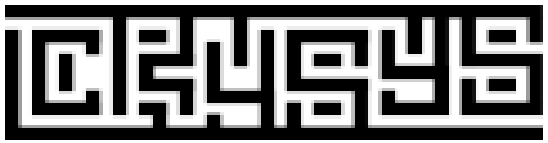

A Web 2.0 platform biztonsági kérdései

Gyöngyösi Péter

Laboratory of Cryptography and System Security (CrySyS)

Budapest University of Technology and Economics

önálló labor-konzulens: Schulz Róbert

Mi a Web 2.0?

- a kifejezés eredete: O'Reilly Media 2004-es konferciasorozatának címe
- népszerű kifejezés, buzzword – Dotcom Bubble 2.0?
- sokféle megközelítés, definíció

- közös nevező mindben:
  - **Paradigmális váltás**
  - **Technológiai váltás**

A Web 2.0, mint paradigma

- új felfogás, **eltérés a** korábbi kliens-szerver illetve **termelő-fogyasztó szemlélettől**
  - az adat, az információ mint fő tartalmi elem
  - az emberek elkezdenek maguk generálni és terjeszteni webes tartalmakat központi ellenőrzés nélkül és szabad újrafelhasználás mellett (tipikus példa: a blogok)
  - a kereskedelmi piac forgalma, a fogyasztás maga megjelenik kommunikációs tényezőként
  - stb.
- A megközelítésből fakadó, **jellegükben új biztonsági kérdések**:
  - a korábbi kis számú, ellenőrizhető szerkesztőbrigád helyett **nagy számban vonjuk be az ismeretlen felhasználókat**
  - **egyetlen rosszindulatú felhasználó ne tehesse tönkre a teljes rendszert** $\Leftrightarrow$ ez a védelem nem állíthat túlzott akadályokat
  - hatalmas számú és nagy aktivitású felhasználó esetén működnek az új szolgáltatások jól $\Rightarrow$ **centralizált kontroll lehetetlen**

A Web 2.0, mint technológiai forradalom

- igazából csak apró újítások, **már létező technológiák váltak kiforrottá**, terjedt el a használatuk -- “JavaScript finally works!”
- főbb **“új” technológiák**, megközelítések:
  - kiforrott Ritch Internet Applicaton-ök: desktop programokkal összemérhető webes alkalmazások (**Flash, Java, ActiveX, JavaScript, AJAX**, XUL stb.)
  - CSS, szabványos, “valid” XHTML-kódolás
  - RSS/Atom feedek, szabványos (pl. XML-alapú) webszolgáltatások használata
- **Új problémák**:
  - **új technikák: mindig új kockázatok**, eddig nem ismert gondok
  - megnőtt felhasználói igény az intelligensebb alkalmazásokra, egyre gyakrabban, **fontos helyeken alkalmazzák az új módszereket**
  - a böngésző “többet tud” --> **több helyen támadható az alkalmazás**
  - egyéb gondok: később...

Biztonsági kérdések – paradigmális váltás

- a tartalom felhasználói megváltoztatása ezen oldalak alapvető funkciója, de **nem megfelelő korlátok esetén ellehetetlenülhet a szolgáltatás**
- főbb problémák:
  - vandalizmus
  - spam
  - tartalom hitelessége, minősége
    - hozzászólások
    - szavazatok
    - vásárlások

Vandalizmus: Wikipedia

- **minden különösebb szándék és cél nélküli destrukció** a felhasználó által amúgy legitimen birtokolt eszközökkel (törlés, módosítás, félrevezető tartalmak stb.)
- védekezni ellene
  - nehéz: **nincs konkrét célja**, ami alapján meg lehetne fogni
  - könnyű: **komolytalan**, egyszeri próbálkozás, nem szisztematikus támadás
- általában **problémás automatizáltan detektálni**
  - egy teljes törlés észrevehető, de...
  - pl. egy oda nem illő erotikus kép egy illusztráció helyén csak emberi ellenőrzéssel szűrhető ki
- emiatt: **szisztematikus emberi szűrés szükséges**
- hatalmas információmennyiség: **muszáj a felhasználókat bevonni!**

Vandalizmus: Wikipedia (folyt.)

- Wikipedia megoldása: **Counter-Vandalism Unit**
  - önkéntes **felhasználók**
  - **szisztematikusan végignézik az összes friss változtatást**, előre lefektetett szigorú irányelvek szerint ellenőriznek
  - fontos: **visszaállítható** és követhető legyen **minden változtatás**!
- a megoldás működik: egy 2002-es IBM-tanulmány szerint már akkor **átlagosan 5 perc** alatt **kijavította** a közösség a vandalizmusokat
- további módszerek:
  - IP-cím tiltás visszatérő esetben
  - különösen “népszerű” szócikkek átmeneti letiltása

Spam: blogok

- az **email-spamek** problémája rég ismert, **tudatosan védekeznek** ellene elég hatékonyan, emiatt egyre kevésbé hatékony
- népszerűbb **blogok**, egyéb webkettes oldalak: hozzászólási lehetőség mellett hatalmas olvasótábor – **ideális terep a spammelésre!**
- elterjedt blog- és portálmotorok (Wordpress, Movable Type, Drupal stb.), nagy szolgáltatók: **automatizálható spammelés**
- két fő fajtája:
  - comment spam: hozzászólásokban
  - referal spam: visszajelzést szolgáló trackback mezővel való visszaélés
- **mostanra komoly probléma**: saját, 3-as pagerank-ű, napi 3-500 látogatós oldalon 50-100 spam lenne védekezés nélkül

Spam: blogok – védekezés

- sztandard blogmotor **apró változtatása**
- **captcha**-k: gyors Turing-teszt
- **előzetes regisztráció**, email-ellenőrzés (macerás, emiatt “központi” regisztrációs oldalak, pl. TypeKey, TinyOrwell)
- **klasszikus spamszűrő technikák**: központi blacklistek, heurisztikus módszerek (pl. Akismet)
- **rel='nofollow'**
  - Google kezdeményezése, sokan csatlakoztak hozzá
  - nem ad pagerank-et az ilyen linkekre, de követi őket
  - pagerank-növelést célzó referal spamet értelmetlenné teszi
  - nem váltotta be a reményeket, komoly kritikák azóta

Tartalom hitelessége, minősége

- **a felhasználók által generált tartalom esetén kiemelten fontos annak ellenőrzése**
- általában érvényes az **1%-os szabály**: a látogatók 1%-a fog várhatóan tartalmat is kreálni, 10%-a szavaz vagy kommentel
- nagy látogatottságú oldalaknál ez jelentős szám: erre lehet **közösségi értékelő és visszajelző rendsze**rt építeni
- három fő kérdés:
  - hozzászólások
  - szavazatok
  - vásárlások

Hozzászólások: Slashdot

Slashdot
News for Nerds. Stuff that matters.

- népszerű **technológiai portál**, erős visszajelző-kommentező rendszerrel
- már 1997 környékén web 2.0-s technikák!
- ezres nagyságrendben érkező hozzászólások --> muszáj valamilyen szűrést alkalmazni
- regisztrált **felhasználók számára** bizonyos szisztéma szerint időnként **moderátori pontok**, amikkel **osztályozhatják a kommenteket**
- felhasználónként beállítható **treshhold**, ami alatti pontszámú hozzászólásokat nem jeleníti meg, de **nincs törlés**
- meta-moderáció: moderációk moderálása
- karma-pont

Szavazatok: Digg

- felhasználói visszajelzés letisztult megjelenítője: a felhasználók linkeket küldenek rövid kommentárokkal, a többiek ezekre szavaznak, a legnépszerűbbek kerülnek ki a nyitóoldalra
- hatalmas forgalom: **nagy gazdasági haszon a visszaélésből!**
- a szavazatok értékelése:
  - változó számú szavazat kell az előrelépéshez
  - eltérő értékű szavazatok: új felhasználóké kevesebbet ér, karma-pont, szűrik a rendszeres csoportos szavazást
  - szavazatok időpontját, eloszlását vizsgálják
  - “kamu” céloldalakat szűrik
- **egyszerű koncepcióból bonyolult rendszer** lett a fair-ség fenntartása érdekében
- ennek ellenére **a 100 legaktívabb felhasználó kontrollálja a** főoldalon megjelent **tartalom 50%-át**

Vásárlások: eBay, Amazon

- nagy **online kereskedelmi oldalak**, erős felhasználói visszajelzéssel (eBay: c2c aukció, Amazon: bolt, de nagyon széles beszállítói körrel)
- közös pont: **a vásárló kevés információval rendelkezik** a megvásárolni tervezett termékről vagy az eladóról
- megoldás: **korábbi vásárlók visszajelzései**
- eBay: minden lezajlott aukció után **az eladó és a vevő értékelheti egymást** (gond: “értékelés-bosszú”)
- Amazon: **bárki írhat értékelést** egy termékről (gond: hamis, torzított, PR-értékelések)
- mindkét esetben **visszakereshető a felhasználók “múltja”**
- **túl sok munkát** igényel a felhasználótól egy korrekt ellenőrzés: nem teljesen önszabályozó a rendszer
- visszaélések kezelése a szolgáltató által, egyesével
- s**ok probléma, rengeteg visszaélés**

Biztonsági kérdések – technológia

- az **új technológiák mindig biztonsági kockázatot hordoznak**, mivel nincs velük sok tapasztalatunk
- több szolgáltatás kell nyújtani, emiatt **komplexebb rendszerek** alakulnak ki
- **egyre fontosabb szolgáltatások** költöznek a webre egyre csábítóbb célpontot jelentve a támadóknak (akik már rég nem script kiddie-k!)
- főbb gondok:
  - XSS: cross-site scripting
  - XML parserek támadása
  - AJAX által felvetett problémák
  - RSS támadása
  - WSDL scannelés
  - SOAP sebezhetőségek
  - Ritch Internet Application-ök

XSS: cross-site scripting

- már a JavaScript kifejlesztésekor is felmerültek a biztonsági kérdések: **egyik oldal kódja ne tudjon hozzáférni a másikhoz**
- alapvető védelem: **same-origin policy**
- **XSS-támadások: ezen védelem kikerülése**, kód elhelyezése egy másik oldalon, bizalmas adatok “kilopása”
- fajtái:
  - **1. típus (DOM-based v. local):** az oldal a hívási paramétereket (GET, POST, URL) építi be saját tartalmába, és nem szűri belőle a scripteket (különösen veszélyes: IE lokális zónában)
  - **2. típus (reflected):** felhasználó által adott tartalmat (pl. form-ok értékét) azonnal felhasználja az oldal scriptszűrés nélkül (pl. keresők) – támadási lehetőség pl. linkküldéssel
  - **3. típus (persistent):** a felhasználói által adott tartalmat a szerver elmenti és kiírja más felhasználók számára (pl. fórumokban) – elég egyszer elhelyezni (MySpace-vírus)
- egyetlen lehetséges védekezés: minden script kiszűrése

XML parserek támadása

- **egyre több helyen** alkalmaznak XML-t adatok átadására, emiatt egyre csábítóbb célpont
- egyre többet tudó, **intelligens XML-parserek**
- szemléletes példa: Sun Unified Development Server XML-parsere, **rekurzió** a DTD-ben:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE foobar [
 <!ENTITY x0 "hello">
 <!ENTITY x1 "&x0;&x0;">
 <!ENTITY x2 "&x1;&x1;">
 ...
 <!ENTITY x99 "&x98;&x98;">
 <!ENTITY x100 "&x99;&x99;">
]>
<foobar>&x100;</foobar>
```

AJAX-technológia által felvetett problémák

- **Asynchronous JavaScript and XML**, egyre gyakrabban használt technológia
- a felhasználó által irányított böngészõben egy **JavaScript rutin indul el aszinkron módon**, információt cserél a szerverrel, majd az eredményt feldolgozva beilleszti az oldalba
- gondok:
  - csak kliensoldali ellenőrzés
  - megnövekedett támadási felület
  - szolgáltatások közeledése a felhasználóhoz
  - új XSS sebezhetőségek
  - **biztonsági tesztelés nehezül** (hagyományos tesztelő-módszerek nehezen alkalmazhatók)

Egyéb gondok: RSS, WSDL scan, SOAP, RIA

- **RSS/Atom támadás:** egyre gyakrabban vesznek át RSS feedeket egymástól oldalak, egy sebezhető oldal támadása átterjedhet könnyen máshova is
- **WSDL (Web Services Definition Language) scannelés:** szolgáltatások gyors felderítésére kitalált rendszer segítségével sok pluszinformáció szerezhető, ami könnyíti a sebezhetőség megtalálását
- **SOAP (Simple Object Access Protocol) üzenetek támadása:** routolás támadása, csomagok módosítása, Xpath beszúrás stb.
- **Ritch Internet Application-ök:** Flash, ActiveX, appletek kliensoldalra letöltődnek, emiatt bár a scripteknél nehezebben, de megváltoztathatóak, patchelhetőek (AJAX-hoz hasonló problémák ha nincs szerveroldali ellenőrzés)

A jövő

- paradigmális váltás terén: **Identity 2.0**
  - jelenleg: **külön “személyiségek” oldalanként**, szolgáltatásonként
  - **ezeket összekötve komolyabb felelősség** --> megbízhatóbb felhasználók
  - fontos: úgy működjön, mint egy **szokványos személyi** – több kibocsájtó, offline ellenőrzés stb. (**skálázhatóság, privacy**)
  - technikai és politikai nehézségek, **senki nem mer belevágni**
  - alakuló kezdeményezés: **OpenID** (AOL, Firefox, Microsoft, Wikipedia)
- technikai váltás terén:
  - **adaptálódó** scannelő, hibakereső **eszközök**
  - adaptálódó fejlesztési metodikák, szabályrendszerek
  - általánosságban: **egyre több tapasztalat**

Kérdések?

?

Köszönöm a figyelmet.

gyp@impulzus.com

http://gyp.impulzus.com/w3c_web20.pdf