tech.net

working with windows

Anytime, anywhere computing

IV. / 02. szám
1344 Ft

ISSN 15865185 02
9 771586 518005

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

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

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

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

Nyomdai előkészítés:
Ars Luna Bt.
Vezető: Dobák Ildikó

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

ISSN 1586-5185

Felelőtlen felelősök

Slimmer: a világ legveszélyesebb férge...

Újabb mérföldkőhöz érkeztek az SQL Server *(+MSDE)* eladások. Az üzembehelyezett rendszerek száma meghaladta azt a bűvös számot, ami felett már érdemes akár réges-régen befoltozott biztonsági hibákra is kódot építeni. E havi bevezetőmben a felelősöket és a felelőtleneket egyaránt megnevezem!

Paradox módon az anyósomtól értesültem a féreg felbukkanásáról. Nem azért, mert ő akkora számítógépguru, hanem mert alaposan figyeli a sajtót, ami rólam korántsem mondható el. *(Ha mindenki annyit nézné a tévét, mint én, a nézettségi index konstans nullát mutatna – Nagy Testvér ide vagy oda.)* Az eset hétfő reggel történt. Izgatottan ütöttem laptopomat *(a Smart Card bejelentkezési cikket írtam éppen)*, amikor anyósom közölte velem, hogy féreg támadta meg az SQL Servert. Ezeket a szavakat még semmilyen szóösszetételben nem hallottam tőle, ezért mindjárt gyanakodni kezdtem, hogy humbug az egész. És nem.

2003 január 26-án valóban leállt a szupersztráda. Denial of Service támadásnak hívják, amikor a rendszerek – túlterhelés vagy egyéb akadályoztatás következtében – képtelenné válnak feladatuk ellátására. Pontosan ez történt. Most megtudtuk, mire képes egy alig 400 bájtos kis bigyó. Ráadásul – bár lehetősége lett volna rá – fizikai kárt nem is okozott. *(Feltehetőleg azért nem, mert az a része még nem készült el, de már kiszabadult.)*

Buffer Overrun *(BO)*

Mi lehetővé teszi, hogy illetéktelenek távolról tetszőleges kódot futtassanak gépünkön? Egy programozói hiba. Az elmúlt hónapban azon ügyeskedtünk *(SQL Injection)*, hogy kódot tároljunk adatként. A BO ennek fordítottja: adatot futtatunk kódként. Hogyan?

Vegyünk egy kódrészletet egy nemlétező, bár Windowsos programnyelven. *(Már programnyelveket is készítek! Felhívnám a figyelmet a kommentkarakterre. Igazán programozóbarát, nemde?)*

```
msgbox negyzet(3)  ▦▦▦ függvényhívás

function negyzet (a)
    semmi string[22]  ▦▦▦ lokális változó
    negyzet=a*a
endfu
```

A függvény meghívásakor a veremre *(stack)* kerül az átadott változók címe, majd a visszatérési cím *(és a stack pointer ott is marad)*. Ezt követ*(het)*ik a lokális változók, szintén a veremben. *(Ezeket a bázispointerrel szokták matatni, hogy a stackpointer mindvégig a visszatérési címre mutasson, így akármikor ki lehet szállni a függvényből.)* Mindezek az adatok „fejjel lefelé", denevérperspektívában kerülnek a verembe, mert a veremkezelő utasítások *(Push, Pop)* így látják a világot. De van itt két bökkenő:

1. A tárolás helyétől és sorrendjétől függetlenül a változókat „normális" irányban töltjük meg adattal
2. Ha nem figyelünk, egy túltöltött változó simán felülírhatja a veremben felette található mentett adatokat.

Ennyi. Ha „megfelelő" hosszúságú és kialakítású adatot tömünk a „semmi" nevű változóba, az szépen felül fogja írni nemcsak az összes lokális változót, hanem a felette lévő visszatérési címet is. Ha a címet „megfelelően" ütjük felül, a függvény végén a RET utasítás nem a hívóhoz repít vissza, hanem oda, ahova akarjuk – tipikusan a stackre, hisz oda tudunk kódot írni.

Itt jegyzem meg, hogy menedzselt kód esetén mindez nem történhet meg. Hajrá .NET!

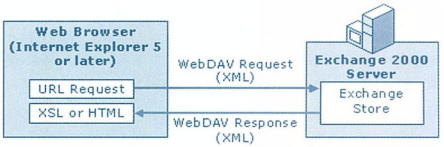
Felelősök

1. Felelős a Microsoft, mert nem két-három évvel korábban hirdette meg a Trustworthy Computingot. Ez a pár év késés pontosan elegendő volt ahhoz, hogy olyan örökség halmozódjon fel, ami évekre ellátja a Tisztelt Fogyasztókat megfelelő számú és mennyiségű buggal. Most már hiába kapálódzik, a múltat eltörölni nem lehet, a régi vevők fele sem telepíti a javításokat.
2. Felelős mindenki, aki a maga pucér mivoltában SQL Servert helyez a netre. Ez az eljárás menthetetlenül hibás. Emlékszem egy esetre *(pár éve történt)*, amikor az egyik hazai vállalat kérésére távolról megnéztem, mennyire biztonságos a rendszerük, és találtam egy saval, üres jelszóval nyíló SQL Servert. Gyorsan csináltam rá egy „A" nevű adatbázist *(CREATE DATABASE A)*, ami a mai napig ott van, mert senki nem tudja, hogy került oda, és mi célt szolgál.
3. Felelős az is, aki Buffer Overrunnal játszik. Valószínűleg az történt, hogy miután a Microsoft tavaly nyilvánosságra hozta a hibát, lelkes amatőrök megpróbálták kihasználni a lehetőséget. A kód félig sem készült el: egyedül terjedni tud. Még az is lehetséges, hogy a legelső működő példány szabadult el, a fejlesztők nem kis bánatára. Nem véletlen az időzítés sem: január vége az egyetemistáknál még a téli szünet része. Akár én is csinálhattam volna, mert a Security Workshop részeként be szoktuk mutatni a Buffer Overrun működését, igaz nem SQL Serveren, hanem RRAS-on, és nem féreggel, hanem egy CMD.EXE elindításával – ami azután nem csinál semmit. Sem rosszat, sem jót.

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

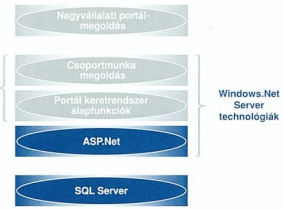
2003. 01. szám

Haladó WebDav-ismeretek



Jelen cikkben az Exchange 2000 adatbázisának alacsony szintű elérését mutatom be. A cikk első részében egy rövid betekintést kínálok azoknak, akik vagy újak az Exchange szerteágazó világában, vagy még semmiféle kapcsolatot nem alakítottak ki vele. Itt lesz szó az ADO használatáról, majd a cikk második *(hosszabb)* részében bővebben kitérünk a WebDav protokollra, amely az adatbázis elérésének leguniverzálisabb és talán legbonyolultabb módja.

4.oldal



MEC 2002

Konferenciabeszámoló

A MEC korábban a Microsoft Exchange Conference kifejezés rövidítése volt, amely később átalakult Microsoft Exchange and Collaboration-re. Azonban idén már nem vesződtek a bejáratott MEC megnevezés blikkfangos értelmezésével, a konferencia hivatalos leírása ez volt: „The essential Microsoft conference for planning, deploying and managing a connected infrastructure".

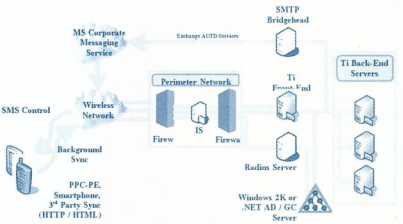
10.oldal

Microsoft Exchange Server 2003

A projekt, amit valaha Titaniumnak hívtak

Hamarosan elkészül az Exchange következő változata, tele számos újdonsággal. Outlook Cached Mode, RPC over HTTP, újraírt Outlook Web Access, integrált mobilszolgáltatások hogy csak néhányat említsünk. Valóban van még új a nap alatt? Tudósítónk jelenti Redmondból.

13. oldal

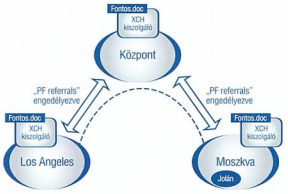


Microsoft Exchange 2000

Nyilvános mappák replikációja

Kis szünet után folytatjuk az Exchange 2000-ről szóló sorozatot. Ebben a részben a nyilvános mappák replikációjával ismerkedhet meg a kedves olvasó. Megvizsgáljuk, mi áll a nyilvános mappák replikációjának hátterében, részletesen foglalkozunk a replikáció folyamatával, a konfliktuskezeléssel, valamint a hibaelhárítással is.

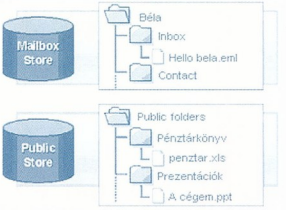
17. oldal



Az Exchange WSS elérése az SPS keresőmotorjával

Az Exchange 2000 és a SharePoint Portal Server külön-külön is rengeteg előnnyel és funkcióval rendelkezik. Ebben a cikkben azt mutatom be, hogy milyen lehetőségek tárulnak elénk, ha a két szervert „összeházasítjuk".

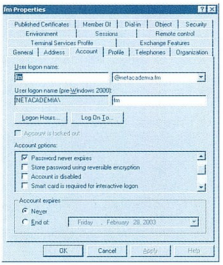
23. oldal



Változó dimenziók kezelése

A vállalati adatraktárak időben általában stabilak, azaz többnyire csak új adatokkal bővülnek; a tények elemzésére szolgáló dimenziók azonban idővel megváltozhatnak. Az eladások adatait elemezhetjük például a „Kereskedők" dimenzió szerint. A cég kereskedőinek adatai azonban személyes okokból, vagy szervezeti átalakítás következtében megváltozhatnak. Az ilyen dimenziót szokás „lassan változó dimenziónak" nevezni.

27. oldal

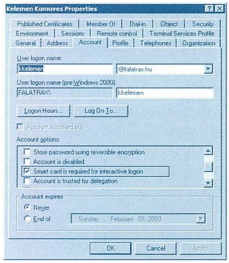


Feltörhetetlen bejelentkezés?

Challenge/Response, Kerberos és Smart Card logon

A szoftveripar nagyjai már sokszor megígérték nekünk, hogy egyszer s mindenkorra véget vetnek a bejelentkezési *(autentikációs)* adatok lopkodásának. Ismerjük a LanMan/NTLM páros szomorú sorsát *(L0PHTCrack)*, s most már az utóbbi években folyamatosan magasztalt Kerberost is utolérte a sorsa *(Kerbcrack)*! Egyetlen megoldási lehetőségnek a jelszavas bejelentkezés kiiktatása, elfelejtése látszik. Az intelligens kártyás bejelentkezés az új ígéret. Ez azonban – elődjeitől eltérően – be fogja váltani a hozzáfűzött reményeket, mert…

28. oldal



Smart Card logon

Lépésről lépésre

Najaink legmegbízhatóbb felhasználóazonosítási módszere az intelligens kártyás bejelentkezés. Ha a megfelelő hardvereszközök rendelkezésre állnak, neki is láthatunk a rendszer bevezetésének. Szükségünk lesz Certificate Serverre, egy Smard Card-író „műhelyre", csoportos házirendek állítgatására és még sok minden másra. Ha bármelyik lépést kihagyjuk, egzotikus hibaüzenetekben gyönyörködhetünk. Ez a cikk a Smart Card logon elkészítésének „szakácskönyve"!

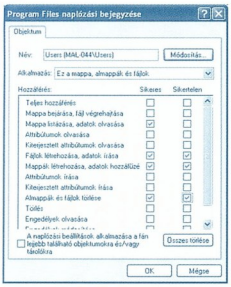
31. oldal

Identitáskezelés

Hivatalos Microsoft-tanulmány

Az összes munkatársra kiterjedő identitások kezelése egy vállalaton belül mindig nagy kihívást jelentett. Napjainkban, az Internet és az elektronikus kereskedelem elterjedésével ez a kihívás megnőtt, mert a vállalatok és kormányzatok partnereik és ügyfeleik számára is kénytelenek hozzáférést biztosítani belső rendszereikhez és alkalmazásaikhoz.

34. oldal



A Microsoft operációs rendszerek biztonsági tényezői

II. rész – Biztonságos hálózati működés

Az előző cikkben eljutottunk a biztonságos vállalati hálózathasználat alapjaiig. Egy tudatosan biztonságosra tervezett operációs rendszert ismertünk meg az NT „személyében". Sőt! Van címtárunk és néhány jól tervezett hitelesítési protokollunk, öröklődő jogosultságaink és stabil fájlrendszerünk is. Mi kellhet még? PKI, a folyamatok követése *(naplózás)* és adatfolyamtitkosítás.

37. oldal

Dupla KU

42. oldal

Haladó WebDav-ismeretek

Jelen cikkben az Exchange 2000 adatbázisának alacsony szintű elérését mutatom be. A cikk első részében egy rövid betekintést kínálok azoknak, akik vagy újak az Exchange szerteágazó világában, vagy még semmiféle kapcsolatot nem alakítottak ki vele. Itt lesz szó az ADO használatáról, majd a cikk második *(hosszabb)* részében bővebben kitérünk a WebDav protokollra, amely az adatbázis elérésének leguniverzálisabb és talán legbonyolultabb módja.

Az Exchange 2000 bemutatása

Az informatikai világ két nagy pártra szakad: struktúrált és nem struktúrált adattárolást helyeslő pártokra, amelyek soha nem fogják megérteni, elfogadni egymás nézeteit az adatok tárolását illetően. Míg az SQL Server struktúrált adattárolást és kezelést tesz lehetővé, az Exchange Server a nem struktúrált párt nézeteit képviseli. Egyszerűbben kifejezve, míg egy SQL fejlesztő az ügyfél követelményeinek hallatán rögtön a CREATE TABLE parancs paramétereit és a szükséges kapcsolatok kapcsolótábláit látja maga előtt, Exchange-es kollégája másképp *(nem biztos hogy jobban)* közelíti meg a problémát.

Többen az Exchange Servert mint levelező szolgáltatásokat nyújtó alkalmazást ismerik. Ezt az Exchange némileg kibővíti egy „Exchange, mint alkalmazás platform" szolgáltatással. Ez annyit jelent, hogy az Exchange támogatja a nyilvános mappáiban és postafiókjaiban adatokat manipuláló alkalmazások futtatását. Szinte minden SQL fogalomnak megvan az Exchange-es megfelelője, tehát ebből a szempontból adatbáziskezelőnek tekinthetjük az Exchange Servert is.

Ahogy az előbbiekből kiderült, az Exchange 2000 két különböző típusú adatbázist foglal magába: postafiókok tárolását és nyilvános mappa funkcionalitást nyújtó adatbázisokat. Ez utóbbi közös munkaterületet jelent a cég dolgozóinak, míg az előbbi minden felhasználó saját adattárát.

Az Adatbázis szerkezete

Az Exchange 2000 és 2003 *(a következő verziójú Exchange, Titanium kódnévre hallgat, amely pillanatnyilag Beta állapotban áll)* adattárolását a Web Storage System nevű szolgáltatás oldja meg, amely szolgáltatást a SharePoint Portal Server szintén, ugyanebből a célból használja.

Hogy miért WEB Storage System a neve? A jól csengő névben szereplő Web szót távoli adatelérési protokolljának köszönheti, amely HTTP kérésekbe burkolt XML struktúra, a neve pedig WebDAV. A rendszergazdák örömére a tűzfalon belül elhelyezett Exchange eléréséhez elegendő a HTTP port engedélyezése, és a hálózaton kívülre irányuló kérések esetében sem szükséges külön odafigyelés, mivel minden kommunikáció HTTP-n keresztül zajlik.

A Web Storage System *(továbbiakban WebStore)* nem struktúrált adatok tárolására és visszakeresésére alkalmas adatbáziskezelő, amely – integrálva az MSSearch indexelő szolgáltatással – támogatja a teljesszöveges *(fulltext)* visszakereséseket. A WebStore architektúráját és szolgáltatásait figyelembe véve főként dokumentumok és az azokhoz tartozó kísérőinformációk tárolására alkalmas, azaz a középpontban a dokumentum áll *(megoldva a WebStore-on futó alkalmazások dokumentumainak tárolását)*, amelyhez leíróinformációk tartoznak.

Mivel a WebStore HTTP alapú, logikusnak tűnhet, hogy a WebStore-ban lévő minden elem *(SQL-es világban rekord)* külön URL-lel rendelkezik. Az URL a szerver, azon belül az adatbázis, azon belül a mappastruktúra, majd az elem nevéből adódik. Így a myexchange szerver public nevű adatbázisában lévő MyFolder-ben elhelyezkedő MyItem.doc elem URL-je http://myexchange/public/MyFolder/MyItem.doc. Ez így teljesen logikus, nem?

Exchange 2000 adatelérési lehetőségek

Visszatérve az Exchange-re: a hosszú évek során sokféle Exchange-adatelérés készült, némelyek közben nevet is változtattak, így annak, aki ma Exchange-fejlesztésre adja a fejét, komoly kihívást jelent a különböző feladatokhoz használatos adatelérési technikák megkülönböztetése és kiválasztása. Ez a cikk nem nyújt effajta iránymutatást más adatelérési technikák használatához, csupán a WebDAV-val foglalkozik, amely eddig a legújabb Exchange adatelérési lehetőség, és használatával majdnem minden megoldható.

A következő lista az Exchange 2000 jelenlegi lehetséges adatelérési változatait mutatja be:

- **MAPI *(Messaging API)*:** alacsonyszintű, kliens-szerver architektúrájú, RPC-n történő kommunikációt végző, eredetileg levelezési szolgáltatások elérésére készített protokoll.
- **ADO MSDAIPP *(Microsoft OLE DB Provider for Internet Publishing)* provider:** WebDAV-ra épülő ADO provider, amelynek használatával a WebDAV kérések összeállítása elkerülhető
- **WebDAV *(Web Distributed Authoring and Versioning)*:** HTTP-n kommunikáló adatelérési szabvány, amely a szerveroldalon futó IIS-sel egy ISAPI komponensen keresztül kommunikál.
- **ADO ExOLEDB *(Exchange OLE DB)* provider:** Csak az Exchange szerveren használható OLE DB provider, amely a WebStore közvetlen elérését teszi lehetővé *(ezáltal megspórolva pl. a HTTP kéréseket)*. Az egyik leggyorsabb adatelérési lehetőség.
- **Microsoft Outlook Object Library:** MAPI-ra épülő komponens, amely magasszintű adatelérést tesz lehetővé *(előnyei: egyszerűbb naptár- és feladatelem kezelés, stb.)*
- **Microsoft CDO *(Collaboration Data Objects)* 1.21 Library:** A MAPI magasabb szintű verziója, amely ADO és ADSI használatával éri el az Exchange Servert.
- **ADSI *(Active Directory® Service Interfaces)*:** Active Directory-specifikus címtárelérési komponens.
- **LDAP *(Lightweight Directory Access Protocol)*:** Szab-

ványos, TCP/IP protokoll felett működő általános címtárelérési protokoll.

- **CDOEX** ***(CDO for Exchange 2000 Server)***: Magasszintű adatbáziselérést biztosító komponens, leginkább levélküldésre használatos.
- **CDOEXM** ***(CDO for Exchange Management Objects)***: Exchange 2000 adminisztráció programozott eszközökkel történő elérését támogató komponens.

MSDAIPP

Az MSDAIPP a WebDAV felett elhelyezkedő adatelérési réteg, amely némileg leegyszerűsíti az adatok elérését. Hátránya, hogy ezidáig csak ADO providerként elérhető, az ADO.NET-ben még nem ismert fogalom. Így ha .NET FrameWorkkel dolgozunk, „wrappelnünk" kell, vagy el kell felejtenünk, és WebDAV-ot kell használnunk. Az ADO-s elérést ezért VB6-os példán keresztül mutatom be. Aki WebStore fejlesztésre vállalkozik, elég gyakran összetalálkozhat COM-os kihívásokkal.
Az MSDAIPP provider nagyban hozzájárul ahhoz, hogy a WebStore-t adatbáziskezelőnek tekintsük: az ADO architektúrájának köszönhetően a keresések futtatása pl. szinte teljesen ugyanúgy történik, mint SQL Server esetében, ráadásul a WebStore egy Transact-SQL nyelvhez hasonló lekérdező nyelv használatát is támogatja.
Ahogy SQL Server esetében is tesszük, elsőként kapcsolatot kell teremtenünk a szerverrel. Itt meg kell adnunk a használt provider nevét, valamint azon mappa URL-jét, amelyhez a kapcsolatot kötjük. A kapcsolaton keresztül történő műveletek ezen mappa hatáskörén belül kell hogy történjenek *(a mappában és – rekurzívan – annak almappáiban)*.

```
Dim objConnection As ADODB.Connection
Set objConnection = New ADODB.Connection

With objConnection
   .Provider = "msdaipp.dso"
   .Open "http://xch2k01/public/NagyonTitkosIratok"
End With
```

Ezzel meg is van a kapcsolatunk. Ha hiba történik kapcsolódáskor, az Outlook Web Accessnek köszönhetően lehetőségünk adódik az URL gyors ellenőrzésére, amelyet beírva a böngészőbe megjelenik az adott mappa tartalma. Ha nem jelent meg, akkor vagy nem létezik *(vagy áll a szerver)*, vagy az Outlook Web Access *(OWA)* nincs megfelelően belőve. A következő lépés egy lekérdezés futtatása, amely visszaad minden olyan elemet a mappánkból, amelynek az IsNagyonTitkos mezőjének értéke True. A kódunkat a következőképpen folytassuk:

```
Dim objRecordSet As ADODB.RecordSet
Dim strQuery As String

'SQL query összeállítása.
strQuery = "SELECT ""DAV:href"" " & _
    "FROM
""http://xch2k01/public/NagyonTitkosIratok"" " & _
    "WHERE ""IsNagyonTitkos"" = true"

'Query futtatása
Set objRecordSet = objConnection.Execute(strQuery)
```

Ha minden jól ment, az objRecordSet objektumban megkaptuk a lekérdezés eredményét, amelyet a szokásos módon bejárva feldolgozhatjuk a tartalmát. Az strQuery változóban összeállított SELECT parancs nem néz ki túl fényesen, de hatékony azáltal, hogy a dupla idézőjelek alkalmazásával idézőjelet írhatunk a sztringbe.

Rövid úton szerzett WebStore tudásunkkal megállapíthatjuk, hogy a WebStore-ban mappa néven emlegetett fogalom megegyezik az SQL-es tábla fogalmával. A WebStore „tábla" fogalma viszont bizonyos szempontból többet is nyújt az SQL-es táblánál, ugyanis lehetőségünk van a mappaszerkezet hierarchikus felépítésére, amely struktúrán rekurzív lekérdezéseket is készíthetünk. Sajnos ez a feature csak nem-MAPI tár esetén áll rendelkezésünkre, azaz se a nyilvános mappákat, se a levelesládákat tartalmazó adatbázisban nem futtathatunk rekurzív lekérdezéseket *(lehetőség adódik további adatbázisok létrehozására, amelyek nem látszanak az Outlook-ban, ezeket non-MAPI store-oknak nevezzük)*. A következő kódrészlet egy nem-MAPI adatbázison végzett rekurzív keresés, amely ugyanabban a mappában, ugyanazzal a kapcsolattal működik, csak nem „SHALLOW" *(sekély)* módon, hanem rekurzívan:

```
...
'Deep SQL query összeállítása.
strQuery = "SELECT ""DAV:href"" " & _
   "FROM SCOPE('DEEP TRAVERSAL OF " & _
   """http://xch2k01/public/NagyonTitkosIratok""')
   " & _
   "WHERE ""IsNagyonTitkos"" = true"

'Deep query futtatása
Set objRecordSet = objConnection.Execute(strQuery)
```

Itt a „DEEP TRAVERSAL OF" kapcsoló a különleges, amely implicit módon az előző keresésnél is jelen volt, csak a DEEP szót ott a SHALLOW helyettesítette, amely csak a megadott mappában tesz lehetővé keresést. Ezen kívül létezik még a HIERARCHICAL traversal, amely az adott mappa alatti almappákat adja vissza, elemekkel nem foglalkozik.
Biztosan feltűnt, hogy az előbbi lekérdezésekben a DAV:href mezőt kérdeztük le a RecordSetünkbe. Ez a mező az elemek URL-jét tartalmazza, amely alapján egyértelműen megnyitható az adott rekord. A következő kódsnippet megnyitja a DAV:href mezőben megadott rekordot, módosítja egy mezőjét, majd elmenti a változtatást.

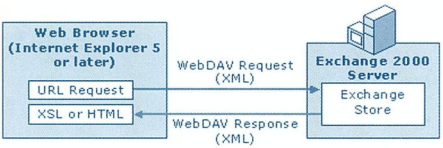
```
Dim objRecord As ADODB.Record

Set objRecord = New ADODB.Record
With objRecord
   .Open objRecordSet.Fields("DAV:href").Value, _
     objConnection, adModeReadWrite
   .Fields("IsNagyonTitkos").Value = False
   .Update
End With
```

Sajnos a WebStore SQL nem teszi lehetővé UPDATE és INSERT használatát, az UPDATE-et az előző példa váltja ki, míg az INSERT-et ugyanitt az Open metódushívás megtoldva egy adCreateNonCollection paraméterrel.

WebDAV

A WebDAV kérések összeállítása a WebStore távoli elérésének legalacsonyabb szintű, és emiatt legtöbb funkciót nyújtó lehetősége. Legfőbb jellemzője a platformfüggetlenség, ugyanis a WebDAV nem más, mint a szaványos HTTP protokoll kiegészítése. A teljes párbeszéd az IIS-en keresztül zajlik. Definícióját az RFC 2518-as számú szabvány tartalmazza *(amely a HTTP 1.1 szabvány kiegészítése alapvető dokumentumkezelő funkciókkal)*.
Még itt az elején egy rövid kitérőt teszek, hogy felhívjam a kedves olvasó figyelmét egy tipikus WebDav-Outlook tervezési kérdésre: a Montana MonFlow nevű Exchange alapú folyamatkezelő terméke majd' teljes funkcionalitását WebDav-ra ala-

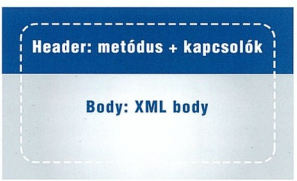
pozva működik. A termék tervezésének kezdetekor nagy kérdés volt a jogosultsági modell kérdése, ugyanis az alkalmazás felhasználói felülete az Outlook-ot kiegészítő COM Add-In-ként kerül megvalósításra, így a felhasználó szabadon választhat a Control Panelen beállított Outlook profiljai közül, míg a WebDav esetében alapértelmezésben a Windows Authentication a nyerő, így a felhasználó két különböző kontextusban szólongathatja a szervert. Az ilyen kérdéseket jó előre látni.



A WebDAV kérés címzettje az IIS, amelyen egy ISAPI DLL végzi a feldolgozást. A kérés lényegi része ezután a WebStore-hoz kerül, aki elvégzi a kérésben megfogalmazott műveletet, majd ad egy választ. A válasz egy másik XML, amely a kért adatok mellett státuszkódot is ad vissza, amelyben információt kapunk a kérésünk eredményéről. Vegyünk nagy levegőt, és bukjunk alá!

```
PROPFIND /public/NagyonTitkosIratok/
 SzigoruanTitkos.doc HTTP/1.1
Host: xch2k01
Content-Type: text/xml
Content-Length: xxx
Depth: 0

<?xml version="1.0"?>
<a:propfind xmlns:a="DAV:">
  <a:prop><IsNagyonTitkos/></a:prop>
</a:propfind>
```

A fenti csúnyaság egy egyszerű WebDAV kérés, amely egy WebStore elem két tulajdonságának *(mezőjének)* értékét kérdezi. Mivel a WebDAV platformfüggetlen, a kérést összeállíthatjuk akár a .NET Framework használatával is, amelyre azon belül sokféle lehetőség adódik: használhatjuk például az XML DOM-ot, valamint a StringBuilder, StreamBuilder, WebRequest és WebResponse osztályokat. Kihasználva a WebDAV platformfüggetlenségét, a kérés összeállítását és a válasz értelmezését megvalósító kódot és tippeket kihagyom, csak magára az adatra összpontosítok.
Elemezzük az előbbi WebDAV kérést: láthatóan két részből áll. A felső a HTTP header, amely szintén két részre bomlik: a legelső sora a metódus:




```
PROPFIND /public/docs/myFile.doc HTTP/1.1
```

Ezek szerint ez a kérés a PROPFIND metódust szeretné futtatni, mégpedig a /public/docs/myFile.doc elemen *(a WebStore világában a dokumentumokat és mappákat egységesen elemeknek nevezzük)*. A HTTP header következő része a kérésben szereplő XML adatra és a metódusra vonatkozó különböző információkat tartalmazza *(pl. az adat típusa és hossza)*. Nevezzük ezeket a header-részeket kapcsolóknak:

```
Host: xch2k01
Content-Type: text/xml
Content-Length: xxx
```

A fejlécből *(headerből)* hátravan még a Depth kapcsoló, amely ha nulla értékű, jelzi a store-nak, hogy csak az adott elem tulajdonságaira kíváncsi, al-elemre *(mert ilyen is van)* nem vonatkozik a kérés:

```
Depth: 0
```

A Depth egy általában minden kérésnél használható kapcsoló, amely az adott metódusra vonatkozóan beállítja a művelet „mélységét" *(hatósugarát)*. A Depth kapcsolóhoz hasonlóan minden WebDAV metódusnak megvannak a maga kapcsolói, ilyen a COPY *(másolás)* esetén az Overwrite *(felülírja vagy sem)*, LOCK *(tranzakcióindítás vagy elem zárolása)* metódus esetén a Timeout kapcsoló *(amely megmondja, hogy mennyi ideig éljen a tranzakció/zárolás)*, stb.
A kérés második része egy XML dokumentum *(Body)*, amely a kéréshez tartozó adatot, jelen esetben a lekérdezendő tulajdonságok neveit tartalmazza:

```
<?xml version="1.0"?>
<a:propfind xmlns:a="DAV:">
  <a:prop><IsNagyonTitkos/></a:prop>
</a:propfind>
```

Minden metódusnak megvan a saját XML body formátuma. A fenti például a PROPFIND metódusé, amely az IsNagyonTitkos tulajdonság értékét kérdezi le. Bizonyos metódusokhoz *(például MOVE, COPY)* nem szükséges body küldése, az eljárás paraméterei néhány kapcsoló használatával a fejlécben beállíthatók. Példaként XML nélküli változatra, a COPY metódust futtató kérés a következőképpen néz ki:

```
COPY /public/NagyonTitkosIratok/
↳ SzigoruanTitkos.doc HTTP/1.1
Host: xch2k01
Destination: /public/EgyaltalanNemTitkosIratok/
↳ SzigoruanTitkos.doc
Overwrite: T
```

A kérést postázva kisvártatva megkapjuk a választ, amely felépítését tekintve ugyanolyan formátumú, mint a kérésünk. A válasz feldolgozása először a státuszkód *(lásd alább)* vizsgálatával kezdődik, majd megfelelő státuszkód esetén az XML értelmezésével és annak bejárásával *(végigiterálásával)* végződik.
Az előbbi PROPFIND metódusunkra a következő választ kapjuk a WebStore-tól:

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml
Content-Length: xxx

<?xml version="1.0" ?>
<a:multistatus xmlns:b="urn:uuid:c2f41010-65b3-
↳ 11d1-a29f-00aa00c14882/"
↳ xmlns:c="xml:" xmlns:a="DAV:">
  <a:response>
    <a:href>
      http://xch2k01/public/NagyonTitkosIratok/
      ↳ SzigoruanTitkos.doc
    </a:href>
    <a:propstat>
      <a:status>HTTP/1.1 200 OK</a:status>
        <a:prop>
```

```
<IsNagyonTitkos b:dt="boolean">1</IsNagyonTitkos>
        </a:prop>
      </a:propstat>
  </a:response>
</a:multistatus>
```

Látható, hogy van egy fejlécrészünk, amelyben státuszkód olvasható, vannak kapcsolóink, valamint egy XML dokumentumunk, amelynek a belsejében rejtőzik a lényeg, az IsNagyonTitkos mező értéke, láthatóan jelezve, hogy boolean típusú adatról van szó. Ez így első ránézésre elég ijesztő lehet, de ijedtségre semmi ok, a következőkben egy kicsit elmazsolázgatunk a részleteken.

Összegezve az előbbieket, a WebDav világában a következő fontos fogalmakkal kell tisztában lennünk:

- Metódus
- Fejléc-kapcsoló
- XML struktúrák

WebDAV visszatérési értékek

Más nyelvekhez hasonlóan a WebDAV-ban is szükségünk van metódushívásaink eredményének ismeretére: vajon sikerült-e a másolás, vagy miért nem adott vissza értékeket a PROPFIND metódusom, stb. A WebDAV-ban ez is a HTTP-hez hasonlóan működik, itt a visszatérési értéket Status Code-nak nevezzük. Persze minden metódusnak saját, rá jellemző visszatérési értékkészlete van. A válaszon belül is létezhetnek különböző státuszkódok. Ha például a PROPFIND metódust több mező egyszerre történő lekérdezéséhez használjuk, minden egyes mezőhöz kapunk státuszkódot, így bizonyos mezőkről eldönthetjük hogy azok nem léteznek, anélkül hogy az egész lekérdezésünk elszállt volna.

A státuszkódokat a válaszban kapjuk meg. Válaszonként két különböző szintű státuszkódot különböztethetünk meg: a kérés egészére vonatkozót, valamint az eredmény különböző részeihez tartozókat. Azoknál a metódusoknál, amelyekhez nem tartozik XML body, csak a fejlécben szerepel státuszkód *(máshol nem is lehetne)*. Ha tehát a PROPFIND kérésünkre kapott válasz státuszkódja 207 *(azaz MultiStatus)*, biztosak lehetünk abban, hogy maga a lekérdezés sikeres volt. Ha a válaszon belüli XML-ben a mezőértékeket tartalmazó node-ok valamelyike a 404-es, jól ismert „Not Found" státuszt viseli, az azt jelzi, hogy a kért mező nem létezik az adott elemnél, de ettől függetlenül a kérésünk lefutott.

A státuszkódok három számjegyű számok, amelyek százas számjegye a kód kategóriájára utal; a 2-vel kezdődő kódok *(200-207-ig)* például minden esetben sikert jeleznek, míg a 4-gyel kezdődők kliensoldali *(pl. nem létező URL esetén)*, az 5-össel kezdődőek szerveroldali hibát jeleznek, stb. A következő válasz egy nemlétező URL-re mutató kérés eredménye:

```
HTTP/1.1 404 Not Found
Content-Type: text/html
Content-Length: xxx
```

Íme egy példa egy olyan lekérdezésre, amely sikeresen lefutott, de a két lekérdezett mezőből az egyik nem létezik:

```
HTTP/1.1 207: Multi-Status
Content-Type: text/html
Content-Length: xxx

<?xml version="1.0"?>
<a:multistatus xmlns:b="urn:uuid:c2f41010-65b3-
↳ 11d1-a29f-00aa00c14882/"
↳ xmlns:c="xml:" xmlns:a="DAV:">
  <a:response>
    <a:href>
      http://xch2k01/public/NagyonTitkosIratok/
      ↳ SzigoruanTitkos.doc
    </a:href>
    <a:propstat>
      <a:status>HTTP/1.1 200 OK</a:status>
      <a:prop>
        <IsNagyonTitkos b:dt="boolean">1
        </IsNagyonTitkos>
      </a:prop>
    </a:propstat>
    <a:propstat>
      <a:status>
        HTTP/1.1 404 Resource Not Found
      </a:status>
      <a:prop><MennyireTitkos/></a:prop>
    </a:propstat>
  </a:response>
</a:multistatus>
```

A válaszból látható, hogy minden DAV:propstat *(a példában a:propstat: hogy miért, arról később szó lesz)* XML tag-hez tartozik egy DAV: status tag, amely a mezőérték lekérdezésének sikerességét mutatja. A WebDAV ezen felül intelligens is, mert a válasz XML-részében az egy státusz alá tartozó eredményeket csoportosítva, ezüsttálcán teszi elénk:

```
...
<a:response>
  <a:href>http://szabodnet/public/Test/asd.EML
  </a:href>
  <a:propstat>
    <a:status>HTTP/1.1 200 OK</a:status>
    <a:prop>
      <IsNagyonTitkos b:dt="boolean">1
      </IsNagyonTitkos>
    </a:prop>
  </a:propstat>
  <a:propstat>
    <a:status>
      HTTP/1.1 404 Resource NotFound
    </a:status>
    <a:prop>
      <c:EzNemLetezik/>
      <c:EsEzSemLetezik/>
    </a:prop>
  </a:propstat>
</a:response>
...
```

Csodálatos! A státuszkódok végére érve egy kis varázslat ismerete jól jöhet, ez pedig nem más, mint a Brief kapcsoló használata: a

```
Brief: t
```

alkalmazásával *(a kérés fejlécében)* PROPFIND és BPROPFIND *(több elem ugyanazon mezőinek lekérdezéséhez használatos)* esetén eltekinthetünk a nem-OK státuszú, PROPPATCH és BPROPPATCH esetén az OK státuszú DAV:propstat XML tag-ektől, magyarul megfogalmazva a kapcsoló használatával elérhetjük, hogy csak a legérdekesebb válaszokat kapjuk meg.

Összegezve a WebDAV státuszkódok világát: megtudtuk, hogy minden válasznak van egy fő státuszkódja. MultiStatus *(több „alválaszt", pl. több lekérdezett tulajdonságot tartalmazó válasz)* esetén a válasz minden külön részéhez tartozik további státuszkód. Azt is megtudtuk, hogy minden metódushoz meg vagyon írva, hogy milyen státuszkódok jelentik a sikert és a kudarcot, de az biztos, hogy a kettővel kezdődők jót jelentenek, a néggyel vagy öttel kezdődők pedig rosszat.

Az XML felépítése

Mivel az XML-t már mindenki jól ismeri, felesleges a válasz XML részének bonyolultabb elemzése. Ami furcsaságszámba

mehet, az a névterek használata. Röviden: ez az a dolog, amitől csúnyán néz ki az XML. Hosszabban: az egész ott kezdődik, hogy a WebStore az elemek mezőit névterekbe csoportosítja, vagyis a WebStore-on futó alkalmazások több ajánlás mellett meg kell feleljenek annak is, hogy az általuk kreált mezőnevek tartalmazzák a gyártó cég nevét, az alkalmazás nevét, majd ezután a mező nevét. Egyszerűbben előadva: a RépaSoft nevű cég Répa 2003 nevű terméke által létrehozott dokumentumok State *(amerikai címekben államot tároló)* mezőjét a következőképpen javasolt elnevezni: *http://www.repasoft.hu/repa2003/State*. Ennek köszönhetően a mező nem fog összekeveredni egy másik, ezt az ajánlást be nem tartó cég State nevű mezőjével, amelyben az adott dokumentum jóváhagyásának állapotát tartják számon. Ebből következik a WebDav kérés és válasz XML-dokumentumában a névterek használata: a következő válasz DAV:multistatus *(a válasz legfelső szintű)* node-ja felsorolja a válaszon belül előforduló névtereket, összerendeli őket egy rövid, egybetűs azonosítóval, majd az XML további részeiben már csak a rövid nevekre hivatkozik.

```
<a:multistatus
  xmlns:b="urn:uuid:c2f41010-65b3-11d1-a29f-
↳ 00aa00c14882/"
  xmlns:c="xml:"
  xmlns:a="DAV:">
```

A WebStore saját mezőneveinél is a fent írt szabványt követi, a *http://schemas.microsoft.com/exchange/security/* névtér például jogosultsággal kapcsolatos információk tárolására használt mezőket fog össze, míg a *http://schemas.microsoft.com/exchange/events/* névtér a WebStore eseménykezelésével összefüggő mezőket. Az előbbiekhez hasonlóan a DAV: is önálló névtérnek számít, amely alapvető adatelérést biztosító és hierarchikus információkat tároló mezőket foglal magába. A WebDAV tehát kihasználja az XML 1.0 szabvány névterekre vonatkozó előírásait, a WebStore névtér fogalmával megfeleltetve azt.

Érdemes még egy-két szót szólni az XML-adattípusokról: amikor boolean típusú mezőértéket kérdeztünk le, a WebStore ezt jelezte felénk *(ugyanígy jelezve volna, ha numerikus vagy dátumtípusról van szó)*. A válasz a következőképpen nézett ki:

```
...
<IsNagyonTitkos b:dt="boolean">1</IsNagyonTitkos>
...
```

A „b" névtér alias-nak egy iszonytatóan csúnya neve van *(kapaszkodni!)*: *„urn:uuid:c2f41010-65b3-11d1-a29f-00aa00c14882/"*. Hogy ez honnan jött, nem tudom, de valószínűsítem, hogy valahol a világ másik oldalán egy GUID generátorból. Ez a névtér jelzi a mező értékének típusát, vagyis ha PROPPATCH *(mezőmódosítás)* metódust használunk, az amúgy boolean IsNagyonTitkos mező értékének numerikus *(példánkban 4 byte-os integer)* értéket adhatunk:

```
...
<d:propertyupdate xmlns:d="DAV:"
  xmlns:typens="urn:uuid:c2f41010-65b3-11d1-a29f-
 00aa00c14882/">
  <d:set>
    <d:prop>
      <IsNagyonTitkos typens:dt="i4">
        0
      </IsNagyonTitkos>
    </d:prop>
  </d:set>
</d:propertyupdate>
```

Ahogy az előbbi példában láthattuk, nem kell minden esetben egybetűs névtér aliasokat használnunk, viszont a WebStore minden törekvésünk ellenére nem tágít az egybetűs gyakorlattól. Talán jobb így, nem terheljük a hálózatot.

A fent látható adattípus elnevezéseket XML-adattípusoknak nevezzük, amelyek listáját a következő táblázat tartalmazza *(az „mv" a multivalue szó rövidítése)*:

Adattípus	Leírás
boolean	Logikai érték, 1 vagy 0 értékként jelenik meg az XML-ben
i2 mv.i2	Integer *(2-byte)*
int mv.int	Integer *(4-byte)*
i8 mv.i8	Integer *(8-byte)*
dateTime.tz mv.dateTime.tz	Dátum és idő
r4 mv.r4	Lebegőpontos szám *(4-byte)*
fixed.14.4 mv.fixed.14.4	Fixpontos szám
float mv.float	Lebegőpontos szám
uuid mv.uuid	Stringformátumú UUID típus, elválasztókarakterekkel vagy azok nélkül
string mv.string	2-byte-os Unicode string
bin.base64 mv.bin.base64	Bináris adat *(base64 encoded)*

WebDav metódusok

A következőkben rövid bemutatásra kerülnek a WebDav főbb metódusai. A B betűvel kezdődő metódusokban a B betű a batch szót jelöli, azaz a B nélküli párjukhoz hasonló feladatot végzik egyszerre több elemmel. A B betűs metódusok nem vehetnek részt tranzakcióban *(ebben az esetben több B nélküli metódushívással lehet megoldani a problémát)*

Metódus	Leírás
COPY, BCOPY	Másolás
DELETE, BDELETE	Törlés
MOVE, BMOVE	Mozgatás
PROPFIND, BPROPFIND	Tulajdonság(ok) lekérdezése
PROPPATCH, BPROPPATCH	Tulajdonság(ok) módosítása
LOCK	Zárolás vagy tranzakció indítása. A tranzakcióban futó kéréseknél fel kell tüntetni a tranzakció egyedi azonosítóját
MKCOL	Mappa létrehozása
NOTIFY	A szerver által hívott metódus, amely – bizonyos eseményekre történő előfizetés esetén – értesítést küld a kliensnek az esemény bekövetkeztekor. Az értesítésre használt protokoll az UDP *(User Datagram Protocol)*.
POLL	Előfizetés után, az adott esemény bekövetkeztének ellenőrzésére, vagy a NOTIFY megtörténtének utólagos ellenőrzésére szolgáló metódus. Mivel az UDP nem biztosítja száz százalékosan a csomagok megérkezését, szükség lehet a kliensről indított periodikus ellenőrzésre.

SEARCH	Keresés
SUBSCRIBE	Előfizetés bizonyos eseményekre
UNLOCK	befejezés vagy tranzakció lezárása *(commit/rollback)*
UNSUBSCRIBE	Előfizetés megszüntetése

Előfizetések

Ahogy az XML mindenki számára ismert fogalom, biztosra veszem, hogy már mindenki használt OWA-t *(Outlook Web Access-t)* is. Ha igen, abban is biztos lehetek, hogy majdnem mindenkinek nagy meglepetésére szolgált élete első OWA-s levélérkezése, amikor a system tray fölött megjelent az „Új levele érkezett" felirat *(az Exchange Service Pack 2 utáni OWA képessége)*. Ha másnak mégsem, nekem biztosan távta maradt a szám. Ez a csúcsszuper feature a WebDav előfizetésfunkciójával van megoldva, amelyet megvalósító metódus a SUBSCRIBE névre hallgat. Előfizetni nem nagy varázslat, még XML-t sem kell összeállítanunk, csupán a header részben kell néhány beállítás, és már mehet is az OWA koppintás! A következő példában nem törünk ilyen nagyra, csupán egy mappában történő változásra fizetünk elő, 10 perc erejéig:

```
SUBSCRIBE /public/GyakranValtozik HTTP/1.1
Host: xch2k01
Notification-type: Update
Depth: 1
Subscription-lifetime: 600
```

Siker esetén válaszként szinte ugyanazt kapjuk vissza, mint amit küldtünk *(200-as státusszal)*, egy előfizetés-azonosítóval *(Subscription-id)* kiegészítve. Ez az az azonosító, amelyre hivatkozva, a POLL metódus segítségével megkérdezhetjük a szervertől az előfizetésünk állapotát. Ezen kívül lehetőségünk van aszinkron értesítés kérésére, ami azt jelenti, hogy ha változás történik (vagy bekövetkezik az, amire előfizettünk), a szerver megpróbál gondoskodni az értesítésünkről *(azaz nem nekünk kell megkérdezni, hogy történt-e valami)*. A „megpróbál" szót azért használtam, mert az értesítés UDP protokoll használatával történik, amely nem garantálja a csomag célba érkezését. Ez utóbbi értesítést megvalósító metódus neve NOTIFY, amelyet minden esetben a szerver hív. A NOTIFY típusú értesítést úgy tudjuk elérni, hogy a SUBSCRIBE metódusban megadunk egy Call-Back nevű kapcsolót, amely a kliensünk „fogadó" címét tartalmazza. A következő válasz egy aszinkron előfizetésre érkezett:

```
NOTIFY http://myclient:8080/510 HTTP/1.1
Subscription-id: 4
```

Tehát mind a POLL, mint a NOTIFY típusú előfizetést a SUBSCRIBE metódussal indíthatjuk, csak a NOTIFY esetében egy kapcsolóval többet kell megadnunk. Ha a POLL metódus használatával szeretnénk lekérdezni az előfizetésünk állapotát, a következő kérést kell összeállítanunk:

```
POLL /public/GyakranValtozik HTTP/1.1
Host: xch2k01
Subscription-ID: 4
Content-Length: 0
```

Megjegyzés: a *POLL* metódust használhatjuk *NOTIFY*-típusú előfizetésnél is. Mivel az UDP-csomagok nem 100%-osan érnek célba, néha előfordulhat, hogy le kell ellenőriznünk az előfizetésünk állapotát.
A Notification-type *(a SUBSCRIBE metódusnál)* kapcsoló lehetséges értékei a következők: update, update/newmember, delete, move, pragma/<http://schemas.microsoft.com/exchange/newmail>.

POLL-típusú Lekérdezésnél a Subscription-ID kapcsolóban – vesszővel elválasztva – megadhatunk több előfizetésazonosítót is. Kérdésünkre a következő választ kapjuk:

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml
Content-Length: xxx

<?xml version="1.0"?>
<a:multistatus
  xmlns:b="http://schemas.microsoft.com/Exchange/"
  xmlns:a="DAV:">
  <a:response>
    <a:href>http://xch2k01/public/GyakranValtozik/
</a:href>
    <a:status>HTTP/1.1 200 OK</a:status>
    <b:subscriptionID>
      <li>4</li>
    </b:subscriptionID>
  </a:response>
</a:multistatus>
```

A fenti válasz azt jelzi, hogy a 4-es számú előfizetésünk bekövetkezett. Ha nem következett volna be, a státusza 204 *(No Content)* lenne. Ha több előfizetésre kérdeztünk volna, státuszonként csoportosítva kaptunk volna egy vagy több DAV:response taget.
Ha lejáróban van az előfizetésünk, lehetőség van a meghosszabbítására, mégpedig egy olyan SUBSCRIBE metódus hívásával, amelyhez csak a Subscription-ID kapcsolót mellékeltük.

Első nekifutásra talán ennyi is elég, a WebStore további izgalmainak felfedezését a kedves olvasóra bízom, ezután csak kihívásokban gazdag WebStore és Exchange fejlesztést kívánhatok. Egyúttal megragadom az alkalmat, és bátorítok minden Exchange-közeli „építészt" és fejlesztőt WebDav-alapú alkalmazások fejlesztésére. A Montana keretein belül jópár WebDavot használó nagyobb méretű, kritikus rendelkezésreállású fejlesztési projektben részt vettem *(pl. VPOP – Vám és Pénzügyőrség Országos Parancsnoksága, KBH – Katonai Biztonsági Hivatal)*, és mindezek után az a véleményem, hogy ha valaki jól ismeri a témát, az Exchange 2000 *(és Titanium)* szinte minden funkcióját gyorsan és hatékonyan kihasználhatja. Ezt bizonyítja a Microsoft következő generációs Exchange adatelérési komponense, amely az XSO *(Exchange Server Objects)* névre fog hallgatni, és teljesen WebDav alapú lesz. A Microsoft jó szokásának az Exchange 2000 esetében is hódolt, elég jó dokumentáció található az Exchange SDK-ban [1]. Exchange fejlesztői témájú cikkek a [2] oldalakon találhatók. A Microsoft levelezőlistáin két WebStore szempontjából érdekes cím: [3] és[4]. A Microsofton kívül megtalálható site-ok közül érdekes lehet a www.wsd2d.com, amely elsősorban a WebStore fejlesztők közti kommunikációt támogatja. Az Exchange 2003 *(Titanium)* Beta 2 bemutatója a következő címen érhető el: [5].

Szabó Dávid
szabo.david@montana.hu
vezető fejlesztő

A cikkben szereplő URL-ek:

[1] http://msdn.microsoft.com/library/en-us/wss/wss/
↳ _exch2k_welcome_to_exchange.asp
[2] msdn.microsoft.com/exchange
[3] microsoft.public.webstore.development
[4] microsoft.public.exchange2000.development
[5] http://msdn.microsoft.com/library/en-us/
↳ _etb2dp/mini_ti/_e2k3_welcome.asp

MEC 2002

Konferenciabeszámoló

A MEC korábban a Microsoft Exchange Conference kifejezés rövidítése volt, amely később átalakult Microsoft Exchange and Collaboration-re. Azonban idén már nem vesződtek a bejáratott MEC megnevezés blikkfangos értelmezésével, a konferencia hivatalos leírása ez volt: „The essential Microsoft conference for planning, deploying and managing a connected infrastructure".

A konferenciáról általában

2002. október 8. és 11. között a Microsoft Anaheimben *(USA, California)*, Los Angeles egyik külvárosában rendezte meg a szokásos csoportmunka technológiai konferenciáját, MEC 2002 elnevezéssel, mintegy 6000 résztvevővel a világ minden részéről, persze elsősorban az Egyesült Államokból.
Noha a konferencia témáinak 50-60 %-a most is az Exchange Serverről szólt, már az elnevezés is egyértelműen jelezte: a Microsoft számára már nem az Exchange a kizárólagos csoportmunka-platform. Az Exchange mellett a konferencia legtöbbet emlegetett témája a SharePoint termékkör *(mind az SPS, mind az STS változat)* és a MIS-alapú mobil megoldások voltak, de sok előadás foglalkozott sokkal általánosabb témakörökkel *(általános rendszermenedzsment, általános fejlesztői lehetőségek)*, mintegy szemléltetve, hogy a Microsoft ma már roppant szélesen értelmezi a csoportmunka *(„Collaboration")* témakört.

Érdekes kettősség jellemezte az egész konferenciát. Egyrészt a Microsoft pillanatnyilag a csoportmunkafunkciók koncepcionális és technológiai újragondolásának fázisában van, nagyjából érezve, hogy az eddig roppant népszerűvé vált Exchange/Outlook rendszer tartalékai kimerülnek, és 2-3 éven belül alapjaiban kell megújulni ezen a területen is. Ezt mutatta, hogy meglehetősen kis számban voltak jelen a hosszú távú terveket ismertető, koncepcionális előadások *(döbbenetes volt, hogy a MEC két Keynote előadása közül egyik sem foglalkozott a Microsoft csoportmunkarendszereinek jövőjével)*, és erre utalt a legbeavatottabbak arcára is kiülő zavart arckifejezés, amikor a meglévő csoportmunkafejlesztések technológiai jövőjére terelődött a szó.
Másrészt viszont egyelőre még ezerrel robog a szekér, az Exchange Server-hez már több mint 100 millió *(!)* klienst adtak el, és az mára de facto ipari szabvánnyá, az SAP mellett a nagyvállalat IT rendszerek standard alkotórészévé vált. A konferencián soha nem látott számú és kifinomultságú Exchange-infrastruktúra és üzemeltetési előadást hailhattunk. A kapcsolódó kiállítások is az Exchange-alapú infrastruktúrális és fejlesztési megoldások döbbenetes választékát mutatták.
A konferencián 6 témakörben 198 előadás hangzott el:

- Collaboration Solutions
- Administration and Management
- Development Tools and Technologies
- Mobility
- Planning and Deployment
- Security

Az előadások közül hosszas tanakodás után a rendelkezésre álló időt maximálisan kihasználva végül 21 előadást hallgattam meg. Ezekből nyújtok át egy csokorravalót, tömörített formában:

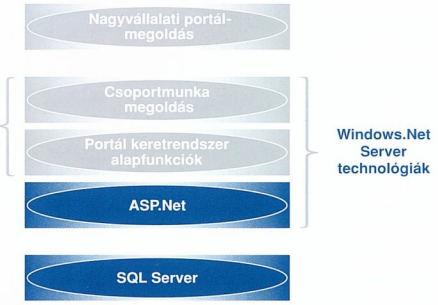
Paul Flessner: The Connected Enterprise *(Keynote)*

A konferencia nyitóelőadását Paul Flessner, a Microsoft .NET Server portfoliójáért felelős alelnöke tartotta. Érdekes technológia víziót láttunk, amely a mindent mindennel *(rendszerek, emberek, eszközök, üzletek)* összekapcsoló funkcionalitáson keresztül mutatta be a Microsoft jövőképét. Számomra három lényeges információ hangzott el:

- az Exchange új verziója a Titanium *(ejstd: tájténiöm; beceneve: Ti - ejtsd: Táj)* 2003 közepén jelenik meg
- szintén megjelenik az új Outlook *(11-es verzió)*
- formálódik a Microsoft új csoportmunkafejlesztő eszközrendszere, az XSO.

Brian Murphy: SharePoint Products and Technologies V2 Overview *(TBR250)*

Ez az előadás volt az egyetlen, amely kifejezetten a SharePoint termékkör új, 2003 közepén megjelenő változatáról szólt. A Microsoft két SharePoint terméke közül a kisebbik testvér, a SharePoint Team Services lett a nyerő, úgy tűnik, az új verzió teljes egészében ennek technológiai platformjára épít *(MS SQL Server adatbázis, webes felület, dokumentumkezelési és általános csoportmunka funkcionalitás)*, amelyet most kiegészítenek a nagyobbik Sharepoint portál- és keresőfunkcióival. Az új verzióban már nem lesz technológiai különbség a két Sharepoint változat között: a nagyobbik testvér mindent tud, amit a kisebbik, csak néhány nagyvállalati és Internetes használathoz szükséges „Enterprise" funkcióval egészül ki.



Látni lehetett, hogy a fejlesztés elsődleges iránya a jobb skálázhatóságra és a robosztus működésre irányul *(természetesen .NET technológiai váltással együtt)*, míg a dokumentumkezelési és tudásmenedzsment funkciók esetében inkább a szintentartás dominál.

Fontos infó volt, hogy az új SharePointnál tovább erősítik a régi STS-ben már megjelent nézeteken és egyedi űrlapokon alapuló általános csoportmunka keretrendszert, amely akár az Exchange Server alternatívája lehet. Az előadás után megkérdeztem az előadót, hogy mégis, akkor most az Exchange Server, vagy a SharePoint lesz az elsődleges platform a csoportmunka-fejlesztésekhez? Enyhén szólva kitérő választ adott, aminek az volt a lényege, hogy igazából mind a két eszköz szükséges lehet. *(A Microsoft maga, házon belül kizárólag az STS-t használja csoportmunkacélokra, sem SPS, sem Exchange-alapú fejlesztései nincsenek!)*

Jensen Harris: Outlook 11: Overview *(TBR200)*

Az Outlook fejlesztői csapatának vezetője *(Lead Program Manager)* ebben az előadásban részletesen ismertette a jövő év közepén az új Office részeként piacra kerülő Outlook 11 újdonságait. Megtudtuk, hogy az Outlook felhasználói felületét alapvetően újratervezték, és az eddig is a világ egyik legkifinomultabb UI-jához még egy sor alapvető újdonságot alakítottak ki. Nagyon fontos szempont az online és offline használat megújítása, amit az előadó egy egyméteres drótvágóval kívánt illusztrálni – sikerrel. A harmadik nagy újdonság, amely az Outlookalapú fejlesztések szempontjából is kimagasló jelentőségű, az úgynevezett „Search Folder" funkció megjelenése, amely lehetővé teszi a különböző fizikai mappákban levő elemek tetszőleges, virtuális nézetekben való egyesítését. Roppant érdekes volt, ahogy az előadó bemutatta, hogy az új Outlook nemcsak az Exchange Serverhez, hanem az új SharePointhoz is rendelkezik közvetlen klienshozzáféréssel, és bizonyos STS-funkciók az Outlookból is elérhetőek. Az egész előadás egyértelműen azt sugallta, hogy maga az Outlook továbbra is a Microsoft csoportmunkamegoldásainak legfontosabb elemét jelenti, noha az előadás után feltett kérdéseimre adott válaszokból kiderült, hogy az Outlook fejlesztői környezetnek *(Outlook űrlapok, VBA)* nem tervezik érzékelhető mértékű továbbfejlesztését.

Leon Warman: Adding People-to-People Workflow to Your Solution *(CLB340)*

Az előadás a Microsoft Exchange Serverbe épített Workflow lehetőségek jól felépített összefoglalása volt. Végigvette a Workflow-motor, és a tervezőeszköz funkcióit, és néhány példával felvillantotta a programozási lehetőségeket is. Tartalmilag nem sok újdonsággal szolgált, ezek nálunk már a MonFlow részeként gyakorlatban használt eszközök. Ami érdekes, hogy az előadás egyértelművé tette: a személyek közötti *(irodai)* workflow-megoldásokra a Microsoft továbbra is egyértelműen az Exchange-környezetet ajánlja *(a BizTalk a korábbi elvárásokkal szemben nem vált a people-to-people workflow megoldások platformjává – maradt a nagy, automatizált rendszerek folyamatainak vezénylőmotorja)*.

Terry Myerson: Exchange Product Roadmap *(TBR201)*

Bátran állíthatom, hogy ez volt a legérdekesebb előadás az egész konferencián, ennek kellett volna az egyik Keynote-nak lennie *(így is kétszer ismételték meg a nagy érdeklődésre való tekintettel)* Terry Myerson, aki a Microsoft Exchange fejlesztői csapat egyik vezetője *(Group Program Manager)* meglehetősen részletesen elemezte az Exchange Serverrel kapcsolatos, a közeljövőben tervezett új fejlesztési terveket.

Először is, hogy mely termékekre épül ma a Microsoft csoportmunka stratégiája: a SharePoint Team Services lesz várhatóan a dokumentumkezelési és az egyedi csoportmunkafunkciók platformja, az SPS a vállalati portálmegoldások és a széleskörű keresőfunkciók hordozója, az Exchange pedig a személyes információkezelés, és a mobilfunkciók mellett a Workflow-lehetőségek tekintetében elsődleges. Új elem a Greenwich kódnevű új szervertermék, amely elsősorban az online együttműködés platformja lesz.

Az egész Exchange 2000-történet legsikeresebb része az új OWA volt. Állítólag a legtöbb ügyfél, aki az Exchange 2000 bevezetése mellett döntött, egyik legfontosabb indoknak az új, minden eddiginél gazdagabb funkcionalitású Outlook Web Access nevezte meg. Éppen ezért a Titanium legnagyobb újítását mindenképpen a webalapú kliens még komolyabb továbbfejlesztése fogja jelenti. Az OWA eddig is messze a legszélesebb funkcionalitású, legkifinomultabb webes levelező-kliensnek számított a világon, ám a bemutatott Titanium demo már messze meghaladott mindent, amit egy webalapú alkalmazásról valaha is el tudtunk képzelni – noha semmilyen kliensoldali huncutság *(Java applet, ActiveX, stb)* nem támogatta ezt – tiszta szerveroldali kód, némi kliensoldalon futó, scripttel felturbózott HTML. Hogy csak néhány példát emeljek ki: végre támogatják a Tasks funkciót, itt is használható lesz a Search Folder, teljeskörűvé válik a Drag&Drop és a jobb egérgomb használata, és működő helyesírásellenőrzés *(!)* is van a rendszerben *(ne feledjük, egy webalapú vékony kliensről van szó!)*. Az egész rendszer felhasználói élménye *(elsősorban a sebesség)* sokkal jobban hasonlított a normál Windows-alkalmazásnál megszokott feelingre. *(Ez a kis demó valószínűleg betekintést engedett a jövő webalkalmazásainak világába – ahol talán kompromisszumok nélkül válthatják ki a böngészőalapú megoldások a hagyományosakat…)*

A másik nagy hanggal beharangozott újítás a sávszélességbarát működési módot támogató új hálózati kommunikációs megoldás, melynek lényege:

- az Outlook és az Exchange közötti kommunikációt optimalizálták *(csak a legszükségesebb információkat küldi és fogadja a rendszer)*
- az adatforgalom tömörített
- az Outlook-kliens automatikus és sima átmenetet ad az online és az offline mód között *(lásd: drótvágós demó az Outlook előadáson)*
- a kommunikáció teljes egészében HTTPS protokollon zajlik.

SharePoint Portal Server (STS)
- Ad hoc csoport-szintű munkaterületek
- Klasszikus dokumentum-tárolás és csoportmunka

SharePoint Portal Server (SPS)
- Nagyvállalati portálok
- Integrált, széleskörű keresőrendszer sokféle forráshoz

Exchange Server
- Személyes információkezelés
- Interperszonális workflow
- Mobil megoldások

Greenwich
- Instant messaging
- Online találkozók
- Alkalmazás-megosztás
- Videoközvetítés
- Élőhang továbbítása

Az offline használat alapját adó kliensoldali komponenst nemes egyszerűséggel csak „Cached Exchange"-nek nevezték. A leírások szerint ennek használatával 30-70%-kal csökken a hálózati forgalom, 50%-kal a CPU terhelés… szerveroldalon.

A harmadik nagy újdonságot pedig a mindent elborító mobilfunkciók integrálása jelenti. A most még külön termékként forgalmazott MIS *(Mobile Information Server)* a következő Exchange verzióban már az alapszolgáltatásokat fogja gazdagítani. Az új Exchange-nek az SMS/MMS/WAP szolgáltatásokat használó mobiltelefonok, és a drótnélküli hálózatba kapcsolt, vagy GPRS kapcsolattal bíró PDA-k éppolyan természetes kliensei lesznek, mint manapság a PC-inken futó Outlookok.

Komolyan megújul az Exchange Server üzemeltetési biztonságát garantáló eszközrendszer is. A Titanium már akár 8 csomópontos fürtbe is szervezhető, és teljesen megújított, optimalizált backup-rendszer kapcsolódik hozzá. Emellett végre teljessé válik a MOM-integráció, és kiteljesednek a biztonsági funkciók is.

Az Exchange Server továbbfejlesztése során az egyik leghangsúlyosabb terület az Exchange Server fejlesztői környezetének megújítása. A CDO-t felváltó új, mindenható hárombetűs rövidítés az XSO. Természetesen ez már a .NET frameworkhöz illesztett, nyelvfüggetlen API.

Az előadás befejezéseként Terry még megpróbált bekukucskálni a Titanium utáni Exchange fejlesztéseket jótékonyan burkoló homály mögé is. Úgy tűnik, a fejlesztés továbbhalad a személyes információkezelés finomításának irányába: szerveroldalon továbbfejlesztett e-mail, Naptár, feladat és kontaktfunkciók várhatók. A kérdésekre adott válaszokból az is egyértelművé vált, amit a suttogó propaganda már régóta terjesztett: a Titanium utáni új Exchange verzió már új adatbázismotort fog használni, ugyanazt, amit az addigra szintén megjelenő új SQL Server számára alakítottak ki.

Mindy Martin: Developing with Exchange Using Visual Studio .NET *(ADM401)*

Mindy Martin a legnagyobb és legbefolyásosabb Exchange fejlesztési guruk közé tartozik. Mint Program Manager az Exchange Server-hez kapcsolódó fejlesztőeszközök kialakításának egyik vezetője, és a mind a mai napig az egyik legjobb Exchange 2000 fejlesztésekkel foglalkozó könyv szerzője *(Programming Collaborative Web Applications with Exchange 2000 Server, magyarul is elérhető)*.

Ebben az előadásában az új .NET-alapú fejlesztőeszközök Exchange-alapú csoportmunka alkalmazásokhoz történő felhasználását járta körül.

Először végignéztük az Exchange 2000-hez illeszthető .NET fejlesztéseket: készíthetünk az Exchange Server szolgáltatásaira épülő Web Service-eket, illetve használhatunk .NET-es kliensalkalmazásokat *(ASP.NET, Win Forms)*, és event sinkek is kialakíthatók .NET eszközökkel. Ami lényeges: abszolút ellenjavalt, és nem támogatott a nyilvános mappákban levő webform-alapú alkalmazások ASP.NET-re való továbbfejlesztése. Ha van oly botor e kies hazában, aki ilyesmivel próbálkozott, az jobban teszi, ha elölről kezdi a fejlesztést a .NET platformra való áttérés során...

Az előadás második részében a most alakuló új Exchange fejlesztői környezet, az XSO lehetőségeit vázolta. Ezt olyan fejlesztési lehetőségnek szánják, amelynek lényege a „managed code class library" és különösebb specifikus Exchange szaktudás nélkül, a .NET nyelvfüggetlen keretrendszerére építve teszi lehetővé a fejlesztést a Titanium környezethez.

Az előadás talán legérdekesebb része az volt, amikor az előadó elmondta, hogy intenzíven gyűjtik az ötleteket az XSO végleges formájának kialakításához. Sajnos az is kiderült ebből, hogy magát az XSO-t igazából még csak tervezgetik, a fejlesztése csak ezután fog kezdődni. Úgy tűnik, ez nem fog megjelenni a Titaniummal együtt, csak később, valamelyik szervizcsomagban, vagy egyedi kiegészítésként.

Egyéb érdekességek

Scott Jamison, a Dell munkatársa az egyik prezentáció során rengeteg gyakorlati példában mutatta be, hogyan lehet az új .NET-es technológiákat felhasználva Office-alapú alkalmazásokat készíteni. Az előadás maradandó mondanivalója volt, hogy az Office-nak már a jelenlegi *(XP)* verziója is ideális frontend az XML-alapú webszolgáltatások számára. Ehhez az Office XP Web Sevices Toolkit nevű csomagot ajánlotta.

Sue Mosher előadásából megtudhattuk, hogy az Outlook nézetek definícióját hordozó információk Outlook 2000-től kezdve egy XML-fájl szerkesztésével módosíthatóak. Az előadás gondosan ismertette ennek kivitelezhetőségét, majd konkrét példákkal, kódrészletekkel mutatta be a gyakorlati használatot.

Matt Gossage az „Exchange 2000: Scalability-Reality vs. Myth" című előadásában nagyon gyakorlatias módon, praktikusan felhasználható formában tárgyalta a nagy terhelésnek kitett Exchange Server méretezési problémáit és megpróbált eloszlatni néhány közkeletű mítoszt. Az Exchange Server számára a szűk keresztmetszetet ma általában a memória jelenti. A terhelés fokozása során először a Windows 2000/Exchange 2000 ismert memóriakorlátjába *(3 GB a Boot.ini kapcsoló beállításával)* fogunk ütközni. A jelenlegi ismeretek szerint a felhasználható processzorok száma, illetve az Exchange Server mögött dübörgő adatbázismotor által kezelt merevlemez területek mérete még tovább növelhető lenne, ha nem ütköznénk a memória korlátaiba...

Az Exchange-alapú fejlesztéseknek a memóriakorláton túl nincs méret/teljesítményhatára. Az Egyesült Államokban sok nagyvállalatnál több száz gigagbájtnyi, milliós léptékű adathalmazokat kezelnek Exchange alapon *(CDO, MAPI, WebDAW)* kifejlesztett egyedi alkalmazásokkal.

Füzessy Tamás
Alkalmazásintegrációs és fejlesztési igazgató
fuzessyt@montana.hu

Microsoft Exchange Server 2003

A projekt, amit valaha Titaniumnak hívtak

Hamarosan elkészül az Exchange következő változata, tele számos újdonsággal. Outlook Cached Mode, RPC over HTTP, újraírt Outlook Web Access, integrált mobilszolgáltatások hogy csak néhányat említsünk. Valóban van még új a nap alatt? Tudósítónk jelenti Redmondból.

Merre tovább Exchange?

Bizony, már három év telt el a Microsoft Exchange 2000 megjelenése óta. Az 5.5-ös verzió utódja, köszönhetően a mélyreható Windows 2000 integrációnak, meglehetősen sikeres pályát futott be, de hagyjuk most a marketinget... A valóságban igenis voltak hullámvölgyek a termék életében, jelenleg az SP3-nál tartunk, a korai gyermekbetegségeken – pl. memóriatöredezettség jelenségek – túljutva, ma már nem kérdés, hogy Active Directory környezetben az Exchange 2000 a vezető üzenetkezelő platform.

Kevesen követhették, de ez idő alatt, valamikor egy éve, az Exchange fejlesztői csoport kijelölte a termék következő generációinak irányvonalát. A nagy kérdés az volt, hogy a közeljövőben kifejlesztendő technológiák vajon javítókészletekbe, vagy egy új szerververzióba kerüljenek-e bele, illetve mikor történjen meg a Nagy Váltás: az adatmotor cseréje egy következő generációs adattároló mechanizmusra. A fejlesztők – egyebek mellett a közeljövőben megjelenő Office hatására – egy új szerververzió kiadása mellett döntöttek, az eredmény egy Exchange Titanium elnevezésű projekt lett. A cél a termékkel nem az volt, hogy az alapvető komponenseket *(adattár, a klienshozzáférési interfészek, vagy a fejlesztői platformszolgáltatások)* lecserélje. A fejlesztők sokkal inkább a mobil kliensek, lassú elérésű távoli telephelyek támogatását, illetve a Windows Server 2003 és Office 11 integrációt jelölték meg elsődlegesnek, szem előtt tartva persze, hogy nem árt, ha az adminisztrátorok életét is könnyebbé teszik néhány apróval tűnő, de nagyon hasznos újdonsággal.

Mielőtt elmélyednénk ezek megismerésében, álljunk meg egy pillanatra. Szép dolgok ezek, gondolhatnánk, de az nem lehet, hogy csak egy-két évre képesek előre gondolkodni a redmondi fejlesztők. Nem is ez történt, ugyanis a Titaniummal párhuzamosan elindult egy másik, igencsak ambíciózus projekt, melynek célja a szerver *(melyet ma Exchange-nek hívunk)* teljes körű újratervezése és integrációja a Microsoft következő generációs adattároló technológiájával *(Yukon)*, kliens operációs rendszerével *(Longhorn)* és produktivitási eszközeivel *(Office11+)*. Itt gyorsan le is zárom ezt a szálat, hiszen ezek a technológiák leghamarabb 2004/2005 körül állnak majd rendelkezésre. Koncentráljunk most a Titaniumra, akarom mondani a Microsoft Exchange Server 2003-ra.

A termék gyártásra kész állapotát – a Microsoftos szleng ezt RTM-nek *(release to manufacturing)* hívja – valamikor ez év júniusában éri el. Jelenleg Beta 2-ben van, ami azt jelenti, hogy funkcionális szempontból többé-kevésbé kész, a júniusig terjedő időszakban csak nagyon kevés új szolgáltatás *(feature)* kerül bele, a fejlesztők javarészt hibajavítással és optimalizálással foglalkoznak majd. A bátor vállalkozók a mellékletben található címről [1] letölthetik, a teszteléshez VMWARE javasolt, a telepítés előtt a követelmények megismerése és a 'release notes' elolvasása kötelező gyakorlat!

Most pedig nézzük az Exchange Server 2003 újdonságait! *(A „hogyan működik" részleteket terjedelmi okok miatt ezúttal megvágtam, teljeskörű technológiai áttekintés egy közeljövőben induló sorozatba kerül majd bele.)*

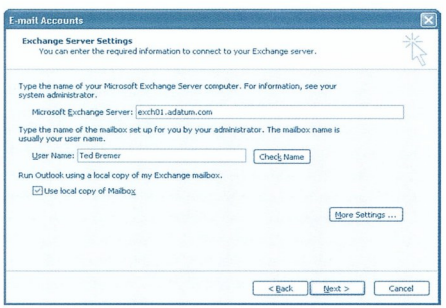
Amit a felhasználók látnak

Az Exchange Server 2003 talán legfontosabb kliensoldali újdonsága az Outlook11 új „Cached Mode"-ja. Ez az üzemmód átmenetet képez az „offline" és „online" Outlook profilok között. Ha bekapcsoljuk, az Outlook 11 felhasználó nem a szerveroldali postaládáját „látja", hanem annak egy lokális változatát, melyet egy intelligens szinkronizációs mechanizmus folyamatosan frissen tart. Miért van erre szükség? A választ erre, mint sok minden másra, egy ma már idejétmúlt tervezői felfogásban érjük utol. Az Exchange első verzióinak tervezésekor az a felfogás alakult ki, hogy egy tipikus hálózat levelező kiszolgálója és kliensei között folyamatosan jó *(LAN)* kapcsolat van, ha pedig ez nem áll fenn, a lassú vonalakon bekapcsolódó felhasználók használjanak „offline" profilt és szinkronizáljanak a kliens igényei szerint. A fenti filozófiát követve fejlesztette ki a Microsoft a MAPI protokollt, mely valójában egy üzenetkezelő infrastruktúrákra optimalizált kliens-szerver interfész. A MAPI egyik fontos tulajdonsága, hogy a konkrét „interprocess" kommunikációra az operációs rendszer RPC mechanizmusát használja.

Erre viszont sajnálatos módon az jellemző, hogy közismerten kapcsolatérzékeny, azaz a kapcsolat a szerver és kliense között - a hálózati forgalom időleges megszűnésével vagy a rendelkezésre álló sávszélesség drasztikus lecsökkenésével – nagyon könnyen megszakadhat *(„timeout-ra fut")*. Ekkor persze a kapcsolatot újra fel kell építeni, ami hatással van a kliensalkalmazás viselkedésére. Az Exchange kliensek, köztük az Outlook, ezt viszonylag nehezen viselik, a korábbi változatok lefagytak, a felhasználóknak újra kellett őket indítani, az OutlookXP-ben ekkor jelent meg az a hírhedt párbeszédablak, amit a mellékelt ábrán látunk.

Az RPC hibák kultúrált kezelése sajnos a mai napig nem megoldott, a fejlesztők szerencsére belátták, hogy egy erősen szegmentált, több telephelyes hálózatban az élet nem habostorta, igenis vannak hálózati kimaradások, sávszélesség-ingadozások. Különösen igaz ez a 802.11b vezeték nélküli hálózatokra, aki kipróbálta már, hogyan viselkedik az Outlook a téreröség megszűnése után egy WindowsXP-s notebook-on, az tudja mire gondolok. Lépni kellett tehát, melynek eredménye az új Outlook Cached Mode, ami átvészeli a hálózat bizonytalanságait. Fő erénye, hogy a hálózati kapcsolat állapotától függetlenül a felhasználó a lehető legfrissebb információkat látja, ha pedig frissítésre kerül sor, lehetőleg a legkevesebb információ menjen át a „dróton". Hogy világos legyen, nézzük, mi történik a háttérben.

A gyorsítótár üzemmódban az Outlook egy lokális OST fájllal tartja a kapcsolatot, melyet egy alacsony prioritású szál folyamatosan aktualizál. A mechanizmus alapjaiban más, mint amit az „offline" szinkronizáció során megismertünk. A kliens az Exchange kiszolgálótól *(amennyiben lehetősége van)* folyamatosan megkapja változásértesítéseket, ha pedig van ilyen, elkezdi a szinkronizálást. Az újdonság itt az, hogy a frissítés során először csak a fejlécek érkeznek meg, a teljes tartalom csak akkor jön le azonnal, ha a felhasználó rákattint egy levélre, ha egyébként nem tesz semmit, a szövegtörzs a rendelkezésre álló CPU idő és a hálózat rendelkezésre állásának függvényében, folyamatosan jön le. Mindebből a felhasználó persze semmit sem lát, ő csak azt érzékeli, hogy a postaládáját folyamatosan „karbantartják", függetlenül attól, hogy éppen van hálózati kapcsolata vagy sem.

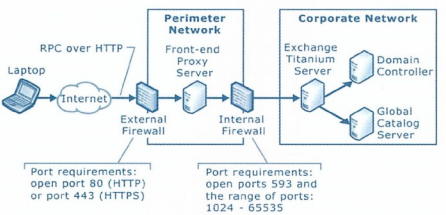


A Cache Mode konfigurálása az Outlook 11-ben

Érdekes még, hogy a szinkronizáció sem megy „simán", mivel az Exchange Server 2003 a leküldés előtt egy GZIP algoritmussal összetömöríti a leveleket. A korai tapasztalatok szerint ez RTF típusú üzeneteknél ~20%, HTML/Text esetén akár ~60%-os kompressziót jelent. Az Outlook Cache Mode-dal kapcsolatban még egy érdekességet említek: az Outlook 11 folyamatosan naplózza a sikeres és sikertelen RPC hívásokat, amit bármikor megnézhetünk a CTRL billentyűvel a tálca Outlook ikonjára kattintva. Mellesleg ezt az információt az Outlook folyamatosan felküldi szervernek, amit később az Exchange System Managerben is megtekinthetünk, ahonnan akár WMI-n keresztül bármikor kiolvasható, hogy aztán az igazán kifinomult adminisztrátorok remek jelentéseket készítsenek a levelezőkliensek oldalán érzékelt rendelkezésre állásról *(például MOM-ban)*. Egyelőre ennyit az Outlook Cache Mode-ról, meggyőződésem, ezentúl alapjaiban másként tervezünk Exchange 2003-as topológiákat, nem lesz szükség külön szerverre a távoli, lassú elérésű telephelyeken, nem beszélve arról a mellékhatásról, hogy az új üzemmódnak köszönhetően a kiszolgálók ugyanazzal a konfigurációval nagyságrendekkel több Cache Mode klienst fognak tudni kiszolgálni, mint online MAPI-s ügyfelet. Most pedig térjünk át egy másik, eddig megoldatlan problémára, nevezetesen hogyan kapcsoljunk internetes MAPI klienseket belső Exchange kiszolgálókra.

Élet a tűzfalon kívül

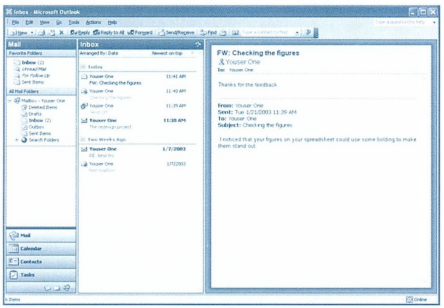
A feladat fogós, a triviális megoldás VPN-t konfigurálni, de ez nem minden vállalaton megy át „csont nékül". A másik megoldási lehetőség, hogy lefixáljuk és kinyitjuk a MAPI-s RPC portokat, amit eddig csak a legbátrabb ISA-adminisztrátorok próbáltak. A dolog megoldható, a Techneten van róla információ, de nem kultúrált, a részletekbe most nem megyünk bele. Pedig milyen jó lenne, ismerve az Outlook Cache Mode képességeit, ha klienseink egy tetszőleges Internetkapcsolat felett is be tudnának menni a tűzfal mögötti postaláda-szervereikre, lehetőleg titkosított csatornán, és persze minimális adminisztratív igénnyel a tűzfalasok felé! Nos, erre született a megoldás az Exchange Server 2003-ben: RPC over HTTP. Segítségével, amint az a nevéből kiderül, egy dedikált front-end kiszolgálót *(Windows Server 2003)* kinevezhetünk ún. RPC proxy-nak. Innen már csak egy lépés, hogy SSL-tanúsítványt kérjünk az Exchange alapértelmezett webjére, publikáljuk a szervert az ISA-ban, s máris jöhetnek a Outlook Cache Mode kliensek, szinkronizálás az Internet felől kipipálva.

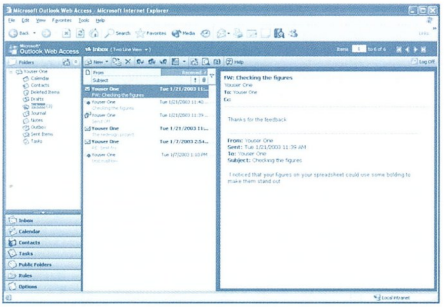


Az RPC over HTTP-infrastruktúra az Exhange 2003-ban

Böngészés, veszteségek nélkül?

A szinkronizálás remek szolgáltatás, de mi van, ha a felhasználóknál nincs Outlook kliens? A válasz: Outlook Web Access. Ez már az Exchange 2000-ben is benne volt, a maga nemében egyedülálló megoldásként. A gond csak az, hogy a böngészőfelület igencsak szűkített képességekkel rendelkezett a Win32-es Outlook klienshez képest *(pl. feladatok, szerveroldali szabályok NEMkezelése)*. Az Exchange Server 2003-ban ez a korszak is lezárul, a fejlesztőcsapat közel egy éve a legszorosabb kapcsolatban van az Office11 fejlesztőivel. Az eredmény az új Outlook Web Access, amely kisebb részletektől eltekintve funkcionálisan azonos az új Outlook képességeivel. Bátran állíthatjuk, az Exchange Server 2003 OWA a Microsoft történetének legösszetettebb és legprofesszionálisabb webes alkalmazása. Csak néhány érdekesség, a részleteket a „Getting Started Guide" [2] kiemerítően tárgyalja: gyorsbillentyűk, drag&drop, keresőmappák, kontextus menük, intelligens előnézet, „flagging", helyesírás-ellenőrző *(magyar sajnos nincs)*, S/MIME támogatás *(!!!)*, csatolt állományok blokkolása stb. Összehasonlításképpen vessünk egy pillantást az új Outlook webes és windows-os változatára.





Teljes szimmetria: Az Outlook 11 Win32-es és webes felülete

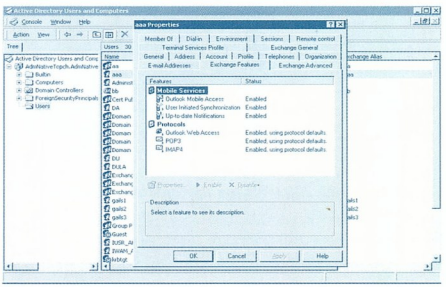
Azt gondolom az ábrák magukért beszélnek, az Exchange Server 2003-ban egy átlagos felhasználó szempontjából teljesen mindegy, melyik klienst használja. Nem érintettük még magát az Outlook 11-et, mely az Office 97 kliense óta vajmi keveset változott. Annál inkább változtak levelezési szokásaink. Nem tudom, az olvasók miként vannak vele, az én postaládámba átlagosan öt másodpercenként érkezik egy levél. Ezt az információáradatot eddig csak kifinomult szerveroldali szabályokkal, szűrt, csoportosított nézetekkel tudtam kezelhetővé tenni. Emellett érdekes jelenség, *(gondolom, sokan igazolnak ebben)*, hogy a felgyorsult kommunikációnk ellenére, naponta valójában nagyon kevés, számunkra ténylegesen értékes levél érkezik. A sok zaj ellenére a valóságban csak ezekkel szoktunk foglalkozni, s közülük is csak kevés az, amit közvetlenül meg tudunk válaszolni, egyszóval a fontos leveleket a felhasználók szeretik félretenni, hogy később, ha már lehetőségük van megválaszolni, elővehessék. Az új Outlook 11 felhasználói felülete épp erre van optimalizálva. A levélmegjelölés és csoportosítás – flagging – a felület központi szolgáltatása. Az új klienst használva a levelek nem ömlesztett levélhalmazként jelennek meg: a postaláda tartalmát a felhasználók egy sokkal 'kifejezőbb' nézetben látják. Újdonság még a keresőmappák bevezetése, melynek szerveroldali komponensei már az Exchange 2000-ben rendelkezésre álltak, kliensoldali integrációja azonban csak az Outlook 11-ben történt meg. A keresőmappákkal tetszőleges feltételrendszer alapján – tennivalók, olvasatlan, nekem jött, magas prioritású stb. – ún. virtuális mappákat definiálhatunk, ami valójában egy folyamatosan aktualizált kiszolgálóoldali lekérdezés. Egyébként ezt eddig is meg tudtuk *(volna)* oldani kliensoldali nézetekkel, a keresőmappák azonban ennél jóval rugalmasabbak: amellett, hogy a kiszolgálóoldali támogatás miatt jóval gyorsabban lefut a lekérdezés, a keresőfeltételhez megadhatjuk, milyen hierarchikus mélységben szeretnénk a lekérdezést futtatni.

Az Outlook 11 felhasználó felületének bemutatását terjedelmi okok miatt most lezárjuk, adjuk most át a helyet egy másik stratégiai fontosságú témakörnek, a mobil támogatásnak.

Mobilitás a dobozban

Mielőtt belemélyednénk, egy történeti áttekintést teszek a mobil technológiák és az Exchange 2000 integrációja körül. Sajnos Magyarországon eddig kevesen tudják, hogy az Exchange 2000-hez a Microsoft kifejlesztett egy több szinten integrált mobilátjárót, a Mobile Information Server 2002-t. A MIS 2002 három mobiltechnológiát vezetett be az Exchange 2000 szolgáltatásai közé, ezek: online hozzáférés WAP 1.0-s eszközökről, kétirányú szinkronizálás a PocketPC 2002 *(ActiveSync)* klienssekkel, valamint egy Exchange szerveroldali szabályokra épülő SMS kiértesítő infrastruktúra. Az MIS az Exchange 2000 megjelenése után két évvel került kereskedelmi forgalomba, célja a vállalati üzenetkezelő rendszerek „felerősítése" mobil technológiákkal.

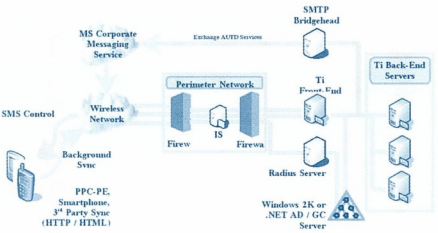
Azóta a mobilitásban lévő óriási momentumnak köszönhetően a „mobil felfogás" a Microsoft termékfejlesztésben alapvető fordulatot vett. A cél ezentúl nem az, hogy újabb és újabb 'mobil' termékeket fejlesszen ki a cég. Ennél lényegesen magasabb rendű szempont, hogy minden közeljövőben elkészülő termék rendelkezzen mobilképességekkel, azaz támogassa a most és közeljövőben megjelenő Microsoftos *(PocketPC 2002, PocketPC 2002 Phone Edition, SmartPhone 2002, Windows-CE.NET)*, valamint a külső gyártók által fejlesztett *(Palm, Nokia stb.)* nem „MS" mobil eszközöket. Cél, hogy ezek a szolgáltatások a jelenlegi *(GSM, GPRS)* és jövőbeli *(G3)* infrastruktúrákon is működőképesek legyenek. Megjegyzem, a cégen belül, valamikor két éve, erre egy külön termékfejlesztési divízió is megszületett. Ezek alapján egyenesen következik, hogy az Exchange 2003-ban a mobil elérést közvetlenül a „dobozból" kapjuk. Ahhoz, hogy postaládáinkat elérjük egy mobiltelefonról, vagy szinkronizáljuk egy PocketPC-s eszközre, nincs szükség extra kiszolgálókra, mindez integráns része a terméknek. Az integráció következménye, hogy a mobilelérés az Exchange Server 2003-ban pusztán egy protokoll, az adminisztrációja a többi klienshez hasonlóan a felhasználók megfelelő AD attribútumainak ki- és bekapcsolásával történik.



Csak semmi bonyodalom: Exchange 2003 mobilkliensek konfigurálása az ADUC-ban.

A „hogyan működik" részt terjedelmi okok miatt ismét elhagyjuk, ez egy későbbi cikk részeként kerül terítékre. Maguk a mobilszolgáltatások többé-kevésbé megegyeznek a MIS 2002-be beépített szolgáltatásokkal – az SMS kiértesítés kivételével. Ez ugyanis kikerült ebből a verzióból. Nézzük előbb az online hozzáférést. Ez azt jelenti, hogy egy WAP-kliens, hasonlóan az OWA-hoz, valamilyen mobil infrastruktúra *(GSM, GPRS)* plusz HTTPS protokoll felett hozzáfér a postaláda tartalmához. Megnézheti például a leveleket, naptárbejegyzéseket, meghívót küldhet, kontaktokban és a GAL-ban keresgélhet, magyarul a legfontosabb Outlook „akciókat" egy egyszerű mobiltelefonról is elérheti. A MIS 2002-es implementációtól ez egy fontos ponton eltér: a WAP-felületet nem egy ISAPI-alkalmazás, hanem a Mobile Internet Toolkit ASP.NET-es keretrendszere generálja. Ez az első Exchange verzió tehát, amely valamilyen formában használja a .Net runtime szolgáltatásait. Figyelem, ez a jelenség a következő változatban még tovább fokozódik!
Az MMIT bevezetése amellett, hogy nagyságrendekkel egyszerűbb implementálni, egyszerűbbé teszi az organikus módon fejlődő mobileszköz-gyártók újabb és újabb produktumainak támogatását. Az eszközadatbázis egy központi helyen tárolódik, amit a Microsoft folyamatosan bővít a jövőben megjelenő eszközök függvényében.

Most térjünk át a véleményem szerint legfontosabb mobilszolgáltatásra, az ActiveSyncre. Aki esetleg nem ismerné, ez egy szerveroldali szinkronizációs szerviz, ami kétirányú szikronizálásra képes egy ActiveSync kliens és az Exchange között. A szinkronizálás történhet manuálisan, vagy időzítve, illetve az Exchange Server 2003-ban kibővül azzal a lehetőséggel, hogy a szerver adott pillanatokban egy speciális SMS-kiértesítő csomag formájában képes „felébreszteni", és szinkronizálásra késztetni a klienst. Ennek következében a felhasználók az ActiveSyncet használva úgy érzik, hogy „valami" folyamatosan frissen tartja a mobileszköz postaládáját *(„always up-to-date sync")*.
Most persze felmerül a kérdés, hogy milyen infrastruktúrális háttér szükséges egy ilyen szolgáltatás bevezetéséhez, hogyan értesíti ki az Exchange Server 2003 a mobileszköz SIM kártyáját, az eszköz hogyan szinkronizálja le a változásokat? *(... és mibe kerül majd ez nekünk? – a szerk.)* A folyamatot az alábbi ábra szemlélteti.



Az Exchange 2003 'always up-to-date' ActiveSync infrastruktúrája

A felhasználói postaládák változásait egy speciális store event sink figyeli, ami egy bejövő levél esetén elkattan, és zárt SMTP csatornán elküld egy „trigger"-üzenetet a vállalat által preferált szolgáltató felé *(MS Corporate Messaging Service)*. A szolgáltatás ezután továbbpasszolja az „ébresztő"-üzenetet az SMS-központba, ahonnan az lejut a felhasználó egységére *(pl. PocketPC Phone Edition vagy SmartPhone)*. Amennyiben az „alszik" *(stand-by mode)*, felébred és elindítja az ActiveSync klienst, ami – HTTPS csatornán történő hitelesítés után – felkéri a szervert, hogy fésülje össze és küldje le a klienssoldalon frissítésre szoruló változásokat *(levelek, naptárbejegyzések, kontaktok)*.
Egyelőre most ennyit az Exchange Server 2003 mobilszolgáltatásairól, most nézzük meg milyen újdonságok könnyítik meg az üzenetkezelő infrastruktúrák adminisztrátorainak közel sem zökkenőmentes életét!

Rendszerfelügyelet: irány a Windows Server 2003!

Helyszűke miatt sajnos ebből is csak csemegézni van lehetőségem. Gondolom, nem mondok újat, hogy ezen a területen a fő fejlesztési cél a Windows Server 2003 új szolgáltatásainak *(Cross Forest Kerberos Authentication, Shadow Copy, Cluster Service)* minél teljesebb körű kihasználása, és néhány tipikus, az adminisztrátorok életét megnehezítő feladat simulékonnyá tétele volt *(pl. „brick level restore", Mailbox Recovery Center, Message Tracking)*. Érdekes még, hogy a fentiek mellett van néhány, csak az Exchange Server 2003-ban megjelenő technológia. Záráskent két kedvencemet említem meg ezek közül:

- Az első egy új DL típus, a Query Based DL objektum, ami tulajdonképpen egy felhasználói objektumokat előállító LDAP lekérdezés. Segítségével a DL hierarchiákat nem kézzel kell karbantartanunk, elég csak keresőfeltételeket megadnunk, amit az Exchange transzport dinamikusan lefuttat.
- A másikodik kedvencem orvosság egy nagyon is valós problémára, szaknyelven szólva a „szpemmelésre". Ez ugyan eddig is benne volt az Outlook kliensben, ám sajnos kevesen használták. Az Exchange Server 2003-ban ezentúl beépített támogatást kapunk a spamforrások szerveroldali adminisztrálására, megadhatunk például külső, ún „black-hole" szolgáltatókat, ahonnan folyamatosan letölthetjük a „hulladéklevél gyártók" listáját, hasonlóképpen ahhoz, ahogy egy antivírus szerviz frissíti a vírusminta adatbázisát az Internetről.

2003. január 27,
Seattle, WA United States

Bátorfi Zsolt
zbatorfi@microsoft.com

A cikkben szereplő URL-ek:

[1] http://www.microsoft.com/exchange/evaluation/ti/default.asp
[2] http://download.microsoft.com/download/e/d/f/edfdeb7f-289d-4e1b-8f2e-663b8dea68db/etb2gsg_pdf.exe

Microsoft Exchange 2000

Nyilvános mappák replikációja

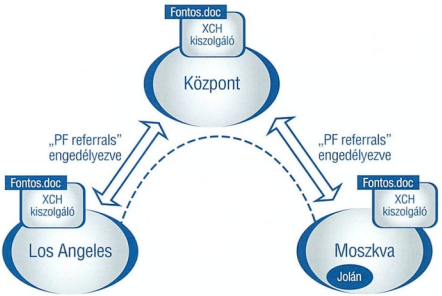
Kis szünet után folytatjuk az Exchange 2000-ről szóló sorozatot. Ebben a részben a nyilvános mappák replikációjával ismerkedhet meg a kedves olvasó. Megvizsgáljuk, mi áll a nyilvános mappák replikációjának hátterében, részletesen foglalkozunk a replikáció folyamatával, a konfliktuskezeléssel, valamint a hibaelhárítással is.

A legtöbb helyen, ahol Exchange-kiszolgálók üzemelnek, a nyilvános mappák adta lehetőségeket is kihasználják. Ahol több kiszolgáló működik, ott máris jelen van a nyilvános mappák replikációja – ha akarjuk, ha nem.
Miért is érdemes a nyilvános mappákat több példányban tárolni?
A nyilvános mappák replikációjával magasabb szintű hibatűrést és rendelkezésreállást tudunk biztosítani. Több telephely esetén a WAN-kapcsolatokat kevésbé terheljük, ha a nyilvános mappákat minden telephelyre lemásoljuk, mintha egy központi helyen próbálnánk őket elérni távoli telephelyekről. Az ügyfelek elsősorban a saját telephelyükön levő Exchange kiszolgálón található nyilvános mappákhoz fordulnak, így mindenki a legközelebbi forrásból olvas, és majd a háttérben történik adatcsere, amely időzíthető.

Ugyanakkor hátrányai is vannak az elosztott rendszernek, hiszen a változások nem jelennek meg azonnal mindenütt, bizony a replikáció időbe telik. Ha túl hosszú az átmeneti idő, gyakran konfliktusok alakulhatnak ki a tartalmat illetően, amelyeket kezelni kell, méghozzá manuálisan. A replikáció is növeli a levélforgalmat a kiszolgálók között, hiszen minden változás levél formájában utazik a kiszolgálók között. Az egyes nyilvános mappákra a többszörözés beállítása a rendszergazdák feladata. Fontos tudni, hogy a legfelső nyilvános mappákra kiadott replikációs beállítás öröklődik a később létrehozott könyvtárakra és objektumokra, de a már meglevőkre nem, azokra minden esetben le kell terjeszteni a beállításokat.
Lehetőség van nemcsak egy Exchange szervezeten belül, hanem különböző szervezetek közt is a nyilvános mappák terjesztésére. Ehhez az Exchange 2000 telepítő CD-n elérhető segédprogramokat lehet használni *(a Support\Exchsync\I386 könyvtárból)*. Ebben cikkben csak a szervezeten belüli nyilvános mappa replikációjával foglalkozunk.

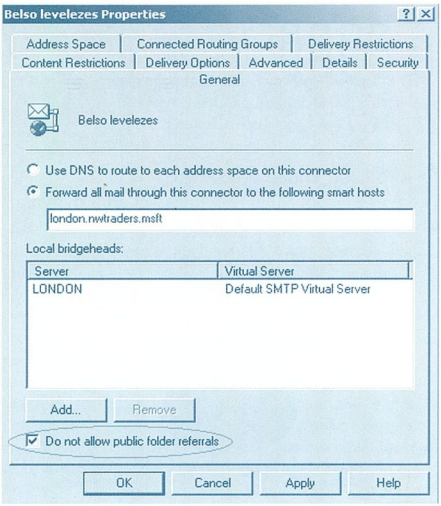
Nyilvános mappák elérése

Hogyan dönti el az ügyfélprogram – mondjuk az Outlook –, hogy melyik kiszolgálóhoz forduljon, amikor egy nyilvános mappában levő levélre van szüksége a felhasználónak?
Lássuk csak: Józer Jolánnak az alábbi képen látható fontos.doc fájlt kellene elérnie a közös mappából.



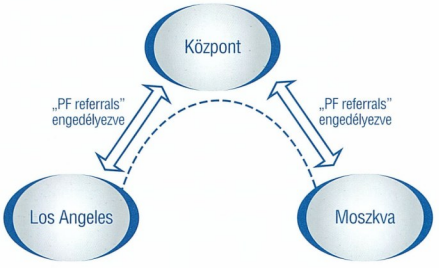
Józer Jolán és a fontos.doc

Józer Jolán postaládája XCH1 kiszolgálón található, Jolán ehhez kapcsolódik, amikor Outlookot használ. Ha a fontos.doc fájl is megtalálható a kiszolgálón, ahol Jolán postaládája van, akkor Jolán máris célba ért, az Information Store rendelkezésre bocsátja a fontos.doc-ot. Ha ugyanez a fájl az azonos útválasztó csoportban levő másik kiszolgálón található, Jolán azt a fájlt is el tudja érni. Ha azonban a fontos.doc egy másik útválasztó csoportban, másik telephelyen található, csak akkor kaphatja meg Jolán az áhított fájlt, ha a „Public Folder Refferrals" - hívhatjuk PF utalásnak - nincs letiltva az útválasztó csoportok közti csatolón. Exchange 2000-ben a PF utalás teszi lehetővé, hogy a nyilvános mappák tartalma útválasztó csoportok közt is elérhető legyen a felhasználók számára.
A „PF referrals" alapértelmezésben engedélyezve van az útválasztócsoportok közt, a csoportok közti csatolók *(Rrouting Group Connector, vagy SMTP connector)* tulajdonságai között lehet letiltani a beállítást.



„PF referrals" tiltása/engedélyezése

Régebbi motorosok az Exchange előző verzióiban, mint „PF Affinity" találkozhattak ezzel a beállítással. Az előző verzióktól eltérően Exchange 2000 esetén a „PF referrals" beállítás tranzitív. Tehát ha a szervezetben legalább három útválasztó csoport van, mondjuk egy központi és két külön telephely, ahogy a képen is az egyik Los Angeles a másik Moszkva, valamint a PF utalás nincs letiltva, Júzer Jolán Moszkvából képes lesz elérni olyan nyilvános mappák tartalmát is, amelyek tulajdonképp Los Angelesben vannak.



Tranzitív „PF referrals"

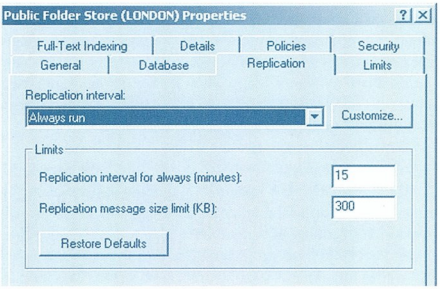
Miért van erre szükség? Mikor Júzer Jolán kapcsolódik a nyilvános mappákhoz az Outlookból, az alapértelmezett jogosultságok mellett az egész nyilvános mappa struktúrát láthatja, mert a hierarchia automatikusan replikálódik. A tartalom azonban nem. A nyilvános mappa utalások segítségével tudják a felhasználók kéréseit a távolabbi helyekről is kielégíteni a kiszolgálók.
Persze ha ez nem szükséges, az a dolga a rendszergazdának, hogy az előző képen is látható jelölőnégyzetet bejelöli az útválasztó csoportok közti csatolón. Ilyenkor Jolán csak a helyben levő mappák tartalmához fog hozzáférni.

Mappák

Az Exchange 2000-ben minden nyilvános mappa-példány egyenrangú. Nincs többé „Home Server", helyette úgynevezett multimaster replikációra váltottak Redmondban.
Az előbb már említettem, hogy a nyilvános mappák hierarchiája *(könyvtárszerkezete)* és a tartalom külön replikálódik. Az elsőként létrejött nyilvános mappa hierarchia – amit úgy is hívunk: MAPI hierarchia-, automatikusan az egész szervezetben replikálódik. A külön létrehozott nyilvános mappa-struktúrákat, mind a tartalmat, mind pedig a hierarchiát „kézzel" kell replikálni, vagyis beállítani a replikációt.
A replikáció folyamata nem túl hatékony módszerrel, levelek formájában történik. Az Information Store egy része, a „Public Folder Replication Agent" – röviden PFRA felelős a replikációs üzenetek küldéséért és a kapott üzenetek feldolgozásáért.

A replikáció beállítása

A replikáció néhány beállítását adatbázisszinten kell megejteni. Beállíthatjuk például, hogy az adott adatbázis tartalma mikor másolódhat. Akkor lehet erre szükség, ha lassú WAN-on keresztül szeretnénk replikációt beállítani.
Ezen kívül beállíthatjuk azt az intervallumot percekben, amikor replikáció elindulhat – ez alapértelmezésben 15 perc.



A replikáció adatbázisszintű beállításai

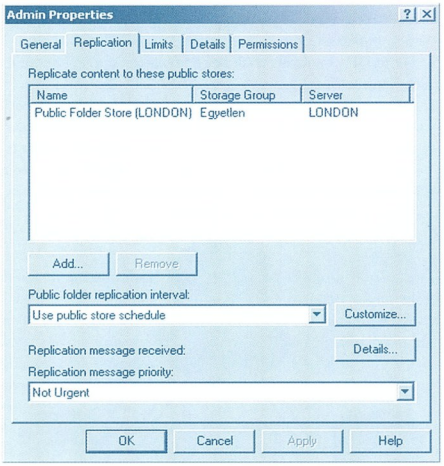
Amikor nyilvános mappa-replikációt tervezünk, érdemes odafigyelni a replikációs levél méretének megválasztására. Ez ugyanis nem a levél méretének maximumát, hanem a minimumát adja meg! Vagyis az itt megadott méretig a PFRA összegyűjtögeti a változásértesítőket, és ha több belefér 300KB-ba, többet küld egyszerre. Ha viszont 10MB-os fájl változik a nyilvános mappákban, a PRFA a változásértesítőbe belerakja az egész fájlt, ráadásul egy replikációs üzenetben fog eljutni a többi kiszolgálóhoz a nagyméretű állomány! Ez az eljárás csupán akkor hatékony, ha sok de nem nagy állományt kell replikálni, ilyenkor nem nőnek nagyra a replikációs üzenetek. A PRFA nem tudja tördelni a fájlokat, ami változik, azt egy az egyben küldi szét a többi kiszolgálónak.
Kicsit javíthatunk ezen az állapoton, ha meghatározzuk az elemek maximális méretét a nyilvános mappákban, vagy ha a replikációt úgy időzítjük, hogy a nagy replikációs levelek ne akadályozzák a normális működést. Ha nem figyelünk, előfordulhat, hogy az átlagos levelezés nagyon lassú lesz, mert épp a nyilvános mappák replikálódnak, lefoglalva a sávszélességet és a kiszolgálók erőforrásait.

Mappaszintű beállítások

Az egyes nyilvános mappákra külön-külön is be lehet állítani néhány dolgot a replikációval kapcsolatosan, illetve van olyan, amit csak és kizárólag a nyilvános mappák szintjén tudunk beállítani. A beállítások minden esetben az egyes nyilvános mappák tulajdonságai közt a Replication tulajdonságlapon találhatók, ahogy az alábbi képen is látható.

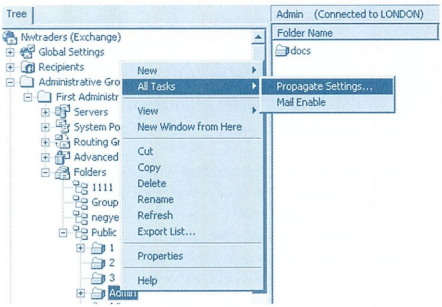
A legfontosabb talán a replikációs partnerek beállítása. Mint már említettem, Exchange 2000 esetén a partnerek egyenrangúak, nincs többé „Home Server" tulajdonság. Akár minden nyilvános mappára külön időzíthetjük a replikációt az alábbi tulajdonságlapon látható „Customize" gomb mögött, de alapbeállításként mindig az adatbázis egészére beállított időzítéssel jut el a tartalom a többi kiszolgálóra.

Ezen kívül állíthatjuk a replikációs üzenetek sürgősségét , illetve megnézhetjük a replikáció állapotát is *(lásd később)*. Most nézzük a beállítások öröklődését, illetve terjesztését!



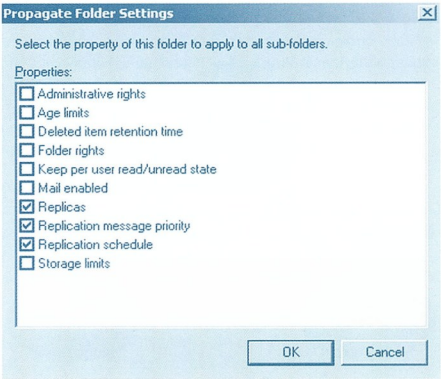
Nyilvános mappák replikációs beállításai

A beállításokat nem öröklik automatikusan a szülőmappák létező gyermekei. Le kell terjeszteni az összes meglevő alsóbb mappára, ha ugyanazokat a replikációs beállításokat szeretnénk látni mindenütt. A létrejövő alsóbb mappák születéskor ugyan öröklik a szülőkönyvtár tulajdonságait, de ha ez később megváltozik, az már csak a kiválasztott mappára lesz érvényes, automatikusan nem öröklődik.



A PF beállítások terjesztése

A beállítások terjesztése *(propagálása)* jól el van dugva a menük közt, de a fenti képen jól látható, honnan lehet előhúzni. Külön lehet a nyilvános mappák különböző beállításait terjeszteni. A replikációnál maradva: külön lehet a replikációban részt vevő kiszolgálók beállítását örökölni – ez a képen is látható „Replicas", külön a replikációs üzenetek sürgősségét, és külön az időzítést is. Persze együtt is lehet terjeszteni a beállításokat, ha mindegyik jelölőnégyzetbe pipát rakunk, ahogy az alábbi képen is látszik.



A replikációs beállítások terjesztése egy menetben

A replikáció folyamata

Érdemes kitérni a replikáció folyamatára, hogy megértsük például az állapotjelzők működését.

Azt már említettem, hogy a replikációért a PRFA vagyis a „Public Folder Replication Agent" felelős. A PRFA nem egy külön szolgáltatás, hanem az Information Store része.

A PRFA feladatai a következők:

- A nyilvános mappa-példányok nyilvántartása
- A replikációs üzenetek képzése és címzése
- A változások figyelése, a beérkezett replikációs üzenetek feldolgozása

De akkor ki küldi a replikációs üzeneteket? Az SMTP kiszolgáló! A replikációs üzenetek winmail.dat bináris mellékletek for-

májában úgynevezett TNEF *(Transport-neutral Encapsulation Format)* formában utaznak a kiszolgálók közt.

A változások követése állapotinformációk *(Message State Information)* alapján történik. Az állapotinformációknak három alkotója van.

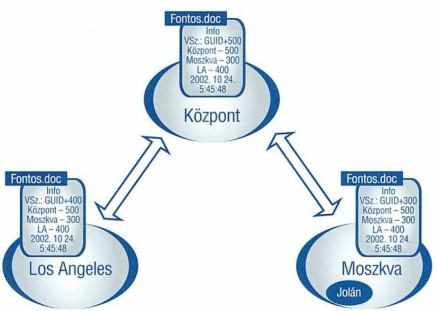
1. Változási szám *(change number)* – egy IS-en belül a változásoknak megfelelően növekvő szám. Egyik része állandó – ez az IS-t azonosító GUID-, a másik része pedig a változáskor mindig növekvő számláló.
2. Változási lista *(predecessor change list)* – minden fájlhoz tartozó lista, amely felsorolja az összes IS-t, amelyben az elem változott *(mellérendelve rögtön a változási számot is)*. Valahogy így nézhet ki:

Moszkva	500
Központ	450
Los Angeles	800
Központ	400
Moszkva	300 stb.

3. Időbélyeg – a létrehozás vagy változás pontos ideje.

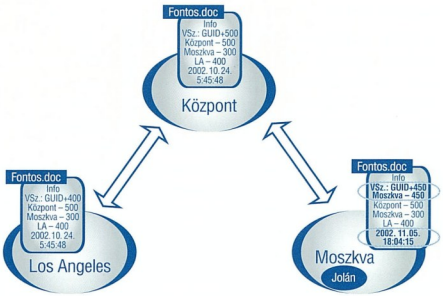
A PRFA az állapotinformációk alapján dönti el, hogy a kapott példány frissebb-e, mint ami helyben megtalálható.

Nézzük a folyamatot, amikor változások replikálódnak a kiszolgálók közt:



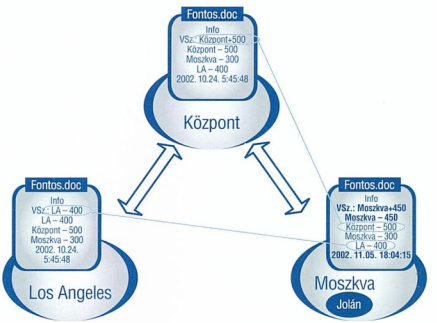
Replikáció előtti kiindulási állapot, amikor minden kiszolgálón ugyanaz a példány szerepel

1. Tovaris Józer Jolán Moszkvában megváltoztatja a nyilvános mappában lévő fontos.doc-ot.
2. A Moszkvában levő PRFA megváltoztatja a fontos.doc állapotinformációit, tehát növeli a változásszámot, frissíti a változáslistát és az időbélyeget is.



Friss állapotinformáció

3. A PRFA létrehozza a replikációs üzenetet, benne a friss változásértesítő és maga a megváltozott fontos.doc szerepel.
4. A PRFA a replikációs beállításoknak megfelelően megcímzi a replikációs üzenetet.
5. Az SMTP kiszolgáló szétküldi a leveleket Los Angelesbe és a Központba is.
6. A fogadó PRFA – mondjuk a Központban – kibontja a replikációs üzenetet. Ha a kapott állapotinformációban levő változási listában szerepel a helyben levő elem változásszáma, úgy érzi, hogy bizony ez egy frissebb elem, és a helyben levő fontos.doc-ot lecseréli a kapott fontos.doc-ra, és az állapotinformációkat is frissíti.



Melyik példány a frissebb?

Jelen esetben a Központba érkező fontos.doc-hoz tartozó változáslistában szerepel a Központban, helyben levő változási szám *(Központ – 500)* ezért a központi PRFA le fogja cserélni a helyben levő fontos.doc-ot a kapottra, majd frissíti az állapotinformációt. Ugyanez történik Los Angelesben is.

A replikáció állapotát a System Managerrel lehet nézegetni. *(Sajnos azonban a replikáció működése miatt nem mindig a valódi állapotot láthatjuk.)*

A replikáció állapotára kétféle jelzőt használ az Exchange:

- **In Sync** – amikor azt gondoljuk, szinkronban vannak a példányok. *(Pedig ez csak azt jelenti, hogy az utolsó változásértesítőket a PRFA elkészítette, és az SMTP kiszolgálónak lepasszolta, és azóta a helybeli példány nem változott. Nem tudhatja a PRFA, hogy a változá-*

sok valóban megérkeztek-e a többi kiszolgálóhoz! Nincs visszajelzés a replikációról.)

- **Local Modified** –a helyben levő példány változott, de a PRFA még nem kürtölte szét a változást.

Viszonylag könnyen előállhat olyan állapot, hogy a System Managerben azt látjuk, szinkronban vannak a nyilvános mappák példányai, de valójában mégsem. Ilyenkor leginkább a kiszolgálók közti üzenetforgalmazást kell ellenőrizni – tehát az SMTP kiszolgálót.

A rendszermappák replikációja

Vannak olyan nyilvános mappák, amelyek a hierarchiában nem látszanak, de a tartalmuk igen fontos. A System Managerben láthatók ezek, ha a Folders konténerben a Public Folders hierarchián állva a gyorsmenüből *(jobb klikk)* a „View System Folders" parancsot választjuk. A következő rendszermappákat találhatjuk itt:

- Events Root: – az Exchange 5.5 kompatibilis scripteket tartalmazza
- EFORMS REGISTRY – Outlookkal használható sablonok tárolója az „Organization Froms Library" tartalma
- OFFLINE ADDRESS BOOK – helyi címlisták tárolója
- schedule+ free/busy – a felhasználók és erőforrások elfoglaltsági információi találhatók itt
- Schema – nyilvános mappák és tartalmuk tulajdonságainak leírása található ebben a mappában
- StoreEvents – Exchange 2000 formátumú scriptek.

A sok rendszermappa közül a Offline Address Bookot és a Free/Busy mappát emelném ki. Mindkét mappa csak az adminisztratív csoportban elsőként létrehozott kiszolgálón található meg, viszont szükség lehet rá bármelyik kiszolgálón. Nem ideális, ha csak egyetlen kiszolgálón tároljuk ezeket, ha ez a kiszolgáló épp nem működik: az Outlookot használók hibaüzeneteket kapnak, amikor egy találkozót próbálnak összehozni, és ehhez megfelelő időpontot keresnek a többiek elfoglaltsága alapján. Mind a két rendszermappa tartalmát érdemes minden kiszolgálóra elterjeszteni. Ez minden esetben a rendszergazdák dolga, nem történik meg automatikusan! Ha több adminisztratív csoport van egy szervezeten belül, érdemes keresztbereplikálni ezen mappák tartalmát, hogy mindenkor elérhető legyen a címlista és az elfoglaltságok is.

A konfliktusok kezelése

Mi a helyzet, ha a változások nem jutnak el egyik kiszolgálóról a másikra, vagyis nincs benne a kapott változási listában a helyben levő példány utolsó változási száma? Példánkhoz visszatérve két dolog lehetséges:

1. A Központban is megváltoztatta valaki a fontos.doc-ot és a változások még nem replikálódtak
2. Nem jutottak el Moszkvába a Központból a változásértesítők miközben a fontos.doc többször is változott.

Az első esetben manuális konfliktuskezelés szükséges, a másodikban pedig elindul automatikusan a „backfill" – vagy hívjuk csak pótlásnak.

A változási listák hivatottak megmondani, hogy frissebb-e a kapott példány a meglevőnél vagy sem. A PRFA ezt úgy próbálja eldönteni, hogy a kapott változáslistában megkeresi a helyben levő példány utolsó változási számát. Ha megtalálja, biztos lehet benne, hogy a kapott példány frissebb, mint a helyben levő. Konfliktushelyzet akkor alakul ki, ha ugyanazt a fájlt közel azonos időpontban két helyen is módosították, és ezek a változások mennek szembe egymással. Mindegyik úgy érzi, ő a legfrissebb példány, de egyik PRFA sem fogadja be a kapott fájlt, mert nem találja a legfrissebb változási számot a kapott változási listában, hisz az még nem érhetett oda.

Ilyen esetben valahogy el kell dönteni, melyik a megfelelő példány. Egyik PRFA sem tudja eldönteni, ehelyett a nyilvános mappa kapcsolattartójához és a módosítást végző személyeknek *(Contact)* elküldi mind a két példányt. A konfliktuskezelés manuális. Az egyik felhasználó dönt az érvényes példányról – vagy nem dönt, és a konfliktus fennmarad.

Nyilvános mappák mozgatása

Ha mondjuk Moszkvából a Központba szeretnénk mozgatni bizonyos nyilvános mappákat és tartalmukat, tulajdonképp ugyanúgy a replikációt kell elindítani először, mintha csak terjeszteni szeretnénk a tartalmat.

Sikeres replikáció után pedig a példányokat tartalmazó listából egyszerűen el kell távolítani a moszkvai kiszolgálót és kész. Igazából nem szükséges megvárni a replikáció befejeződését, mert csak a replikáció befejeztével fogja az IS letörölni a nem kívánt példányt, bár ilyenkor a replikáció alatt nem érhető el a nyilvános mappák tartalma. Mégis érdemes először replikálni, és csak aztán kivenni a kiszolgálót a listából, mert ez *(talán)* biztonságosabb. Replikáció után véletlenül se töröljük a kiszolgálóról a nyilvános mappát, hisz a törlés is egy változás, ami replikálódni fog a többi példányhoz. Nyilvános mappa mozgatásakor a kiszolgálók listáját változtatjuk a nyilvános mappa replikációs tulajdonságlapján!

Amikor törlünk egy nyilvános mappát, az IS törli a tartalmát, majd a PRFA változásértesítőt küld a többi kiszolgálónak, és minden kiszolgáló maga törli le a helyi példányát.

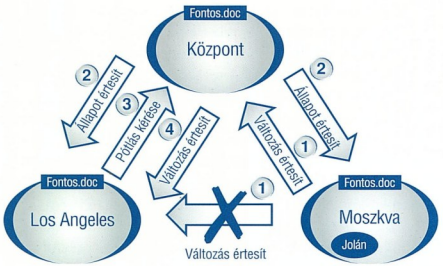
A pótlás *(Backfill)* folyamata

A kiszolgálók nem tudják hogy az általuk küldött replikációs üzenetek valójában megérkeztek-e a többi kiszolgálóhoz, nincs semmilyen visszacsatolás. Amint a PRFA átadta az SMTP-nek az üzeneteket, azonnal In Sync állapotba teszi a nyilvános mappa állapotát.

A vakság miatt akkor is mennek a nyilvános mappákról állapotüzenetek *(naponta)*, ha nem változott semmi. Ha a helyben levő változási lista tartalmazza a kapott változási számot, ez azt jelenti, tényleg szinkronban van a nyilvános mappák tartalma. Ha a kapott változási szám nem szerepel helyben, a küldőhöz visszamegy egy kérés, hogy pótolni kellene az újabb fájlokat. Az utolsó változási számok alapján a kiszolgálók tartalma szinkronizálódik.

A pótlási folyamat akkor indulhat el,

- ha egy kiszolgálót régebbi mentésből állítottunk vissza,
- ha hosszabb ideig nem volt hálózati kapcsolat a kiszolgálók között
- ha elvesztek a replikációs üzenetek az éterben.



Utólagos feltöltés folyamata

Mondjuk Jolán Moszkvában megváltoztatja a fontos.doc fájlt.

1. A moszkvai kiszolgáló változás-értesítőt küld minden kiszolgálóhoz, de ez Los Angelesbe nem jut el, persze erről Moszkva mit sem tud.
2. A Központig eljut a változás, idővel a központi kiszolgáló állapotértesítőt küld szét a fontos.doc-ról akkor is, ha nem változott.
3. Moszkvában minden rendben, de LA észreveszi, hogy a központban frissebb verzió van, mint nála, ezért kéri a pótlást.
4. A központi kiszolgáló elküldi a frissebb fájlt.

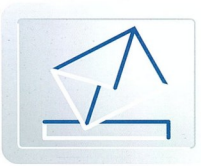
De mi van, ha Jolán véletlenül letörli a fontos.doc-ot a moszkvai kiszolgálóról? A rendszergarázda fogja, és visszaállítja a nyilvános mappa-adatbázist a tegnapi mentésből. Egy idő után a mentés óta változott fájlok replikálódnak a visszaállított kiszolgálóra, mert a backfill automatikusan beindul. Ennek eredményeképp a fontos.doc is egyszer csak ismét eltűnik a nyilvános mappából. Mit lehet tenni ez ellen? Például nem az éles kiszolgálóra állítjuk vissza a nyilvános mappa adatbázist, hanem egy elkülönített masinára, majd onnan exportáljuk pst-be, és importáljuk az éles kiszolgálóra. Kicsit körülményes, de működik.

Folytatjuk…

Dorner Csilla
dorner.csilla@netacademia.net
A szerző a NetAcademia oktatója, MCSE

Kapcsolódó tanfolyamaink:
1572 – Exchange 2000 Server üzemeltetése

Az Exchange WSS elérése az SPS keresőmotorjával



Az Exchange 2000 és a SharePoint Portal Server külön-külön is rengeteg előnnyel és funkcióval rendelkezik. Ebben a cikkben azt mutatom be, hogy milyen lehetőségek tárulnak elénk, ha a két szervert „összeházasítjuk".

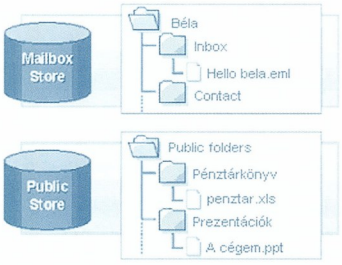
Az Exchange2000 Server és a Web Storage System

Az elmúlt évben néhány cikken keresztül részletesen kitértem arra, hogy a WSS adatai hogyan, milyen módszerekkel érhetők el saját alkalmazásunkból. Ezeket most nem kívánom újra megismételni, azonban azon Olvasók kedvéért, akik az említett cikkeket nem olvasták és nem járatosak a témában, néhány szóban bemutatnám, miről is van szó.

A WSS *(Web Storage System)* az Exchange2000 Server *(és a SharePoint Portal Server)* adattárolási mechanizmusa, amely sokoldalú hozzáférést biztosít az adatokhoz *(HTTP, SMTP, IMAP4, POP3, stb.)*. Felépítése hierarchikus: a különböző fájlok, dokumentumok mappákba szervezhetők, s ezek a mappák *(folderek)* tetszőleges mélységig rekurzívan egymásba ágyazhatók.

A kétféle store-típus közül a Mailbox store olyan adatbázis, amelynek mappáihoz és dokumentumaihoz egyetlen felhasználó, a tulajdonos fér hozzá. E-maileket, hozzájuk csatolt állományokat, mappákat, dokumentumokat stb. tárolhatunk itt. Természetesen minden felhasználóhoz egyéni Mailbox store rendelhető.

A Public store legfőképpen abban különbözik a Mailbox store-tól, hogy nem egyetlen felhasználó kizárólagos tulajdona, megosztottan többen is hozzáférhetnek, használhatják, rendelkezhetnek tartalma fölött – persze a megfelelő jogosultságok alapján.



A Web Storage System felépítése

A SharePoint Portal Server és a Web Storage System

A SharePoint Portal Server hasonló módon tárolja adatait, természetesen más céllal: itt nem a levelezésen és az ahhoz tartozó funkciókon, hanem a dokumentumkezelésen van a hangsúly.

Addig beszélünk dokumentumkezelésről, amíg a végfelhasználó a Word, Excel, Access és egyéb dokumentumait szerkeszti a számítógépén. A tevékenység kapcsán sok egyéb mellett olyan kérdések merülnek fel, mint a dokumentumoknak központi tárba történő mentése, ottani tárolása és nyilvántartása, a felhasználó által onnan való kikölcsönzése és ezen idő alatt mások számára való zárolása, a dokumentum verzióinak kezelése, a dokumentumnak a külvilág számára történő publikációjának támogatása, vagy a jóváhagyási procedúra. E folyamatok közben a dokumentum tartalma megváltozhat.

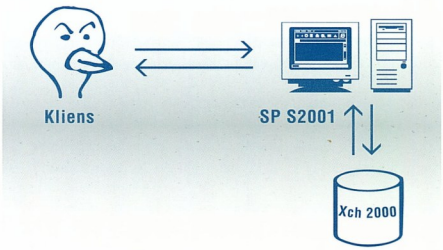
A SharePoint Portal Server *(SPPS)* e funkciókat hivatott megvalósítani, háttérben a WSS-sel, ahol az elsődleges adattárolás és -feldolgozás történik.

SPPS-ben az információk, adatok, dokumentumok tárolására a tartalomforrások *(content source)* szolgálnak. Kliensoldalon mindezt egy URL segítségével érhetjük el. Tartalomforrásként nemcsak a WSS mappái, hanem egyéb típusú helyek is megadhatók:

- weboldalak
- megosztott file-ok
- Exchange2000 Public Folderek
- Exchange 5.5 Public Folderek
- Lotus Notes adatbázisok
- SPPS munkaállomások *(akár távoliak is)*.

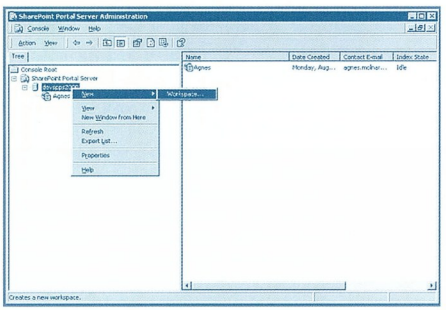
(Amennyiben olyan tartalomforrás-típust szeretnénk elérni, amelyet nem támogat a SharePoint Portal Server, az interfészek kiterjeszthetők, azaz saját protokollkezelőket is létrehozhatunk.)

Az SPPS adatai különböző módszerekkel érhetők el. A fenti források közül most azt szeretném bemutatni, hogy egy Exchange2000 mappa hogyan indexelhető be, illetve kezelhető az SPPS keresőmotorjával. A kérést tehát az SPPS kapja kliensünktől, és ő az indexei alapján küldi a választ – de az Exchange2000 Web Storage System alapján.



A lekérdezés menete az SPPS keresőmotorjával

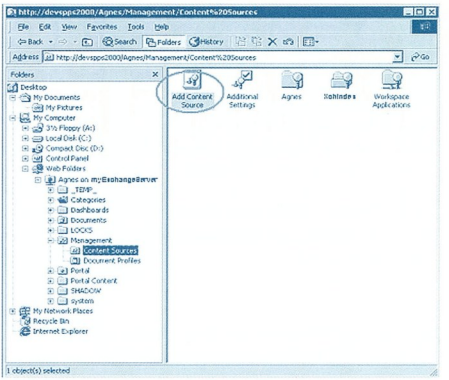
Lássuk tehát, mi mindent kell beállítanunk és elvégeznünk a fenti architektúra megvalósítása érdekében. Először is hozzunk létre egy munkaállomást *(workspace)* az SPPS-en, és adjunk neki egy találó, fantáziadús nevet, legyen például Agnes. Ezt az SPPS adminisztrációs eszközével tehetjük meg *(Start → Programs → Administrative Tools → SharePoint Portal Server Administration)*. A bal oldali ablakrészben a fából válasszuk ki a kiszolgálónkat, majd jobb egérgombbal rákattintva a New → Workspace menüpontot válasszuk ki, és máris indul a varázsló.



Új workspace létrehozása SharePoint Portal Serveren

Következő lépésként nyissuk meg ezt a munkaállomást *(My Computer → Web Folders)*. Ezzel a SharePoint Portal Server már alkalmas arra, hogy rajta különböző tartalomforrásokat hozzunk létre.

Ahhoz, hogy egy Exchange2000 WSS mappa *(vagy bármely más típusú tartalomforrás)* tartalmát elérhessük SPPS-en keresztül, annak tartalmát be kell indexelni valamely munkaállomás alá *(ennek küldjük majd később a hozzáférési kéréseket a kliensről)*. Ennek érdekében a megnyitott munkaállomás Management\Content Sources ágát kifejtve klikkeljünk az `0` ikonra *(lásd az alábbi ábrát)*.
Ekkor egy varázsló indul el, amelynek segítségével különböző tartalomforrásokat tehetünk elérhetővé. Első lépésként ennek típusát kell kiválasztani, azaz a fent említett lista valamely elemét. Válasszuk ki az „Exchange 2000 Server public folders" opciót. Második lépésként az elérendő Web Store mappa URL-jét kell megadni *(legyen ez például* `http://myExchange2000server/public tree/myfolder/`*)*. Adjuk meg, hogy a mappát milyen mélységig szeretnénk indexelni: csak az adott mappát *(Only this folder)* vagy minden almappájával együtt *(This folder and all subfolders)*. Végül adjunk egy nevet az indexmappának *(pl. myXchINdex)*, majd indítsuk el az indexelést. Ez néhány percet biztosan igénybe vesz, közben nyugodtan olvassuk tovább ezt a cikket .



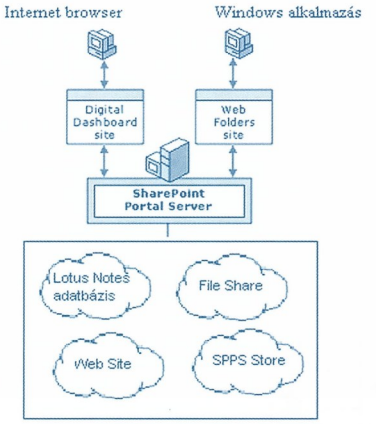
Tartalomforrás hozzáadása

Ha kész az indexmappánk, azon jobb egérkattintással további dolgokat állíthatunk be. Ezek közül az egyik legfontosabb lehetőség a frissítés ütemezése, melynek két típusa létezik. A teljes frissítés minden alkalommal a teljes Exchange-mappán *(és a beállításoknak megfelelően annak összes almappáján)* végigmegy, és annak minden elemét teljes egészében újraindexeli az SPPS-en, függetlenül attól, hogy azon történt-e változás a legutóbbi frissítés óta. Ezzel szemben inkrementális frissítés esetén csak a módosítások, illetve az új dokumentumok kerülnek át az indexbe, ezáltal csökkentve egyrészt az adatforgalmat, másrészt a frissítésre fordítandó időt is.
Az adaptív frissítés az inkrementális frissítés egyik válfaja, amely különböző statisztikákat, heurisztikákat használ a hatékonyság növelésének érdekében. Rögzíti például, hogy milyen gyakran változik a tartalom, és ennek függvényében végzi a frissítéseket.
Az utolsó, leghatékonyabb módszer az értesítő frissítés, amikor is változás esetén a tartalomforrás értesítést küld az SPPS kiszolgáló indexének. Ezen esemény hatására indul el a frissítés a tartalomforrás indexén. Az SPPS ezt használja alapértelmezett beállításként, hátránya viszont, hogy csak az SPPS saját store-jában illetve NTFS fájlrendszerben elhelyezett dokumentumokra használható.

A frissítés típusán kívül az időzítést is beállíthatjuk *(milyen időközönként induljon automatikusan az index frissítése)*, illetve különböző szabályokat is megadhatunk az indexek létrehozásához. Háromféle szabálytípust különböztethetünk meg:

1. *„Site path" szabályok:* az indexbe felvehető dokumentumok számának korlátozására *(maximum hány dokumentum szerepelhet az indexben)*.
2. *„Mapping" szabályok:* Hogyan jelenítse meg az SPPS a keresések eredményeit, és hogyan férhessenek hozzá a felhasználók a tartalomhoz az indexelés után.
3. *„File type" szabályok:* Különböző fájltípusok bevétele az indexbe, illetve kihagyása onnan *(például a* `.txt` *és* `.doc` *típusú dokumentumokat indexelje, de az* `.exe` *és* `.dll` *kiterjesztésű fájlokat ne)*. Csak a munkaállomáson kívüli fájlokra érvényes.

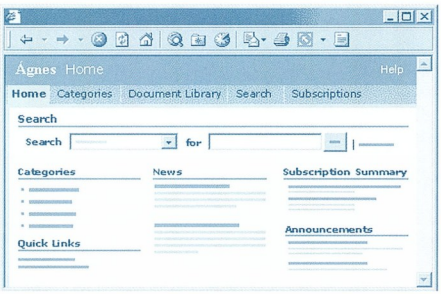
Az alábbi ábra a ShrePoint Portal Server elérési környezetét mutatja:



Tartalomforrások

Az SPPS környezete

Az ábra alsó harmadában a különböző tartalomforrásokat láthatjuk, amelyeket a SharePoint Portal Server le tud indexelni. A szerver szolgáltatásai kétféle módon érhetők el. Egyrészt a digitális műszerfalon *(digital dashboard)* keresztül, ami egy webes felületet biztosít, amelyen keresztül elérhetjük a SPPS munkaállomásait *(egyszerűen csak írjuk be a böngészőbe a megfelelő címet, fenti példánkban ez* `http://mySPPSserver/Agnes`*)*.



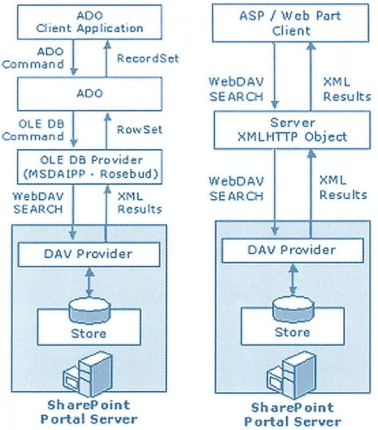
Az SPPS dashboard site

A másik elérési mód az, ha az SPPS adattárait webes mappaként fogjuk fel, és a Windows-alkalmazásokban saját URL-jükkel hivatkozunk rájuk. A továbbiakban ez utóbbi módszerre térek ki részletesen.

Az SPPS adatainak elérése programozott módon

Az SPPS keresőmotorja ADO-n és WebDAV-on keresztül érhető el.

Az alábbi ábrán jól látható, hogy a szerverhez mindkét esetben XML formátumú kérés jut, és a választ is ebben a formátumban kapjuk. ADO esetében az OLEDB Provider feladata a kérés és a válasz feldolgozása, így ő játssza a „közvetítő" szerepét a kliens és a szerver között. WebDAV esetén közvetlenül az XML formátumú kérést adjuk ki, így a hozzáférés sokkal gyorsabb és hatékonyabb lehet.



Az SPPS elérése ADO-val és WebDAV-val

A módszer megválasztásán túl egyéb megfontolások alapján is növelhetjük keresésünk hatékonyságát. Először is a szabadszöveges, illetve a nagy mélységben, vagy távoli kiszolgálón történő keresések mind nagyban növelik a végrehajtási időt. Szintén lassítja a lefutást a különböző metaadatokra történő keresés, hiszen a jellemzők értékét az úgynevezett „property store"-ból kell kiolvasni *(bár ez cache-elhető a memóriába, ami jelentősen gyorsíthatja a futást)*. Ha ésszerűen korlátozzuk az eredményhalmaz méretét, azaz azt, hogy a keresés maximum hány találatot eredményezzen, lényegesen növelhetjük a hatékonyságot. A keresőmotor számára azt is megszabhatjuk, hogy legfeljebb mennyi ideig próbálkozzon a lekérdezés végrehajtásával. A rendezés *(ORDER BY)* viszont nincs hatással a lekérdezés idejére, hiszen vele egy időben kerül végrehajtásra.

Fontos tényező, hogy keresések esetén a különböző attribútumokhoz különböző súlyokat rendelhetünk. A nagyobb súllyal rendelkező attribútumok számunkra „fontosabb" információt jelölnek.

Például az alábbi kódrészlet azt mutatja, hogy a `Subject` jellemző súlya nagyobb *(0.8)*, mint a `Title` súlya *(0.5)*.

```
// ...
"urn:schemas-microsoft-com:office:office#
 Subject":0.8,
"urn:schemas-microsoft-com:office:office#
 Title":0.5,
// ...
```

Az Exchange2000Server WSS-ének elérése az SPPS keresőmotorjának segítségével

Ennyi bevezető után lássuk, miként „házasíthatjuk" össze a SharePoint Portal Server keresőmotorját az Exchange2000 Server Web Storage Systemében tárolt adatokkal. *(Az SPPS WSS hasonlóan érhető el, a különbségekről az [1] címen található bővebb információ.)*

A programrészleteket C#-ban adom meg, azt prezentálva, hogy .NET alatt hogyan oldható meg a probléma. A kód magja, a WebDAV-kérés azonban bármilyen platformról továbbítható a kiszolgáló felé.

Ha az indexelést a fentiek alapján sikeresen elvégeztük, máris megteremtettük az alapfeltételeket, így koncentrálhatunk a feladat lényegére. Programunkban először is adjuk meg az SPPS-munkaállomás és az alá indexelt WSS mappa elérési útvonalát:

```
// SPPS munkaállomás elérési útvonala:
string sURL = "http://mySPPSserver/Agnes/";

// Az Exchange2000 WSS mappájának
// elérési útvonala:
string location =
"http://myExchange2000server/public tree/
myfolder/";
```

Következő feladatunk a WebDAV-lekérdezés megadása:

```
StringBuilder sQuery = new StringBuilder();

sQuery.Append(@"<?xml version='1.0'?>");
sQuery.Append("<g:searchrequest xmlns:g='DAV:'>");
sQuery.Append("<g:sql>");
sQuery.Append("SELECT \"DAV:href\",
\"DAV:displayname\"");
sQuery.Append("FROM SCOPE('SHALLOW TRAVERSAL OF
\"");
sQuery.Append(location);
sQuery.Append("\"')");
sQuery.Append("</g:sql>");
sQuery.Append("</g:searchrequest>");
```

A szokásos SELECT lekérdezésről van tehát szó. A SHALLOW TRAVERSAL kulcsszóval adjuk meg, hogy csak magában a WSS mappában keressen, annak almappáiban ne. Ha ez utóbbi funkcióra lenne szükségünk, használjuk a DEEP TRAVERSAL kulcsszót. Figyeljük meg, hogy itt még csak az Exchange mappát jelöltük ki, azaz a keresendő információ forrását, hogy kinek küldjük a lekérdezést, az itt még nem jelenik meg. Így ugyanezt a lekérdezést akár magának az Exchange2000 kiszolgálónak is elküldhetnénk. A SharePoint Portal Server keresőmotorja azonban szélesebb körű szolgáltatásokat nyújt számunkra *(ilyen például a szabadszöveges keresés az összes mappatípusra)*, ezért esetenként célszerűbb lehet az összetett megoldás a WSS-beli adatok elérésére.

Most következhet a kérés elküldése a megfelelő helyre, esetünkben az SPPS Agnes nevű munkaállomásának:

```
try
{
WebRequest request = WebRequest.Create(sURL);
request.Method = "SEARCH";
request.Credentials = new
NetworkCredential("user", "password", sURL);
request.ContentType = "text/xml";
request.Headers["Depth"] = "1,noroot";
request.ContentLength = sQuery.Length;

StreamWriter writer = new
StreamWriter(request.GetRequestStream());
writer.Write(sQuery);
writer.Close();

WebResponse response = request.GetResponse();
Stream stream = response.GetResponseStream();
}
catch
{
hibaUzenet = „Nem sikerult...\n";
hibaUzenet += „Ágі szabadságon van.\n";
hibaUzenet += „Anyukája e-mailcíme nem publikus.";
}
```

A kiszolgálóhoz történő hozzáférést érdemes egy `try..catch` blokkban elhelyezni, hiszen itt különösen sok a hibalehetőségek száma *(pl. a hálózati forgalomnak köszönhetően)*.

A WebRequest kérést a következőképpen paraméterezzük: a kérést az sURL *(azaz a* `http://mySPPSserver/Agnes`*)* címre küldjük majd, amely esetünkben egy SPPS munkaállomás. A metódus ezúttal keresés *(SEARCH)*, és a `user` nevű felhasználó, `password` jelszavával kérünk beléptetést. Nagyon fontos még a kérés hosszának megadása is *(*`request.ContentLength`*)*, hiszen anélkül *(vagy annak rossz beállításával)* hibaüzenetet kapunk vissza a szervertől.

Ha mindezzel megvagyunk, egy `StreamWriter`-t hozunk létre, és ennek segítségével küldjük el az sQuery kérést az SPPS kiszolgálónak, a fenti beállításoknak megfelelően.

A választ egy `WebResponse` típusú változóba kapjuk, amelyet a `GetResponseStream` metódussal egyszerűen streammé alakíthatunk, amely a továbbiakban tetszőlegesen feldolgozható. *(Erre már láthattunk példát korábbi, Exchange2000 Web Storage Systemmel foglalkozó cikkemben).*

Összegzésül

Mindezzel tehát elértük azt, hogy az Exchange2000 Server keresőmotorját a SharePoint Portal Server keresőmotorjával helyettesítjük, ezáltal az előbbi funkcionalitását kibővítjük néhány újabb lehetőséggel. Természetesen mint mindennek, ennek is ára van: az SPPS-keresések ugyan különböző módszerekkel gyorsíthatók, azonban számolnunk kell a két kiszolgáló közötti hálózati forgalom sebességcsökkentő hatásával. Helyi hálózatok esetében persze ez nem számottevő, azonban nagy adatmennyiség vagy távoli kiszolgálók esetében ezt is mérlegelni kell alkalmazásunk infrastruktúrájának tervezésekor.

Molnár Ágnes
agnes.molnar@t-systems.com

A cikkben szereplő URL-ek:

[1] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wss/wss/sps_store_diffs.asp

Változó dimenziók kezelése

A vállalati adatraktárak időben általában stabilak, azaz többnyire csak új adatokkal bővülnek; a tények elemzésére szolgáló dimenziók azonban idővel megváltozhatnak. Az eladások adatait elemezhetjük például a „Kereskedők" dimenzió szerint. A cég kereskedőinek adatai azonban személyes okokból, vagy szervezeti átalakítás következtében megváltozhatnak. Az ilyen dimenziót szokás „lassan változó dimenziónak" nevezni.

A lassan változó dimenziók kezelésének két elterjedt módja a felülírás (*„restate history"*) és a változásnaplózás (*„track history"*). Ralph Kimball a „The Data Warehouse Toolkit" című híres könyvében ezeket a dimenziókat 1-es és 2-es típusúnak nevezte. [1]

Példa lassan változó dimenzióra

Az előzőekben említett „Kereskedők" dimenzió táblája egyszerűsített formában a következő lehet:

Kulcs	ID	Terület	Kezd
1	1239	Kelet	1999-01-01
2	8328	Kelet	2000-07-01
3	4290	Nyugat	2000-12-01

Az „ID" oszlop a kereskedő „természetes" azonosítója, amely állandó a foglalkoztatás ideje alatt, és célszerűen nem allokáljuk más alkalmazottnak még akkor sem, ha jelenlegi birtokosa már elhagyta a céget. A „Kulcs" oszlop az a mesterséges azonosító, amellyel ez a dimenziótábla a ténytáblához kapcsolódik.

A valóságban ez a dimenziótábla sok egyéb oszlopot is tartalmaz, például a kereskedő nevét és egyéb személyes adatait. A személyes adatok változása többnyire 1-es típusú: nyugodtan felülírhatjuk a régi értéket az újjal.

Mi történjen, ha a cégen belüli átszervezés következtében az egyik kereskedő területe megváltozik?

Ha felülírjuk a dimenziótábla adatait és így illesztjük össze a ténytáblával, a 8328 azonosítójú kereskedő korábbi eladásai is hirtelen egy másik körzetbe kerülnek.

Ez többnyire helytelen, mert meghamisítja a területi eladások adatait. A helyes megoldás a változásnaplózás.

Kulcs	ID	Terület	Kezd	Befejez	Aktuális?
1	1239	Kelet	1999-01-01	NULL	Igen
2	8328	Kelet	2000-07-01	2002-11-24	nem
3	4290	Nyugat	2000-12-01	NULL	Igen
4	8328	Nyugat	2002-11-25	NULL	igen

A dimenziótáblát egy új sorral bővítettük, amely az új helyzetet mutatja. A ténytáblában a kereskedő korábbi eladásai a 2-es kulccsal szerepelnek, az új eladások a 4-es kulcshoz fognak tartozni.

Felülírás és változásnaplózás egyszerre

Olykor felmerül az igény, hogy egy dimenzió esetén mindkét változáskövetési módszert megvalósítsuk.

A példában változásnaplózás megfelelő a területre történő felösszegzés szempontjából, de nem jó a kereskedőnek, akinek a prémiuma az eladásoktól függ. Ha gyorsan szeretnénk megtalálni, hogy a különböző kulcsok közül melyek azonosítják ugyanazt a személyt, célszerű felvenni még egy oszlopot, amely az adott ID-hez tartozó első kulcsot tartalmazza. Redundáns táblát kapunk, hiszen az [ID] és a [Kezdés] oszlopok alapján az [Első Kulcs] előállítható, de ezt a redundanciát a későbbi feldolgozási sebesség érdekében vállaljuk.

Kulcs	ID	Terület	Kezd	Befejez	Aktuális?	Első Kulcs
1	1239	Kelet	1999-01-01	NULL	igen	1
2	8328	Kelet	2000-07-01	2002-11-24	nem	2
3	4290	Nyugat	2000-12-01	NULL	igen	3
4	8328	Nyugat	2002-11-25	NULL	igen	2

A tábla változatlanul a historikus megközelítést mutatja, de könnyen építhetünk rá egy aktuális nézetet.

```
CREATE VIEW [KereskedőkAktuális] AS
SELECT [Első kulcs], [ID], [Terület], stb.
FROM [Kereskedők]
WHERE [Aktuális?] = 'igen'
```

Ha a ténytáblát hozzá tudjuk kapcsolni a [KereskedőkAktuális] viewhoz, készen is vagyunk. Kisebb méretű ténytábla esetén egyszerűen felvehetünk egy új idegen kulcsot tartalmazó oszlopot. Nagy, sok milliárd soros ténytábla esetén azonban célszerű takarékoskodni a hellyel. Minden bájt, ami hozzáadódik a ténytábla szélességéhez, megszorzódik a sorok számával. A megoldás ismét egy view lehet:

```
CREATE VIEW [TényKétNézetben] AS
SELECT [Tény].*, [Kereskedők].[Első Kulcs]
FROM [Tény]
INNER JOIN [Kereskedők]
ON [Tény].[Kulcs] = [Kereskedők].[Kulcs]
```

Az Analysis Servicesben ezután két dimenziót készíthetünk. A [Kereskedők] tábla alapján készített dimenzió a változásnaplózást mutatja. A [KereskedőkAktuális] view alapján készített dimenzió pedig a felülírással aktualizált állapotnak felel meg. *(Ez utóbbi dimenziót célszerű „Changing Dimension"-nek deklarálni.)* Az OLAP-kockánk ténytáblája a [TényKétNézetben] view lesz. A viewban szereplő join egy kicsit lassíthatja a feldolgozást, de nem jelentősen. Ha a többi dimenzió miatti joinokat „kioptimalizáljuk" , a kocka feldolgozása még így is nagyon gyors lesz.

Joy Mundy (Microsoft) írása alapján

Kószó Károly
karolyk@microsoft.com

A cikkben szereplő URL:

[1] http://www.ralphkimball.com

Feltörhetetlen bejelentkezés?

Challenge/Response, Kerberos és Smart Card logon

A szoftveripar nagyjai már sokszor megígérték nekünk, hogy egyszer s mindenkorra véget vetnek a bejelentkezési *(autentikációs)* adatok lopkodásának. Ismerjük a LanMan/NTLM páros szomorú sorsát *(L0PHTCrack)*, s most már az utóbbi években folyamatosan magasztalt Kerberost is utolérte a sorsa *(Kerbcrack)*! Egyetlen megoldási lehetőségnek a jelszavas bejelentkezés kiiktatása, elfelejtése látszik. Az intelligens kártyás bejelentkezés az új ígéret. Ez azonban – elődjeitől eltérően – be fogja váltani a hozzáfűzött reményeket, mert...

Mielőtt belemerülnénk a kártya dugdosásába, röviden tekintsük át mind az NTLM Challenge/Response, mind a Kerberos-protokollt. Előbbit azért, hogy lássuk, miért volt nála százszor jobb a Kerberos V5. Utóbbi azért, mert a kártyás bejelentkezés nem helyettesíti, hanem átalakítja a Kerberost. A hitelesítés folyamata 95%-ban megegyezik a jelszavas és a kártyás Kerberosnál is.

Jelszótárolás

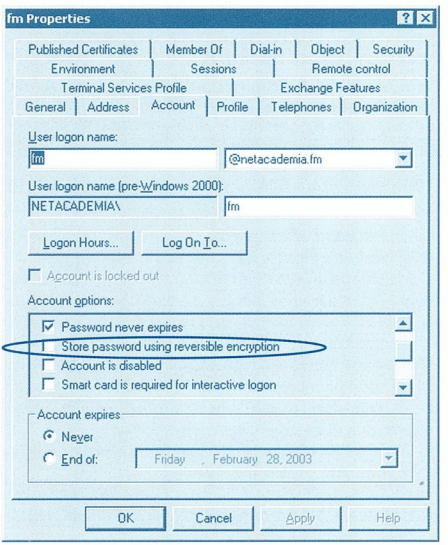
A hitelesítési mizéria megértéséhez külön kell választanunk a hálózati bejelentkezési protokollok működését a jelszavak tárolásának módszereitől. Először foglalkozzunk a jelszótárolással. Az általános hiedelemmel ellentétben a Windows **nem** tárolja a felhasználók jelszavait, csak az abból készített hash-eket. Ezt azonban mindjárt két változatban: egy gyenge *(LanMan)* és egy erős *(NTLM)* formátumban.

Mivel a jelszavak nem tárolódnak, felmerül a kérdés, hogy a csudába lehet azokat visszafejteni. A hash-algoritmus olyan, mint egy húsdaráló: felül bemegy a nyuszi, alul kijön egy rá jellemző fasírt, de a folyamat fordítva **NEM** végezhető el! Fasírtból nem lehet nyulat csinálni.

A kérdés általánosítható: mit jelent egy hash-algoritmus feltörése? Addig feszegetem a húsdarálót, amíg el nem kezd visszafelé működni? Nem! A cél olyan kiinduló adat *(nyuszi)* találása, amit ha ledarálunk, ugyanahhoz a fasírthoz jutunk. Ezresével daráljuk a nyuszikat, hátha az egyik „jó" fasírtot ad.

Mivel a mai napig senki sem bizonyította be a hash-algoritmusok *(MD5)* működőképességét *(csak használjuk, és bízunk benne, hogy különböző adatokra különböző kimenetet ad)*, a nyers erővel *(brute force)* végzett tömeges nyúldarálás végén lehet, hogy nem az eredeti jelszóhoz jutunk, hanem egy olyan karaktersorozathoz, mely – véletlenül – pontosan ugyanazt a hasht adja, mint a keresett jelszó. Az esetek 99,9999999999999%-ában egyébként a valódi jelszó ad jó megoldást, tehát a hashalgoritmusok *(többsége)* jó munkát végez.

Lehetőség van a jelszavak eredeti, olvasható alakjának megőrzésére is, ez külön erőfeszítést igényel a rendszergazda részéről: minden egyes felhasználón be kell pipálni, hogy az illető jelszavát őrizze meg a Windows. Erre akkor lehet szükség, ha a DC-nek olyan harmadik kiszolgáló felé kell azonosítania minket, akivel a Basic *(clear text)* azonosítás a közös nevező. Az alábbi ábrán láthatjuk, hol lehet ezt kérni az Active Directorytól:



Az Active Directory lehetővé teszi a felhasználói jelszavak megőrzését, de ezt külön kérnünk kell

LanMan és NTLM

Az NTLM-hash nem más, mint MD4, szerencsére éppen ezt használja a Kerberos protokoll is, így harmadik jelszótárolási formátumra már nincs szükség.

Érdekesebb jószág a LanMan-hash. Az itt alkalmazott „varázskódolásnak" köszönhető, hogy az NT4 biztonsága oly sok kívánnivalót hagy maga után: a „varázskódolás" *(security by obscurity)* sohasem hozza meg a kívánt eredményt. Egy jó debugger és megfelelő mennyiségű idő birtokában akármelyik egyetemista tetszőleges, ismeretlen DLL-ről felkommentezett forráskódot ad. Nem meglepő, hogy a LanMan-eljárás is pórul járt. Íme a LanMan lépései a jelszóalapú bejelentkezés során:

```
1. A jelszó nagybetűsítése (UPPERCASE)
2. A jelszó kiegészítése szóközökkel, 14 bájt
hosszúságra
3. A 14 bájtos jelszóból (14 * 8 = 112 = 2 * 56
bit) 2 db DES kulcs készül
4. A két 56 bites DES kulccsal titkosítjuk a
0x4B47532140232425 "mágikus" számot
5. Az eredmény 2*8 bájt, összesen 16 bájtos "HASH"
```

A fenti lépéssorból kiolvasható, hogy ha a jelszó rövidebb, mint 8 karakter, a „hash" második fele mindig 0xAAD3B435B51404EE, hisz konstanssal *(csupa szóközzel)* titkosítjuk a szintén mindig állandó „mágikus" számot. Ennek köszönhető, hogy a L0PHTCrack *(és társai)* hamarabb megfejtik a nyolcadik karakter utáni jelszórészt, mint az elejét, hisz egy már meglehetősen bonyolultnak számító 9 karakteres jelszó erőssége megegyezik egy 7+2 karakteresével. A kétkarakteres hátsó fél feltöréséhez másodpercek *(sem)* kellenek.

De figyelemre méltó az UPPERCASE is, mert emiatt felesleges a jelszót kis- és nagybetűkkel tarkítani – úgyis elvész ez az információ.

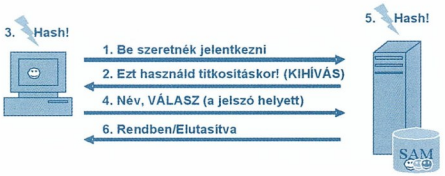
Ha a LanMan-jelszótárolás engedélyezve van, akár ne is erőltessük a 7 karakternél hosszabb és bonyolult jelszavakat.

Erre a két jelszótárolásra, mint „biztos" alapra épülnek a hálózaton keresztüli felhasználóazonosítás protokolljai, a Challenge/Response és a Kerberos. Ez utóbbinál csak az NTLM-hash dolgozik *(MD4)*, emiatt ez a hitelesítés egy jó nagyságrenddel biztonságosabb, mint a Challenge/Response – de ez is rogyadozik már.

Challenge/Response

Az NTLM-alapú Challenge/Response-protokoll volt a Windows-világban az első hitelesítési eljárás, melyet – legalábbis egy darabig – biztonságosnak lehetett tekinteni. Maga a Microsoft is tett olyan kijelentéseket, hogy ennek feltörése puszta ábránd, fikció, hisz a jelszavak kódolására használt MD4 hashalgoritmus állja a hackerek rohamát *(ez ma már nem igaz)*. A bejelentkezési adatok titkosításához használt, állandóan változó Challenge pedig eléggé összekuszálta a szálakat ahhoz, hogy emberi elme ne tudja kibogozni. Hogyan is működik?

A Challenge/Response erejét az adja, hogy minden egyes bejelentkezés előtt a hitelesítést végző számítógép átküld a kérelmezőnek egy nyolcbájtos „kihívást", amit annak megfelelően titkosítania kell. Tehát nem a jelszót, hanem a jelszóVAL titkosítunk egy véletlenszerű adatot. Az alábbi ábra ezt a folyamatot mutatja:



A Challenge/Response folyamata

Hogyan zajlik a kihívás titkosítása? A felhasználó által begépelt jelszó alapján a munkaállomás előállítja a LanMan és NTLM-hasheket. Mint azt fentebb láttuk, ezek mindegyike 16 bájtos. Ha hozzácsapunk még 5 bájtot *(csupa nullát)*, 21 bájthoz jutunk, ami testvérek között is három darab DES-kulcs. Nos, mindhárommal *(mindhattal, hisz két jelszóváltozatunk is van)* titkosítjuk a kihívást, így összesen 8+8+8+8+8+8 bájtnyi válaszhoz jutunk. Ezt küldjük vissza a DC-nek, aki saját oldalán kiássa a címtárból mind a LanMan, mind az NTLM-hasht, szintén elvégzi a titkosítást, és ha azonos eredményre jut, mint a kérelmező, megadja az engedélyt.

A Challenge/Response ellen remekül használható az úgynevezett known-plaintext kriptoanalízis, hisz a kihívás *(az eredeti adat!)* birtokában kell megtalálnunk azokat a DES-kulcsokat, amivel betitkosították. *(Tehát nem az adatot, hanem a kulcsokat keressük.)* A kulcsok összessége nem más mint az NTLM és a LanMan-hash. És itt a vége.

A Challenge/Response halálát a LanMan „titkosítás" okozza, mivel a két jelszóhash kölcsönösen gyengíti egymást. A LanMan-hash alapján könnyűszerrel visszafejthető a jelszavak nagybetűs változata. Az MD4-hasht már csak ennek különböző kis- és nagybetűs kombinációival kell „megetetni", és máris miénk az eredeti jelszó.

Ezt a gyengeséget azzal tetézi a Challenge/Response, hogy minden egyes új hálózati kapcsolatnál újra és újra lefut, tehát egy adott felhasználó jelszavai*(ból képzett adatok)* naponta kismilliószor jelennek meg a hálózaton, ráadásul itt nem részletezett okokból kifolyólag *(Pass Thru)* gyakorlatilag a hálózat minden zugában felbukkanhatnak. Ha öt percig futtatunk egy sniffert, valószínűleg a dolgozók felének jelszavához hozzájutunk. Ezzel szemben a Kerberos…

Kerberos

A Kerberos úgynevezett jegyalapú hitelesítést használ, amelynek az a lényege, hogy egy kezdeti *(reggel 8)* azonosítás után a hálózati erőforrások eléréséhez már nem jelszóhasht, hanem jegyeket használ. Ebben a cikkben nem térek ki részleteiben a Kerberosra, hisz ezt korábban *(2 éve)* már megtettük, s a cikk kikerült a NetAcademia tudásbázis webre is [1].

Ami azonban fontos, az a kezdeti azonosítás. Ehhez bizony továbbra is jelszóalapú adatra van szükség, hacsak ki nem váltjuk ezt Smart Carddal. De egyelőre ne váltsuk ki!

A Kerberos-azonosítás első lépése az Authenticator (hitelesítő) adatokkal

A kezdeti azonosítást az Authentication Service végzi. Az eljárás hasonlatos a Challenge/Response szisztémához, tehát nem a jelszót, hanem a jelszóVAL titkosítunk valamit, nevezetesen a munkaállomás rendszerórájának állását. Ezt a tartományvezérlő megpróbálja kibontani a nála tárolt jelszóhash segítségével. Sikeresség esetén még egy ellenőrzés történik, nevezetesen a kibontott csomagban található óraállást a tartományvezérlő

összeveti a saját óraállásával. Ha az eltérés nagyobb öt percnél, ijedten eldobja a csomagot és rendőrért kiált. Ez logikus védekezés az úgynevezett visszajátszási *(replay)* támadás ellen, amikor egy gonosztevő, aki nem tudja a Vezérigazgató jelszavát, beküldi az Authentication Service-nek a Vezér által korábban küldött bejelentkezési adatokat.

Ilyen huncutságra tehát összesen öt percünk van. Félek, hogy előbb-utóbb lesz megfelelő eszköz, amivel abszolválható e tempó, s ez ráadásul a Smart Card-bejelentkezésre is veszélyessé válhat. Ebből a szempontból a Challenge/Response erősebb, mert egy adott kihívásra *(emlékezzünk: 8 bájtos adat)* csak egyetlenegy *(helyes)* választ fogad el, így a visszajátszás tökéletesen értelmetlen; az a vonat elment.

KerbCrack

Igaz ugyan, hogy a Kerberos naponta csak egyszer fed fel bármilyen adatot a felhasználó jelszaváról, de akkor megteszi. Van már ügyes eszköz a Kerberos AS-forgalom figyelésére és elkapására, ez a KerbCrack néven futó *(letölthető)* tünetegyüttes. Nincs mit tenni, be kell vallani, hogy az MD4-hash ma már, a 2 GHz-es személyi számítógépek korában halottnak tekinthető.

Egy korábbi cikkemben említettem a jelszóparadoxont. Egy jelszó legyen:

- Bonyolult
- Hosszú
- Megjegyezhető

Sajnos a valóságban ebből a három feltételből rendszerint maximum kettő teljesül. Több kiút is lehetséges, ezek közül napjainkban a Smart Card-alapú bejelentkezés került elérhető közelségbe. A kártyák és olvasók ára zuhanórepülésbe kezdett, a többi infrastrukturális elem *(Certificate Server stb.)* pedig „ingyen" van annak, aki megvásárolta a Windows 2000 Servert.

Smart Card logon

És most térjünk rá a Smart Card logonra, az intelligens kártyával végzett bejelentkezésre. A felhasználó szemszögéből a változás annyi, hogy már nemcsak egy jelszójellegű izét kell tudnia *(PIN-kód)*, hanem valamivel rendelkeznie is kell – a kártyával *(vagy USB-tokennel)*. A PIN-kód nem megy át a hálózaton, hanem a Crypto API-n egyenesen áthaladva a kártyához tartozó Crypto Service Provider segítségével lejut az eszközbe. Ez az útvonal gyakorlatilag lehallgathatatlannak tekinthető, így a PIN-kódnak korántsem kell olyan hosszúnak és bonyolultnak lennie, mint egy hagyományos jelszónak.

Ráadásul több kártya is rendelkezik önmegsemmisítő szolgáltatással, vagyis x darab hibás bejelentkezés után használhatatlanná válik.

Mit tudnak az intelligens kártyák? Mindenféle titkosítási algoritmusokat futtatni. A hash-algoritmusok közül általában tudják az MD4, MD5, SHA változatokat, a szimmetrikus titkosítóalgoritmusok közül a DES, DESX, 3DES, RC4, RC5 típust, a nyílt kulcsúak közül pedig az RSA-t és a DSA-t.

Erre a bázisra alapozva lehetővé válik a Kerberos kezdeti hitelesítési lépésének átterelése a kártyára, mégpedig a következő módon: az azonosítás során az Authenticator mezőt nem a jelszóból képzett adattal, hanem a kártyán lévő – és azt soha el nem hagyó – privát kulccsal titkosítjuk. Ezt küldi el a munkaállomás a tartományvezérlőnek. A DC a beérkezett csomagból kiolvassa a felhasználó nevét és előkeresi a címtárban publikált tanúsítványok közül azt, amelyikben a megfelelő publikus kulcs rejtőzik. Ezzel kinyitja az Auth. mezőt, és innentől minden ugyanúgy zajlik, mint a korábbi, jelszavas bejelentkezéskor.

Feltörhető?

A security levelezési listán kérdezte valaki, hogy a Smart Card bejelentkezés is csak időleges megoldás-e, magyarán: feltörhető lesz-e valaha?

Erre egyértelmű választ adhatok. A Kerberos Auth. csomag éppúgy hajlik a „known plaintext" visszafejtés előtt, mint a Challenge/Response. De ne feledjük, mi a jutalma annak, aki végigpróbálgatja a feltöréshez szükséges hatalmas kulcsmezőt: szert tesz arra a kulcsra, amellyel a csomag bontható.

Szimmetrikus titkosítás esetén, amikor a titkosító kulcs megegyezik a kibontókulccsal a megfejtő jutalma egy jelszó.

Nyílt kulcsú titkosítás bevetésével a helyzet gyökeresen megváltozik. A véres verítékkel, próbálgatásos módszerrel megfejtett kibontókulcs nem más, mint az RSA-kulcspár publikus tagja, amihez minden erőfeszítés nélkül is hozzájuthatott volna a nyomorult hacker – hisz azért publikus!

Ami veszélyesnek tűnik az a visszajátszásos támadás *(replay attack)*. Emlékezzünk vissza: öt percünk van arra, hogy beküldjünk egy helyes Auth. csomagot!

Biometrikus megoldások

Végül ejtsünk néhány szót a biometrikus azonosítási módszerekről is. Akármelyikkel találkoztam *(pl. ujjlenyomatolvasó)*, azt tapasztaltam, hogy ezek csak jelszópótlóként használhatók. Az ujjlenyomatolvasó azonosítja becses személyemet, ez alapján kikeresi a saját dugi raktárából a jelszavamat, és elküldi a tartományvezérlőnek. Hangsúlyozom: elküldi! A jelszavam! A hálózaton át!

Ezek tehát csak akkor érnek valamit, ha megszigorítjuk a jelszavas bejelentkezést: letiltjuk a LanMan-jelszótárolást stb. Amíg a biometrikus eljárások támogatása be nem épül a tartományvezérlőkbe, addig ez a módszer játék marad.

(Az IDENTIX cég rendelkezik kiszolgálóoldali ujjlenyomatfelismerővel, de annak ellenére nem adta ki tesztelésre, hogy regisztrált ujjlenyomat-olvasó felhasználó vagyok.)

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

A cikkben szereplő URL-ek:

[1] NetAcademia tudásbázis, a Kerberos-protokoll
http://www.netacademia.net/secu/kerberos

Kapcsolódó tanfolyamaink:

SECU – Hálózatbiztonság (12 óra)
PKI Workshop (2 nap)
2150 – Windowsos hálózatok biztonsága (40 óra)

Smart Card logon

Lépésről lépésre

Najaink legmegbízhatóbb felhasználóazonosítási módszere az intelligens kártyás bejelentkezés. Ha a megfelelő hardvereszközök rendelkezésre állnak, neki is láthatunk a rendszer bevezetésének. Szükségünk lesz Certificate Serverre, egy Smard Card-író „műhelyre", csoportos házirendek állítgatására és még sok minden másra. Ha bármelyik lépést kihagyjuk, egzotikus hibaüzenetekben gyönyörködhetünk. Ez a cikk a Smart Card logon elkészítésének „szakácskönyve"!

Egy jó szakácskönyv a hozzávalók felsorolásával kezdődik, hogy a háziasszony mindent a keze ügyébe készíthessen. Én is követem ezt a szerkesztési módszert.

A Smart Card logon feltételei

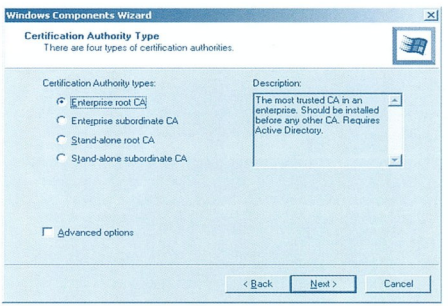
1. Olyan operációs rendszerek kellenek mind ügyfél-, mind pedig kiszolgálóoldalon, amelyek támogatják ezt a bejelentkezési formát. Javaslom az NT4, mint őskövület elfelejtését, bár hallomásból tudom, hogy több kártyagyártó rendelkezik NT4-en is működő megoldással. Ebben a szakácskönyvben csak a Windows 2000 *(és utódai)* által gyárilag támogatott, Kerberos-alapú kártyás bejelentkezéssel foglalkozom.
2. Ha már ragaszkodom a Kerberos-alapú megoldáshoz, ezzel azt is meghatároztuk, hogy Active Directoryt kell használnunk. Ez a címtár – sok egyéb adat mellett – a felhasználók tanúsítványainak tárolására is fel van készítve.
3. Szükségünk lesz a Windowshoz tartozó Enterprise Certificate Authorityra is, amelyik két fontos dologra képes. Egyfelől csak az Enterprise-változat képes és a megfelelő tanúsítványsablonok *(template)* kiállítására, másfelől a kész tanúsítványokat *(benne ugyebár a publikus kulccsal)* automatikusan elhelyezi a címtár megfelelő rekeszében *(a megfelelő felhasználónál)*.
4. A kártyák készítését nem a végfelhasználó, hanem egy úgynevezett Smart Card kiállító személy végzi. Ennek az első pillantásra szokatlan, kényelmetlen megoldásnak biztonsági okai vannak: ezzel elkerülhető, hogy boldog-boldogtalan kártyakiállítói jogosultsággal rendelkezzen. A Jogosult Személy a Kártyakiállító Központban végzi feladatát, vagyis egy erre a feladatra elkülönített PC előtt ül. A PC-re természetesen tegyük fel a kártya kezeléséhez szükséges eszközöket, programokat, eszközmeghajtókat. A kártyagyártó személy tetszőleges felhasználó nevében készíthet új kártyát. Ha ezzel a lehetőséggel visszaél, az egész rendszer nem ér semmit.
5. Előkészített kártyák, tokenek. A kártyák formázását, a PIN- és PUK-kódok beállítását jobb előre elvégezni, így gördülékenyebb a kártyagyártás. Sok kártyánál *(és a PKI Workshopon használt USB-tokennél is)* alapértelmezett kulcstárrá kell tenni azt a rekeszt, amelyikben a logon-kulcspár csücsül.

Ha ez mind megvan, sorozatban lehet generálni a kártyákat. Most lássuk részletesebben a harmadik pontot!

Certificate Server telepítése

A Windows 2000 telepítőlemeze tartalmazza a Tanúsítványkiállító Központot, a Certificate Authority szolgáltatást. Windows-komponensről lévén szó, telepítése megegyezik – mondjuk – a DHCP Server telepítésével, azaz Control Panel Add/Remove Programs Windows Components. Amire vigyázni kell: a Certificate Server bebetonozza a kiszolgáló aktuális állapotát. Ha feltettük, többé nem változtatható sem a gép neve, sem tartományi tagsága!

Mivel Enterprise CA-t szeretnénk telepíteni *(és ez Active Directoryt igényel)*, a kiszemelt CA-kiszolgálónak a telepítés pillanatában vagy tartományi tagnak, vagy egyenesen DC-nek kell lennie. Ha ez nem teljesül, sajnos csak önálló *(Standalone)* CA-t tudunk telepíteni – az Enterprise… gombok szürkék lesznek. Jelen leírásban egyszerűségre törekszem, így minden további hókuszpókusz nélkül válasszuk az Enterprise Root CA telepítési változatot.



Enterprise Root CA telepítése

Tanúsítványsablonok *(certificate template)*

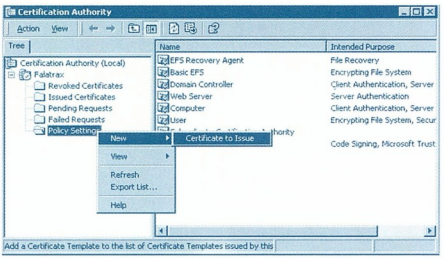
Miért is ragaszkodunk az Enterprise opcióhoz? Mert csak ez a változat képes a kártyás bejelentkezéshez szükséges tanúsítványtípusok előállítására. A sablon meghatározza a kulcsok felhasználási területeit. Három különleges tanúsítványsablonra lesz szükségünk:

1. A kártyák készítését egy úgynevezett Enrollment Agent végzi *(hús-vér ember)*. Említettem, hogy ezt a feladatot csak a megfelelő jogosultság birtokában lehet elvégezni. Nos, a jogosultságot stílszerűen egy megfelelő tanúsítvány birtoklásával bizonyítja az ügynök. Smart Card íráshoz tehát a következő tanúsítvánnyal kell rendelkezni: Enrollment

Agent *(EA)*. Ehhez a tanúsítványhoz nyilván tartozik egy kulcspár is, amelyet akár Smart Cardon is tárolhatunk, de ez nem kötelező. Alapértelmezésben kizárólag a Domain Admins csoport tagjai kérhetnek EA-tanúsítványt, de ez megváltoztatható az Active Directory Sites and Services eszközzel *(ennek bemutatása túlmutat a cikk keretein)*.

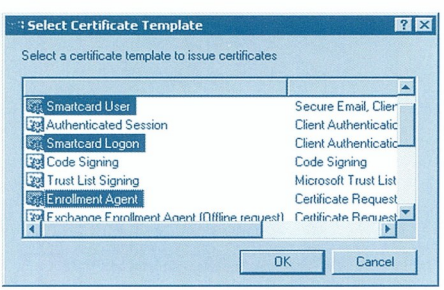
2. A kártyák felhasználói pedig olyan tanúsítványokat kapnak, amelyek a bejelentkezéshez szükségesek. Vagy Smart Card Logon *(csak bejelentkezésre)* vagy Smart Card User *(bejelentkezés+email)* típusú tanúsítvánnyal kell rendelkezniük.

E három tanúsítványtípus sajnos nem kerül be gyárilag a Certificate Server által kiosztott sablonok közé, külön engedélyeznünk kell ezek használatát a Certificate Authority MMC-konzol segítségével. Az alábbi ábrán jól látható, hogy a Policy Settings ágon kell jólirányzott klikkeléseket elhelyezni:



Certificate Authority: további sablonok engedélyezése

A megjelenő ablakban egyszerre akár mindhárom tanúsítványtípust is kijelölhetjük, csak tartsuk szorosan fogva a CTRL billentyűt!



Az Enrollment Agent és a két Smart Card-sablon egyidejű engedélyezése

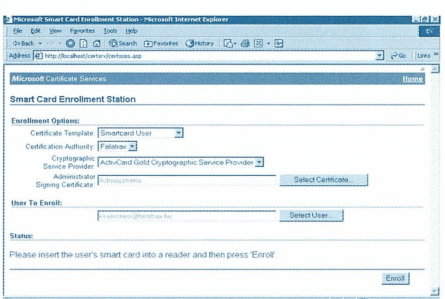
Kártyakiállító központ

A megfelelő sablonok engedélyezése után hamarosan indulhat a munka, előtte azonban a kártyákat kiállító személynek szert kell tennie egy Enrollment Agent tanúsítványra, mert ezzel fogja bizonyítani, hogy jogosult más nevében kártyákat kiállítani. Ehhez elindít egy böngészőt, és megnyitja a Certificate Servert futtató gépen a http://gepneve/CertSrv lapot. Jól jegyezzük meg ezt az útvonalat, ez a tanúsítványkérési webhely címe, mindig ide kell visszatérnünk, ha a kiszolgálótól tanúsítványt kérünk. Az alábbi ábrán összesűrítettem az Enrollment Agent tanúsítvány kérésének lépéseit *(egyébként Advanced Request, Using a Form...)*:

Enrollment Agent tanúsítványkérés (Az utolsó ábrán rossz helyen áll a kurzor. Fáradt voltam...)

Kártyák készítése

A kártyák készítése kísértetiesen hasonló módon történik, csak a web egy másik ágán *(Advanced Request, For Smart Card on behalf of another user...)*, azzal a nem elhanyagolható különbséggel, hogy most a tanúsítvány típusa Smart Card *, a kulcspár helye nyilván a kártya, a célszemély pedig egy hagyományos felhasználó *(Kőműves Kelemen)*. Az utolsó lépés képernyőjét mutatom, mert szépen összefoglalja mindazt, amit ebben a bekezdésben írtam:



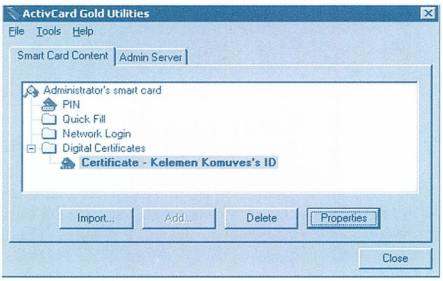
Smartcard User tanúsítvány kérése Kőműves Kelemen részére, ActivCard Gold kártyára

Az Enroll gomb megnyomása után a kiválaszott Crypto Service Provider *(esetünkben ActivCard Gold)* érzékeli a kulcsgenerálási kérést, és feldob egy PIN-kód ablakot. Ha sikeresen eltaláljuk a PIN-kódot, a kulcsgenerálás megtörténik lent az eszközben, majd a publikus kulcs elvándorol a Certificate Serverbe, ahol rákerül egy digitális aláírás a CA privát kulcsával, vagyis előáll egy tanúsítvány. Ezt azután a CA automatikusan

beledobja az Active Directoryba is – nomeg vissza is küld egy példányt nekünk. Ezt az op'rendszer letölti a kártyára, így azon a műveletsor végére három objektum található:

- egy privát kulcs
- egy hozzá tartozó publikus kulcs
- egy a publikus kulcs hitelességét igazoló X.509 tanúsítvány.

Mindez meg is tekinthető a kártyához adott kezelőszoftver segítségével:



Kőműves Kelemen bejelentkezési tanúsítványa ActivCard Gold kártyán

Itt jegyzem meg, hogy – jól láthatóan – a tanúsítvány NEM a „Network Login" bugyorba került – mert nem az ActivCard saját rendszerét használjuk, hanem a Windows 2000-ét. Ebből különböző galibák szoktak származni abban az esetben, ha több tanúsítvány is van a kártyán. Aki játszott S/MIME-mal vagy SSL-lel, tudja jól, hogy mindig pontosan be kell állítani, melyik tanúsítványt használja a rendszer. A Smart Card logonnál nincs ilyen beállítási lehetőség. Ez mindig az **alapértelmezett** tanúsítvánnyal és kulcspárral próbálkozik – még ha az nem is megfelelő típusú! Tegyük alapértelmezetté a logon-tanúsítványt!

A kártya használata

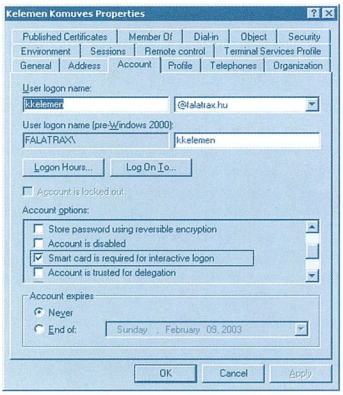
Talán feltűnt, hogy ezidáig csak a Certificate Serverrel játszadoztunk, a tartományvezérlőnek nem kellett elmagyaráznunk a Smart Card logon mibenlétét. Ez azért van így, mert ez a fajta bejelentkezési mód az Active Directory alapértelmezett hitelesítési eljárásai közé tartozik! Immár három éve!

De lássuk tovább a folyamatot: a felhasználó kézhez kapja a kész kártyát, elballag vele a saját gépéhez, ám a bejelentkezőképernyőnél nem CTRL+ALT+DEL akkordot nyom, hanem beilleszti a kártyát az olvasóba *(vagy bedugja a tokent az USB-hubba)*, majd a megjelenő ablakba begépeli a kártyához tartozó PIN-kódot.

- a PIN-kód nyitja a kártyát
- a kártya titkosítja a Kerberos Auth. mezőt
- a gép elküldi a tartományvezérlőnek a KRB_AS_REQ csomagot
- a DC kiolvassa a felhasználó nevét a csomagból, és megkeresi a számára publikált Smart Card * tanúsítványokat
- a megfelelő tanúsítványból kiolvassa a megfelelő publikus kulcsot
- kinyitja vele az Auth. mezőt
- a csomag küldője tehát rendelkezik a privát kulccsal. Login.

További beállítási lehetőségek

Az egyik legfontosabb beállítási lehetőség a hagyományos *(jelszavas)* bejelentkezés tiltása. Mivel ennek csak azoknál a felhasználóknál van értelme, akik már rendelkeznek megfelelő kártyával, ez a beállítás felhasználói fiókonként egyedi. Keressük meg a célszemélyt az Active Directoryban, és a tulajdonságlapján pipáljuk be a „Smart Card is required..." pipát:

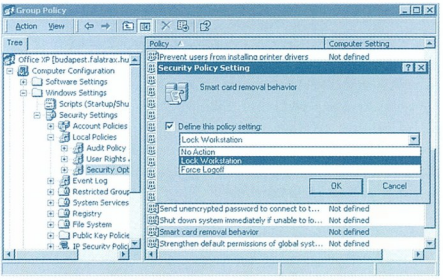


A kártyahasználat kötelezővé tehető!

Egy másik fontos beállítási lehetőség a kártya eltávolítására adott válaszlépés. Alapértelmezésben a válasz: hallgatás. Ha kihúzzuk a kártyát, nem történik semmi. De lehetőség van arra, hogy a kártya kihúzásakor

- zárolódjon a munkaállomás *(Lock Workstation)* vagy
- a felhasználót a rendszer jelentkeztesse ki *(Force Logoff)*.

Ez a két lehetőség különösen ajtónyitóval integrált kártyák esetén váltja be a hozzá fűzött reményeket, mert a felhasználó kénytelen magával vinni a kártyáját, ha elhagyja a szobáját. Ez a beállítás csoportos házirenddel *(Group Policy)* módosítható:



Mi történjen a kártya kihúzásának hatására?

Érdekes, hogy ez a beállítás gépszintű *(lásd az ábrán)*, azaz egy adott PC-re, és nem egy adott személyre vonatkozik. Olyannyira nem, hogy ha nem is kártyával jelentkeztünk be, akkor is él. Ha egy akármilyen kártyát bedugunk, majd kihúzunk, a logoff beindul. Vajon miért?

Fóti Marcell
marcellf@netacademia.net

Identitáskezelés

Hivatalos Microsoft-tanulmány

Az összes munkatársra kiterjedő identitások kezelése egy vállalaton belül mindig nagy kihívást jelentett. Napjainkban, az Internet és az elektronikus kereskedelem elterjedésével ez a kihívás megnőtt, mert a vállalatok és kormányzatok partnereik és ügyfeleik számára is kénytelenek hozzáférést biztosítani belső rendszereikhez és alkalmazásaikhoz.

Ha az egyre magasabb fokon integrált környezetekben hatékonyan kívánjuk kezelni a hozzáférési engedélyeket, vállalati identitáskezelő rendszer bevezetése válik szükségessé. Az Active Directory, a Microsoft Metadirectory Services, valamint a Microsoft többi identitáskezelő megoldásával és termékével a rendszergazdák biztonságosan, költséghatékonyan és minimális bonyolultság mellett felügyelhetik a felhasználók hozzáférését a kritikus vállalati alkalmazásokhoz és adatokhoz.

Gyakorlatilag minden céges adat elektronikus változata megtalálható a vállalati hálózaton. Éppen emiatt egyre fontosabb és nehezebb annak garantálása, hogy csak a jogosultsággal rendelkező felhasználók férhessenek hozzá ezekhez az információkhoz. A vállalatok számára ugyanakkor elengedhetetlen, hogy a munkatársak, ügyfelek és üzleti partnerek kényelmesen érhessék el a számukra szükséges adatokat. A két alapkövetelmény közti egyensúlyozás az identitáskezelés legnagyobb kihívása. A témakört körbejárva a rendszergazdák az alábbi kérdésekkel szembesülnek:

- **Biztonság:** Az alkalmazottak, szerződéses és üzleti partnerek miatt módosultak az adatok és az alkalmazások elérésével kapcsolatos igények. A vállalatok számára létfontosságú, hogy csak az arra felhatalmazott felhasználók férhessenek hozzá az érzékeny céges adatokhoz.
- **A felügyelet bonyolultsága:** A korszerű vállalatok számos, különféle platformon futó célrendszert használnak. A felhasználók és a rendszerek számának növekedésével az egységes felhasználói hozzáférési házirendek kidolgozása is egyre bonyolultabbá válik.
- **Költségcsökkentés:** Még az egyszerűbb hozzáférési házirendek fenntartása is költséges lehet, ha több alkalmazás, rendszer és platform üzemeltetéséről kell gondoskodni, és ezek különálló, saját hozzáférési listát használnak. Például 10000 felhasználó jogainak módosítása 20 rendszeren legalább 200000 bejegyzés átírását teszi szükségessé.

Ezeket az identitáskezeléssel kapcsolatos kulcsfontosságú kérdéseket lekezelve a vállalkozások nagyobb termelékenységet, költségcsökkenést, illetve magasabb fokú üzleti integrációt érhetnek el.

Az identitáskezeléssel kapcsolatos biztonsági kihívások

A biztonságos adatelérés megoldása a hitelesített felhasználók számára egyre problémásabb. A legtöbb vállalatnál az egyedi alkalmazások és rendszerek saját felhasználói adatbázissal vagy címtárral rendelkeznek, ezek alapján ellenőrzik, hogy ki használhatja az adott erőforrást. Ahogy a hozzáférési jogok odaítélése egyre inkább decentralizált jellegűvé válik, a biztonság sérülésére is lényegesen nagyobb esély mutatkozik. Például:

- A távozó alkalmazottak, ügyfelek és üzleti partnerek gyakran megőrzik hozzáférésüket a rendszerekhez, amit egészen a következő frissítésig, illetve az érvénytelenné vált hozzáférések megszüntetéséig használni tudnak.
- A nem egységes házirendek eredetileg meg nem adott hozzáférésre adhatnak módot *(pl. humánerőforrás-adatbázis).*
- A gyenge hitelesítési adatok, a szegényes vagy nem létező jelszóházirendek, illetve a felhasználók által megjegyzendő nagy számú azonosító és jelszó miatt sebezhetőbbek a rendszerek.

A felügyelet bonyolultsága

Ahogy a vállalatok és a kormányzatok egyre több célrendszert használnak – például hálózati erőforrás-címtárakat, levelezőkiszolgálókat, emberierőforrás-adatbázisokat, hangpostakiszolgálókat, és bérfeldolgozási alkalmazásokat –, egyre összetettebb feladattá válik a felhasználói jogok kezelése. A vállalat egyes részlegei sok esetben különbözőképpen kérvényezik és osztják szét az erőforrásokat. Emellett a legtöbb vállalatnál minden rendszer saját eszközöket használ a felhasználói fiókok kezelésére. Sokszor külön jelszóra és bejelentkezésre van szükség a felhasználók azonosításához. A fenti problémák mindegyike hozzájárul az informatikai felügyelet bonyolultságának növekedéséhez. Például:

- Az elosztott és eltérő hitelesítési rendszereket más módszerekkel kell felügyelni, kezelni és naplózni.
- A címtárak és más identitástárolók sokszorozódása miatt a módosításokat többféle módon, többféle tárolóban kell végrehajtani.
- A felhasználókat kellemetlenül érinti, ha több azonosítót és jelszót kell észbentartaniuk a különféle alkalmazásokhoz és rendszerekhez.
- Ahogy a vállalatok bővítik rendszereiket – hiszen már nemcsak a munkatársakat, hanem az ügyfeleket és az üzleti partnereket is ki kell szolgálni az Interneten keresztül –, a problémák csak sokszorozódnak.

Költségcsökkentés?

Sok szervezetnél az egyes rendszerek különleges rekordok és adatbázis-bejegyzések szigeteiként viselkednek, és felügyeletükről egyedileg kell gondoskodni. Ezek a rendszerek általában saját felhasználói identitás-definícióval rendelkeznek *(név, beosztás, azonosítók, szerepek és csoporttagságok).* Minél na-

gyobb egy szervezet, annál többféle tárolót találni benne, és annál nagyobb költséget és erőfeszítést igényel ezek naprakészen tartása.

- A termékmenedzserek, informatikai szakemberek és emberierőforrás-vezetők rengeteg időt töltenek űrlapok kitöltésével, a felhasználók adatainak bevitelével és frissítésével, a fiókok beállításával és az elfelejtett jelszavak módosításával.
- Az új alkalmazottak és szerződéses partnerek gyakran napokat várnak arra, hogy hozzáférhessenek a kritikus alkalmazásokhoz és információkhoz – eddig tart, míg minden rendszergazda beállítja hozzáférésüket.

Megoldások az identitások kezelésére

Ha a növekvő számú alkalmazáshoz, rendszerhez, információforráshoz és céghez rendelt hozzáféréseket költséghatékony és biztonságos módon akarják kezelni, a vállalatoknak identitáskezelő infrastruktúrát kell kiépíteniük. Az infrastruktúra kiépítése átfogó felhasználó- és erőforrás-kezelő rendszer kifejlesztését teszi szükségessé, illetve a biztonság felügyeletéről is gondoskodni kell. Az identitáskezelő megoldások alapja több összetevőből áll:

- Identitástároló címtár, amely biztonságos, hiteles adatforrásként szolgál.
- Az identitás hitelesítése, amellyel ellenőrizhető, hogy a felhasználó valóban az-e, akiknek kiadja magát.
- A hitelesítés révén biztosítható, hogy a felhasználók hozzáférése az erőforrásokhoz, alkalmazásokhoz, fájlokhoz és funkciókhoz szabályos és engedélyezett legyen.
- Platformok közötti együttműködési szolgáltatások, amelyek megfelelő időn belül lehetővé teszik a más rendszereken és platformokon végrehajtott identitásváltozások felismerését.
- A jogok biztonsági házirendek szerinti biztosításának és visszavonásának kezelése, illetve a kapcsolódó rendszerek értesítése felhasználók hozzáadásáról és törléséről.

Identitástároló

A felhasználók és az általuk szabályosan igénybe vehető erőforrások halmazának tárolására általában egy címtárat használunk. A címtárban tárolt információk magukba foglalják az illető profiljába tartozó adatokat – pl. név, cím, munkakör –, hozzáférési jogait az egyes erőforrásokhoz, az ezek használatával kapcsolatos házirendeket, illetve a biztonsági adatokat, például a jelszavakat. Az identitástároló nemcsak felhasználói adatokat raktároz, hanem az erőforrások, számítógépek és alkalmazások identitásadatait is, ugyanis a biztonsági házirendek ezekre az erőforrásokra is kiterjednek. Az identitásoknak a hitelesítéssel és a jogosultságok kiosztásával való integrálása lehetővé teszi a beléptetést és a címtárral kezelt erőforrások használati jogának egyidejű biztosítását. Az Active Directory a Microsoft biztonságos, magas rendelkezésre állást nyújtó, elosztott központi identitástárolója, amely szorosan együttműködik a Windows 2000 operációs rendszerrel. Egyszerre szolgál a felügyelet és a felhasználói fiókok, a hitelesítés, a biztonsági házirendek és a vállalati erőforrások – számítógépek, nyomtatók, kiszolgálók – nyilvántartásának központi pontjaként.

Az Active Directory egyedülálló értéke azon képessége, hogy a felhasználói identitásokat és a különféle erőforrásokhoz való hozzáféréseket különféle rendszereken és platformokon tudja kezelni. Így egy szervezet az identitásokkal, az azonosítással és a hitelesítéssel kapcsolatos, más alkalmazásokra, rendszerekre és platformokra is kiterjeszthető adatok megbízható tárolójaként használhatja az Active Directoryt.

Az identitások hitelesítése

Mély kapcsolat fedezhető fel az Active Directory és az általa támogatott Windows 2000 biztonsági modell között. Az Active Directory tárolja a tartománnyal, a hálózattal és a felhasználókkal kapcsolatos összes biztonsági házirend adatait. Ez a kapcsolat csak az Active Directorynak a Windows 2000 operációs rendszer biztonsági infrastruktúrájába való tökéletes integrálásával valósítható meg.

A rendszer erőforrásaihoz való hozzáférések mindegyike hitelesített. A Windows 2000 hitelesítőszolgáltatása – amely az operációs rendszer része – egybeépül az Active Directoryval. Ez azt jelenti, hogy a felhasználók egyetlen jelszó segítségével bármely munkaállomásra bejelentkezhetnek, és ugyanez a jelszó biztosítja hozzáférésüket a megfelelő erőforrásokhoz is. Az integráció biztosítja a felhasználók eltávolítását is az Active Directoryból, ha hozzáférésük megszűnik.

> Az Active Directory egybeépülése az operációs rendszerrel lehetővé teszi, hogy megbízható hitelesítési adattárolóként szolgáljon.

Az Active Directory rendkívül rugalmas, és a legújabb internetes szabványokra épül. Rugalmassága különösen abból a szempontból érdekes, hogy sokféle hitelesítési módszert támogat – így minden vállalat a számára szükséges védelmi szintet valósíthatja meg: egyszerű azonosító és jelszóalapú, erősebb, tanúsítványalapú, vagy rendkívül biztonságos, intelligens kártyák használatára vagy biometriára alapuló hitelesítést használhat. Az Active Directory a Csoportházirend szolgáltatással együtt felhasználói- vagy csoportalapon lehetővé teszi a különféle hitelesítési módszerek keverését is. Egy vállalatnál előfordulhat például, hogy bizonyos felhasználók intelligens kártya, mások pedig csupán felhasználónév és jelszó segítségével azonosítják magukat.

Az Active Directory egybeépülése az operációs rendszerrel lehetővé teszi, hogy megbízható hitelesítési adattárolóként szolgáljon. Az egymással együttműködő, szabványokon alapuló hitelesítési eljárások révén a vállalatok más platformokra és rendszerekre is kiterjeszthetik az Active Directory hatáskörét.

Hitelesítés

A Windows 2000 biztonsági modellje a felhasználó identitása alapján minden objektumra egységesen megvalósítja a hozzáférés-vezérlést és a hitelesítést. Minden objektumhoz tartozik egy tulajdonos, aki jogokat tud adni az objektumnak. Ha egy felhasználó hitelesítése megfelelően lezajlott, a hozzáférés-vezérlési mechanizmus meghatározza, hogy a felhasználó mely objektumokat és hogyan érhet el.

A hozzáférés-vezérlési listák *(Access Control List, ACL)* adják meg azokat az előírásokat, amelyek alapján az operációs rend-

szer gondoskodik egy-egy objektum védelméről. Az ACL-alapú megközelítés előnye nyilvánvaló: minden objektumot – ami lehet felhasználó, nyomtató, fájl, kiszolgáló vagy egyéb hálózati erőforrás – külön ACL védhet, az ACL-lel összerendelt identitások tárolását és kezelését pedig az Active Directory végzi. Ha egy identitást az Active Directory kezel, lehet tisztán Windows-alapú, de használható más platformokon is, mindenképpen fog rá vonatkozni egy ACL. Ha egy felhasználó nem tudja megfelelően azonosítani önmagát, nem fog hozzáférni az erőforrásokhoz. Így az identitás a hitelesítésnél és a jogok megadásakor egyformán fontos. Emiatt az Active Directory a vállalati hitelesítés központi elemének tekinthető.

Különböző platformok közti együttműködés

Az együttműködés alapfeltétele egy olyan metacímtár, amely biztosítja a címtár és az egyes szervezetek nagy számú címtára, adatbázisa és egyéb tárolója által kezelt információk közötti kapcsolatot. Segítségével akkor is egységesen kezelhetők egy szervezet felhasználói, ha az adatok több címtárban és felhasználói adatbázisban találhatók.

A metacímtár – mivel szoros kapcsolatban áll az identitástárolóval – gondoskodik az identitásokkal kapcsolatos változások terjesztéséről is. A változások terjesztése automatizálja és egyszerűsíti a felhasználói fiókok és jogok hozzáadását, módosítását és visszavonását. A folyamatok automatizálásával csökkenthetők a költségek, és jelentősen növelhető a termelékenység. Garantálható a házirendek és eljárások egységes betartása is. A visszavonások terjesztése szintén kulcsfontosságú. Amikor egy alkalmazott kilép, a változásokat terjesztő rendszer gyorsan és egységesen törölni tudja a felhasználói fiókokat, illetve vissza tudja vonni a hozzáférési jogokat. A Microsoft olyan együttműködési megoldásokat kínál, amelyek kielégítik a vállalatok különféle igényeit. Ezek a termékek a következők:

- Microsoft Metadirectory Services *(MMS)* – lehetővé teszi több rendszeren és platformon tárolt identitásokkal kapcsolatos információk integrálását. Az MMS számos csatolóval rendelkezik, amelyek többek közt Sun ONE Directory Server, Novell NDS és eDirectory, Lotus Notes és cc:Mail rendszerrel integrálják az identitásokkal kapcsolatos adatokat. Az MMS támogatja az egyre népszerűbb internetes adatformátumokat, például az XML és a DSML formátumot. Az MMS széles körű együttműködési lehetőségek révén növeli az Active Directory értékét: integrációs lehetőség különféle identitástárolókkal, az identitásokkal kapcsolatos adatok terjesztése és szinkronizálása többféle tárolón keresztül, az identitásokkal kapcsolatos adatok módosulásának automatikus érzékelése és megosztása a rendszerek között.
- Services for UNIX – UNIX-os felhasználókat, csoportokat és állomásokat feleltet meg Windows-alapúaknak. A UNIX felhasználók és csoportok tehát a Windows felhasználókkal és csoportokkal azonos módon kezelhetők. Ezáltal közös névtér jön létre, és csökken a két különböző névtér és címtár felügyeletéből fakadó többletterhelés.
- Services for Netware – rendkívül rugalmas szolgáltatás, amely teljeskörű együttműködési megoldást kínál az Active Directory és a Novell NDS és bindery címtárai között. Az Active Directory és a Novell rendszerek közötti kétirányú szinkronizációval csökkenti az identitáskezelés költségét. A jelszavak szinkronizálását is lehetővé teszi az Active Directory címtárból a Netware felé.
- Services for Macintosh – a Microsoft Windows 2000 Server integráns része, lehetővé teszi a Windows és Macintosh OS rendszert futtató számítógépek közti fájl- és nyomtatómegosztást. Emellett a Windows 2000 Server AppleTalk-útválasztóként is használható. A Services for Macintosh tartalmazza a Microsoft User Authentication Module-t *(MSUAM)*, amely biztonságos és egyszeri bejelentkezési lehetőséget kínál a Macintosh ügyfeleknek a Windows 2000 Servert futtató kiszolgálókra.
- Host Integration Server 2000 *(HIS)* – az alkalmazások, adatok és hálózatok integrációjával a hagyományos állomásokra is kiterjeszti a Windows hatókörét. A HIS a Windows alapú és a hagyományos rendszeren is automatikusan hitelesíti a felhasználókat, így egyszerűsödnek a folyamatok. A HIS képes arra is, hogy az állomás biztonsági rendszerében végrehajtott módosítások nélkül is átültesse a Windows alapú felhasználói azonosítót és jelszót a hagyományos rendszerbe.

> Az együttműködés alapfeltétele egy olyan metacímtár, amely biztosítja a címtár és az egyes szervezetek nagy számú címtára, adatbázisa és egyéb tárolója által kezelt információk közötti kapcsolatot.

Felügyelet

Az MMS felügyeleti és adatterjesztési lehetőségei mellett a felhasználók és számítógépek csoportjaira vonatkozóan a Windows 2000 Csoportházirendjével is definiálhatók és betartathatók a felhasználókra és a számítógépekre vonatkozó beállítások.

A Csoportházirenddel a vállalatok a következőkre írhatnak elő házirendet:

- Rendszerleíró alapú, a Windows 2000 operációs rendszerre, összetevőire és az alkalmazásokra vonatkozó házirendek.
- Biztonsági beállítások, amelyek a helyi számítógépre, a tartományra vagy a hálózatra vonatkoznak.
- Szoftvertelepítési és karbantartási beállítások, amelyekkel központilag kezelhető a szoftverek telepítése, frissítése és eltávolítása.

Megfelelő identitáskezelő megoldást kidolgozva a vállalatok egyetlen helyről kezelhetik az összes felhasználói jogot. Így megoldhatók a legfontosabb – már említett – identitáskezeléssel kapcsolatos problémák. ■

A Microsoft operációs rendszerek biztonsági tényezői

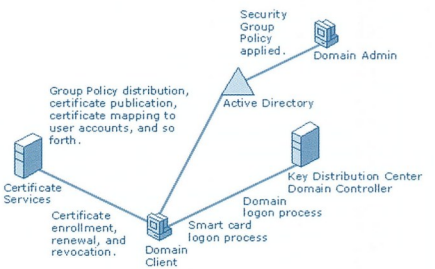
II. rész – Biztonságos hálózati működés

Az előző cikkben eljutottunk a biztonságos vállalati hálózathasználat alapjaiig. Egy tudatosan biztonságosra tervezett operációs rendszert ismertünk meg az NT „személyében". Sőt! Van címtárunk és néhány jól tervezett hitelesítési protokollunk, öröklődő jogosultságaink és stabil fájlrendszerünk is. Mi kellhet még? PKI, a folyamatok követése *(naplózás)* és adatfolyamtitkosítás.

Nyilvános kulcsos architektúra

Már eddig is számos alkalommal utaltunk a Windows 2000 nyilvános kulcsú infrastruktúrájára, amelyet rengeteg alkalmazás használ. Most egy kicsit tekintsük át, mit is jelent a PKI a Windows környezetben.
A PKI nem egy szolgáltatás, inkább egy keretrendszer, amelynek elemeit és elemi szolgáltatásait különböző más szolgáltatások igénybe veszik. A Windows 2000-ben az alábbi beépített szolgáltatások építenek PKI alapokra:

- A levelező szoftverek, amelyek az S/MIME szabványt alkalmazva aláírt, és/vagy titkosított leveleket készítenek és továbbítanak. A módszer segítségével meggyőződhetünk arról, hogy valóban a feladótól kaptunk levelet, azt nem módosították a kézbesítés során, és illetéktelenek nem férték hozzá a küldemény tartalmához. *(Letagadhatatlanság, hitelesség, bizalmasság).*
- Biztonságos webhelyek, amelyek képesek a felhasználókhoz a tanúsítványuk alapján különféle engedélyeket rendelni.
- A biztonságos webes kommunikáció, amely SSL és TLS protokollok használatával titkosított kapcsolatot épít fel az ügyfél és a kiszolgáló között, azonosítja a belépőt a tanúsítványa alapján, egyúttal biztosítja a szolgáltatást igénybe vevőt, hogy nem egy „álhelyre" lépett *(Kölcsönös hitelesítés).*
- Szoftverkód aláírása, amely biztosítja a felhasználót, hogy az aláírt szoftver *(pl. eszközmeghajtó, makró, stb.)* megbízható helyről érkezett.
- Intelligens kártyás azonosítás, amely a felhasználók tanúsítványát és titkos kulcsát tárolva helyi és távoli bejelentkezést tesz lehetővé.
- Az IPSec protokoll, amely képes a felhasználókat vagy egyéb objektumokat a kiadott tanúsítvány alapján hitelesíteni egy IPSecre épülő kommunikáció során.
- A titkosított állománynrendszer *(EFS)*, amely tanúsítványokat használ mind a felhasználók, mind a helyreállító ügynökök esetén.

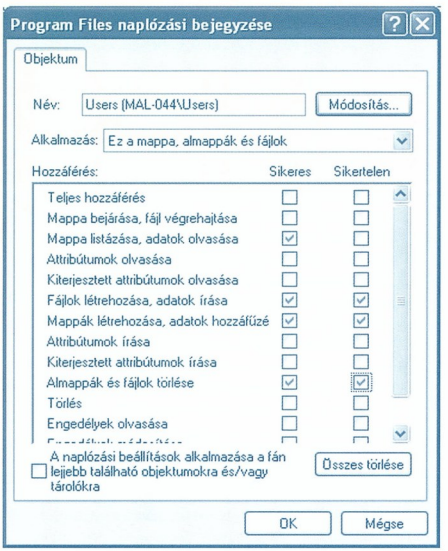


A felsorolás csak az alaprendszerre vonatkozik, amely természetesen rugalmasan kiegészíthető más gyártók által forgalmazott megoldásokkal.
A PKI erőteljesen integrálható a címtárral, kölcsönösen használják egymás képességeit. A PKI által előállított tanúsítványokat a címtár segítségével lehet a felhasználókhoz, vagy számítógépekhez rendelni, és a csoportházirendeket valamint a tanúsítvány-sablonokat is a címtár tartalmazza. Ugyanakkor a címtár a PKI tanúsítványokra támaszkodik, amikor kártyás belépteést használ vagy a Kerberos hitelesítéshez kulcsokat oszt ki. A címtár-integráció azonban nem kötelező. A tanúsítványkiosztó szolgáltatást kétféle módon lehet telepíteni: integrált és nem integrált módon. Ha valamely szervezet nem használja az Active Directoryt, a nem integrált módot kell választania.
A Microsoft PKI megoldása jól skálázható, és a címtárral integrálva komoly versenytársa a hasonló terméket gyártó cégeknek, mivel minden szerverterméknek része ez a komponens. Ezzel együtt nem kizárólagos megoldás. Ha egy szervezet úgy dönt, hogy a szolgáltatások nem elégítik ki az igényeit, bevezethető más PKI megoldás is, például az E-Trusté. Tekintve, hogy egy ilyen rendszer költségeinek nagyobbik része a szervezet és a folyamatok kialakítása, csak alapos mérlegelés és előkészítés után szabad az igényeinknek megfelelő megoldás mellett dönteni.

Naplózás

A Windows NT és így a Windows 2000 fejlesztői is azt a célt tűzték ki, hogy az üzemelés során előforduló eseményeket egyetlen közös helyre gyűjtsék. Ezt a helyet eseménynapló

(event log) névvel látták el. A törekvés mögött az a biztonsági elvárás húzódik meg, hogy bármely objektum bármely tevékenysége naplózható, ezáltal nyomon követhető legyen. A szándék csak többé-kevésbé valósult meg. Vannak olyan események, amelyekről ugyan készül naplóbejegyzés, de nem a központi eseménynaplóban, hanem egy adott alkalmazás saját állományában *(pl. az Internet Information Server vagy a fürtszolgáltatás az ilyen renitensek közé tarozik)*. A tervezők kénytelenek voltak kompromisszumot kötni a naplóbejegyzések és a rendszer teljesítménye között. Az elvi lehetőség ugyan adott a teljes nyomkövetésre, ha azonban valóban minden eseményt naplózni szeretnénk, olyan jelentős teljesítménycsökkenéssel kell számolni, amely már nem teszi kifizetődővé az információk begyűjtését. Ráadásul a megnövekedett bejegyzésszám feldolgozásáról, tárolásáról is gondoskodni kell.



Végeredményben a következő megoldás született: az eseménynaplót három nagy tárolóra osztották: a rendszernaplóba kerülnek azok az események, amelyek az operációs rendszer alapszolgáltatásaival, eszközkezelőivel és a hardvererőforrásaival kapcsolatosak. A rendszeradminisztrátorok nem módosíthatják ezen tárolón a naplózás részleteségét.

Az alkalmazás naplórészébe kerülnek az operációs rendszer kiegészítő szolgáltatásainak *(WINS, DHCP stb.)* eseményei valamint a telepített Microsoft vagy harmadik gyártótól származó alkalmazások bejegyzései, ha erre a szoftvert egyébként felkészítették *(a „Designed for Windows 2000" logó megszerzésének egyik feltétele, hogy az alkalmazás teljesítse ezt a kritériumot)*. Az események részletességét az adott alkalmazásban lehet szabályozni.

A harmadik tároló a biztonsági napló. Tagkiszolgáló és munkacsoportban működő szerverek esetén ez a napló nincs bekapcsolva, a rendszer nem rögzít eseményeket. Ha a szerver tartományvezérlő, a sikeres és sikertelen hitelesítési eseményeket a rendszer itt tárolja. Szemben a másik két naplóréssel, itt konfigurálhatjuk a legrugalmasabban azt, hogy milyen eseményeknek maradjon nyoma. Az állományokhoz és a mappákhoz, amelyeket a felhasználók a megosztásokon keresztül érnek el, napló *(audit)* beállítások rendelhetők. Tulajdonképpen egy objektumra elvégezhető minden egyes művelet sikerességéről vagy sikertelenségéről készülhet bejegyzés. A naplóbeállítások – akárcsak a jogosultságok – öröklődnek a mappahierarchiában *(címtár esetén a címtár-hierarchiában)*, és az öröklődés éppúgy megszakítható vagy újraindítható, ahogy azt korábban megszoktuk.

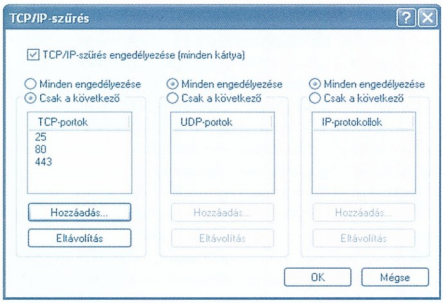
Az engedélyek *(permissions)* betartása vagy megszegési kísérlete mellett a jogok *(rights)* igénybevételéről vagy megszegéséről is készülhet bejegyzés. A naplórend ezen részét a helyi biztonsági házirendben vagy tartományi gépek esetén a biztonsági csoportházirendben szabályozhatjuk. Minden egyes jog sikeres és/vagy sikertelen használatát naplózhatjuk.

Mód van a naplózás egyéb paramétereinek központi beállítására is. Meghatározhatjuk a napló méretét, vagy hogy mennyi ideig őrizze a rögzített eseményeket. A biztonsági napló kiemelt fontossága miatt lehetőségünk van arra, hogy amennyiben betelik a naplózáshoz szükséges hely, a rendszer leállítsa magát, és mindaddig ne induljon el, amíg a napló adatait ki nem mentettük. Ez természetesen veszélyes beállítás, mert ha nem ürítik rendszeresen a naplót, a szerver váratlanul leállhat. Ha azonban megfelelő eljárásrendet alkalmazunk, biztosítható, hogy egyetlen, biztonsági szempontból kritikus eseményt se mulasszunk el. A naplóban található események törléséhez rendszeradminisztrátori jogok szükségesek, ám ez szintén szabályozható. Az egyes bejegyzések külön nem szerkeszthetők, csak a teljes napló törölhető, ám az üres napló első bejegyzése azt rögzíti, hogy a naplót ki és mikor törölte. Ezek az óvintézkedések azt a célt szolgálják, hogy az üzleti felhasználók ne legyenek kiszolgáltatva saját rendszergazdáiknak.

Hálózati biztonság és titkosítási algoritmusok

A Windows 2000 operációs rendszer család számos lehetőséggel rendelkezik a hálózati forgalom védelmére *(titkosítás)* vagy a hálózati forgalommal szembeni védelemre *(NAT, csomagszűrés stb.)*. Nézzük a képességeket és a felhasználási területüket.

Csomagszűrés. Általában minden TCP/IP implementáció része a csomagok elfogadásának vagy elutasításának lehetősége. A Windows 2000 alaprendszer csak TCP kapuszámok alapján képes csomagok szűrésére. Az ábrán látható módon hálózati csatolónként adhatjuk meg, hogy mely kapukat kívánjuk nyitva hagyni. Számos alkalommal ez jó első védelmi vonalat jelent, tűzfalnak és kizárólagos megoldásnak azonban nem alkalmas. Nemcsak azért, mert a támadásokról nem érkezik riasztás, hanem azért is, mert nem pontosan úgy működik a csomagszűrés, ahogy az egy tűzfalnál megszokott. Egy TCP kapcsolat-felvételi kérésére a kiszolgáló „reset connection" csomagot küld, ezzel elárulva, hogy ő tulajdonképpen létezik, csak az adott kapu le van zárva. Ezt a védelmet statikusnak nevezzük, mert szituációtól függetlenül változatlan.



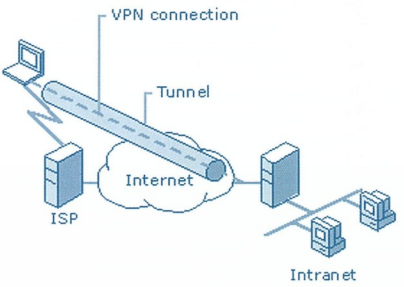
Címfordítás *(Network Address Translation, NAT).* A Windows 2000-ben debütáló szolgáltatás csak részben tekinthető biztonsági elemnek. Nem történik szűrés, csupán egy adott *(általában privát)* TCP/IP hálózati címtartományt egy másik *(többnyire nyilvános)* címre fordít le a rendszer. A jelentősége mégis nagy, ugyanis a NAT elrejti a belső hálózat struktúráját, és a külső szemlélő *(behatoló)* számára kívülről csak egy vagy néhány állomásnak tűnik. A NAT alapja a Microsoft és más gyártók tűzfal megoldásának, például az ISA Servernek.

Távoli hozzáférés *(Remote Access)*. Az egyik legfontosabb – és biztonsági szempontból kritikus – szolgáltatás a távoli hozzáférés biztosítása a felhasználóknak. A távoli hozzáférés biztosítja, hogy munkatársaink bárhol és bármikor képesek legyenek hozzáférni a helyi hálózatokban tárolt információkhoz. A biztonságos távoli hozzáférést az alábbi fejlesztések szolgálják:

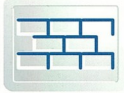
- **Biztonságos felhasználóazonosítás:** A rendszer lehetővé teszi a felhasználót azonosító adatok titkosított cseréjét. Számos szabvány áll rendelkezésre ehhez, többek között az Extensible Authentication Protocol *(EAP)*, a Microsoft-féle Challenge Handshake Authentication Protocol *(MS-CHAP)* 1-es és 2-es verziója, a Challenge Handshake Authentication Protocol *(CHAP)* valamint a Shiva Password Authentication Protocol *(SPAP)*.
- **Kölcsönös azonosítás:** Az egyoldalú azonosítás mellett beállíthatunk kölcsönös, tehát mind az ügyfél, mind a szerver oldaláról kikényszerített azonosítást is. Ezt az EAP-Transport Level Security *(EAP-TLS)* protokoll vagy az MS-CHAP 2-es verzió biztosíthatja.
- **Adattitkosítás:** Ha nem elégséges csak a hitelesítési információ titkosítása, hanem a teljes adatforgalmat kell védenünk, erre is van mód. Az EAP-TLS és az MS-CHAP2 protokollok kiegészítő beállítása lehet az adatfolyam titkosítás. A Windows 95 és az utána kibocsátott Microsoft rendszerek ismerik a Microsoft Point-to-Point Encryption Protocol *(MPPE)* szabványt. Az MPPE RSA RC4 titkosítást használ 40, 56 vagy 128 bites titkos kulccsal. Az MPPE kulcsok az MS-CHAP vagy az EAP-TLS hitelesítési folyamat során generálódnak.
- **Visszahívás:** Egy egyszerű biztonsági beállítás. Lényege, hogy a hitelesítés után a szerver bontja a kapcsolatot, és egy előre meghatározott számot tárcsáz. A kapcsolat bármely telefonszámról kezdeményezhető, de csak egyetlen számot hív vissza a rendszer, tehát csak itt használható.
- **Hívó-fél azonosítás:** Ha a telekommunikációs rendszerelemek lehetővé teszik, beállítható, hogy a betárcsázást csak egy adott számról lehessen kezdeményezni. Ez egy ritkán használt szolgáltatás, a visszahívásnál annyival szigorúbb, hogy már a kezdeményezés sem történhet akárhonnan.
- **Fiókzárolás távoli hozzáférés esetén is:** Különösen VPN kapcsolat során fontos, hogy bizonyos számú próbálkozás után a rendszer kizárja a bejelentkezni igyekvőt, feltételezve, hogy illetéktelenül próbál valaki behatolni. A rendszergazdák beállíthatják, hogy hány sikertelen próbálkozás után zárja ki a rendszer a felhasználót, továbbá azt is, hogy mennyi idő után kapcsolja vissza a kizárt fiókot.

A rugalmasabb konfigurálás elősegítése érdekében a Windows 2000 a távoli hozzáférések szabályozására is tartalmaz házirendet. Ennek segítségével olyan szabályokat definiálhatunk, hogy „A pénzügyi csoport reggel nyolc és délután négy között nem tárcsázhat be, egyébként pedig a betárcsázás után csak a levelező szerverrel kommunikálhat". A szervezet működését megismerve számos olyan szabályt lehet létrehozni, amely csökkentheti az esetleges támadások mozgásterét.

Virtuális magánhálózat *(Virtual Private Network, VPN)*. A betárcsázás helyett egyre gyakoribb a VPN kapcsolatok kialakítása. A Windows 2000 ugyanazzal a komponenssel *(Routing and Remote Access)* oldja meg mindkét igényt, így a biztonsági szolgáltatások is nagyon hasonlóak, sokszor ugyanazok. A VPN azonban kiegészül három olyan szabvánnyal, amelyet a RAS nem ismer. Ezek a PPTP, az L2TP és az IPSec.



A PPTP *(Point to Point Tunneling Protocol)* a Windows 95-től kezdve valamennyi Microsoft operációs rendszer által ismert szabvány. MPPE módszert használ a VPN csatorna kialakítására. MS-CHAPv2-vel és erős jelszóval biztonságos kapcsolat építhető fel vele. Könnyű telepíteni és konfigurálni, integrálható smart-kártyás hitelesítési módszerrel *(EAP-TLS segítségével)* és együttműködik a NAT szolgáltatással.
Az L2TP *(Layer 2 Tunneling Protocol)* az IPSec *(Internet Protocol Security)* szabvánnyal együtt képes a PPTP-hez hasonló szolgáltatást nyújtani. Alkalmazásuk esetén lehetőség van a felhasználó mellett a számítógép azonosítására is, továbbá minden egyes csomagot ellenőriz a rendszer integritása és hitelessége szempontjából. A kétségtelen biztonsági előnyökkel szemben azonban számos nehézségbe ütközhetünk egy ilyen VPN rendszer kialakításakor. Az L2TP/IPSec feltételezi egy PKI infrastruktúra előzetes felállítását, amely kis szervezetek számára költséges lehet. A Windows 2000-ben található imple-



mentáció nem működik együtt a NAT szolgáltatással, mert a NAT lecseréli a csomag IP címét, ami az integritás megsértését jelenti. A teljes igazsághoz hozzátartozik, hogy mind a NAT, mind az IPSec egy Internetes szabvány, amelyet a Microsoft pontosan követ. Az inkompatibilitás tehát a szabványok összeférhetetlenségéből adódik, és nem szoftvercég hibája. *(A Windows .Net Server VPN implementációját felkészítik a NAT-tal való együttműködésre egy új szabványtervezetet figyelembe véve.)* További probléma, hogy az L2TP/IPSec együttest csak a Windows 2000 és Windows XP ügyfelek ismerik. A Microsoft 2002 augusztusában kiadott egy „Microsoft L2TP/IPSec VPN Client" szoftvert, amely a Windows 98, Windows Millennium Edition és Windows NT 4.0 rendszerekhez nyújt megoldást.

Az IPSec. Az Internet Security Protocol nem csak a VPN megoldásoknál használható. Egy olyan nyílt szabvány implementációjáról van szó, amely két célt tűzött ki maga elé: védeni az IP csomagokat, és védelmet nyújtani a hálózat elleni támadásokkal szemben. Az IPSec a következő támadásokkal szemben képes hatékony védelmet nyújtani:

- **Hallgatózás *(sniffing)*:** Az IPSec az IP csomagokat titkosítja, tartalmuk harmadik fél számára nem értelmezhető.
- **Adatmódosítás:** Az IPSec kriptografikus szabványok szerint előállított kulcsokat használ, amelyeket csak a kommunikációban résztvevő számítógépek birtokolnak. Ezen kulcsok segítségével titkosítja, és digitális ellenőrzőösszeggel látja el az IP csomagokat, így azonnal kiderül, ha bármilyen módosítási kísérlet történt.
- **Hamis azonosság, szolgáltatás-túlterhelés:** Az IPSec segítségével lehetséges olyan kommunikáció, amelyben csak előzetesen azonosított és megbízható felek vesznek részt.
- **Man-in-the-Middle:** A kölcsönös azonosítás kizárja, hogy egy harmadik fél a szerver felé ügyfélnek, az ügyfél felé pedig szervernek adja ki magát.

Az IPSec nyílt szabvány, ami nem jelenti azt, hogy a Microsoft ne adhatna többletszolgáltatást a megoldásához. Az IPSec a Windows 2000 tartományi modelljét használja, az IPSec házirend pedig Kerberos v5 hitelesítést alkalmaz, hogy azonosítsa a kommunikációban résztvevő számítógépeket. A tartománymodell segít az IPSec alapú kommunikáció kialakításában.
Ahogy azt már korábban más szolgáltatásoknál is megszokhattuk, az IPSecre vonatkozóan is kialakíthatunk házirendeket, amelyek azután a tartomány minden számítógépén *(amely a házirendet értelmezi)* érvényre jutnak. Nem kell tehát minden egyes gépet külön-külön beállítani.
Szerencsés módon az IPSec „átlátszó" szolgáltatás, – nincs szükség az alkalmazások újraírására. A felsőbb szintű alkalmazások tulajdonképpen nem is érzékelik, hogy az IPSec működik. Minden kommunikáció, amely TCP/IP alapú, biztonságosabbá tehető vele.
A Windows 2000 rendszerben az Internet Key Exchange *(IKE)* dinamikusan cseréli és kezeli a titkosító kulcsokat a kommunikáló felek között.
Az IPSec integrálható a PKI infrastruktúrába, a hitelesítéskor használhatók a kiadott tanúsítványok. Így azokkal a számítógépekkel is biztonságosan lehet kommunikálni, amelyek nem tagjai egy közös erdőnek, de megfelelő tanúsítvánnyal rendelkeznek. Ha azonban nincs PKI, az sem probléma, mert be lehet állítani egy előre megadott közös jelszót, amely a hitelesítés alapjául szolgál. Természetesen ez elviekben kevésbé biztonságos, hiszen meg kell oldani, hogy a közös jelszó egy biztonságos csatornán keresztül kerüljön a két félhez.

IPSEC „tűzfal". Az IPSec szabvány alkalmazásából „adódik" egy lehetőség a hálózati csatolókon zajló kommunikáció szabályozására. Lényegében egyfajta csomagszűrést lehet megvalósítani. A korábban ismertetett megoldáshoz képest azonban további előnyök is jelentkeznek. Egy-egy IP címre és hálózati csatolóra külön megadható szabályokról van szó. A beállítás nem igényel újraindítást. A lezárt kapukra érkező csomagok eldobásával a szerver „rejti magát", – viselkedése megegyezik a valódi csomagszűréssel.
Előnyös tulajdonságai ellenére ez a megoldás sem tekinthető tűzfalnak. Nem dinamikus, nem kapunk riasztást támadás esetén, és nem integrálható a NAT-tal. Ennek ellenére a módszert lehet ajánlani, mert jó alapot nyújt az egyéb megoldások mellett.

A beépített alkalmazások biztonsága

A Windows 2000 rengeteg beépített alkalmazást tartalmaz, amelyek részben az operációs rendszerre támaszkodva saját biztonsági funkciókkal is rendelkeznek. Röviden tekintsük át, melyek ezek az alkalmazások, és milyen védelmi képességekkel ruházták fel őket.

Internet Information Server *(IIS)*

Az Internet Information Server *(IIS)* a Windows 2000 beépített webszervere. Tulajdonképpen három fő szolgáltatása van: a web *(HTTP)*, a levél *(SMTP)* és a hírlevél *(NNTP)*. Mi most tág értelemben használjuk az IIS kifejezést, mindhárom szolgáltatást beleértve.
Az IIS biztonsági alrendszerét szorosan integrálták a Windows 2000-hez. Képes titkosított csatorna kiépítésre a webes ügyfelekkel a HTTPS és a TLS protokollokat használva. A webszerveren keresztüli állomány-hozzáféréseket a Windows 2000 engedélyrendszerével szabályozhatjuk.
Az IIS többféle módon képes az ügyfelek hitelesítésére. A nyilvános adatokhoz elégséges azonosítatlan hozzáférés. Ekkor az ügyfél nem azonosítja magát, a webszerver pedig az IUSR_<szervernév> elnevezésű felhasználóhoz tartozó kontextusban jár el. A basic hitelesítésnél már azonosítja a felhasználót, de az identifikációhoz szükséges jelszó kódolatlanul halad át a hálózaton. Ha a teljes adatforgalmot előzőleg nem titkosítottuk, ez a módszer nem biztonságos. A harmadik módszer az integrált Windows hitelesítés, melynek előnye az előző eljáráshoz képest a biztonságos jelszóátvitel, hátránya azonban, hogy csak az Internet Explorer böngésző alkalmas a hitelesítés elvégzésére. További aggály a módszerrel szemben, hogy gyenge jelszó és szofisztikált támadás esetén mégiscsak mód van a jelszó illetéktelen megszerzésére. A negyedik módszer megoldást jelenthet a problémára, ekkor ugyanis az előre kiosztott felhasználói tanúsítvány segítségével azonosítja a böngésző a szerver felé a felhasználót. Bár messzemenően ez a legbiztonságosabb eljárás, csak korlátozottan használható, mert feltételezi a nyilvános kulcsú infrastruktúra meglétét.
Az IIS-ben szabályozhatjuk, hogy a szerveroldali kódokat a felhasználók végrehajthatják-e, és ha igen, milyen feltételek mellett. Ez minden egyes virtuális könyvtár esetén külön-külön megtehető. Meglévő PKI infrastruktúrát feltételezve az IIS ellátható szerveroldali tanúsítvánnyal, amely segítségével az ügyfelek meggyőződhetnek arról, hogy a kiszolgáló nem egy

imposztor. A hitelesítéssel együtt így megvalósulhat a kommunikációban résztvevő felek kölcsönös azonosítása.
A számos biztonsági funkció mellett is az egyik „leglyukasabb" Microsoft alkalmazás az IIS. Profi hackerek és jószándékú hibakeresők gyakran fedeznek fel olyan, általában programozás-technikai hibából adódó biztonsági réseket, amelyek komolyan veszélyeztethetik a rendszerre bízott adatokat. A teljes igazsághoz az is hozzátartozik, hogy ez az az alkalmazás, amely a „nyílt Interneten", a frontvonalban van, tehát a leginkább kitett mindenféle próbálkozásnak.
A helyzet más szempontból is súlyos. A Microsoft következetes fejlesztési politikájának eredménye az, hogy az azonos funkcionalitású komponenseket csak egyszer fejleszti ki, és inkább függőségeket teremt egymástól egyébként „függetlenül" fejlesztett alkalmazások között. Az IIS telepítése szükséges előfeltétele az Exchange 2000 telepítésének, mert az Exchange 2000 az IIS számos szolgáltatását *(SMTP, NNTP, web alapú hozzáférés, Metastore stb.)* használja. Az IIS sebezhetősége tehát egyben a rá épülő alkalmazások sebezhetőségét is jelentheti.

Dynamic Host Configuration Protocol *(DHCP)*

A Dynamic Host Configuration Protocol szabvány majdnem kizárólag szórt üzenetcsomagokkal dolgozik a harmadik és negyedik hálózati rétegben. Az IPSec módszert nem számítva nehéz biztonságossá tenni ezt a szolgáltatást. Pedig nagy szükség lenne rá, mert egy elindított, de véletlenül vagy szándékosan rosszul konfigurált szerver megakadályozhatja, hogy a számítógépek képesek legyenek bármiféle kommunikációra. Egy Windows 2000 tartománystruktúrában úgy próbálják elejét venni kalóz DHCP szerverek megjelenésének, hogy kötelezővé teszik a tartományban működő valamennyi DHCP szerver működésének engedélyezését. Ha ilyen engedéllyel nem rendelkezik a szerver, akkor a szolgáltatás automatikusan leáll. Az engedély megadását enterprise administrator jogkörhöz rendelték. A DHCP szolgáltatás „szabadidejében" kalóz DHCP szervereket keres, és ha talál ilyet, naplóbejegyzés formájában értesíti az adminisztrátorokat. Bár mindenféle DHCP-n alapuló támadásra ez nem jelent megoldást, mégis nagy segítség a hálózat-felügyeletben.

Domain Name System *(DNS)*

A Windows 2000 fejlesztői a címtár struktúráját erőteljesen integrálták a DNS névfeloldási szabványhoz. DNS nélkül nincs Active Directory sem, másképp fogalmazva, a DNS szolgáltatás elleni támadás a címtár, és így minden címtárral integrált alkalmazás működését teszi lehetetlenné.
Az operációs rendszerben implementált DNS megoldás teljes egészében szabványos, de a programtervezők néhány speciális funkcióval is kiegészítették a névfeloldást. A funkciók egy része a DNS biztonságát hivatott növelni. Az egyik ilyen funkció a zónák Active Directoryban való tárolási lehetősége *(Active Directory Integrated DNS Zone)*. Egy ilyen integrált zóna írható és olvasható is, ezáltal megszűnik a DNS szabványból fakadó gyenge pont, miszerint egy zóna fájlnak csak egyetlen írható példánya van. Ráadásul a címtárban már megismert jogosultságrendszer kiterjeszthetővé válik a zónákra is, így azok adminisztrációját szigorú módon szabályozhatjuk. Egy zóna bevonása a címtárba azt is jelenti, hogy a zóna replikációja a címtár replikációs infrastruktúráját fogja használni, amely minden esetben redundáns, tehát a működés szempontjából biztonságos. Ha pedig ez a replikáció nem megbízható, vagy nem ellenőrzött adatcsatornán történik, akkor a címtárba épített védelmi képességeket az integrált zónák is élvezik.
A zónák frissítése automatikusan történhet, mert erre a DNS szervert felkészítették. Beállíthatjuk azonban, hogy csak olyan számítógépek frissíthessék a zónát, amelyek azonosítása korábban már megtörtént, tehát illetéktelen zónafrissítés kizárható. Ennek a funkciónak azonban van tervezési hibája: ha egy új altartományt akarunk létrehozni, akkor a DNS zóna egy olyan tartomány tartományvezérlőjét keresi, amely még nem létezik, hisz épp most készül el. A hitelesítés sikertelensége így a zónafrissítést is meghiúsítja, végeredményben nem jön létre az új altartomány. Jelenleg tehát a biztonságos zónafrissítés nem minden esetben használható szolgáltatás.

Fürtszolgáltatás *(Cluster)*

Némi biztonsági funkciót a nagy rendelkezésre állást biztosító fürtszolgáltatásba is beépítettek. A fürtadminisztrátor alkalmazást ugyan bárki elindíthatja, de az erőforrások menedzselését csak az végezheti el, akinek erre külön joga van. A biztonsági funkciók azonban nem túl részletesek, nincs mód erőforrásonként engedélyek kiosztására, sőt, ezt még csoportonként sem tehetjük meg, kizárólag a teljes fürtre vonatkozó jogokat definiálhatunk.

Lepenye Tamás, MCSE 2000
lepenyet@mal.hu

Dupla KU

Hiberfil.sys

K: *Windows XP-t telepítettem és csakhamar elfogyott a hely a rendszerpartíción. Megvizsgáltam a rejtett fájlokat és találtam a gyökérben egy hiberfil.sys fájlt. Gondolom ez a hibernáláshoz kell, de én sosem használtam ezt a funkciót. Hogy tudnám eltávolítani a fájlt? Letörölni nem tudtam!*

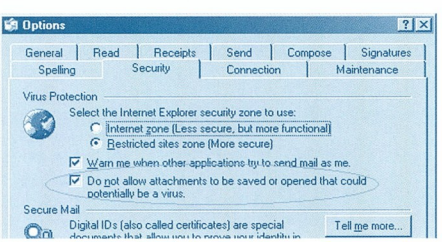
V: Valóban, a hiberfil.sys a hibernáláshoz használatos, a gép hibernálásakor ebbe a fájlba menti el a teljes memóriatartalmat a Windows XP. Ezért aztán a hiberfil.sys mérete mindig megegyezik a fizikai memória méretével. Ha ezt a helyet szeretnénk felszabadítani, ki kell kapcsolni a hibernálás funkciót, amely alapértelmezésben be van kapcsolva:
Control Panel → Energiagazdálkodás *(Power Options)* → A Hibernálás vagy Hibernate tulajdonságlapon ki kell kapcsolni a hibernálást.

Az OK megnyomása után a hiberfil.sys fájl el fog tűnni a rendszerpartícióról és felszabadul a hely.

Mellékletek mentése Outlook Expressből

K: *Outlook Express 6-os verziója nem engedi a mellékletek tartalmát lementeni. Eddig nem volt ezzel probléma, csak mióta a 6-os verziót használom. Hogy tudnám menteni mégis a mellékleteket?*

V: az Outloook Express 6-os verziója biztonsági okokból alapértelmezésben nem engedi lementeni a mellékleteket, ezzel akadályozva a vírusok terjedését. A következő módon lehet kikapcsolni ezt a védelmet:
Outlook Express Eszközök *(Tools)* menü → Beállítások *(Options)* → Biztonság *(Security)* tulajdonságlap.
Itt ki kell venni a pipát a képen bekeretezett beállítás elől.



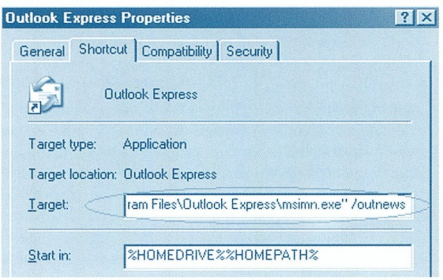
A mellékletek mentésének engedélyezése

Az OK után a beállítás azonnal érvényre jut.

Outlook Express, csupán hírolvasó ügyfél *(NNTP kliens)*

K: Outlook 2000-et használok levelezőprogramként, és Outlook Expresst hírolvasóként *(vagyis NNTP kliensként)*. Nincs szükségem az Outlook Express levelező szolgáltatására, csupán hírolvasóként szeretném használni. Van erre valami megoldás?

V: Meg lehet szabadulni a levelező kliens funkcióktól, ehhez csupán annyit kell tenni, hogy a /outnews kapcsolóval kell indítani az Outlook Expresst.
A Start Menü Futtatás ablakából az **msimm /outnews** paranccsal indíthatjuk: nem lesz Inbox és az Accounts menüből eltűnik a Mail menüpont.
Természetesen a parancsikont is megváltoztathatjuk, ilyenkor a parancs végéhez az idézőjeleken kívülre kell írni a /outnews kapcsolót, például így:



OE indítása hírolvasóként

Windows 2000 DNS Forwarder

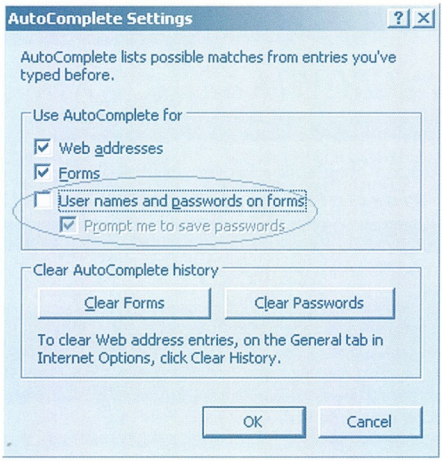
K: *Frissen telepítettem Windows 2000 Active Directoryt az első tartományvezérlővel, a DNS Active Directory-integrált, mindent maga csinált. A DHCP-n beállítottam, hogy a munkaállomások ezt a szervert használják DNS-ként, innentől kezdve a klienseken nem működik az Internetes névfeloldás. Gondoltam beállítok forwardert a DNS szervernek, de ezt meg a Windows 2000 DNS nem engedte. Most akkor mit lehet tenni, vagy Internet vagy Windows 2000 AD? Mind a kettő kellene!*

V: Az Active Directory telepítő varázsló, ha nincs közvetlen internetes kapcsolat, létrehoz egy root zónát a DNS-ben a normál zóna mellett. Egyszerűen le kell törölni a . ***(pont)***nevű zónát a zónák listájából, ezután már be lehet állítani a forwardert.

Internet Explorer AutoComplete

K: *Szeretném minden felhasználó esetén letiltani, hogy az Internet Explorer megjegyezze a beírt jelszavakat. Azt is szeretném megakadályozni, hogy ezt egyáltalán felajánlja a felhasználóknak. A környezet igen vegyes, Active Directory ugyan már működik, de a Windows 2000-ek és XP-k mellett még vannak Windows NT 4.0 kliensek is. Remélem, van lehetőség központilag tiltani ezt a „fícsört".*

V: Az alábbi képen az látható, hogy mindezt hogyan tudjuk tiltani az Internet Explorer beállításai közt:



Internet Explorer AutoComplete kikapcsolása

Ez az ablak az Internet Explorer Tools → Intenet Options panel → Content tulajdonságlap → AutoComplete gomb mögött bújik meg. Jól el van rejtve.
Ha kivesszük a pipát, nem fogja felajánlani a felhasználóknak a beírt neveket és jelszavakat, viszont a már beírtakat nem fogja kitörölni. A korábban beírt jelszavakat és neveket csak manuálisan lehet kitörölni. Legegyszerűbben az egész profil újragenerálásával lehet talán megszabadulni tőle.
Ez eddig nyilván nem az automatikus beállítás. Az egyes operációs rendszerekre és böngészőverziókra más-más módszer használható.
Windows XP és Windows 2000-es ügyfelek esetén viszonylag egyszerű a megoldás. A csoportos házirendben tudjuk szabályozni az Internet Explorert, hogy ne ajánlja fel a nevek és jelszavak megjegyzését. A csoportos házirendek közt a User Configuration →Administrative Templates → Windows Components → Internet Explorer alatt található két beállítás amely ezt szabályozza:

- Disable Autocomplete for Forms – ha ezt Enabled státuszba állítjuk, az IE nem fogja kiegészíteni az űrlapokban a neveket és a hozzá tartozó jelszavakat. *(Ez a felső pipa a kép bekeretezett részén)*
- Do not allow AutoComplete to save passwords – ha ezt is tiltjuk, az Internet Explorer nem fogja felajánlani a jelszavak mentését. *(ez az alsó pipa a kép bekeretezett részén)*

Ha ezeket a beállításokat a Default Domain házirendben állítjuk be, nagy valószínűséggel kivétel nélkül mindenkire hatni fog a beállítás. Természetesen, ha csak bizonyos felhasználókra szeretnénk ezt a tiltást beállítani, külön szervezeti egységet képzünk nekik – vagy csoporttagság alapján is szabályozhatjuk a házirendet.
Egyébként a beállítás az Internet Explorer 5.01-es verziójától kezdve változatlan.

Windows NT 4.0 esetén kicsit más a megoldás, ilyenkor talán közvetlenül a regisztrációs adatbázis módosításával legegyszerűbb ezt a beállítást kierőszakolni. Ehhez létre kell hozni a következő .reg kiterjesztésű fájlt.

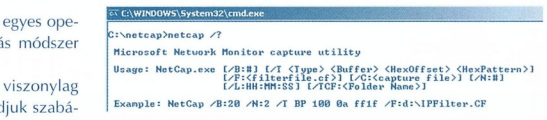
```
REGEDIT4
[HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\Main]

"FormSuggest PW Ask"="no"

"FormSuggest Passwords"="no"
```

Ezt a fájlt mondjuk a login scriptből lehet a felhasználónak eljuttatni *(a regedit /s <fájl név> paranccsal)*. Kérdés, hogyan tudjuk megtiltani a felhasználóknak, hogy ezt a beállítást kézzel visszaváltoztassák. Ehhez az Internet Explorer Administration Kit-ből kinyerhető .adm fájlra lesz szükségünk, amelynek segítségével szabályozhatjuk, hogy mit tud átállítani a felhasználó az Internet Explorer beállításai közt.

Windows XP és a Network Monitor driver

K: *Szeretném Network Monitorozni az egyik Windows XP-t a hálózatunkon, ehhez kellene Network Monitor agentet telepítenem, de sehol nem találom. Windows 2000 esetén fel lehetett telepíteni, a Windows 2000 része volt. Windows XP-ben már nincs ilyen?*

V: A Windows XP már nem tartalmazza a Network Monitor drivert, de külön fel lehet telepíteni. A telepítő CD-n található Support Tools része a netcap.exe program, ami többek közt a Windows 2000 Network Monitor driver megfelelője is.
A Netcap egy parancssori progam, amely a telepítő CD-n a Support\Tools könyvtárban található support.cab fájlból kibányászható.



A netcap.exe kapcsolói

Első futtatáskor kérdés nélkül telepíti a Network Monitor drivert, ami elengedhetetlen a monitorozáshoz. Ezen túlmenően parancssorból monitorozhatunk is vele az adott gépről, tehát nem csak kliensként tud működni. Netcap /? segítségével kaphatjuk meg a használható kapcsolókat. Jó szórakozást!

NETACADEMIA MESTERQRZUSOK

2003. Tavaszi program

A tavaszi előadások vendégelőadói értékes ajándékokkal is meglepik a kedves résztvevőket!

Kérjük, hozza magával a korábban kiosztott kedvezményes jegyet!

2003. március 27. 13:45 – 18:00

13:35 – 14:00 Regisztráció

14:00 – 15:00 **Bevezetés a hálózati forgalom elemzésébe**

Minél összetettebb egy vállalati hálózat, minél többféle szolgáltatást futtatunk, annál furcsább hibajelenségeknek lehetünk időnként tanúi. A hálózati forgalom elemzésével olyan hibák felderítésére is lehetőség nyílik, amelyekről semmit sem mond az Eseménynapló!

(Fóti Marcell)

(Kapcsolódó tanfolyamaink: NetMon Workshop, TCP/IP Workshop)

15:00 – 16:00 **A nyílt kulcsú technológia építőkockái**

A digitális aláírási törvény megjelenésével egyre több vállalat fordul érdeklődéssel a nyílt kulcsú technológiák felé. Mi kell ahhoz, hogy elérhetővé váljon számunkra a PKI adta sok-sok lehetőség?

(Vendégelőadó: NetLock Kft.)

(Kapcsolódó tanfolyamaink: PKI Workshop)

16:00 – 16:20 Kávészünet

16:20 – 17:20 **Nyílt kulcsú architektúra a Windowsban**

A PKI felhasználási területei a titkosított hálózati kommunikációtól a digitális aláíráson keresztül a felhasználó-azonosításig terjednek. Kiszolgálóoldalon többek között a RAS, VPN, IPSec, és IIS, felhasználói oldalon a Smart Card Logon, dokumentumok aláírása, S/MIME a témaválaszték. A műsorváltoztatás jogát az újonnan megjelenő szolgáltatások miatt fenntartjuk!

(Fülöp Miklós)

(Kapcsolódó tanfolyamaink: 2150 – Windowsos hálózatok biztonsága, 2159, ISA Server)

17:20 – 18:00 **Gyakorlat: Felhasználói és kiszolgálóoldali PKI-gyakorlatok**

Ajándék: Hiteles NetLock tanúsítványpár elektronikus aláíráshoz és titkosításhoz. A kulcstároló eszközök a helyszínen megvásárolhatók.

2003. május 29. 13:45 – 18:00

13:35 – 14:00 Regisztráció

14:00 – 15:00 **Az elektronikus levelezés védelmének lehetőségei Exchange Serveren**

Az elektronikus levelezés áldás és átok egyben. A veszélyes leveleket nem szabad a felhasználókig eljuttatni. Ennek eszköze lehet az Outlook-ügyfelekhez készített központi korlátozóeszköz, vagy egy víruskereső.

(Fóti Marcell)

(Kapcsolódó tanfolyamaink: 1572 – Exchange Server üzemeltetése)

15:00 – 16:00 **SOAP** (Előadó: Soczó Zsolt)

A webszolgáltatások alapjait a SOAP-szabvány rakta le. Az előadásban áttekintjük a SOAP történelmi gyökerét, és az XML-technológiákon keresztül megnézzük gyakorlati felhasználását is.

(Soczó Zsolt)

(Kapcsolódó tanfolyamaink: .NET fejlesztési tanfolyamok)

16:00 – 16:20 Kávészünet

16:20 – 17:20 **A Windows rendszerek támadható, és védelmi felületei**

A vírusvédelem egyik faktora nyilván a jó, friss víruskereső program. A másik faktor az okos felhasználó, illetve az okos rendszergazda. Ebben az előadásban megismerkedünk a vírusok által használt támadási trükkökkel, így soha többé nem fogunk Administratorként levelezni…

(Vendégelőadó: VirusBuster Kft.)

(Kapcsolódó tanfolyamaink: 2150 – Windowsos hálózatok biztonsága, 2159, ISA Server)

Gyakorlat: Felhasználói és kiszolgálóoldali PKI-gyakorlatok

Ajándék: Virus Buster Personal Edition, 1 éves előfizetéssel

A MesterQrzusok helyszíne: NetAcademia Kft., Budapest, Andrássy út 62.
Jelentkezés: http://www.netacademia.net/mesterq

Microsoft .NET fejlesztői képzéssorozat a SZÁMALK Továbbképzésnél

25% kedvezménnyel

Kedvezményes konstrukció, rugalmas időbeosztás • A kezdő szinttől a haladó ismeretekig.
Párhuzamosan sajátíthatja el a Visual Basic .NET és a C# nyelv használatát.
Szerezzen nemzetközi (MCAD/MCSD) minősítést!

Microsoft CERTIFIED Application Developer

A Microsoft .NET fejlesztői képzés hat darab hivatalos Microsoft tanfolyamra épül, összesen öt hétben. A tanfolyamok minden hónap egy-egy hetében kerülnek megrendezésre, vagyis 3–5 napot jelentenek havonta.

Microsoft CERTIFIED Solution Developer

Kezdési időpont: 2003. március 10.

2559/2609-es tanfolyam (Visual Basic .NET és C#.NET):	2003. március 10–14.
2071+2389-es tanfolyam (adatkezelés, SQL 2000, ADO.NET):	2003. március 31–április 4.
2555/2565-ös tanfolyam (Windows alkalmazásfejlesztés):	2003. április 22–24.
2310-es tanfolyam (webes alkalmazásfejlesztés):	2003. május 26–30.
2524-es tanfolyam (XML/webszolgáltatások fejlesztése):	2003. június 30–július 2.

A tanfolyamok **biztosan indulnak**. Erre nyugodtan tervezhet ... amíg a szabad helyek tartanak!

A felsorolt tanfolyamok (külön-külön) összköltsége akár a 750.000 Ft-ot is elérheti.

Nálunk most az öt tanfolyam **kedvezményes ára:** 595.000 Ft/fő.

Az összeget tanfolyamonként több részletben is lehetőség van kifizetni.

Ismerje meg a **jövő fejlesztőeszközét** és **technológiáit** a SZÁMALK-nál!

Microsoft CERTIFIED Technical Education Center

SZÁMALK TOVÁBBKÉPZÉS

SZÁMALK

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