

K: Miért van az, hogy amelyik munkaállomás tud Internetezni, az nem látja az Active Directory-t, és vica versa?

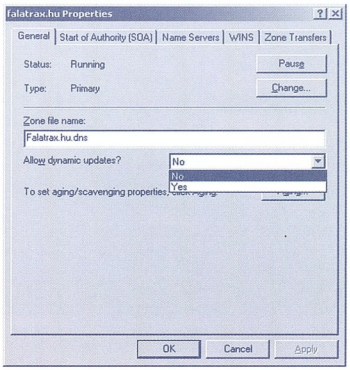
V1: Látod a netet, de nem látod az AD-t? Mert ha a gépen nem a belső DNS Server van beállítva a TCP/IP paraméterek között, akkor nyílván nem fogják megtalálni a szükséges erőforrásrekordokat, hisz az Internetszolgáltatónk DNS Server-e nem fog tudni a mi belső AD telepítésünkről.
V2: Látod az AD-t, de nem látod a netet? Olvass tovább, a belső DNS Server-rel lesz még tennivalód.

K: Bele lehet-e ugrani a DCPROMO-ba telepített DNS Server nélkül?

V: Bele. Ha majd azt kérdi, csináljon-e neked egyet, mondj YES-t. *(Ez ugyan root DNS-t telepít, de sebaj. Olvass tovább)*

K: Ha már rég fent van a DNS, és a megfelelő zóna is ott van, miért nem jegyzi be magát a nyavalyás?

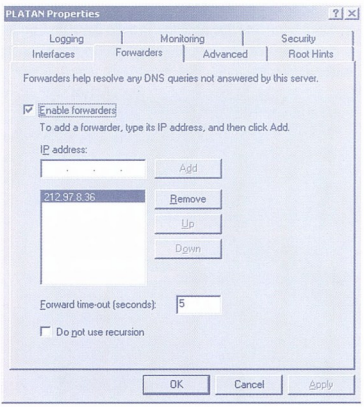
V: Mert le merem fogadni, hogy a zónán nem engedélyezted a dinamikus frissítést. Jobbklatty, properties, Allow Dynamic Update, Yes, Ok.



K: De engedélyeztem, és mégsem ír bele
V1: Mert elgépelted a tartományod nevét *(hogy ez milyen gyakori! Legalább tíz esetet láttam, ahol a jól felkészített falatraksz.hu zónába az Istennek sem kerültek be a falatrax.-hu tartomány bejegyzései :-)*
V2: Mert türelmetlen vagy. Az újraboot után jó néhány perc, mire megjelenek az SRV rekordok. Ha unod a várakozást
NET STOP NETLOGON
NET START NETLOGON

K: Az igaz, hogy elsőre elbaltáztam, de most már tele a zóna, ennek ellenére minden hibajelenség ugyanúgy fennáll. Megőrülök!
V1: Vár még tíz percet, és minden hirtelen megjavul! :-O
V2: Hát igen. Ez - hogy is mondjam nyomdafestéket tűrő módon – baj. Ugye hiába búsol a Tech.Net CD-t, semmi válasz? Hát tudd meg, itt összeesküvés van. A TCP/IP stack újításai közül a legzavaróbb a minden 2000 gépen jelen lévő DNS gyorsítótár *(cache)*, mely az egyszer már begyűjtött DNS rekordokat betárazza, hogy legközelebb ne kelljen ugyanazért a rekordért felkeresni a DNS Server-t. Ám a cache úgynevezett negatív bejegyzéseket, tehát válaszhiányokat is tárol, így hiába javul ki a zóna, még 5-10 percig a régi semmi van a gyorsítótárban. Tekintsd meg:
IPCONFIG /DISPLAYDNS
Majd irtsd ki:
IPCONFIG /FLUSHDNS
és már megy is minden, mint a karikacsapás!

K: Miért nem látjuk az Internetet? Eddig olyan szépen ment!
V: Mert a DNS szervered, amelyre minden ügyfél mutat, nem lát ki, hisz senki sem mondta meg neki, merre keressen tovább, ha egy kérdésre nem tudja a választ. Be kell állítunk úgynevezett forwarder DNS Server-t, ezzel minden ismeretlen kérést átirányítunk például az Internetszolgáltatónkhoz. Jobbklatty a DNS Server-en, Properties, Forwarders fül.



K: Ott vagyok éppen, de mind a Forwarders, mind a Root hints fülön szürke minden beviteli mező. Most mindennek vége?
V: Nem, dehogy. Most kezdődik a tánc! Az a baj, hogy – véletlenül vagy szándékosan – root DNS-t sikerült telepíteni. Beszéljük le a DNS szerverünket a Root Name Server szerepről! Menj vissza a zónákhoz, állj rá a . *(pont)* nevűre, és töröld le. *(Ne kérdezd miért, csak tedd amit mondok.)* Most nyomj F5-öt. Na most menj vissza a Forwarder beállításra. Mellesleg a Root Hints is magától feltöltődött!

Hát hirtelen ennyi, és akkor hol van még a zónadelegáció, az NS, MX és ki tudja még milyen rekordok! Mindenkinek melegen ajánlom DNS cikksorozatunkat! És most következzen a szokásos tallózás a NetAcademia nyilvános levelezési listáinak forgalmából. ***(Feliratkozás: http://lyris.netacademia.net)***

K: Sajnálattal tapasztaltuk, hogy az SQL Server 6.5-ös a tárolt eljárásokat TÁROLT ELJÁRÁSKÉNT meghívva HOSSZABB idő alatt futtatja le, mintha egyszerű sql scriptként futtatnánk ugyanazt. Emiatt fejlesztés és tesztelés közben egy kb. 200 - 250 ezer rekordot érintő szűrés 1,5 perc alatt futott végig scriptként, ám tárolt eljárásban a válaszidő 5 és 7 perc között volt. Sajnos a tárolt eljáráson már nem lehet egyszerűsíteni, mert az már a használhatóságát vonná kétségbe *(indexek vannak, és az algoritmus kötelezően bonyolult)*. Érdekes módon SQL7-es alatt a lekérdezés ugyanezen tábláknál ugyanekkora adatbázisban a kétféle módszerrel egyformán 35 másodperc alatt futott végig. De sajnos nem az a feladat, hogy SQL7-esre térjünk át. Mit tegyek?

V: Van egy jó meg egy rossz hírem, s a kettő ugyanaz: térj át SQL Server 7.0-ra, vagy méginkább 2000-re. A dolog magyarázata az SQL Server 6.5 és 7.0 memóriakezelése, lekérdezés optimalizálója, joinstratégiái, egyszóval belső felépítése közötti különbségekben rejlik. Lehet ugyan játszadozni a tárolt eljárás meghívásakor a WITH_RECOMPILE opcióval, amely minden alkalommal új végrehajtási tervet készíttet a lekérdezéshez, de az eredmény több mint kétséges. Az SQL Server 6.5 sohasem volt képes dobogós helyet elfoglalni a független szakértők által zsűrizett teljesítménytesztekben **(http://www.tpc.org)**, míg a 7.0 verzió már béta korában megtette ezt. Fontos tudni, hogy az SQL Server 2000 egyenesen áll a képzeletbeli dobogó legfelső fokán, mi több, lassan az első öt helyen SQL 2000-t fogunk találni.
Forrás: NetAcademia SQL lista

K: Vállalatunknál az előző, azóta elment rendszergazda telepítette az Exchange Server 5.5-öt mégpedig sajnos úgy, hogy az Administrator lett a Site Service Account. Ha most megpróbálom megváltoztatni az Admin jelszavát, akkor a következő alkalommal nem indul el az Exchange. Később észrevettem, hogy az Exchange Admin programmal „után lehet húzni" a jelszót, de ez akkor is kényelmetlen. Nem lehetne lebeszélni az Exchanget arról, hogy így jelentkezzen be?

V: De. Control Panel->Services. A megjelenő listában minden egyes „Microsoft Exchange...“ kezdetű szolgáltatáson a StartUp nyomógomb mögött meg lehet változtatni azt a felhasználói fiókot, amivel a szolgáltatás fut. Persze előtte az új Servicce felhasználónak adni kel egynémely jogot: az NT-ben „Log on as a Service“, Act as part of the op. System“ és „Backup Files and Directories“, míg az Exchange-ben minden szinten *(Organization, Site, Configuration)* „Service Account Admin“ szerepet kell adni *(majd az Organization szintjén el kell venni tőle a Search jogot a ServAcctAdmin szerephez képest, ha előtte sem volt beállítva)*. Talán mégis könnyebb lett volna eleve jól telepíteni :-)

Forrás: NetAcademia Exchange lista

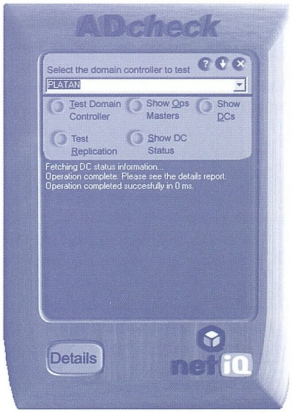
K: Létezik az Active Directoryhoz valami könnyen használható ÉS ingyenes elemzőprogram?

V: A **http://www.netiq.com/adcheck/** címről letölthető eléggé egyedi felhasználói felületű *(olyan, mint egy PDA)* NetIQ ADcheck termék a megfelelő eszköz az Active Directory vizsgálatára.

A következő tesztek hajthatók végre a segítségével:

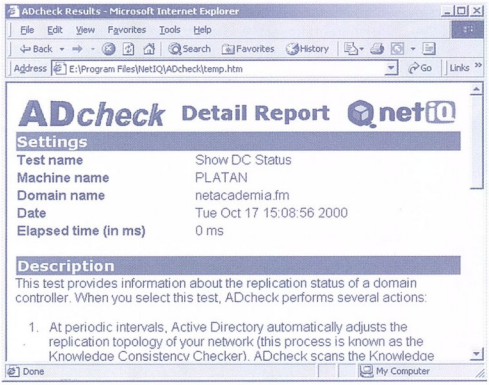
- Hálózati kapcsolat, és egyéb beállítások helyessége
- Tartományvezérlők listázása, működésük elemzése
- FSMO szerepek listázása
- Egyes tartományvezérlők közötti replikáció sikeressége
- A replikáció aktuális állapota *(mely gépek vannak lemaradva, és mennyivel)*

A tesztek eredményét pedig a nagy Details feliratú gomb megnyomása után HTML-ben kapjuk!



K: Az lenne a feladatom, hogy sok száz gépen állítsak be azonos NTFS jogokat. Ezt egyszer még könnyű végigkattintgatni, de nem lehetne erre valahogy egy batch fájlt készíteni, azaz parancssorból állítani az NTFS jogosultságait?

V: A Windows NT/2000 része a CACLS.EXE parancssori segédprogram, mellyel a feladat könnyen elvégezhető. Majdnem mindent tud, például a Windows 2000 verzió már arra is képes, hogy ne egyszerűen felülírja a jogosultságlistát, hanem



beleszerkesszen. Ezzel az ingyenes változattal azonban csak a szokásos jogosultságok beállítása valósítható meg, azaz Read, Write, Change *(write)* és Full control állítható be. Ha ennél is finomabban kell állítani a jogosultságokat, akkor a Windows 2000 Resource Kiten található XCACLS.EXE-re lesz szükség, amely sokkal finomabban hangolhatóvá teszi a jogosultságállítást, hisz az előbbieken túl Change Permissions, Take Ownership, Execute, egyedi Read, Write, Delete és üres *(!)* ACE is állítható vele. Természetesen VBScript is írható, külső gyártók szoftverei is használhatók, de az XCACLS.EXE funkcióinál aligha lesz szükség többre.

Forrás: NetAcademia Windows 2000 lista

K: Azt szeretném elérni, hogy az Encrypting File System titkosítás egy egérkattintásnyi közelségbe kerüljön a felhasználókhoz, ne kelljen állandóan a Properties->Advanced ablakba beszaladni, valahányszor titkosítani akarok valamit.

V: A Windows 2000 Beta 3 még tartalmazott EFS menüpontot a fájlok gyorsmenüjében, ám a végleges változatból ezt sajnos kivették. Visszacsempészni a következő módon lehet: a

`HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced`

kulcs alá fel kell venni a

`EncryptionContextMenu=1 (REG_DWORD)`

Értéket, ekkor újraindítás után megjelenik a gyorsmenüben az Encrypt/Decrypt!

Forrás: NetAcademia Security lista

XML – kezdetnek nem rossz!

Az októberi különszámunkban megjelent SQL annotált sémák késztettek e cikk megírására. A visszajelzések alapján ugyanis világossá vált számunkra, hogy égető szükség lenne egy tudományos fantasztikumtól mentes, kizárólag értelmes szavakat tartalmazó XML cikk megjelentetésére, mert amíg valaki meg nem írja élete első XML/XSL dokumentumpárosát, addig bizony nehezen éli bele magát a bonyolultabbnál bonyolultabb XML alkalmazások *(például BizTalk Server)* lelkivilágába. Azután itt vannak a szokásos kérdések: az XML az egy új programozási nyelv? Nem. Új adatbázis-formátum? Nem-nem!

A teljesség igénye nélkül

Félre hát az SGML-lel és az URN-ekkel, készítsünk hirtelen olyan XML dokumentumot, amely nem több mint három sor - készítsük el a „Hello World!" alkalmazást XML-ben! Elő a „Visual" Notepaddel!

```
<gyoker>
      <duma>Hello World!</duma>
</gyoker>
```

Ennyi. Save. Legyen HELLO.XML a neve. Nyissuk meg Internet Explorerrel. Hát nem gyönyörű? Ha Internet Explorer 5-össel nézzük, akkor valami hasonlót kell látnunk:

Korábbi böngésző használata esetén nem tudom, mi jelenik meg :) Aki ki szeretné próbálni a cikk további példáit, használja fel a képernyőképekről leolvasható URL-eket, hisz a fájlok az újság weblapján is elérhetők. *(Az egyszerűség és olvashatóság kedvéért az összes XML példám ékezetmentes, mert UTF-8 kódolással kellett volna a Notepaddel elmentenem a fájlokat, amire a Notepad képes ugyan, de kerülni próbáltam a konfliktushelyzeteket.)*

De mit is csináltunk? Látható, hogy XML kódocskánk nagyon hasonlít a HTML nyelvre, ugyanúgy <tag>-ekkel *(magyarul: HTML tagok)* dolgozunk, mint hagyományos weblapkészítés közben. A drámai különbség az, hogy kinek szólnak e tagok: tisztán látszik, hogy nem a böngészőnek *(nem, nem azért lett piros!)*, hisz érintetlenül megjelentek a képernyőn. Honnan is tudná az Explorer, miként kell formázni a <duma> tagot? Az XML fájlban a tagok nem az adatok megjelenítését vezérlik, hanem a végfelhasználónak *(embernek vagy gépnek)* nyújtanak információt a tartalom logikájáról. Valaha a HTML nyelv is ügyelt erre, hisz nem formázó, hanem tartalmi leírótag például a <title>, ám az évek folyamán az egyre giccsesebb weblapok megjelenésével ez a vonulat elsorvadt. Hajdanában a <title> tagot valóban a dokumentum címével illett feltölteni, hogy a keresőprogramoknak könnyebb dolguk legyen, ám mivel ez jelenik meg az ablak tetején a böngésző címsorában, mi sem kézenfekvőbb, minthogy ideírjuk marketingakcióink időpontjait.

Az XML azonban visszahozza a dokumentumokba a tartalmi leírást, a metaadatokat. Egy XML dokumentumban sohasem találunk közvetlen megjelenítési, formázási információt, vagy ha igen, az baj. Az XML fájl legyen adat központú, hisz sohasem tudhatjuk, hogy mi a végső célja: ha bankkártya adatokat cipel két vállalat között *(született az SAP-ben, majd a BizTalk Serveren átvonulva elnyeli az SQL Server 2000)*, akkor beláthető, hogy a külalaki információk tökéletesen feleslegesek! Más a helyzet, ha böngészőben meg kell jeleníteni. Ilyenkor segítenek a stíluslapok, az XSL fájlok - de erről majd később.

Most térjünk vissza az XML-hez, és vizsgáljuk meg, mitől több ez, mint spanyol viasz.

Kiterjeszthető

XML=eXtensible Markup Language. Bármit le lehet írni vele. Íme a kedvenc Billjeim:

```
<gyoker>
      <bill>Bill Clinton</bill>
      <bill>Bill Gates</bill>
      <bill>Billy Idol</bill>
      <bill>Buffalo Bill</bill>
</gyoker>
```

Ha jellemezni kellene őket, ugyanúgy paramétereket adhatok nekik, mint például HTML-ben egy betűtípusnak. Hasonlítsuk össze ezt:

```
<font face="arial" size="2"
color="red">Hello World!</font>
```

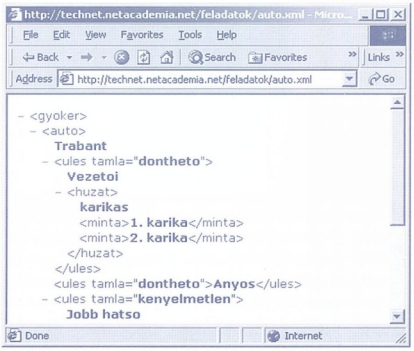
ezzel:

```
<bill munkahelye="Fehér Ház" kora="44"
biztos_ez="nem">Bill Clinton</bill>
```

Hierarchikus

Vegyünk egy a valós életből ellesett, szigorúan hierarchikus példát! Írjuk le egy autó összes ülésének, támlájának, huzatának mintáit egyetlen fájlban!

A hierarcikus jelleg olyan esetekben igazán hasznos, amikor például egy összetett, apa-fiú adatbázis-lekérdezés eredményét kell eljuttatni egy munkaállomásra.



nehézségi fok: ●●●○○

Az XML előtti világban ez csak több, táblázatos lekérdezés egymás utáni átvitelével volt lehetséges. *(Illetve szabványmentes módon egy-két gyártó megvalósította ugyan, de piaci sikert senki nem ért el vele.)*

Szabványos

Ennek fontosságát azt hiszem, nem kell ecsetelnem. Végre mindenfajta konverzió nélkül szót ért egymással SQL Server és Oracle, Exchange Server és majd a konkurensek is, ha végre észbe kapnak.

Szigorú

A szigorúság alatt azt értem, hogy az XML szintaktikája sokkal kötöttebb, mint hinnénk, és ez többnyire jót tesz neki. Amitől viszont minden Windowshoz szokott XML guru el fog hűlni: megkülönbözteti a kis- és nagybetűket *(case sensitive)*. A <tag>-nak nem bezárója a </Tag> :-(Úgy tűnik egy nagyhangú junixos is helyet kapott a szabványtestületben. Viszont igen lényeges szigorítás, hogy minden tagot minden körülmények között be kell zárni, még akkor is, ha nincs értéke. A HTML-ben simán elhelyezhetünk lezárás nélkül egy önálló
 (sortörés) tagot, XML-ben viszont ennek így kell kinéznie:

```
<önlezárás/>
```

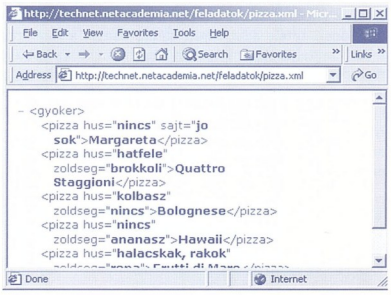
Olvasható

Végre itt a világ legjobban debuggolható formátuma! Az XML fájl ugyanis értelmes TXT, olvasható! Mindenféle segédeszköz nélkül tudunk benne hibát keresni. A formátum nyilván csak egy adott befogadható adatmennyiségig élvezetes olvasmány, de hibakeresésre tetszőleges szövegszerkesztő használható. Ugyanakkor e szószátyárság miatt jó nagy is, ám pont emiatt...

Remekül tömöríthető

Persze, hisz a redundancia foka közel végtelen amikor már egymilliomodszor ismételjük el a fájlban a tag nevét:

```
<gyoker>
<pizza hus="nincs" sajt="jo
sok">Margareta
</pizza>
<pizza hus="hatfele" zoldseg="brokko-
li">Quattro Staggioni
</pizza>
</gyoker>
```



Megjelenítés XSL stíluslapokkal

Evezzünk vadabb vizekre! Próbáljuk Internet Explorerben csinosan megjeleníteni az adatokat! Ehhez az XML fájl mellett szükségünk lesz egy másik fájlra is, mely a megjelenítési információkat hordozza - ez az XSL. Ha az XSL-t valóban megjelenítésre használjuk *(miért, mire lehet még...?)* akkor belső szerkezete HTML jellegű. Képzeljük el úgy, mint egy mintát, melyet az XML adattartalmával kell kitölteni. Ha az XML-ben 1000 féle pizza van, amelyet egy HTML táblázatban szeretnénk látni, akkor az XSL tartalma logikailag a következő:

```
<html>
<table>
Ezt ismételd meg annyiszor, ahány
pizza van:
      <tr>
<td>Na ide jöhetnek a pizzanevek</td>
<td>Ide pedig a sajtosság foka</td>
            <td>stb....</td>
      </tr>
      ciklus vége
</table>
</html>
```

Az igazsághoz hozzátartozik, hogy ez a fajta procedurális megközelítés csak az egyik, mégpedig a primitívebb XSL megjelenítési lehetőség, mely kizárólag well-known, azaz jól ismert struktúrájú adathalmazon használható. Amennyiben sánta fa és egyéb érdekfeszítő adathalmaz feldolgozására volna szükség *(lásd később)*, akkor elő kell*(ene)* vennünk halmazszemléletű XSL tudásunkat...

Térjünk vissza mindkét lábunkkal a földre, és fejezzük be az előbbi XSL-t úgy, hogy az előbb ékes magyar nyelven beírkált utasításokat lefordítjuk XSL transzformációs tagokra:

```
Ezt ismételd meg annyiszor, ahány piz-
za van
      ...
ciklus vége
```

helyett

```
<xsl:for-each select="gyoker/pizza">
      ...
</xsl:for-each>

Na ide jöhetnek a pizzanevek
```

helyett

```
<xsl:value-of /> (Önlezáró.)

A <pizza> tagok belső attribútumait
(hus, zoldseg) pedig a következőképpen
írathatjuk ki:

<xsl:value-of select="@hus"/>
<xsl:value-of select="@zoldseg"/>
```

ahol a kukac karakter jelzi, hogy itt nem egy érték, hanem egy attribútum tartalmára vagyok kíváncsi. Végül lássuk el a fájlt a hagyományos XSL fejléccel,

```
<?xml version='1.0'?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

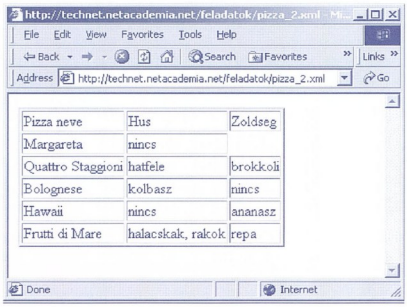
melyet megérteni nem kell, csak flopin a zsebünkben hordani, hogy ha hirtelen kellene XSL-t alkotnunk, ne álljunk bután és tehetetlenül a probléma megoldásának utolsó mozzanata előtt. Megfigyelhető, hogy az XSL fájl is XML nyelven íródott, csak egyfajta szabályos, rögzített szintaxisú XML-lel van dolgunk.
Apropó flopi: mentsük rá azt a sort is, amelyik az XLM-XSL összerendelést végzi – az sem triviális!

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="pizza.xsl"?>
```

Az XSL végleges formában:

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl=
"http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <BODY>
        <TABLE BORDER="2">
          <TR>
            <TD>Pizza neve</TD>
            <TD>Hus</TD>
            <TD>Zoldseg</TD>
          </TR>
          <xsl:for-each select="gyoker/pizza">
            <TR>
              <TD><xsl:value-of /></TD>
              <TD><xsl:value-of
select="@hus"/></TD>
              <TD><xsl:value-of select="@zold-
seg"/></TD>
            </TR>
          </xsl:for-each>
        </TABLE>
      </BODY>
    </HTML>
  </xsl:template>
</xsl:stylesheet>
```

Az XML ezek után így jelenik meg:



Hogy az alapokból semmi se maradjon ki, most készítsünk halmazszemléletű XSL megoldást is, ehhez azonban először szükségünk lesz egy táblázatba nem foglalható, masszívan hierarchikus XML adathalmazra:

```
<gyoker>
<mozi>Muvesz
<terem>Chaplin
            <film>Matrix
                  <kezdet>14:00
                  </kezdet>
                  <kezdet>16:00
                  </kezdet>
            </film>
            <film>Forrest Gump
                  <kezdet>17:00
                  </kezdet>
            </film>
      </terem>
      <terem>Huszarik
            <film>Mumia
                  <kezdet>23:59
                  </kezdet>
            </film>
      </terem>
</mozi>
<mozi>Kobanya
      <film>Rain Man
            <kezdet>17:00
            </kezdet>
            <kezdet>18:30
            </kezdet>
      </film>
      <szakkor>Vasutmodellezes
            <nap>Szombat
            </nap>
            <kezdet>13:00
            </kezdet>
      </szakkor>
</mozi>
</gyoker>
```

Könnyen belátható, hogy ez a struktúra kényelmesen nem jeleníthető meg kétdimenziós táblázatban, hisz a Művész mozi esetén mozi->terem->film->kezdet egymásba ágyazást láthatunk, míg a Kőbánya esetén a terem megkülönböztetésének nincs értelme, továbbá ott nemcsak filmeket vetítenek, hanem szakkörök is vannak – egyébként tavaly bezárt :(. A leghelyesebb az lenne, ha nem is kellene ciklikusan végigfutkározni a sorokon, és rákeresni, hogy vajon éppen miféle adat került elénk, hanem a formázást a lehető legteljesebb mértékben rábízhatnánk az XSL értelmezőre. Ennek semmi akadálya. Az XSL képes arra, hogy az általunk definiált formátummintákat automatikusan, ciklikusan, rekurzíve stb. „ráfesse" az adatokra. Nem kell mást tennünk, mint úgynevezett templateket definiálnunk, hogy hogyan is nézzen ki egy filmcím, legyen az bármilyen mélyen is a hierarchiában,

```
<xsl:template match="film">
      <font  color="green"><xsl:apply-
templates/></font>
</xsl:template>
```

azaz a filmcím legyen zöld. Az xsl:apply-templates arra szólítja fel az XSL értelmezőt, hogy ezen a ponton vesse bele magát az XML fáljba, és keresse ki az adatokat. Érdemes megjegyezni, hogy az xsl:apply-templates rekurzív hívás, azaz ha az előásott adatoknak gyermekadataik vannak *(itt a filmeknek kezdetük van)* akkor azokra ismét megadhatunk templatet, valahogy így:

```
<xsl:template match="kezdet">
      <b><xsl:apply-templates/></b>
</xsl:template>
```

A kezdet félkövér lesz. Itt azonban két esetet meg kell különböztetnünk: ha a kezdet egy film gyermeke, akkor a rekurzív hívás miatt <xsl:apply-templates> örökli a szülő formátumát is, emiatt nemcsak félkövér, hanem zöld is lesz. Ha a kezdet nem filmhez tartozik, hanem például egy szakkörhöz, akkor fejlesztésünk jelen stádiumában nincs mit örökölnie a szülőjétől *(mert arra még nem írtunk megjelenítési mintát)*, ezért egyszerűen félkövér lesz. Írjunk mintát a szakkörre is:

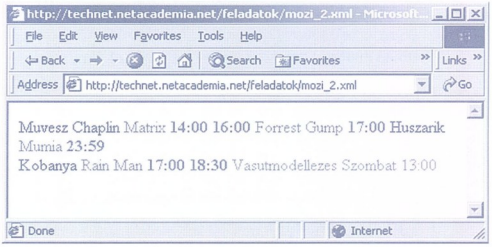
```
<xsl:template match="szakkor">
      <font    color="red"><xsl:value-
of/></font>
</xsl:template>
```

Vajon ettől pirossá válik-e a szakkör alatti kezdete tag kijelzése is? Igen, mert az <xsl:value-of> nemcsak az adott tag értékét írja ki az adott mintával, hanem a gyermekekét is. Ez néha súlyos bonyodalmak forrása lehet, mert ha tényleg csak az adott szint értékét kérjük, azt másképp, XSL függvénnyel kell megadni, így: <xsl:value-of select="text()"/>. Ez leállítja a rekurziót. Végül hadd mutassam meg a dzsoli dzsóker mintát, amely szintén az előbbi text() függvény használatával minden egyes további, mintával el nem látott, illetve „mintás", de <xsl:value-of> taggal ki nem íratott értéket egyszerűen a meghívás helyén *(rekurzióról van szó!)* kidobál a kimenetre:

```
<xsl:template match="text()">
      <xsl:value-of/>
</xsl:template>
```

Ez a minta minden halmazszemléletű XSL fájban legyen benn, hogy tehesse azt, amire való: jelenítse meg a kifelejtett, félredefiniált halmazok adatait is. Természetesen itt is használhatunk HTML formázást, de vigyázzunk: minden fent említett elemre hatással lesz ez a minta!
És már majdnem készen is vagyunk, még kell egy globális template az egész folyamat beindításához *(match="/")* és save.
A mozi.xsl egy kicsit túl nagy ahhoz, hogy ide illesszem, ezért letölthetővé tettem. A cikkben előfordult összes XML és XSL fájl letölthető a következő címről:
http://technet.netacademia.net/feladatok/xml.zip

A „formázott" dokumentum pedig így néz ki:



A zip remek lehetőséget nyújt arra, hogy példáimat ki-ki ízlése szerint átírkálja, próbálkozzon vele. A további tanulást segíti az msdn.microsoft.com címen található XML tutorial gyűjtemény, amely a cikk által felszínre kerülő XML, XSL kérdésekre megnyugtató válaszokat nyújt. Természetesen a levelezési listán is szívesen válaszolok.
Végezetül hadd jelezzem, hogy az XSL fájlok nemcsak megjelenítésre valók, hanem ennél átfogóbb szerepük van: általános XML transzformációra alkalmasak. Megfelelő módon alkalmazva az XSL-ek képesek A.XML-ből B.XML-t faragni, ahol A.XML például egy autógyártó cég rendelési formátuma, míg B.XML egy alkatrészbeszállító cég gyártósorvezérlő adata. Ez azonban már egy másik történet: a BizTalk Server legendája...

Fóti Marcell
MCT, MCSE, MCDBA

Telepítés, adminisztráció és tervezés

Az Office Server Extensions azért készült, hogy lehetővé váljon a munkacsoporton belüli valódi együttműködés, rendszerfelügyeleti oldalról pedig az volt a cél, hogy minél egyszerűbb legyen kezelni, s ahol lehet, használjuk ki a meglévő technológia előnyeit. Most a második részben az Office Server Extensions használatának tervezésével foglalkozunk, megnézzük a telepítést és a felügyeleti funkciókat. További információk a Microsoft Office 2000 Resource Kit-ben olvashatók.

A munkaállomások hardverigénye

Olyan számítógépre van szükség, amely képes a Microsoft Office 2000-t, Netscape Navigator 3.0-t, az Internet Explorer 3.0-t, vagy ezeknél újabbat futtatni.

A munkaállomások szoftverigényei

A használhatóság nagyban függ a Microsoft Office és a böngésző verziójától. Az alapfunkciókhoz csak egy webböngésző szükséges. A Netscape Navigator-t vagy Internet Explorer 3.0-t használók az Office Server Extensions megbeszélési és előfizetési funkcióit a Kezdőoldalon keresztül, a Discussions eszköztár keretekre épülő verziójával érhetik el. A Microsoft Office 2000-et és Internet Explorer 4.01-et használók közvetlenül az Office alkalmazásból vagy a böngészőből használhatják. Csak a DAV kiszolgálóra történő közzétételhez szükséges Internet Explorer 5.0.

A kiszolgáló hardverigényei

Az Office Server Extensions hardverigényei nagyban függenek az adott telepítési környezettől. Az egyidejűleg dolgozó felhasználók várható száma alapján számított igényeket a Microsoft Office 2000 Resource Kit tartalmazza.

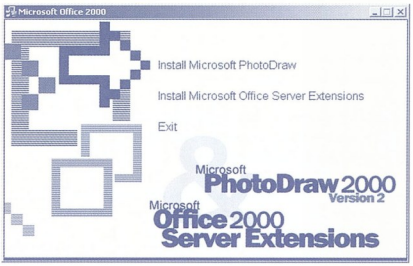
A kiszolgáló szoftverigényei

- Windows NT Server vagy Workstation 4.0 Service Pack 4-gyel, Internet Information Server 4.0 vagy újabb, a Windows NT Option Pack-ről telepített Index Server *(a kereséshez)* és Internet Explorer 4.01 vagy újabb.
- Microsoft Exchange Server vagy más SMTP levelezési kiszolgáló a hálózatban *(az értesítésekhez)*.

Az Office Server Extensions telepítése

Az Office Server Extensions olyan számítógépeken használható, amelyek Windows NT Workstation-t vagy Server 4.0-t vagy újabbat *(Service Pack 4-gyel)* futtatnak. A telepítés az Office 2000-éhez hasonlít, és felügyelet nélküli telepítésre is lehetőség van.

A Telepítő az Office 2000 Telepítőtől független alkalmazás, neve Setupse.exe, és vagy a Microsoft Office 2000 Professional-en *(1. CD)* vagy az Office 2000 Premium-on *(3. CD)* található meg. Ha az OSE telepítőt az Autorun.exe-ből futtatjuk a 3. CD-ről *(Office Premium)*, az meghatározza, hogy rendelkezünk-e az OSE telepítésének feltételeivel – az Internet Explorer 4.01-gyel, a Windows NT Options Pack-kel és a Windows NT SP4-gyel – és segítséget nyújt ezen összetevők telepítéséhez. *(Az OSE telepítőnek ez a funkciója nem használható az Office 2000 Professional-ből.)*



Az OSE telepítő ezután azt a varázslót futtatja, amely beállítja az Office Server Extensions-t, és lehetővé teszi a Webkiszolgálón történő közzétételt, a webmegbeszéléseket és a webelőfizetéseket. A varázsló futása után a web készen áll arra, hogy az Office 2000 felhasználói együttműködjenek, Office dokumentumokat és weboldalakat tegyenek közzé. A Windows NT rendszergazda a telepítést egy varázsló segítségével végzi el és a szokásos módon felhasználói nevet és licencinformációkat kell megadnia. Ezután a telepítés helyét jelöli ki. Az Office Server Extensions telepítés részeként a következő elemek némelyike vagy mindegyike települ:

- FrontPage 2000 Server Extensions
- Microsoft Data Engine *(MSDE, de csak akkor, ha nincs a gépen SQL Server)*
- Windows Installer Service Pack.

A telepítés végén a beállítási varázsló elkezdi a webkiszolgáló bővítését az Office Server Extensions-szel.

A varázsló olyan alapértelmezett értékeket tartalmaz, amelyek a legtöbb telepítés esetében használhatóak. Ezek után befejeződik az Office Server Extensions-t tartalmazó kiszolgáló beállítása, és így alkalmassá válik arra, hogy segítségével az Office 2000 felhasználói együttműködjenek és dokumentumokat tegyenek közzé. A rendszergazda a következőket állíthatja be:

- Az adatbázis nevét és jelszavát, ha a Microsoft Data Engine van telepítve. Ha a webkiszolgáló már rendelkezett korábban telepített SQL Server adatbázissal, egy meglévő SQL Server adatbázisnevet, SQL Server felhasználó-

nehézségi fok:

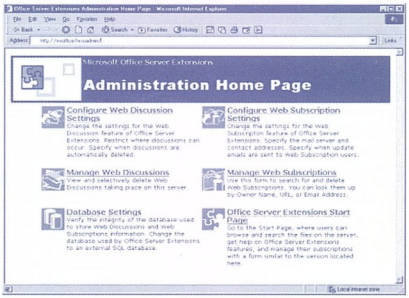
nevet és SQL Server jelszót kell megadni. Ha a Microsoft Data Engine adatbázisról SQL Server-re frissítünk, nincs szükség semmilyen beavatkozásra.

- Létrejöjjenek-e felhasználói csoportok és rendszergazda fiók az NTFS adatvédelemhez. Ezzel üres rendszergazda *(admin)*, szerző *(author)*, böngésző *(browser)* és együttműködési *(collaborator)* csoportok jönnek létre az Office Server Extensions kiszolgálón.
- Ki férjen hozzá a webmegbeszélés funkcióhoz. A már létrehozott, ismert Windows NT fiókok és csoportok állíthatók be, vagy mindenki – akár Windows NT fiók nélkül is. Az alapértelmezés az ismert NT fiókok.
- Engedélyezzük-e az Internet Explorer-en kívüli böngészőket *(alapértelmezés szerint engedélyezzük)*. Ez a beállítás egyúttal az alaphitelesítést is bekapcsolja.
- Engedélyezzük-e az AutoNavigation oldalakat a webhez *(alapértelmezés szerint engedélyezzük, és ezzel bekapcsoljuk a könyvtárböngészést is)*.
- A levelezési beállításokat az előfizetéshez.

A varázslóban megadott beállítások később az Office Server Extensions Administration Home Page használatával változtathatók meg. A varázsló beállításait meg kell adni az Office Server Extensions első futtatása előtt. Ha ezeket nem adjuk meg, az előfizetés és értesítés funkció nem működik a levelezési beállítások hiányos volta miatt.

Rendszerfelügyelet

Az Office Server Extensions-t tartalmazó kiszolgáló karbantartását az Office Server Extensions adminisztrációs eszközei segítik, amelyek az Administration Home Page-ről elérhető webalapú űrlapok. A FrontPage adminisztrációs eszközökkel szabályozhatjuk, hogy egy munkacsoport-webre ki tehet közzé dokumentumokat.

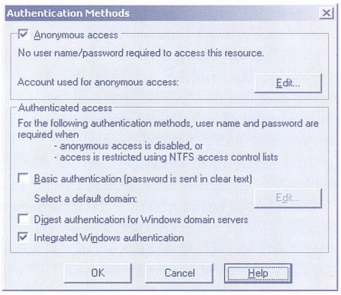


Adatvédelem

Az adatvédelmet a Windows NT Server, az IIS, a FrontPage Server Extensions és az Office Server Extensions kombinációja biztosítja. A Windows NT Server adatvédelem használatához NTFS-t kell használni.

Az adatvédelem beállítása az igényeknek megfelelően kiszolgálónként különböző lehet. Az alábbiakban az adatvédelem elemeinek szerepét tekintjük át:

- **Windows NT Server**: Hozzáférést biztosít a kiszolgálókhoz és a fájlmegosztásokhoz. NTFS használatával egyedi jogokat rendelhetünk a felhasználókhoz és csoportokhoz. A Windows NT User Managerrel, illetve Windows 2000 esetén az Active Directory Users and Computers eszközzel hozhatunk létre felhasználókat és csoportokat, majd a megfelelő jogokat ezeknek adhatjuk ki.
- **Internet Information Server** *(IIS)*: Az IIS a böngészők felé biztosítja az adatvédelmet.



A hitelesítés négy szinten állítható be:

- Az **anonim** *(Anonymus)* hitelesítéssel minden felhasználó hozzáférhet a webhelyszínhez, még azok is, akik nem rendelkeznek Windows NT fiókkal. *(Ez tulajdonképpen a hitelesítés kikapcsolását jelenti.)*
- Az **alap** *(Basic)* hitelesítés nem titkosítja a felhasználónevet és a jelszót. A Basic hitelesítést sok webkiszolgáló támogatja, az IIS továbbítani tudja a bejelentkezést a Windows NT Server-nek. Ha a Basic hitelesítést a Secure Sockets Layer-rel *(SSL)* kombináltan használjuk, a hitelesítés gyors és biztonságos lehet. *(Az SSL titkosítja a továbbítást, így illetéktelenek nem tudják az adatokat olvasni.)*
- A Windows NT Challenge/Response *(más néven NTLM, vagy integrated Windows)* biztonságosabb módszer, mint az alaphitelesítés. Az NTLM nem működik tűzfalon keresztül, nem továbbítja a bejelentkezést másodlagos kiszolgálóknak, és Internet Explorer 4.01-et vagy újabbat igényel. Ha a kiszolgáló Windows 2000, akkor Kerberos hitelesítés is történhet.
- A Digest hitelesítés viszonylag új, szabványos forma, amelyet viszont csak az Internet Explorer képes egyelőre használni. 80-as porton kommunikál, így tűzfalakon is keresztüljut.

Lehetőség van arra, hogy mind az alap, mind a Windows NT Challenge/Response hitelesítést engedélyezzük. Ha a webböngésző támogatja a Windows NT Challenge/Response-t, ezt a hitelesítési módszert, egyébként az alaphitelesítést használja. Az IIS azt is lehetővé teszi, hogy egyes TCP/IP címeket *(vagy számítógépeket)* letiltsunk. Ha minden felhasználó Internet Explorer 4.01-et vagy újabbat használ, az alaphitelesítést kikapcsolhatjuk. A Netscape nem támogatja az NTLM-t, így az ilyen felhasználók igénylik az alaphitelesítés meglétét.

- **FrontPage Server Extensions:** Mind a FrontPage, mind az Internet Information Server lehetővé teszi, hogy a rendszergazda több különálló webhelyszínt hozzon létre egy kiszolgálón. A FrontPage azt is lehetővé teszi, hogy minden webhez különböző adatvédelmi szerepet állítsunk be: böngészőt, szerzőt vagy rendszergazdát. A FrontPage-dzsel egy weben belül alwebeket is létrehozhatunk, amelyekkel az adatvédelem tovább szabályozható. Például egy webhelyszín az egész vállalat számára nyilvános, de egy éppen létrehozás alatt lévő dokumentumhoz csak egy adott munkacsoport férhet hozzá.
- **Office Server Extensions:** Az Office Server Extensions telepítésekor minden webhez speciális felhasználói csoportok is telepítődnek, melyek egymásra épülnek – az együttműködő böngészhet és csoportmunkában vehet részt, a szerző pedig dokumentumokat írhat, csoportmunkában vehet részt és böngészhet. Az admin, author és browser csoportok megegyeznek a FrontPage Server Extensions-ben beállítottakkal – az Office Server Extensions az collaborator szerepet a megbeszélési funkcióhoz használja. A collaborator szerep lehetővé teszi, hogy a felhasználók csoportjainak anélkül adjunk jogot a webmegbeszélésekben való részvételre, hogy az egész webhez korlátlan hozzáférést adnánk nekik.

Az Office Server Extensions tehát a Windows 2000 Server adatvédelmére és a FrontPage Server Extensions szerepekre osztott adatvédelmi modelljére épít. Amikor az Office Server Extensions-t telepítjük a webkiszolgálóra, a rendszergazdának csak felhasználókat és csoportokat kell adnia az Office Server Extensions author és/vagy admin csoportjaihoz. Az alapértelmezések használata esetén a böngésző és együttműködő csoportokhoz az Everyone *(mindenki)* csoport, az admin csoporthoz pedig a LOCALMACHINE\Administrators csoport adódik.

Javaslat a Webkiszolgáló engedélyeinek kiadásához

Az Office-szal bővített biztonságos webkiszolgáló telepítésének legegyszerűbb és legkönnyebb módja az, ha a Windows NT Challenge/Response-t *(NTLM)* használjuk. Az NTLM egyszerűvé teszi a bejelentkezést, elegendően biztonságos hitelesítési módszer, viszont csak Internet Explorer-rel használható. Ha néhány felhasználó nem rendelkezik Internet Explorer 4.01-gyel vagy újabbal, telepítsük mind az alap, mind a Windows NT Challenge/Response hitelesítést. Ez az Office Server Extensions alapértelmezett beállítása. A használandó módszert a kiszolgálóhoz hozzáférő böngészők alapján választhatjuk ki. Ezután a felhasználókhoz egyedi jogokat úgy adhatunk, hogy a FrontPage-dzsel további webeket telepítünk.

Néhány webhelyszín ennél jobban szabályzott adatvédelmet igényel. A FrontPage beépített biztonságkezelő funkcióinak kikapcsolásával a FrontPage-dzsel bővített webre kézzel is beállíthatunk engedélyeket.

A webmegbeszélések adminisztrálása

A webmegbeszéléseket az Office Server Extensions Configure Web Discussions Settings oldal használatával engedélyezhetjük és tilthatjuk le.

Azt is meghatározhatjuk, hogy a felhasználók részt vehetnek-e távoli kiszolgálók dokumentumairól szóló megbeszéléseken. Alapértelmezésként az Office Server Extensions engedélyezi ezt, de ez az adatbázis méretének csökkentése érdekében korlátozható. Az adatbázis méretének csökkentése az automatikus törlés *(Automatic Deletion)* használatával is elvégezhető. Ekkor a megbeszélési elemek a megadott idő elteltével automatikusan törlődnek. A rendszergazda a megbeszélésre használható dokumentumok listáját is megtekintheti, majd kézzel is törölhet elemeket.

A webelőfizetések és értesítések adminisztrálása

A webelőfizetéseket az Office Server Extensions Administration Home Page használatával engedélyezhetjük vagy tilthatjuk le. Ha engedélyezzük a webelőfizetéseket, meg kell adnunk annak az SMTP levelezési kiszolgálónak a nevét, amely a leveleket küldeni fogja, valamint be kell állítanunk a használni kívánt küldő és címzett címeket is. Azt is meghatározhatjuk, hogy a felhasználók milyen időközöket választhatnak az értesítési e-mail fogadására. Beállíthatjuk, hogy a módosítástól számított pár percen belül, naponta egyszer vagy hetente egyszer kapjanak levelet. Az is beállítható, hogy mikor küldődjön el – ha az értesítési e-mail-t csúcsidőn kívülre időzítjük, a küldést nem zavarja más hálózati forgalom. Az aktív előfizetéseket is megtekinthetjük, különböző szempontok szerint szűrhetjük és szükség szerint egyes előfizetéseket törölhetünk is.

Ha nem rendelkezünk SMTP kiszolgálóval, vagy fontos az adatvédelem, ezt a funkciót letilthatjuk.

Fejlett rendszerfelügyelet

A Windows NT, az Internet Information Server és a FrontPage Server Extensions további felügyeleti eszközöket is tartalmaz:

- **Performance Monitor:** Valósidejű kiszolgáló-statisztikák összegyűjtéséhez a Microsoft Database Engine-en vagy az IIS-en.
- **FrontPage Server Administration Tools:** Az adatvédelem kezeléséhez, alwebek létrehozásához és kezeléséhez, a közzétételi funkciók engedélyezéséhez és letiltásához és egyéb FrontPage webfunkciókhoz. Az eszközök különböző formátumúak lehetnek – MMC snap in, parancssori és Windows alkalmazás. Az egyik ilyen funkció például egyszerűvé teszi az Office Server Extensions további webekhez adását, és az adminisztrációs beállítások megváltoztatását.
- **HTML kód:** HTML kóddal és az Office Server Extensions objektummodellel az AutoNavigation oldalak testre szabhatók.

Microsoft Office 2000

Tervezési szempontok

Ha az Office Server Extensions-t kis munkacsoportban használjuk, a tervezés igen egyszerű. Ellenőrizzük, hogy a számítógépek megfelelnek-e a rendszerigényeknek és hogy megfelelő engedélyekkel rendelkezünk-e a szoftver telepítéséhez és karbantartásához.

Nagyobb szervezetben egy kicsit körültekintőbben kell terveznünk, különösen a webmegbeszélések esetében. Itt az adatvédelmet és a teljesítményt kell megvizsgálni. Az Office Server Extensions telepítésekor a következőket kell megtervezni:

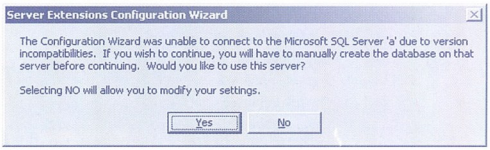
- **Rendszerigények:** A Microsoft Office 2000 Resource Kit használatával ellenőrizzük, hogy a hardver kielégíti-e az egyidejű felhasználók számához igazodó igényeket. Azt is ellenőriznünk kell, hogy rendelkezünk-e a szükséges szoftverekkel és service pack-ekkel.
- **Funkciók:** Mely funkciókat akarjuk engedélyezni? Le akarunk-e tiltani bizonyos funkciókat?
- **Webmegbeszélések:** Le akarjuk-e töröltetni automatikusan a régi megbeszéléseket? Engedélyezzük-e a távoli dokumentumokkal kapcsolatos megbeszéléseket?
- **Előfizetés és értesítések:** Rendelkezünk-e SMTP levelezési kiszolgálóval? Ha nem, egyszerűen telepíthetjük. Felmerülnek-e adatvédelmi problémák, ha egy felhasználó előfizethet olyan mappára, amelyhez nem rendelkezik olvasási engedéllyel?
- **Adatvédelem:** Az Office Server Extensions author, browser és collaborators csoportjaihoz megfelelő felhasználói csoportok adásával használjuk ki a már meglévő adatvédelmet. Ha a web egyes részei ettől különböző adatvédelmet igényelnek, hozzunk létre még egy webet. Döntsük el, hogy az együttműködők csoportnak alaphitelesítéssel adunk-e helyi bejelentkezési jogot. Döntenünk kell az együttműködők szerepéről is. Alapértelmezésként a beállítási varázsló mindenkinek együttműködési hozzáférést ad. Ha ezt el akarjuk kerülni, a beállítást az Office Server Extensions telepítésekor kell megváltoztatnunk. Később ez csak kézzel végezhető el. Mivel az AutoNavigation oldalak a könyvtárböngészés engedélyezését igénylik, ez az alapértelmezés. A nagyobb biztonság érdekében a könyvtárböngészést és az alaphitelesítést ki is kapcsolhatjuk.
- **A megbeszélési adatbázis:** Ha a webkiszolgálón már rendelkezünk SQL Server adatbázissal, és a mentéshez és a karbantartáshoz ki akarjuk használni a meglévő beállítások előnyeit, az SQL Server-t használjuk az adatbázishoz. Ha nincs SQL Server, az Office Server Extensions a Microsoft Data Engine adatbázist telepíti. Ha egy távoli SQL Server adatbázist akarunk használni és SQL Server még nincs helyileg telepítve, a telepítőt a „no database" paraméterrel futtassuk, és az adatbázist később határozzuk meg.

Integráció a FrontPage Server Extensions-szel

A FrontPage Server Extensions a webkiszolgáló olyan programjainak készlete, amely a felhasználók számára lehetővé teszi, hogy a FrontPage alapú webet – amely a webhelyszint felépítő oldalakat, képeket, alkönyvtárakat és egyéb fájlokat tartalmazza – távolról adminisztrálják. A FrontPage Server Extensions emellett azt is lehetővé teszi, hogy a felhasználók a weboldalakon dinamikus funkciókat *(például számlálókat és űrlapkezelőket)* alkalmazzanak. A Server Extensions a szabványos webkiszolgáló bővítő felületeket *(például a CGI-t és az ISAPI-t)* használja, és gyakorlatilag minden elterjedt webkiszolgálóval használható.

- Az Office Server Extensions a FrontPage Server Extensions-re épül, mely képes például a Webhelyszín dokumentumaiban automatikusan karbantartani a hivatkozásokat.

Talán szomorú hír, de tapasztalati tény, hogy az OSE telepítő sajnos nem tud mit kezdeni az SQL Server 2000-rel, és verzióhibákra hivatkozva nem képes elkészíteni az OSE adatbázist. Így sajnos egyelőre marad az MSDE, vagy az SQL Server 6.5 és 7.0 használata.



Ez a bosszantó mellékkörülmény elkerülhető, ha a Server Extensionst az SQL Server telepítése előtt tesszük fel, így nem talál rá a meglévő SQL Serverre, és felteszi az MSDE-t, amit azután egyszerűen frissíthetünk SQL 2000-ré. A másik megoldás az lehet, hogy – tekintettel az SQL Server 2000 többpéldányos telepítési lehetőségére, és arra, hogy akár párhuzamosan is futni képes egy 7.0 verzióval – SQL Server 7-re telepítjük, majd onnan a beépített Data Transformation Services segítségével áthúzzuk a kitöltött adatbázist a 2000-es adatbáziskezelőre.

Játék!

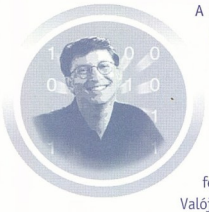
Ilyen még nem volt, de ha sikeresnek ígérkezik, akkor legközelebb is lesz: válaszoljon az alábbi négy kérdésre, és ha szerencséje van, bekerül azon boldog IT guruk körébe, akik már rendelkeznek NetAcademia bögrével. A bögretulajdonosok arról híresek, hogy valahol, valamikor egyszer valami nagyon okosat mondtak, s ezt honoráltuk Magyarország talán leginformatikusabb bögréjével, mely hatalmas befogadóképességének köszönhetően fél literig skálázható.

A kérdések:

1. **Mit jelent az XML leírónyelvben a <bill> tag?**
2. **Miért nem érdemes az Exchange Server 5.5 telepítésénél meghagyni a felkínált Administrator fiókot, mint Site Service Account felhasználót?**
3. **Mi a baj a régi típusú *= illesztő operátorral?**
4. **Mitől sokkal jobb választás a Windows 2000 beépített Kerberos azonosítása mint a hagyományos NTLM akármelyik verziója?**

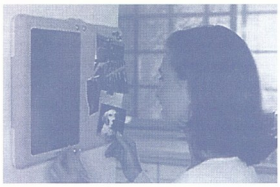
A válaszokat a **http://technet.netacademia.net** oldalon, a novemberi számhoz tartozó lapon várjuk szeretettel a decemberi szám megjelenéséig.

Bill Gates mondja...
Miért nem halnak ki a PC-k?

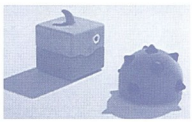


A számítástechnikai eszközök ezerféle alakba fognak megjelenni, de mindig szükség lesz általános célú számítógépekre.

A személyi számítógépek korszakának vége – halljuk évről évre a sajtóban, majd a PC eladások ismét minden várakozást felülmúlnak. Ebben az évben ugyanez történt. Az első negyedév eladási előrejelzései igen kilátástalannak tüntették fel a piacot, azt hihettük: most már tényleg vége. Valójában az egész évre vetített adatok egészséges, 19%-os növekedést mutatnak. Több, mint százmillió PC-t adtak el például 1999-ben, ez azt jelenti, hogy legalább annyi személyi számítógép kel el, mint amennyi színes televízió.



A PC mindenkinek kezébe adja azt a számítási teljesítményt, ami 10 évvel ezelőtt kizárólag nagyvállalatok számára volt elérhető. Azonban az emberek ezt már teljesen természetesnek veszik – és még többet akarnak. A PC szolgáltatásai mindennapjaink részévé váltak, emiatt ezeket a mai ember mindenütt használni szeretné függetlenül attól, hogy éppen hol tartózkodik, vagy milyen eszköz áll rendelkezésére – palmtop, webes telefon, autó PC, vagy webTV – egyre megy. Okos szoftverekkel, erőteljes processzorokkal, drótnélküli kommunikációval rendelkezünk – ez a kívánalom hamarosan teljesíthetővé válik. A legtöbbünk számára nyilván a PC marad az elsődleges munkaeszköz, hisz továbbra is szükségünk lesz nagyméretű kijelzőre és normális billentyűzetre levélíráshoz, webböngészéshez, és ugyanígy szükség lesz a processzorok hatalmas teljesítményére házi digitális fotólaborunk, és a család kedvenc játékprogramjai számára. Azonban a jövő az együttműködő számítógépeké. Eszközeink képessé válnak majd arra, hogy naptárunkat, elektronikus leveleinket, címlistáinkat tökéletesen átvigyük egyik géptípusról a másikra, s még csak vesződnünk sem kell majd a beállításokkal, hisz mindez automatikusan fog megtörténni. Ha meg szeretnénk tudni egy új autó legkedvezőbb beszerzési árát, majd azonnal le szeretnénk ellenőrizni bankszámlánkat, hogy vajon elegendő pénz van-e rajta a vásárláshoz, ezt mind megtehetjük majd azzal az eszközzel, amit mindenhova magunkkal fogunk hordani – nem nevezném e készüléket telefonnak. Akárhol vagyunk, akármilyen információra van szükségünk, hozzá fogunk férni! Ezzel egy időben – ne tévesszük szem elől a trendeket – a PC-k egyre erőteljesebbé, megbízhatóbbá, és egyszerűbben használhatóvá válnak. Történik mindez annak ellenére, hogy a bennük lévő hardver és szoftver egyre bonyolultabbá válik, ez a „bonyodalom" már nem jut el a felhasználóig, nem fogja őt állandóan zaklatni. Hozzánk igazodó felhasználói felületek jelennek meg hangfelismeréssel élő nyelvi vezérléssel.



Természetesen azonnal rendelkezésre fognak állni, ahogy bekapcsoljuk őket *(instant on)*, és nem kell majd perceket várni a bootolásra. A PC válik az otthoni hálózat központjává *(esetleg egy nagyobb kiszolgálóhálózat részeként, mely teljesítményfigyelést, szoftverfrissítést és egyéb nyalánkságokat nyújt automatikusan)* és természetesen mindennél egyszerűbb lesz a kezelése. Azután tanúi lehetünk majd a PC-k alakváltozásának, hisz minden alkatrész egyre kisebbé és kisebbé válik: a Tablet PC alakját például egyértelműen a kijelző, s nem a többi hardvereszköz mérete és formája határozza meg.

Ez a modell létfontosságú szerepet játszik a jövő „mindentmindenhol" számítástechnikájában. A személyi számítógépek alacsony árnagy volumen megközelítése át fog terjedni eme kis masinák piacára is. Az innovációk, újítások ekkor simán kifizetődnek, még jobban, mint a mai kis sorozatú, ám horribilis árú kézi eszközöknél. A széleskörűen elfogadott szabványokon alapuló eszközökről pedig már a vásárlás pillanatában tudni fogja a boldog tulajdonos, hogy masinája együtt fog működni meglévő eszközeivel. A PC-plus világban a kommunikációé a főszerep, így a szabványok követése teljesen természetes és logikus lépéssé válik.

A PC-k teljesen új utat mutattak a világnak abban, hogyan is kell hatékonyan dolgozni, kommunikálni és önfeledten szórakozni. A PC-plus korszak legalább ekkora forradalmat hoz: A PC-k ereje a világ minden zugában elérhetővé válik olyan eszközökkel, amelyekről még csak álmodni sem merünk. Az én pozíciómban talán meglepő hogy így lelkesedem. A Microsoft jövőjét bizony erre a lapra tettem fel!

Microsoft® SQL Server

Microsoft Back Office® Family Member

1994-ben ismerkedtem meg a Microsoft SQL Server-rel, 4.0a volt a verziószáma. Mai szemmel nézve persze döbbenetesen primitív volt, ám már akkor is rendelkezett azokkal a képességekkel, amelyek miatt az ember örömmel dobta el a Clipper-t: tranzakciókezelés, ügyfél-kiszolgáló felépítés, többszálú végrehajtás stb. Félelmetesen összetett banki rendszereket készítettünk akkoriban 486DX4-100 processzoros, 16 megabájt memóriával felvértezett „szervereken". Az elmúlt évek során az SQL Server megfiatalodott: fürge lett és erős. S ahogy verzióról verzióra fiatalodott, úgy vált egyre okosabbá is. Zseniális lekérdezés optimalizálója segítségével szakavatott kezekben szárnyakat kap. A TPC-C teljesítménytesztek világbajnoka rendszerfelügyeleti szempontból is kiváló termék. A 7.0 már lehetővé tette az adatbázis által összegyűjtött adatok okos elemzését, hisz megjelent benne az OLAP kiszolgáló. S ami napjainkban lázba hoz, az a mesterséges intelligencia határmezsgyéjén található adatbányászat. Az adatok rejtett összefüggéseinek feltárása minden adatbázis tartalmát életre kelti. A december elején a NetAcademia Kft, a BST Kft. és a Microsoft Magyarország közös szervezésében megrendezésre kerülő

SQL Server Napon

Megpróbálok valamennyit átadni azokból a tapasztalatokból, melyek az évek során „rám rakódtak". Természetesen egyetlen nap nem lehet elegendő a teljes információhalmaz átadására. Mindenkit szeretettel várok SQL Server 2000 tanfolyamaimon, melynek helyszíne a Business Software Training Hivatalos Microsoft Oktatóközpont (CTEC).

Jelentkezés a konferenciára és részletes tudnivalók a
http://www.netacademia.net/SQLnap
vagy
http://www.bst.hu/SQLnap
címen!

Az SQL Server nap rövid tematikája a következő:

- ***Adatbázisok a rendszergazda hozzáértő kezében.*** *Azaz mi mindent tehetünk a teljesítmény fokozása érdekében még akkor is, ha készen vett alkalmazással van dolgunk?*
- ***Az SQL 2000 újdonságai programozók számára.*** *Ismerkedjünk meg az SQL 2000 fejlesztőknek szóló fantasztikus újdonságaival: XML támogatás, felhasználói függvények, indexelhető nézetek, instead-of triggerek, Kaszkád referenciális integritás*
- ***Adatbányászat A mesterséges intelligencia*** *határmezsgyéjén húzódó technológia az adatok között megbúvó rejtett összefüggések feltárására.*
- ***Online Analitical Processing.*** *Adatelemzés mesterfokon*

BST Business Software Training

Hivatalos SQL 2000 Server vizsgaelőkészítő tanfolyamok a NetAcademia Kft. és a Business Software Training Hivatalos Microsoft Oktatóközpont (CTEC) közös szervezésében!

2072 Administering a Microsoft SQL Server Database, 5 nap. Időpontok: 2000 november 20-24, 2001 január 22-26

2073 Programming a Microsoft SQL Server Database, 5 napIdőpontok: 2000 november 6-10, 2001 február 5-9

Jelentkezés:
www.netacademia.net/tanfolyamok illetve **www.bst.hu**

www.netacademia.net

A legjobbakat tanítjuk.

1105 Budapest, Ihász utca 13. • Tel.:263-2732

NETACADEMIA

2000.11.

ISSN 1586-5185
11
9 771586 518005