Békési Gábor*

BIZTONSÁGOS WEBSZOLGÁLTATÁSOK

Az informatikában az elmúlt évtized egyik leglátványosabb fejlődését az elosztott számítástechnika produkálta. Ebben kétségtelen szerepet játszott

- a világháló elterjedése, használatának általánossá válása,
- az XML[1] és a rá épülő protokollok (pl. a SOAP[2]) megjelenése és gyors elterjedése,
- a vezető szoftvergyártók közös törekvése a plattform-független elosztott rendszerek megvalósítására.

A technológia első sikereit a webalkamazások jelentették. A böngészőt használó, „vékony" kliensek és a tartalmat felkínáló, elérhetővé tévő webszerverek valóban operációs rendszertől és a programozási nyelvtől független kommunikációt biztosítottak. Az ügyfelek által igénybevehető szolgáltatásoknak a böngészők lehetőségei szabnak határt: a letöltött információ elsődlegesen megjeleníthető (ideértve a nyomtatást is), saját rendszerekbe nem, vagy csak körülményesen adható tovább. (Ebből a szempontból a levelező rendszerek sem adtak többet, feladatuk a fájlok cseréjében kimerült.) Az elektronikus kiskereskedelem (B2C) számára a webalkalmazások elfogadható megoldást jelentettek, de alkalmatlanok vállalatok közötti kapcsolatok fenntartására. Ez a terület ugyanis ügyfélalkalmazás – szerver szintű együttműködést kíván meg. Itt azonban a plattform-függetlenséggel van gond. A cégek belső informatikai rendszerei nagyon különbözőek, a használt adatbáziskezelők is eltérnek. Viszont az egyre élesedő versenyben a vállalatok internetes kapcsolattartásáról végzetes hiba lett volna lemondani.

Egyre sürgetőbb igényként jelent meg egy új, osztott számítási modell megalkotása. Ezt ismerték fel az ezredforduló éveiben a legnagyobb szoftvergyártók (IBM, Microsoft, Sun, BEA stb.), akik lázas kutató- és szervezőmunkába kezdtek egy olyan koncepció megvalósításáért, melyben

- a kommunikációs hálózat az internet,
- az összekapcsolt programok mérete nem jelent korlátozást,
- a kapcsolattartás nem függ a kapcsolódó gépek típusától, operációs rendszerétől, a rajtuk futó programok programnyelvétől és
- nyitott a szabványok változásával szemben.

A *webszolgáltatások* ezeknek a kritériumoknak tesznek eleget.

---

* *Főiskolai tanár, Általános Vállalkozási Főiskola*

[1] *eXtensible Markup Language – kiterjesztett jelölő nyelv (1998). A karakteres adattartalom jelentését, jellemzőit a tartalmat kísérő jelölő elemek és attribútumok írják le.*

[2] *Simple Object Access Protocol – egyszerű objektum-hozzáférési protokoll (2000). A „boríték" az üzenetirányítási információkat tartalmazza, a „törzs" a megcélzott objektum(ok) metódusainak meghívása ill. a hívás eredménye, mindez XML-ben megadva.*

A webszolgáltatásokat két további fontos tulajdonság is jellemzi:

- a webszolgáltatás *leírható*,
- a webszolgáltatás *felkutatható*.

Az első tulajdonság ma a szolgáltatáshoz tartozó WSDL[3] dokumentum révén valósul meg, míg a második jellemző teljesítése az UDDI[4] nyilvántartókra hárul. A WSDL az alapja a szolgáltatást igénybevevő és a szolgáltató közötti szerződésnek is. A webszolgáltatások zömében a SOAP protokollt használják. A szerződések anyagi konzekvenciákkal is járnak, így a szerződő felek közös érdeke, hogy a szolgáltatás „fogyasztásából" az (illetéktelen) harmadik felet kizárják és a szolgáltatást sikeressé tegyék. Ezek a törekvések valósulnak meg a *biztonságos webszolgáltatások*ban. A SOAP nem foglalkozik a biztonság kérdésével, így ezt a problémát alkalmazás-szinten kell megoldani. (A webalkalmazásoknak van biztonsági protokollja, ilyen a HTTPS, mely az SSL[5] megvalósítása HTTP felett.)

A biztonság megteremtésével kapcsolatos első kérdés: megbíznak-e a felek egymásban? A webszolgáltatások területén ehhez az ügyletben résztvevőknek igazolni kell kilétüket. Az igazoláshoz használt dokumentumok a tulajdonos hitelességének elismerését általában térben, időben és a megbízhatóság „erejében" is korlátozzák. (Az igazoló dokumentumot hitelesnek elismerő területet megbízhatósági tartománynak nevezik. Ez hálózatok együttese is lehet.)

Akkor mondunk egy webszolgáltatást biztonságosnak, ha az alábbi ismérvek közül legalább egy teljesül:

- az információcsere bizalmas, azaz titkos,
- a kapcsolatban álló felek hitelesek,
- maga a hálózat biztonságos.

Általában a szolgáltatás határozza meg a biztonsági igényeket. Például tanúsítvány iránti kérelem benyújtásakor elég a hitelesség megkövetelése, de egy e-banki tranzakciónak ezenkívül titkosnak is kell lennie. A hálózati biztonságot biztonsági protokollokkal (pl. HTTPS), vagy alkalmas hálózati szoftverrel (pl. Kerberos) valósítják meg. Mindkét esetben a hitelesítésnek meg kell előzni az információs csatorna titkosítását.

A biztonságos elektronikus kommunikáció első két kritériuma a titkosság (rejtjelezés) és a hitelesség. Ezekre a feladatokra a kriptográfia ad megoldást és az algoritmusokat szoftverekbe építve forgalmazzák az informatikai piacon.

A hitelességnek két aspektusa van: hitelesnek ismerjük-e el egy dokumentum tulajdonosát illetve magát a dokumentumot? Ha a tulajdonos hitelességéről megbizonyosodtunk, a dokumentum eredetét (és sértetlenségét) tulajdonosának aláírása alapján ellenőrizhetjük. A webszolgáltatások milyen igazolásokat fogadnak el a felek hitelességének vizsgálatakor? A bizalmi elv alapján szinte bármi elfogadható, a gyakorlatban azonban három igazolástípus terjedt el: a felhasználónév/jelszó, a Kerberos-jegy illetve az x509 szabványos tanúsítvány.

---

[3] *Web Services Description Language – webszolgáltatás-leíró nyelv. A leírás XML-ben történik, a szolgáltatás programozási jellemzőivel (adattípusok, eljárások, paraméterek) és elérhetőségével.*

[4] *Universal Description, Discovery and Integration – általános leírás, felkutatás és beillesztés. Speciális struktúrákban tárolja a szolgáltatás jellemzőit (köztük a WSDL-t), valamint szolgáltató adatait, gyors keresési eljárásokat biztosítva hozzájuk.*

[5] *Secure Socket Layer – biztonságos kapcsolódás. A böngésző a szerverrel folytatott „előzetes" párbeszéd során egy titkosított csatornát hoz létre a kommunikáció számára.*

A *felhasználónév/jelszó* érvényességi területe addig terjed, amig ezt a párost azonosítani tudják. (A password hálózatban nyíltan sosem továbbítható!) Windows környezetekben egy felhasználó ismert egy tartományban, ez Active Directory használata esetén akár intranet fürtökre is igaz. Internetet közbeiktatva az ismertség csak úgy biztosítható, ha a felhasználót partnerünk korábban felvette egy adatbázisba. (Tanúsítvány-szolgáltatók esetén ez gyakori, de ennek előfeltételei vannak.) Az időbeni érvényességet a Windows rendszerszinten ellenőrzi. A megbízhatóság erejét tekintve ez a megoldás Windows alatt megfelelő, az interneten keresztüli használata csak egyedi esetekben javasolt. Alkalmazása csak ügyfél-oldalon értelmes.

A *Kerberos-jegy* (csak Windows-környezetben használva) a username/password-el azonos terülei érvénnyel rendelkezik (leszámítva, hogy külső adatbázisokban nem tárolható), a jegyek lejárati idejét a rendszer ellenőrzi. Élő webszolgáltatásokban, interneten keresztül alkalmazva a jegy csak egy jelsorozatot jelent, tehát a username-hez hasonlóan gyenge aláíró/hitelesítő eszköz.

Az *X509v3 szabványú tanúsítvány* érvényes mindenhol, ahol kibocsátóját[6] megbízhatónak elismerik. Lejárati ideje a dokumentumban benne van. Az üzleti világban ma a legerősebb hitelességi igazolás.

A webszolgáltatások miért nem biztonsági protokollt (pl. HTTPS-t) használnak? Válaszként két fő okot említhetünk:

- A webszolgáltatások általában nem webszervereken futnak, igy nincs nyilvános IP-címük. Az interneten keresztül közvetlenül nem érhetők el.

- Általános gyakorlat az üzenetirányítás, azaz az üzenet több csomóponton halad keresztül.

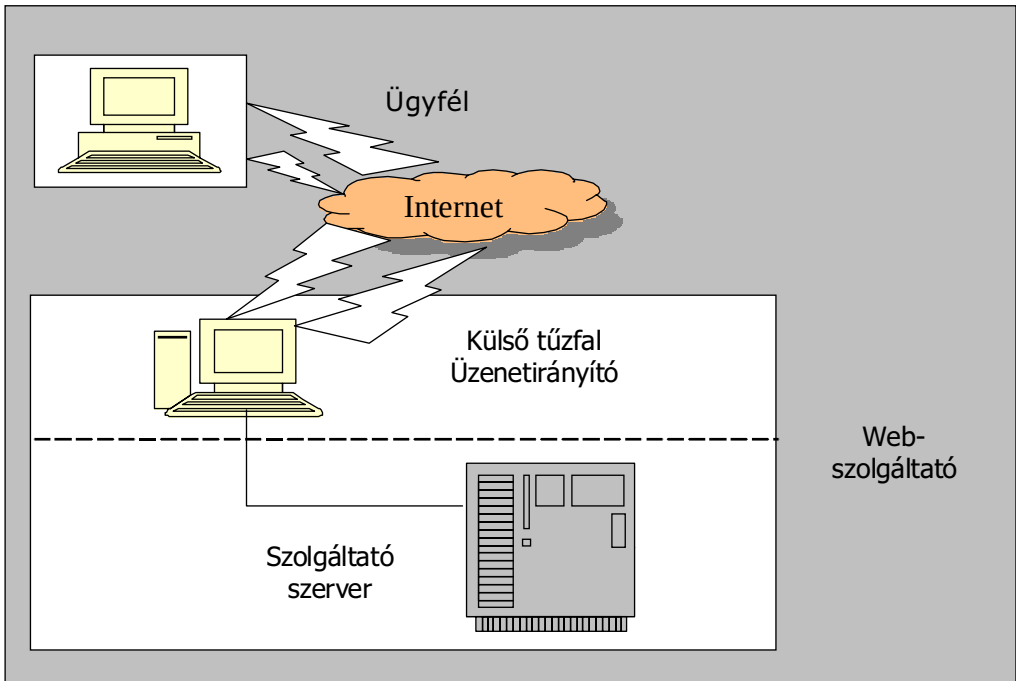
Csak a második megjegyzésünk igényel magyarázatot. A titkosított csatorna (pl. a HTTPS megvalósítása) mindig két végpont között jön létre és a végpontokban az üzenet nyílt. Ez komoly veszélyforrás. Több csomópont esetén a titkos csatornát az egymást követő párok között újra létre kell hozni, ami erőforráspazarló, azonkívül rendkívül lassú művelet.

Az üzenetirányítási probléma szemléltetésére nézzük az 1. ábrát. A lokális hálózatban lévő, nem-webszerver szolgáltatónk előtt egy szerverként telepített (az internet felé látható) külső tűzfal helyezkedik el. Ennek a feladata – a tűzfal-szerep mellett – az ismert (szerződött!) ügyfelek azonosítása és kéréseik továbbítása a szolgáltató felé ill. a szolgáltatás eljuttatása a kliensekhez. Az üzenetirányítókból több is lehet, egy-egy feladatra szakosodva. Ami lényeges: nem férhetnek hozzá az üzenet tartalmához. Tevékenységüket kizárólag a SOAP boríték számukra készült bejegyzése alapján végzik.

---

[6] *A kibocsátók ma már világméretű hierarchikus szervezetet alkotnak (PKI). A felsőbb szintek igazolják aláírásukkal az alárendelt kibocsátók tanúsítványait.*

1. ábra
ÜZENETIRÁNYÍTÁS TŰZFALLAL



(Forrás: Békési: 2006)

A szolgáltatásleírások mellett a webszolgáltatók biztonsági stratégiákat is előírnak: a SOAP üzenetek mely részei, milyen módszerrel legyenek titkosítva/aláírva. (Nagy állományok titkosítására használhatunk szimmetrikus eljárást is!) A stratégiák megvalósításához osztálykönyvtárak állnak rendelkezésre különböző plattformokon (a Microsoft legelterjedtebb gyűjteménye SSPI néven ismert), azonban az eljárások programozása rendkívül fáradságos és, úgymond, „programozót próbáló" munka.

Az IBM 2003 májusában egy szabványt bocsátott ki a biztonsági stratégiák leírására (WS-Policy), mely az évek során többször módosult[7] . Lényegében egy XML alapú parancsnyelvről van szó. Az üzenetek titkosítására/aláírásra már elfogadott biztonsági szabványok megadására alkalmas szkript- (futásidőben feldolgozandó) fájlokat készíthetünk, mind az ügyfél, mind a szolgáltató számára, a jövőben az üzenetirányítók számára is. Ez a módszer rendkívüli rugalmasságot biztosít a felek biztonsági követelményeinek megváltozása esetén, ugyanis nem kell programot módosítani.

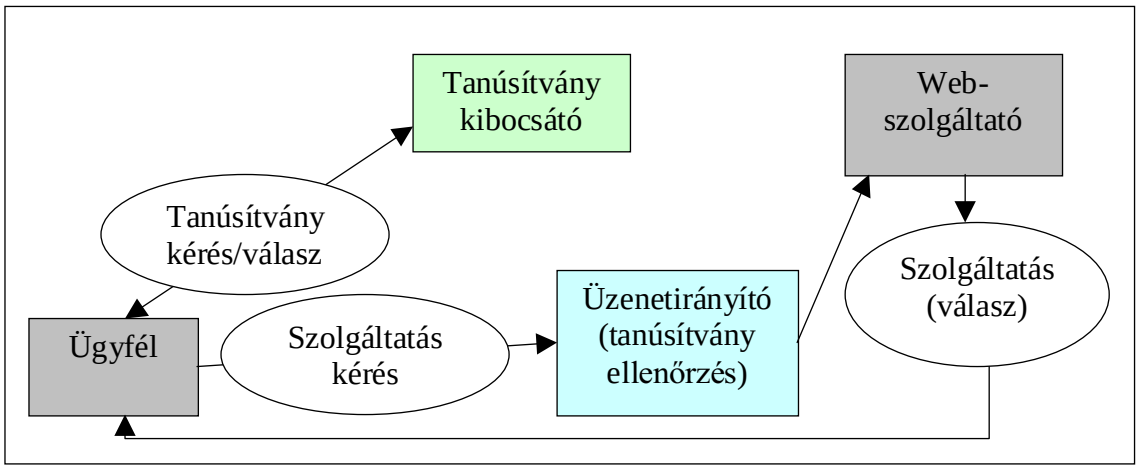
Megjelentek a szoftverpiacon olyan keretrendszerek, melyek a konfigurációs fájlokban tárolt stratégiai előírásokat alig néhány soros programozási munkával képesek érvényesíteni, automatikus üzenetfeldolgozást téve lehetővé. Ilyen a Microsoft WSE 2.0 (2005) szoftvere, mely a Visual Studio 2003 fejlesztőrendszerbe is beépíthető. Az irodalom (Békési, 2006) referált tanulmánya ehhez a rendszerhez kapcsolódóan a stratégia-szkriptek készítéséhez járul hozzá.

Az alábbi egyszerű – oktatási célból készült – modellen (2. ábra) mutatjuk be a biztonsági előírások használatát WSE 2.0 alkalmazása mellett.

---

[7] *Lásd az irodalomjegyzékben a „Web Services Policy 1.5 – Framework" hivatkozást.*

2. ábra
WEBSZOLGÁLTATÁS TANÚSÍTVÁNY KÉRÉSSEL ÉS TANÚSÍTVÁNY ELLENŐRZÉSSEL



A webszolgáltató ügyfeleit x509-es tanúsítványokkal fogadja el és ezt felhasználva szolgálja ki. A mi kliensünk ilyennel még nem rendelkezik, be kell szerezni egyet. Ezt az ügyletet a tanúsítvány kibocsátóval UserId/password segítségével hitelesíti és tanúsítványát majd ezzel titkosítják. Az üzenetirányító esetünkben csak a szerver áthelyezhetőségét biztosítja. Az ügyfél tanúsítványának, aláírásának ellenőrzése és a titkosítás kezelése a szolgáltatónál történik. (Ha a hitelességvizsgálatot jelenleg az üzenetirányítónál végeznék, tetemes programozási munkára számíthatnánk.)

Az x509 tanúsítványok előállításához a nyilvános kódú OpenSSL programot használtuk, mivel csak mintaprogramról volt szó. A tanúsítványok telepítése manuális tevékenység, így modellünk két önálló részre bomlott: az ügyfél tanúsítvány beszerzése és (a telepítését követően) a webszolgáltatás igénybevétele. Mindkét modellrészben stratégialeírással adtuk meg a biztonsági követelményeket és csak az ügyfél Windows-os felülete valamint a szolgáltatás kezelése igényelt programozást.

Mire számíthatunk a jövőben?

A SOAP üzenetek kriptográfiai szabványainak véglegesedése, a stratégialeírások használatát támogató rendszerek elfogadottsága és elterjedése meghatározta a jövő biztonsági koncepcióját. Néhány általánosan támogatott technika jellemzi a webszolgáltatások igénybevételét. Ez tükröződik (a Microsoft szemszögéből) a WSE új verziójában, ahogy arról Skonnard 2006 júniusi cikkében beszámol. (Skonnard, 2006)

A WSE 3.0 hat szabványos biztonsági stratégiát támogat:

1. Username OverTransport
   A szolgáltató hitelesítve van, az ügyfél nincs, a szállítási réteg titkosít („titkos csatorna”, ilyen a HTTPS).

2. Username ForCertificate
   A szolgáltató x509-e tanúsítvánnyal hitelesíti magát és ezt használja a titkosításhoz, az ügyfél UserId/password-öt alkalmaz ugyanezekhez.

3. AnonymousForCertificate
   Az ügyfél anoním, a szolgáltató x509-es tanúsítványt használ. (Nincs titkos csatorna!)

4. MutualCertificate10
   Mindkét fél x509-es tanúsítvánnyal rendelkezik az aláíráshoz/titkosításhoz.
   (a WS-Security 1.0 specifikáció szerint használják).

5. MutualCertificate11
   Hasonló a 4-eshez, de a WS-Security 1.1-es specifikáción alapszik, azaz használja a szimmetrikus titkosítást is.

6. Kerberos hitelesítés
   Ha mind a szolgáltatás, mind a kliens azonos Windows tartományban van, használható ez a változat, de csak Active Directory alatt szervezhető. Az 5-öshöz hasonlóan szimmetrikus algoritmussal dolgozik, biztonsági erejük is azonos.

A SOAP azáltal, hogy a biztonsági követelmények is az üzenetek részévé váltak, a hálózati kommunikáció protokoll-független formáját teremtette meg. Ennek előnye már a WSE 2.0-ban is megmutatkozott, bár a gyakorlatban csak a HTTP protokollon használták. A Microsoft webszolgáltatásokat támogató új plattformja a WCF[8] a HTTP-n kívül számos protokollt foglal magába, többek között a Microsoft MQ-t és a vezeték nélküli kommunikációt is. Vagyis a jövőben az ügyfél/szolgáltató kapcsolatában a hálózat megvalósítási jellemzői már nem játszanak szerepet.

A webszolgáltatás leírása, a mai WSDL, nem alkalmas a biztonsági stratégia megadására. Ez ugyanis a szolgáltató közleménye. Ha specifikálnánk is a webszolgáltató biztonsági követelményeit, az ügyfél hasonló elvárásairól mit sem tudhatunk. A Microsoft a WCF-ben itt is változást igér: a WSDL-ben megadott kapcsolati információk alapján egy biztonságstratégiai szkriptet is generálunk mindkét fél számára. A kliensnek ezen még lehetősége van változtatni, tárgyalásos alapon. Az előírt biztonsági ellenőrzések (az un. SoapFilter-ek) egy része szolgáltatás-függő, tehát automatikusan generálható, a lokális előírások (a tárgyalás eredményeként) paraméterek formájában kerülnek a konfigurációs állományokba.

A fentieket figyelembevéve – legalábbis a Microsoftnál érzékelhető tendenciák alapján – a jövőben a webszolgáltatások

- biztonságos webszolgáltatások lesznek,
- a szolgáltatásleírás (a mai WSDL) lesz a biztonsági stratégia meghatározója,
- a kommunikációs stratégiákat konfigurációs állományok tartalmazzák,
- a szolgáltatások felkutatása változatlanul az interneten történik, az igénybevételnél viszont bármilyen hálózat szóba jöhet.

Persze mindehhez a Microsofton kívüli „világnak” is lesz még szava.

IRODALOM

Békési Gábor (2006): *Webszolgáltatások biztonsági modelljei* (Kutatást záró dokumentáció), ÁVF, 2006. szeptember.

Web Services Policy 1.5 – Framework (Working Draft), July 2006.
http://www.w3.org/TR/2006/WD-ws-policy-20060731/

Skonnard, Aaron (2006): *WSE 3.0, SOAP Tranports, and More*, June 2006.
http://msdn.microsoft.com/msdnmag/issues/06/06/ServiceStation/default.aspx

[8] *Windows Communication Foundation, már a Vista része.*