Békési Gábor*

BIZTONSÁGOS WEBSZOLGÁLTATÁSOK FEJLESZTÉSE MICROSOFT PLATFORMON

Visszatekintés

Ahhoz, hogy a későbbiek érthetők legyenek, szükségünk van a tárgykörben egy rövid terminológiai/fogalmi ismeretfrissítésre, igaz, jószerével a kulcsszavak szintjére szorítkozunk.

A webszolgáltatásokban és azok biztonságában az alábbi fogalomkörök játszanak szerepet:

– Szervizközpontú architektúrák (SOA)
  A SOA-rendszerek az üzleti folyamatokat szolgáltatások sorozatára képezik le. Legelterjedtebb megvalósításuk a webszolgáltatások.
– Webszolgáltatásokról akkor beszélünk, ha a szolgáltatásra teljesülnek az alábbiak:
  - a felek XML-alapú kommunikációt folytatnak,
  - platformfüggetlenség,
  - szolgáltatásleírásuk van és felkutathatók,
  - a hálózatuk (többnyire) az internet.
– Az adatbiztonság kritériumai:
  - a felhasználó hitelesíthetősége,
  - a felhasználóhoz jogosultságok tartoznak,
  - az eseményeket folyamatosan ellenőrzött naplók rögzítik,
  - a kényes információkat titkosítás védi,
  - az üzenetek sértetlenségét azok aláírása garantálja,
  - a szolgáltatás mindig rendelkezésre áll.
– A WCF és fejlesztői környezete
  A 2007-ben kibocsátott Windows Communication Foundation (WCF) v3.5 a Microsoft szolgáltatásalapú rendszerek létrehozását támogató terméke. A WCF olyan, a .NET Framework[1] lehetőségeit kihasználó, gazdag hálózati megvalósítást biztosító keret, amely a Visual Stúdió 2005 fejlesztőeszközök körébe illeszkedik.

* *főiskolai tanár, Általános Vállalkozási Főiskola*

[1] *Legfontosabb jellemzője, hogy az ellenőrzött kód-szerelvényekből felépülő programokat egy futtató rendszer hajtja végre.*

Biztonságos szolgáltatásalapú rendszerek fejlesztési folyamata

A következő táblázat a biztonsági elvek megjelenését mutatja a szoftverfejlesztés fázisaihoz kapcsolódva:

1. táblázat
A szoftverfejlesztés szakaszaihoz kapcsolódó biztonsági teendők

Fejlesztési szakaszok	Biztonsági teendők
Követelmények, helyzetfelmérés	Biztonsági elvárások
Tervezés, rendszerspecifikáció készítése	A fenyegetettség-modell felállítása, biztonsági terv készítése
Fejlesztés (programspecifikáció, kódolás)	A biztonsági feladatok érvényesítése (konfigurálás/kódolás)
Tesztelés modul-, ill. rendszerszinten	Biztonsági teszt
Installáció	A biztonsági környezet installálása

Forrás: Patterns & practices (2008)

Biztonsági elvárások: Adatvédelmi kritériumok, amelyeknek teljesülniük kell. Üzleti tevékenységekhez kapcsolódnak. A tevékenységek biztonsági zónák között zajlanak.

Fenyegetettség-modellezés: Azonosítja a rendszer biztonsági réseit, az ellenük intézhető támadásokat, a sérüléseket és a védelmi lépéseket.

A biztonsági feladatok érvényesítése történhet kódolással illetve konfigurálással. Mérlegelni kell a megoldások előnyeit, hátrányait. (Például a konfigurálás rugalmasabb, de nehezebben védhető meg.)

Biztonsági teszt: Fenyegetettség-modellünk tesztelése a kész rendszerben. Támadások szimulálása.

A biztonsági környezet installálása az alkalmazás, a gazdarendszer (*host*) és a hálózattal szembeni biztonsági követelmények érvényesítését jelenti. Az üzemszerű működéshez szükséges hiteles tanúsítványok beszerzése, a felhasználói azonosítók adatbázisának létrehozása stb.

Biztonsági kategóriák

Az alábbi tevékenységek és fogalmak a szolgáltatás adatbiztonságának megteremtéséhez járulnak hozzá. Ismeretük, alkalmas helyen történő használatuk a biztonság elleni támadások kivédésének alapja.

2. táblázat
Biztonsági kategóriák

Biztonsági kategória	Jelentése
Naplózás és monitorozás	A biztonsági eseményeket naplóállományokba írják, és a tartalmukat ellenőrzik.
Autentikáció, (felhasználóhitelesítés)	A felhasználó hitelesítése, azonosítása egy folyamat, melyben az egyed személyes azonosítóival (pl. felhasználónév és jelszó) igazolja magát.
Jogosultságkezelés (*authorization*)	A szolgáltatások jogosultságkezelése révén történik az erőforrásokhoz, műveletvégzéshez szükséges hozzáférési jogok ellenőrzése.
Konfigurációkezelés	A konfigurációkezelést a működési környezet beállítására használják (pl. az adatbázis-kapcsolatok megadása, tanúsítványok azonosítása stb.)
Kivételkezelés	A kivételek kezelése, különösen a hibákra adott üzenetek tartalma miatt, biztonsági rést jelent.
Megszemélyesítés, delegáció	Azon módok összessége, melyeken a felhasználó, a feldolgozási folyamat későbbi állomásain azonosítani képes magát.
Üzenettitkosítás	Az üzenet tartalma rejtjelezett.
Üzenetismétlés felderítése	Fel kell tudni ismerni az ismételten beküldött üzeneteket.
Üzenet aláírása	A digitális aláírás az üzenet hitelességét és érintetlenségét igazolja.
Érvényességellenőrzés	Az üzenetek input/output adatainak validációját jelenti (pl. méretre, karakterkészletre, tartalomra ellenőrzik).
Érzékeny adatok kezelése	Érzékeny adatok azok a felhasználói, ill. alkalmazásra vonatkozó adatok, melyek titkosan kezelendők (pl. a jelszó).
Munkamenet (*session*)-kezelés	Egy munkamenet alatt az ügyfél és a szolgáltató hosszú távú kapcsolatban állnak, több üzenetet is váltanak. (A munkamenet azonosítójának rossz szándékú megszerzése különösen veszélyes.)

Forrás: (Patterns & practices, 2008)

Fenyegetettség és támadási lehetőségek

Táblázatunk az előzőekben ismertetett kategóriák biztonsági réseit és az ellenük intézhető támadásokat foglalja össze[2].

[2] *A biztonsági résekkel és támadásokkal részletesen foglalkozik Andrews – Whitaker (2007).*

3. táblázat
Biztonsági kategóriák és fenyegetettségük

Biztonsági kategória	Fenyegetettség/támadás
Naplózás és monitorozás	▪ A naplófájlok meghamisítása. ▪ Felületes monitorozás.
Autentikáció (felhasználóhitelesítés)	▪ A hálózat lehallgatása. ▪ „Nyers erő”-n alapuló támadás. ▪ „Szótár” alkalmazásos támadás. ▪ „Süti visszajátszás”-os támadás. ▪ Azonosító- (pl. jelszó) lopások.
Jogosultságkezelés (*authorization*)	▪ Jogosultságnövelés. ▪ Érzékeny adatok kiadása. ▪ Adathamisítás.
Konfigurációkezelés	▪ Az adminisztrációs felület jogosulatlan elérése. ▪ A konfigurációs tárolóhoz történő jogosulatlan hozzáférés. ▪ A nyers szöveg elérése. ▪ A titkos szöveg manipulálása. ▪ Nem valós fiókazonosító használata.
Kivételkezelés	▪ A rendszer, ill. az alkalmazás részleteinek felfedése. ▪ Szolgáltatás megtagadás (*Denial of service*).
Megszemélyesítés, delegáció	▪ Jogosultság emelése.
Üzenettitkosítás	▪ Információkiadás.
Üzenetismétlés felderítése	▪ Horizontális és vertikális jogosultság kiterjesztése.
Üzenet aláírása	▪ Adathamisítás.
Érvényességellenőrzés	▪ Puffer túlcsordítása. ▪ Idegen parancs végrehajtása (*Cross site scripting*). ▪ SQL-befecskendezés. ▪ Kanonizációs támadás.
Érzékeny adatok kezelése	▪ A tárolókban lévő érzékeny adatok elérése. ▪ A hálózat lehallgatása. ▪ Információkiadás.
Munkamenet- (*session*) kezelés	▪ Munkamenet-eltérítés. ▪ Munkamenet-ismétlés. ▪ Köztes lehallgatás (*Man-in-the-middle*) támadás

Forrás: (Patterns & practices, 2008)

Hogyan védekezhetünk?

A védekezést az határozza meg, hogy milyen környezetben fut szolgáltatásunk, illetve milyen felhasználókra számíthatunk. De van néhány alapelv, amelyet minden SOA-rendszer esetében ajánlatos figyelembe venni. A legfontosabbak:

Alkalmazzunk többszörös védelmi zónát a támadók ellen. Ezt a katonai terminológia *mélységi védekezés*nek nevezi. Így az első vonalakat áttörő támadó egy következő szinten valószínűleg már megállítható lesz.

Próbáljuk meg minél korábban hitelesíteni a felhasználót.

Gondoljuk végig, mi lehet a támadás célja. Próbáljunk meg az egyes szolgáltatásokhoz minimális, a felhasználói fiókokhoz engedélyezett jogosultságokat rendelni.

Legyenek biztonsági alapértékeink! Az alapértelmezett fiók mindig a legalacsonyabb jogosultságokkal rendelkezzen. Figyeljünk rá, hogy jogosulatlan próbálkozások esetén a küldött üzenetek ne tartalmazzanak információt a rendszerhez tartozó belső adatokról.

Nem szabad megbízni az ügyfelektől származó adatokban! Lehetőleg minden inputot ellenőrizzünk az összes rendelkezésre álló információ alapján.

Osszuk a szolgáltatási rendszert bizalmi zónákra! A zónák határain ellenőrizzük újra a fiók hitelességét és jogosultságait!

Ha a szolgáltatást már nem használják, zárjuk le a kapcsolatot!

A szerver és a hálózat biztonságát is vizsgáljuk meg!

A *webszolgáltatásokra* kialakult egy biztonságtechnikai forgatókönyv (bővebben ld. Patterns & practices, 2008: 48–50), amely kérdések formájában járja végig a biztonsági problémákat. Szemléltetésül ki ragadtunk néhányat a 3. táblázat kategóriáiból, és az alábbi kérdéseket tettük fel. Tudnunk kell válaszolni rájuk!

A fiók hitelesítése

- Milyen személyes azonosítókat használhatnak ügyfeleink?
- Honnan (internet/intranet) hívhatják a szolgáltatást?
- Hol és hogyan tároljuk az ügyfelek védett adatait?

Jogosultságkezelés

- Szolgáltatásunk milyen jogosultságokkal vehető igénybe?
- Vannak-e különösen védett szolgáltatási elemek?
- Ki végezze a jogosultság-ellenőrzést? (Ez lehet pl. a tűzfal, a szerviz vagy a szolgáltatást magába foglaló üzleti szint.)
- Szükséges-e az ügyfélnek személyesen hozzáférni háttérerőforrásokhoz (pl. adatbázisok)?
- Szerveztünk-e jogosultság alapján ügyfél csoportokat?

Konfigurációkezelés

- Milyen biztonsági környezetben fut a szolgáltató? (Internet/intranet.)
- Vannak-e adatbázis-kapcsolatok és kell-e védeni a kapcsolódási információt?

Az üzenetek tartalmának ellenőrzése

- Hogyan ellenőrzi szolgáltatásunk a beérkező SOAP-üzeneteket?
- Hogyan ellenőrizzük az input paramétereket?
- Ellenőrizzük-e az ügyfeleknek küldött információkat?
- Ellenőrizzük-e a belső forrásokból (adatbázis, fájlrendszer) származó adatokat?
- Gondoltunk-e a hálózatban mozgatott adatok biztonságára?

A kérdések megválaszolásakor figyelembe kell vennünk a 3. táblázat adott kategóriájára vonatkozó fenyegetettségeket, a támadások okozta lehetséges sérüléseket, károkat; illetve ezeket egybevetnünk a velük szembeni ellenintézkedésekkel. Ehhez hasznos útmutatónak ajánlható az irodalomjegyzékben is hivatkozott *Patterns & practices* (2008) második és harmadik fejezete.

A WCF nyújtotta lehetőségek

Az előző pontban számba vettük a fejlesztendő rendszer biztonsági réseit, az ezek ellen irányuló lehetséges támadásokat és a védekezés lehetőségeit. Ahogyan ezt már jeleztük, szolgáltatásinkat Microsoft környezetben nyújtjuk, a WCF szolgáltatási technológiára alapozottan. Tekintsük most át – elsősorban adatbiztonsági szempontból – azt az eszköztárat, amely itt rendelkezésünkre áll.

A WCF osztályai a fejlesztői környezetből elérhetők, így a kívánt funkciók programozottan is megvalósíthatók. A rugalmasság, az XML-ből következő olvashatóság és könnyű áttekinthetőség azonban a konfigurációs megoldások mellett szól. A kapcsolatok (*bindings*) és a viselkedések (*behaviors*) azok az elemek, amelyekben az adatátvitel biztonságát és a biztonsági kategóriákat beállíthatjuk, leírhatjuk.

A kapcsolatok specifikálják az átviteli protokollt, az üzenetek formáját, kódolását, megadják a szolgáltató és az ügyfél közötti kommunikációs csatorna leírását, míg a viselkedés elemekben találjuk az ügyfél autentikációjára, illetve jogosultságaira vonatkozó információkat. A következő táblázatunk bemutatja a WCF úgynevezett standard kapcsolatait. (Érdemes megfigyelni, hogy a kapcsolati protokollok között számos nem http-alapú is szerepel.)

4. táblázat
Standard kapcsolatok a WFC-ben

Kapcsolat típus	Konfigurációs cimke	Leírás
BasicHttpBinding	basicHttpBinding	Elsősorban a korábbi webszolgáltatásokkal (pl. ASMX) való kompatibilitást szolgálja.
WSHttpBinding	wsHttpBinding	A legutóbbi WS*-standardokat valósítja meg.
WSDualHttpBinding	wsDualHttpBinding	Kétoldalú kommunikációt támogató WSHttpBinding.
WSFederationHttp-Binding	wsFederationHttp-Binding	A WS*-standardokat támogató, megegyezéses kapcsolat.
NetNamedPipeBinding	netNamedPipe-Binding	Az azonos gépen futó alkalmazások közötti kapcsolatot szolgálja.
NetTcpBinding	netTcpBinding	Gépek közötti kapcsolat TCP-protokoll felett.
NetPeerTcpBinding	netPeerTcpBinding	Az üzenetszórás TCP-alapú kapcsolata.
NetMsmqBinding	netMsmqBinding	Hálózatbiztos üzenetváltást garantál MSMQ-használatával.

Forrás: (Bustamante, Michele Learning WCF, O'Reilly, 2007)

Az adatátvitel biztonsága megvalósítható szállítási szinten, illetve üzenet szinten. Az előbbi eljárás SSL (újabban TLS) néven ismert és az ügyfél és a szerver között hoz létre titkosított csatornát, HTTPS-protokollon, míg az üzenet titkosításához a résztvevők azonosítóit, tanúsítványait használják, és a protokollok XML-alapúak. A többszörös üzenetváltást biztonságosan megvalósító megbízható munkamenetek (*reliable sessions*) csak üzenetszinten hozhatók létre.

A WCF nagyban épít a Windows-os környezet Active Directory-t használó autentikációs és szerepalapú jogosultságcsoportokkal dolgozó technikáira. Ez megkönnyíti a megszemélyesítés/delegáció[3] alkalmazását (ilyenkor a fiók a szolgáltatás alsóbb szintjein is saját azonosítóit használhatja), illetve hogy az egyszer már beléptetett felhasználót az alrendszerek is hitelesnek ismerjék el. Ha nincs Active Directory vagy a platform heterogén, az azonosíthatóság érdekében a tanúsítványokat az üzenethez kell csatolni.

A szolgáltatások igénybevételéhez ismernünk kell azok leírását (a felkutatásnak előzőleg meg kell történnie, hisz a kapcsolatokban a szerver címe már szerepel). A leírásból az ügyfél számára egy proxy (közvetítő) osztály készül, a szolgáltatást ezen osztály metódusain keresztül érjük el.

WCF alapú szolgáltatásmodell intraneten

Amennyiben a szolgáltatón kívül az ügyfelek is lokális Windows hálózatban vannak, a leghelyesebb netTcpBinding kapcsolatot használni Windows autentikáció mellett. Ilyenkor alapértelmezetten szállítási szintű biztonságot követelünk meg, vagyis a kommunikáció titkosított

[3] *Megszemélyesítést használunk, ha a részszolgáltatás is ugyanazon a gépen van; delegálunk, ha azt egy távoli gépen érjük el.*

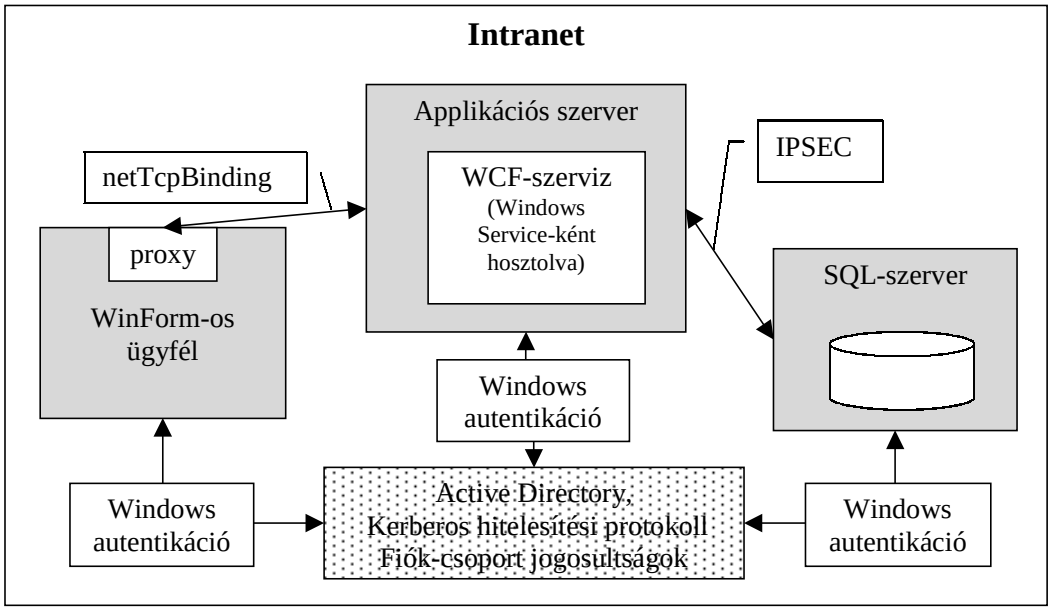
csatornákon folyik. A fiókok személyes azonosítóit az SSPI néven ismert szolgáltatás kezeli. A netTcpBinding támogatja a bináris üzenettartalmat is; ez fontos hatékonysági szempont lehet.

A webszolgáltatásokra jellemző, hogy az ügyfél az igényelt szolgáltatást saját rendszerében felhasználja, abba beépíti. Erre a webalkalmazások – ahol a kliens egy böngészőt használ – kevéssé alkalmasak. Ezért választottunk olyan sémát, amely az intranetekben gyakori: Winformos ügyfél („vastag kliens") igényli az applikációs szerveren futtatott szolgáltatást, amely adatbázis használattal is jár.

Az 1. ábra vázolja modellünket. Bár a legtöbb információ a vázlatból leolvasható, a legfontosabbakat kiemeljük:

- A felhasználók azonosítása és jogosultságainak kezelése az Active Directory-t használó Kerberos-rendszerre épül.
- A Windows Service rendszerszolgáltatásként futtatott WCF-szervizünk önálló fiókazonosítóval rendelkezik, mely az SQL-szerver felhasználói között a szükséges jogosultságokkal szerepel.
- A netTcpBinding alapértelmezés szerint biztosít titkosított adatcsatornát (ehhez a Kerberos-jegyekben tárolt kulcsokat használja fel).

1. ábra
Intranetben megvalósított WCF-szolgáltatás



A klinesek proxy osztályait a szolgáltatásleírásból (WSDL) a fejlesztői környezethez tartozó SvcUtil.exe segédprogrammal generálhatjuk le. A szolgáltatás Windows Service-ként történő futtatása biztosítja az intranten belüli elérhetőségét.

A modell részleteiről és egy oktatási célú megvalósításáról számol be az irodalomban hivatkozott Békési (2008).

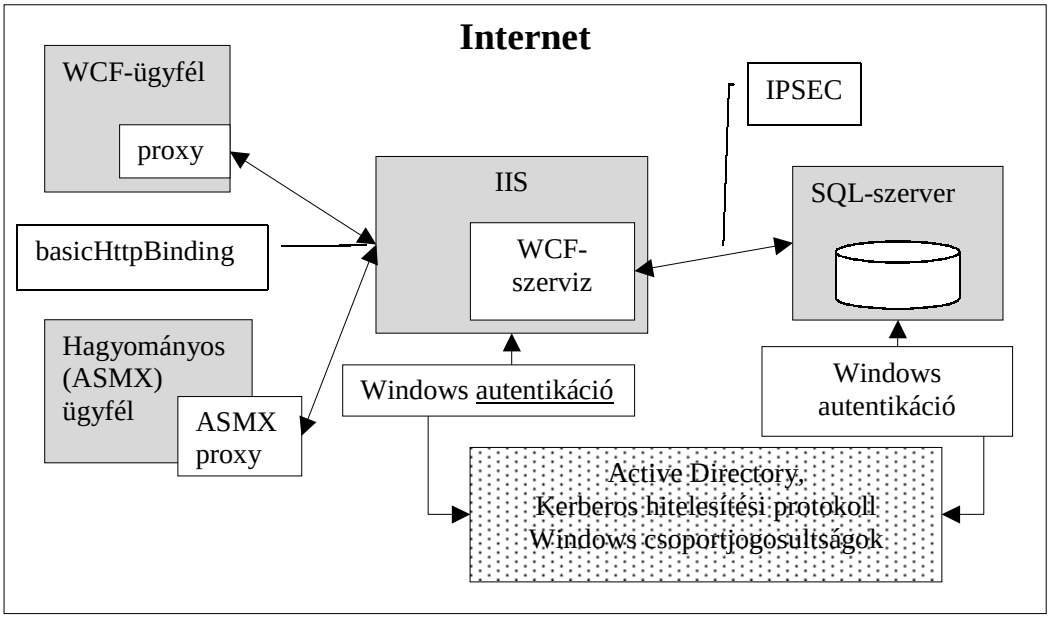
Internetet használó WCF-szolgáltatásmodell

A webszolgáltatók Microsoft környezetben sem feltétlenül az MS Internet Information Services (IIS) bizonyos verziói, modellünk a platform homogenitása okán ezt tételezi fel.

A modellt bevezetendő egy megjegyzést kell tennünk. A WCF megjelenéséig a Microsoft platformon webszolgáltatásokat fejlesztők egyik elfogadott technológiája a WSE[4] volt. Nagyon sok kliensalkalmazás készült erre alapozottan (a Visual Stúdió 2005-ös fejlesztő rendszer ma is támogatja ezt az irányzatot), ezért a WCF-es szervizek felhasználói köréből nem szabad kizárni ezeket a „maradi" ügyfeleket. Modellünk tehát mind WCF-el készült, mind a hagyományos, ASMX-típusú kliensek kiszolgálására alkalmas.

Az IIS valójában egy ASP.NET technológiára épülő dll (aspnet_isapi.dll), amely támogatja az alkalmazások elszigetelésének stratégiáját[5]. Ez a technológia a webtartalmat dinamikusan állítja elő, úgy hogy minden http-kérést az ASP.NET munkafolyamat vesz át és futtat végig az úgynevezett ASP.NET futószalagon. A munkafolyamat létrehozza a kérés környezetét leíró objektumot, amelyet aztán egy ugyancsak dinamikusan előállított alkalmazásobjektumba épít. Az alkalmazásobjektumok végigfutnak a futószalag állomásain, a végeredmény a válaszként letöltött weblap lesz. A 2. ábrán egy IIS-re telepített szolgáltatást láthatunk.

2. ábra
Interneten hosztolt WCF-szolgáltatás



[4] *Web Services Enhancements – a Microsoft által is támogatott WS*-specifikációk megvalósítása IIS-en, SOAP-protokoll alatt.*

[5] *Az alkalmazáselszigetelés eredményeként egy „lefagyott" alkalmazás nem állítja le a szerver munkáját, a többi alkalmazás tovább fut.*

A 2. ábra tanúsága szerint az ügyfelek – mivel közöttük ASMX típusúak is vannak – basicHttpBinding kapcsolatot használnak. Alkalmazhatunk szállítási szintű titkosítást (SSL), de az ügyfél személyes azonosítóit mellékeljük az üzenetfejben – érvényességüket a WCF-szolgáltatás fogja ellenőrizni egy ügyfélnyilvántartásból. WCF-szervizünk az ASP.NET munkafolyamat azonosítója alatt, annak jogosultságaival fut, és ezt az azonosítót vegyük fel az SQL-szerver felhasználói közé is. Az autentikációt és a szerep szerinti hozzáférés engedélyezést most már a Windows szerverre bízhatjuk.

A modell biztonsága javítható, ha az IIS-t egy tűzfal mögé helyezzük, intranetes webszerverként. Ehhez viszont az ügyfélellenőrzést a tűzfal gépen kell elvégezni.

Felhasznált irodalom

Andrews, Mike – Whitaker, James (2007): *Hogyan törjünk fel webhelyeket*. Budapest, Kiskapu Kft.

Békési Gábor (2008): *Komplex szállítási igények kezelése és a Goggle Maps-re alapozott távolsági szolgáltatás*. ÁVF Tudományos Közlemények 20, 57–74.

Patterns & practices (2007): *Improving Web Services Security*. Microsoft.