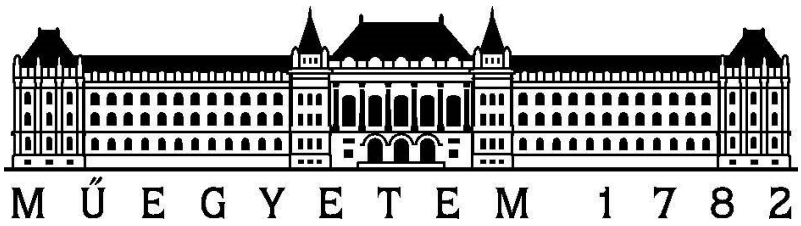


Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Híradástechnikai Tanszék

Webes anonimitást és privátszféra-barát identitásmenedzsmentet nyújtó alkalmazás tervezése

Készítette: Dargó Sándor
Konzulensek: Schulcz Róbert (HIT), Gulyás Gábor György (HIT)

Nyilatkozat

Alulírott Dargó Sándor, a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatója kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem el és abban csak az irodalomjegyzékben felsorolt forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva a forrásokból átvettem, egyértelműen, a források megadásával megjelöltem.

2009. december 11.

…….

Dargó Sándor

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani Gulyás Gábornak mindazért a sok óráért, amit rém fordított a konzultációkon és azokon túl, illetve a számtalan hasznos tanácsért és ötletért, de legfőképpen azért, hogy végig hitt a határidő tarthatóságában és ezt elhitette Velem is. Továbbá köszönet illeti Valahovits Szilviát aki lektorálta a diplomatervet, segített felfedezni az elgépelési és helyesírási hibákat, illetve azokat a stílusbeli hibákat, amiket igazán csak egy női szempár láthat meg. Köszönetet szeretnék még mondani Csapó Ádámnak, aki kegítő kezet nyújtott a Kivonat angol nyelvű elkészítésekor.

Kivonat

Az információs társadalom kialakulása, a kommunikáció idő- és térbeli határainak leomlása a számos előny mellett új veszélyeket is magában hordoz. A kéretlen levelek és ajánlatok, a számos világháló segítségével elkövetett csalás, mind a privátszféra-védelem fontosságának irányába mutat.

Diplomatervem első, elméleti részében bemutatom a világhálón adatainkat támadó szereplők céljait és eszközeit, az általuk használt különböző módszereket. Ezek után megvizsgálom a veszélyekre adható különböző válaszokat, bemutatom a rendelkezésre álló védekezési lehetőségeket, ezen belül foglalkozom a szerepalapú identitásmenedzsment témakörével.

A fennálló veszélyek, a lehetséges védekezési módszerek ismeretében felállítok egy ideális identitásmenedzsment szolgáltatásra vonatkozó elméleti kritérium rendszert.

A diplomaterv második, gyakorlati részében a megfogalmazott elméleti kritériumok figyelembe vételével megtervezek egy saját identitásmenedzsment alkalmazást, amely megoldást nyújt az adatainkra leselkedő veszélyekre. A tervezett alkalmazás teljesíti az anonimitás követelményeit, valamint lehetőséget nyújt az adataink szerepalapú identitásmenedzsment általi védelmére.

Ezek után bemutatom a tervek alapján általam implementált tesztrendszert, aminek a segítségével a tesztesetek során megmutatom, hogy a megfogalmazott elméleti modell alapján elkészített tervek teljesítik a támasztott kritériumokat, továbbá, hogy az implementált tesztrendszer a felhasználó számára garantálja az anonimitást, valamint a privátszféra barát identitásmenedzsment lehetőségét.

Abstract

With the evolvement of the information society and the gradual disappearance of time-wise and space-wise constraints in info-communications, new horizons are opened up for humanity. However, the immense technological developments we are witnessing also bring to surface a wide variety of new dangers, including spam and other web-based solicitations, web-based fraud, etc. For this reason, we are approaching an era in which the protection of privacy will become increasingly important.

In the first, theoretical part of this thesis, the motives and methods of those who attack user privacy on the Internet are briefly presented. This will be followed by a summary of countermeasures that can be taken to alleviate these dangers, with special attention to role-based identity management. With this introductory knowledge in hand, it is possible to specify a theoretical set of criteria for an ideal identity management service.

In the second, practical part of the thesis, an identity management application developed by the author is presented. The application satisfies the theoretical criteria laid out in the previous sections, and also allows for the protection of data through role-based identity management. It will be shown that the application gives considerable protection against the loss of privacy while maintaining the user's anonymity.

The final section introduces a test system developed to evaluate the performance of the identity management based protection application. Through a wide variety of text cases, it will be demonstrated that the application satisfies the theoretical model presented earlier, and that the system guarantees the user's anonymity, thus allowing for private sector friendly identity management.

Tartalomjegyzék

1 Bevezetés: veszélyek, privátszféra-védelem ... 1
1.1 Az adatainkra leselkedő veszélyek a Interneten ... 1
1.2 Lehetőségek a veszélyek kivédésére ... 2
2 A Web veszélyeinek bemutatása ... 4
2.1 A támadók bemutatása: célok és lehetőségek ... 4
2.1.1 Profilkészítés ... 4
2.1.2 Identitáslopás ... 5
2.1.3 Phishing és Whaling ... 5
2.1.4 Egyéb visszaélések ... 6
2.2 A támadók technikai lehetőségei ... 6
2.2.1 Alapvető eszközök a támadás során ... 7
2.2.2 A bemutatott eszközökre épülő módszerek ... 9
2.2.3 Egyéb módszerek ... 11
2.3 Védekezési lehetőségek ... 14
2.3.1 Anonim proxy ... 14
2.3.2 Anonim böngészők ... 15
2.3.3 Anonim böngészők privátszféra-barát identitásmenedzsmenttel ... 15
2.3.4 Az azonosíthatóság behatárolása: Nymity slider ... 19
3 Egy IDM rendszerrel szemben állított kritériumrendszer ... 21
3.1 Az anonimitás kérdése ... 21
3.2 Böngészővel szembeni követelmények ... 22
3.3 Identitásmenedzsment ... 24
4 Egy IDM alkalmazás megtervezése ... 26
4.1 Identitásmenedzsment funkciók kialakítása ... 26
4.1.1 A profilhierarchia lehetséges megvalósításai ... 26
4.1.2 Szerepsémák ... 29
4.1.3 Identitások közötti váltások ... 24
4.1.4 Kényelmi funkciók kialakítása ... 24
4.2 Biztonsági beállítások ... 26
4.2.1 A sütimenedzsment lehetőségei ... 26
4.2.2 Veszélyforrások blokkolása ... 28
4.2.3 Beállítások profilonkénti mentése ... 28
4.3 Az anonimitás megőrzése ... 29
5 Az implementált IDM rendszer bemutatása ... 30
5.1 Kezelőfelület ... 30
5.1.1 Tervezési szempontok, szóba jött változatok ... 30
5.1.2 A lehetőségek kombinációja ... 33
5.1.3 Értékelés ... 34
5.1.4 Továbbfejlesztési javaslatok ... 34
5.2 Egy kiválasztott modul: a Flash blokkoló beállításai ... 35
5.2.1 Követelmények a blokkolóval szemben ... 35
5.2.2 A blokkolóhoz tartozó kezelőfelület ... 36
5.2.3 Fájlkezelés ... 37

5.2.4 A blokkoló működése 37
5.2.5 Értékelés 38
5.2.6 Továbbfejlesztési javaslatok 38
5.3 Profilbeállítások 39
5.3.1 Tervezés 39
5.3.2 Profilhierarchia 40
5.3.3 Profiladatok 43
5.3.4 Profilhoz tartozó oldalak 47
5.4 Anonim mód 50
5.4.1 Az „anonim mód” felépítése 50
5.4.2 Az „anonim mód” működése 50
5.5 Továbbfejlesztési javaslatok 51
5.5.1 A privátszférát erősítő továbbfejlesztési javaslatok 51
5.5.2 A felhasználó-barát működést segítő javaslatok 56
6 Az implementált tesztrendszer működése 57
6.1 Profilváltások vizsgálata 57
6.2 Flash blokkoló tesztelése 59
6.3 Anonim mód tesztelése 64
6.4 A tesztrendszer értékelése 66
7 Összefoglalás 69

Rövidítésjegyzék

HTML	HyperText Markup Language	Hiperszöveges jelölőnyelv
IDM	Identity Management	Identitás Menedzsment
IP	Internet Protocol	Internet Protokoll
LSO	Local Shared Object	Helyben tárolt megosztott objektum
PIE	Persistent Identifier Element	Megmaradó azonosító elem
PRIME	Privacy and Identity Management for Europe	Európai Uniós projekt, amit a privátszféra-védelem fejlesztésére irányul
RBP	Role Based Privacy	Szerepalapú privátszféra-védelem
XML	Extensible Markup Language	Kiterjeszthető leíró nyelv
XSS	Cross Site Scripting	Kereszt oldal szkriptelés
XUL	XML User Interface Language	XML felhasználói felület nyelv

1 Bevezetés: veszélyek, privátszféra-védelem

A mindennapok során komoly figyelmet szentelünk személyes adataink védelmére, erre nevelnek mind a családunkban, mind a különböző közösségeinkben. Figyelünk arra, hogy ne adjuk oda illetékteleneknek születési adatainkat, ne adjuk ki címünket, banki adatainkat, amelyekkel mások visszaélhetnének. Azonban mára azok a hagyományos praktikák, hogy például nem tesszük a tárcánkba a bankkártya mellé egy annak a PIN kódját tartalmazó papír cetlit, mára már messze nem elegendőek.

Ahogy a modernkor vívmányai egyre nagyobb hatást gyakorolnak életünk minden egyes területére, úgy válnak egyre inkább elégtelenné a hagyományos adatvédelmi módszereink. Az információs társadalom fejlődésével az internet egyre jobban behálózza mindennapi életünket, egyre több helyen egyre több adatunkat vagyunk kénytelenek megosztani idegenekkel, ebből kifolyólag egyre komolyabb veszélyek fenyegetik az adatainkat, így elengedhetetlenné vált egyrészt ezen fenyegetések azonosítása, másrészt a fenyegetésekre adandó védekezési mechanizmusok kidolgozása.

1.1 Az adatainkra leselkedő veszélyek a Interneten

Mint említettem rengeteg veszély fenyeget minket, illetve egészen pontosan a személyes adatainkat az interneten barangolva. Amennyiben nem vigyázunk, számtalan manipulatív hirdetésnek, rengeteg kéretlen levélnek leszünk kitéve, rosszabb esetben pedig kézzel fogható anyagi károk is érhetnek minket.

Érdemes már azzal is vigyáznunk, hogy a különböző közösségi portálokon regisztrálva milyen minőségű és mennyiségű adatot adunk meg magunkról, mert rengeteg rosszindulatú felhasználó juthat hozzá azokhoz. Gondoljunk csak bele, hogy vajon megadnánk-e a telefonszámunkat egy nem éppen bizalomgerjesztő külsejű idegen személynek, aki szembe jön velünk az utcán. De ennél ártalmatlanabbnak tűnő esetek is a megadott adataink korlátozására sarkallhatnak. Vegyük azt az esetet, amikor egy számítástechnikai világcég egy alkalmazottját azért rúgták ki a munkahelyről, mert egy közösségi honlapon olyan képeket publikált, melyeken a tengerparton pihen, miközben

betegszabadságon volt [1]. Éppen ezért érdemes nagyon gondosan elvégezni az adatainkra vonatkozó beállításokat, megelőzendő a nagyobb kárt. Ahogy a közmondás tartja, jobb félni, mint megijedni.

Azonban balgaság lenne a közösségi portálokat egyedüli és legfőbb bűnbakként megtenni az adataink biztonságát, vagy inkább biztonságuknak hiányát illetően. Akárhányszor megadjuk a nevünket, e-mail címünket, telefonszámunkat, vagy a bankszámlaszámunkat, komoly veszélynek tesszük ki magunkat, ezért érdemes többször is átgondolni, hogy kinek milyen adatunkat adjuk meg.

Az előző bekezdésekben olyan veszélyekről írtam, amik elsősorban a felhasználó figyelmetlensége által jelennek meg, kevésbé technikai jellegűek. Ezzel szemben vannak olyan ellenünk leselkedő veszélyek, melyek sokkal jobban kihasználják a technika által adott lehetőségeket. Ilyenek lehetnek az ügynevezett „tracking cookiek” – melyekről a 2.2.1.2 fejezetben lesz bővebben szó –, vagy a webpoloskák (2.2.2.1), de egy a böngészőnk által megjelenített weboldal majd minden részletében lehetnek olyan elemek, melyek ellen jóval nehezebb a védekezés, mint a fenyegetések először ismertetett fajtái ellen.

1.2 Lehetőségek a veszélyek kivédésére

A különböző veszélyek egy igen jelentős részét megfelelő odafigyeléssel és beállításokkal kiszűrhetjük, bár a helyzet koránt sem olyan egyértelmű, mint első látásra tűnhet.

Mint ahogyan azt már korábban említettem, a mi felelősségünk, hogy különböző oldalakon, milyen adatokat adunk ki magunkról, illetve, hogy kik számára tesszük azokat hozzáférhetővé. Tehát mondhatjuk, hogy a józan gondolkodás, mérlegelés a legegyszerűbb védekezési mechanizmus, úgy, mint az életnek annyi más területén is.

Azonban az inkább technikai jellegű fenyegetések területén már nem ilyen egyértelmű a helyzet. Vegyük alapul a sütiket, melyekkel később bővebben is foglalkozom majd. Nem mondhatjuk, hogy a sütik használata önmagában káros lenne, hiszen nélkülük minden egyes alkalommal úgy térnénk vissza egy korábban már látogatott oldalra, mintha legelőször járnánk ott. Ez bizonyos esetekben hasznos is lehet akár, de beláthatjuk, hogy

a sütik – amik bevezetik az állapotot az alapvetően állapotmentes HTTP protokollba – használata nélkül az internet adta számos lehetőséget csak részben, vagy egyáltalán nem tudnánk kihasználni.[1] Ezzel csak arra próbáltam rávilágítani, hogy önmagában az, hogy a böngészőnkben a legszigorúbb beállításokat alkalmazzuk, nem jelent felhőtlen örömöt. De sajnos még teljes biztonságot sem.

Egyrészt az emberi felelőtlenséget nem, vagy csak részben tudjuk biztonsági beállításokkal kivédeni, másrészt pedig vannak olyan informatikai vívmányok, melyek ellen a ma használatban lévő alapböngészők nem védenek, mint ahogyan azt majd láthatjuk a webpoloskák esetében.

Minthogy ez nem egy teljesen új keletű probléma, ezért vannak azok kiküszöbölésére szánt megoldások. Ezek közül a legfontosabbak, amikkel én is részletesen foglalkozni fogok a következőkben: az identitásmenedzsment, illetve annak egy válfaja a szerepalapú privát-szféravédelem. Az általuk nyújtott elvi alapokhoz nyúlva teszek kísérletet egy olyan alkalmazás elkészítésére, amely a felhasználót teszi a saját személyes adatai áramlásának fő vezérlőjévé.

[1] A HTML5 már sütik használata nélkül is képes megoldani a HTTP protokoll állapotmentes volta okozta hiányosságokat

2 A Web veszélyeinek bemutatása

Már az előző fejezetben röviden említést tettem a ránk leselkedő veszélyekről, illetve megoldási lehetőségekről, azonban ahhoz, hogy megtervezzek és implementáljak egy a következő fejezetekben meghatározott specifikációknak megfelelő biztonságú rendszert, mindenképpen szükséges részletesebben megvizsgálni ezt a területet.

Mivel a felhasználó felelőtlenségét, az emberi tökéletlenséget lehetetlen kizárni, ezért itt nem kívánok foglalkozni azzal, hogy mennyire is fontos önmagában az, hogy egy idegen személynek ne adjuk meg személyes adatainkat, ezzel szemben inkább a téma technikai ismertetésével foglalkoznék.

2.1 A támadók bemutatása: célok és lehetőségek

Első lépésben azt érdemes megvizsgálni, hogy kik azok, akik adatainkat veszélyeztetik, és milyen célok motiválják őket. Általánosságban elmondható, hogy a támadók közvetett, vagy közvetlen célja mindig a pénzszerzés, amelyet – ha közvetve is - a gyanútlan felhasználók finanszíroznak.

Ennek igazolásaként tekintsük meg a következő példákat. Egy web áruházat üzemeltető cég a személyre szóló eladásain keresztül közvetlenül a vásárlótól tesz szert a magasabb bevételre. Ezzel szemben a profil készítésével, adatbázisok adás-vételével foglalkozó cégek, személyek a hasznukat a profilok, adatbázisok vásárlóin realizálják, azonban végső soron ezt a pénz közvetve mégis a felhasználók fogják finanszírozni, hiszen adatbázisok vásárlói abban a reményben szerzik be ezeket a termékeket, hogy azok használata révén extra bevételre tehet szert.

2.1.1 Profilkészítés

Profilkészítés alatt azt értjük, hogy a cégek nyomon követik, tárolják az egyes ügyfelek böngészési szokásait, azt hogy mikor, mit néznek meg, miket rendelnek, milyen termékek, szolgáltatások iránt érdeklődnek.

A profilkészítés eredményeként rendelkezésükre álló profilokat például a dinamikus árazáshoz használhatják fel. Dinamikus árazás alatt azt értjük, hogy a cégek az ügyfelek preferenciái szerint egyedi árakat hoznak létre, egyedi ajánlatokkal kecsegtetik őket.

A dinamikus árazás mellett az elkészített profilok felhasználhatják célzott hirdetések, személyre szabott szolgáltatások elkészítésére is. Ezeknek – hasonlóan a dinamikus árazáshoz – az az alapja, hogy a szolgáltató tisztában van a személyes preferenciáinkkal, meglepő pontossággal tudja, hogy mely termékcsoportok iránt érdeklődünk, milyen szolgáltatásokat szoktunk igénybe venni, és ezek alapján tud minket megkeresni olyan ajánlatokkal, amelyektől a bevétele növekedését várja.

Ennél a visszaélési fontos megjegyezni, hogy a profilkészítésnek lehet jó oldala is, hiszen a felhasználó számára előnyös lehet, hogy adott esetben csak olyan terméket kínálnak számára, ami releváns a számára, azonban az áruházak célja ebben az esetben is a nagyobb bevétel elérése.

2.1.2 Identitáslopás

Identitáslopásról akkor beszélhetünk, amikor valaki más személynek adja ki magát abból a célból, hogy ezáltal pénzt vagy egyéb előnyöket szerezzen. Az identitáslopás nem új fogalom, már az Internet elterjedése előtt évtizedekkel is ismert volt, számos országban törvény szabályozza a kérdést és bűnténynek minősül, ha valaki más személynek adja ki magát.

Ha csak a számítógépekkel kapcsolatos identitáslopásokat vesszük számba, akkor is számos lehetőségük van a támadóknak a gyanútlan – és sok esetben felelőtlen – felhasználók identitásának megszerzésére. Az identitás ellopása egyre több ember számára jelent valós veszélyt, amit a legjobban az jelez, hogy Nagy-Britanniában a felhasználóknak már lehetőségük van úgynevezett „identitáslopás-biztosítások” kötésére [2].

2.1.3 Phishing és Whaling

A phishing (magyarul:adathalászat) egy olyan eljárás, amelynek során a támadó egy jól ismert cég hivatalos oldalának hű másolatát létrehozza és ezen keresztül próbál meg

bizonyos személyes adatokat, például jelszót, bankkártya-számot illetéktelenül megszerezni.

A támadó ahhoz, hogy az áldozata meglátogassa ezt a hamis oldalt, általában e-mailt küld, amiben az eredeti cég nevében megpróbálja rávenni a címzettet a hivatkozás követésére.

A whaling a phising egyik változata. Whaling esetében a támadók olyan célcsoportokat vesznek célba, akiken keresztül jóval nagyobb haszon elérését remélik. Például magasabb beosztású dolgozókat, vagy olyan személyeket keresnek meg, akik jogosultak nagyobb vállalati pénzek kezelésére, adományok nyújtására, így magasabb bevételre tehetnek szert a támadók még relatíve kevés sikeres támadás esetén is.

2.1.4 Egyéb visszaélések

A bemutatott főbb támadási lehetőségek mellett természetesen más formában is veszélybe sodródhat privátszféránk biztonsága. Egyre többet hallani munkahelyi megfigyelési botrányokról, számos helyen előfordul, hogy monitorozzák a dolgozók e-mailjeit és figyelik, hogy milyen honlapokat látogatnak meg mialatt munkahelyükön tartózkodnak.

Azonban a munkahelyi megfigyeléseknél is mind erősebben érdekeltek az államok a megfigyelésekben. Egyre sűrűbben hallhatunk tiltakozó hangokat az ellen, hogy számos országban titkosszolgálati szerveknek lehetőségük van a személyes elektronikus adatforgalom megfigyelésére. A nagyhatalmak titkosszolgálatai óriási pénzeket ölnek az ilyen lehallgató-központok fejlesztésébe [3], amelyek egyrészt a becsületes állampolgárok védelmét szolgálja és a központok használatának lehetősége erőteljesen korlátozott, de a tudat, hogy bármikor olvashatják a levelet, amit éppen elküldtünk, aggasztó lehet.

2.2 A támadók technikai lehetőségei

Az alábbi fejezetben sorra veszem azokat az eszközöket és ezekre épített módszereket, melyek segítségével a támadók megvalósíthatják a fentebb bemutatott céljaikat.

2.2.1 Alapvető eszközök a támadás során

Az következő pontokban ismertetem azokat az eszközöket, melyekre alapozva a támadók különböző módszerek révén fenyegetik adataink biztonságát.

2.2.1.1 IP (Internet Protocol) Cím

Az IP cím egy egyedi hálózati azonosító, amelyet az Internet Protocol segítségével kommunikáló számítógépek egymás azonosítására használnak. Minden egyes számítógép rendelkezik ilyen címmel, amelyik csatlakozik az Internetre, a cím állandó (statikus IP) vagy folyamatosan változó volta (dinamikus IP) szolgáltatás-függő.

Az IP cím azért bír számunkra jelentőséggel, mivel az Internetet használó számítógépek minden tranzakció során (például egy oldal betöltése során) megadják IP- címüket a kiszolgáló számára, ami alapján elvileg be lehet azonosítani a gép helyét és a felhasználó több egyéb adatát, melyekre a későbbiekben még kitérek.

Mivel az IP címből a felhasználó számos adata kideríthető, azt több országban – így Magyarországon is – személyes adatnak nyilvánították, és törvényben szabályozták, hogy a szolgáltatók mely esetben adhatják ki, hogy az egyes felhasználók milyen IP címeket használtak.

2.2.1.2 Cookie-k

A cookie-k (magyarul: sütik) az alapvetően állapotmentes HTTP protokollba bevezették az állapot fogalmát. Így a sütik által, amikor visszalátogatok egy oldalra, vagy egy másik aloldalra lépek, akkor nem úgy kezel minket a szerver, mintha először járnánk az adott oldalon, hanem már visszatérő vendégként. Azon technológiáknak, amik ezt a kényelmi, de a biztonságot csökkentő funkciót lehetővé teszik, kétségkívül vannak előnyei és hátrányai is, mint azt az alábbiakban olvashatjuk.

Először is azt a kérdést kell tisztáznunk, hogy mi is az az úgynevezett süti. A süti voltaképpen nem más, mint egy egyszerű string [4], ami betöltődik a böngésző memóriájába és a számítógépen mentésre kerül. Egy süti 4 részből épül fel:

- név
- érték

- lejárati idő
- elérési útvonal.

Az elérési útvonal kapcsán fontos megjegyezni, hogy csak azok a szerverek képesek kezelni a sütit, amelyek az elérési útban szerepelnek. Biztonsági okokból egy szerver, csak a saját elérési útvonalát állíthatja be, másét nem.

A sütik önmagukban nem rejtenek veszélyt, ám ha rosszhiszeműen használják fel, akkor hathatós fegyver lehet, adatvédelmi szempontból számos aggályt kelthet. Általuk feltérképezhetőek a böngészési szokásaink, a sütit létrehozó oldal tulajdonosai profilt építhetnek rólunk, melyet a későbbiekben kihasználhatnak például marketing-kampányok segítségével. A nyomon követésre használt sütiket a szakirodalom „tracking cookie” névvel illeti [5].

A sütiknek az alábbi főbb fajtáit különböztetjük meg [6]:

- **Sima Cookie**
  A sima időkorlátolt süti arra szolgál, hogy egy webes felületen a beállításainkat elmentse két belépés között. Addig él, amíg a lejárati ideje szól.
- **Session Cookie**
  Ez az úgynevezett egyszer használatos süti, addig él, amíg a böngésző fut, ezek a sütik nem tárolódnak el a felhasználó gépén, a böngésző bezárásával törlődnek.
- **First Party Cookie**
  Minden olyan süti, amely az adott oldal domainjéről származik.
- **Third Party Cookie**
  A neve is tal arra, hogy ez a sütitípus egy harmadik féltől származik. Ez alatt pontosabban azt értjük, hogy egy másik szerver belinkelt képei, médiái, oldalai, amelyet a website-on kívülre mutatnak is adhatnak sütit. Reklámbannerek tartalmazhatnak például ilyen sütiket, statisztikai adatok gyűjtésére használhatják őket, tehát leginkább ezek a Third Party Cookiek használatosan a fentebb megfogalmazott marketingcélokra.

2.2.1.3 Local Shared Object (LSO)

Az LSO-k (magyarul: helyben tárolt megosztott objektumok) a sütikhez hasonlóan munkamenet-azonosítók, illetve egyéb hasznos adatok tárolására használatosak, azonban ezekben az objektumokban csak a Flash-ben írt programok tárolhatnak információkat.

Az LSO-k egyik legfőbb újdonsága, hogy képesek a törölt sütik visszaállítására, ebben jelent többet, újdonságot az egyszerű sütikhez képest [7].

A sütikhez hasonlóan az LSO-k is lehetnek hasznosak, ugyanakkor rosszindulatúak is. A nyomon követésre használt LSO-kat PIE-nak (Persinstent Identification Element, megmaradó azonosító elem), magyarul a süti elnevezés analógiájának megfelelően pitének hívják.

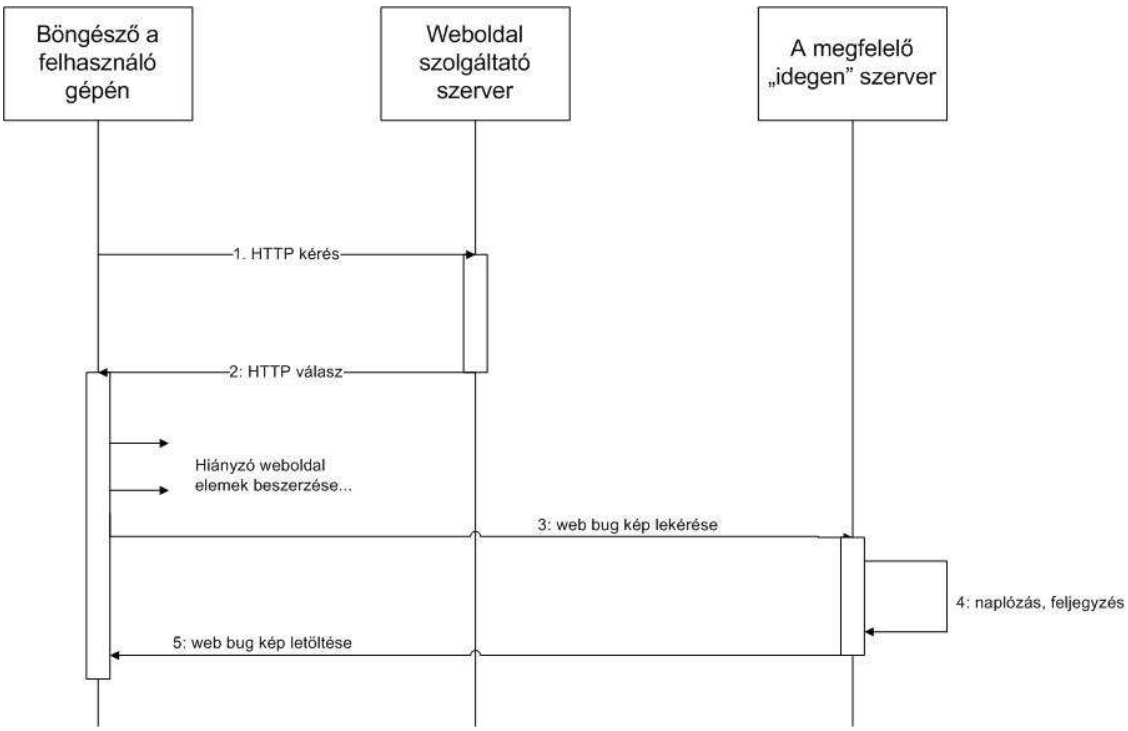
2.2.2 A bemutatott eszközökre épülő módszerek

A következőkben az fentiekben bemutatott eszközökre építő módszereket veszem sorra, megvizsgálom, hogy az egyes módszerek milyen veszélyeket jelentenek privátszféránkra.

2.2.2.1 Webpoloskák

A webpoloskák már elnevezésükkel utalnak azokra a poloskáknak nevezett eszközökre, amiket lehallgatásra használnak. A webpoloskák, amelyek általában 1*1 pixeles grafikus elemek és HTML oldalakon, vagy e-mailekben kerülnek elrejtésre, arra szolgálnak, hogy a felhasználót nyomon kövessék [8].

Röviden összefoglalva a nyomon követés a következőképpen lehetséges. Mint említettem a webpoloska egy kisméretű, átlátszó kép, így meglapulhat az oldal bármely részén, jó eséllyel észre sem vesszük, erre csak akkor van némi esélyünk, ha az oldal HTML kódját vizsgáljuk. A nyomon követés azért valósulhat meg, mert a képet egy harmadik fél szerveréről töltődik be a kép. Amikor a kép letöltésre kerül, a poloska tulajdonosa (a harmadik fél) tudomást szerez arról, hogy mikor, milyen IP címmel, milyen böngészővel rendelkező felhasználó töltötte le a poloskával ellátott oldalt. A webpoloskák letöltésének folyamata az 1. képen látható.



1. ábra Webpoloskák működési mechanizmusa [8]

Ha a poloskát sütik használatával kombinálják, akkor például egy vásárlás során a harmadik fél számára lehetővé válik a vásárló és a vásárolt termék egymásnak való megfeleltetése.

2.2.2.2 Hirdetések

A hirdetések is felhasználhatók rosszindulatú támadások eszközéül. Ez általában akkor lehetséges, amikor egy weboldal hirdetési felületét nem közvetlenül értékesíti a weboldal üzemeltetője, hanem egy harmadik fél által, aki általában kifejezetten hirdetések közvetítésével foglalkozik. Ebben az esetben nagyobb valószínűséggel előfordulhat, hogy a rosszindulatú hirdető átcsúszik ellenőrzési pontok és hamis reklámot tud megjelentetni. Ilyen eset volt, amikor egy New York-i napilap honlapján megjelent egy hirdetés, ami ál antivírus program letöltésére sarkallta a felhasználókat, majd később folyamatosan ajánlatokkal bombázta a letöltőket [9]. Ennek analógiájára előfordulhat, hogy nem olyan

csak ajánlatokkal árasztják el a felhasználót, hanem olyan programot telepítenek, ami kiadja a felhasználó személyes adatait.

2.2.2.3 Audit szolgáltató hivatalos poloskái

A már korábban bemutatott nyomon követési technikák felhasználásával – legális – auditálási szolgáltatások is elérhetők a Weben. Ezen szolgáltatás például alkalmas lehet arra, hogy egy honlap látogatottságát, illetve az aloldalak látogatottságát mérje a szolgáltató.

Ilyen auditálásra használatosak például a Yahoo! webpoloskákhoz hasonlatos web beaconje [10], melyeket ha letöltünk a letöltő böngészőt egy süti jelöli meg. Ezek a kis fájlok alkalmasak arra, hogy mérjék az egyes a Yahoo! honlapján belüli aloldalak látogatottságát, de a felhasználói szokásainak monitorozására is alkalmazzák őket, ennek segítségével nyújtva egyre inkább személyre szabott szolgáltatást. A web beaconöket a felhasználók Yahoo! oldalán kívüli böngészés során is használják, azonban erről csak aggregált, anonim információkat tárolnak el.

2.2.3 Egyéb módszerek

Az alábbiakban bemutatok olyan eszközöket és módszereket, amelyek között megtalálhatók – korábban ismertetettekhez hasonlóan – nem feltétlenül rosszindulatúak, de támadásra is lehet őket használni, illetve olyan technikákat is, melyeket kifejezett rosszindulatú cselekedetek végrehajtására használnak.

2.2.3.1 URL referrer

Az URL referrer a webpoloskával ellentétben jóindulatú technológiaként terjedt el, azonban használata mégis csökkenti biztonságunk szintjét, így nem érdemes minden körülmények között használni és pedig mindenképpen tárgyalnunk kell.

Az URL referrer a HTTP protokoll egyik fejléce, amelybe a böngésző annak az oldalnak a címét írja, ahonnét az újonnan megnyíló oldalra hivatkoztak. Tehát ha egy linkre kattintok, akkor a böngésző megküldi a letöltendő oldalnak annak az oldalnak a címét, amelyik a linket tartalmazza.

Egyszerűen belátható, hogy a referrer a látogatottsági statisztikák fontos eleme, segítségével meg lehet tudni, hogy honnan érkezik a legtöbb látogató egy weboldalra, illetve milyen útvonalat követnek az oldalon belül; emellett a jogosulatlan lekérések és a hamisított lekérések kiszűrésére is használják.

Egyes böngészőkben, illetve egyes tűzfalak segítségével le lehet tiltani vagy meg lehet változtatni a referrert, ezt referrer spoofingnak [11] hívják. Néha úgy próbálják egy oldal látogatottságát vagy keresőbeli helyezését javítani, hogy mesterségesen kreálnak más oldalakon látogatásokat, és referrerként a reklámozandó oldalt adják meg; ez a referrer spam.

2.2.3.2 Böngésző által megadott adatok gyűjtése

Amint az már az 2.2.1.1 fejezetben említettem, az IP cím felfedése által beazonosíthatóvá válik helyünk, és még számos adat, amihez egyszerűen hozzáférhetnek a támadók. Amint azt a későbbiekben az 6.3 fejezetben látni fogjuk, erre alkalmas honlapok segítségével megtekinthetjük ezeket az információkat.

Az IP cím, valamint a böngésző által megadott egyéb adatok szinte minden adatot felfed a felhasználó számítógépéről. Csak a böngésző által megadott adatok alapján óriási adatbázisokat lehet összeállítani, amelyek alkalmasak felhasználói profilok összeállítására, amiket a későbbiekben akár rosszindulatúan is felhasználhatnak.

2.2.3.3 Több kiszolgáló logjainak összehasonlítása

Amikor letöltünk egy weboldalt, a kiszolgáló szerveren ennek nyoma marad, a log fájlban bejegyzésre kerül, hogy mikor milyen IP címről töltöttek le egy adott oldalt. Az ilyen log fájlok is alkalmasak a visszaélésekre, profilok építésére, amennyiben két kiszolgáló megvizsgálja, hogy a közös metszetét a két log fájlnak. Az összejátszó kiszolgálók így képet nyerhetnek a felhasználók böngészési szokásairól.

2.2.3.4 Spyware (magyarul: kémprogram) alkalmazások

Kémprogramnak nevezzük az olyan, interneten terjedő számítógépes programokat, amelyek arra törekednek, hogy törvénytelen módon megszerezzék a megfertőzött számítógép felhasználójának személyes adatait.

Ezen programok települése általában a felhasználó számára észrevétlenül történnek, vagy a felhasználó figyelmetlenségét, és/vagy a böngésző rossz beállításait kihasználva. Ezek mellett olyan magukat álcázó programok is léteznek, melyek a felhasználó közreműködésével települnek rosszindulatú honlapokról. A megszerzett információk felhasználhatók profilok készítésére, ugyanakkor lopásra is, ha például a bankkártya-számunk is a megszerzett adatok között vannak.

2.2.3.5 JavaScript gyorsítótárazási problémák kihasználása

A webpoloskák módszeréhez hasonló módon a JavaScript gyorsítótárazott fájljai alkalmasak nyomon követésre. Ebben az esetben a honlaphoz egy JavaScript állományt csatolnak, amelyben egy változó segítségével eltárolnak egy azonosító értéket, ami a későbbi látogatások alkalmával elérhető lesz egészen addig, amíg a gyorsítótár nem frissül.

Ezt a módszert lehet használni az azonosítót tartalmazó süti pótlására. Minden egyes betöltéskor, amikor nincs jelen azonosító süti, az létrejön a JavaScriptből, fordított esetben pedig, a süti írja a JavaScript állományba a megfelelő értéket [12].

2.2.3.6 Cross-site scripting

A Cross Site Scripting (továbbiakban XSS) napjaink egyik leggyakoribb webes veszélyformája, ami a rosszindulatú támadók számára lehetővé teszi kliensoldali scriptek beszúrását mások által is látogatott weboldalakba. 2007-ben a weboldalakat érintő dokumentált veszélyforrásoknak hozzávetőlegesen 80 százalékát tette ki az XSS [13].

Az XSS-t lehetővé tevő biztonsági lyukakon keresztül a támadóknak lehetővé válik a kliens oldali biztonsági mechanizmusok kijátszása. Azzal, hogy rosszindulatú scripteket tudnak beszúrni a támadók egy oldalba, hozzáférést szerezhetnek nem publikus tartalmakhoz, megszerezhetnek session cookie-kat, illetve egyéb más privát adatokat, melyeket a böngésző a felhasználóról tárol.

Megkülönböztethetünk perzisztens és nem perzisztens XSS-eket. Utóbbiaknál a perzisztens XSS-ek jóval veszélyesebbek, mivel beillesztés után is a rendszerben maradnak, általában beleíródnak az oldal adatbázisába és minden letöltéskor

megjelennek, míg a nem perzisztens XSS-ek általában URL, nem pedig állandó kérés formában jelennek meg.

Az XSS-eket leginkább sütik ellopására használják, de sok egyéb dologra is használható, azonban számunkra ez utóbbi bír a legnagyobb fontossággal.

2.2.3.7 RSS/Atom olvasásának figyelése

Az RSS és az Atom hasonló célra szolgáló XML állományformátumok családja. Segítségükkel a felhasználók megkímélhetők attól, hogy az ezeket a technológiákat használó weboldalalakat folyamatosan frissíteniük kelljen az új tartalom ellenőrzése végett. A manapság használatos böngészők szinte mindegyike alkalmas arra, hogy ellenőrizze az RSS vagy Atom csatornákon kapott üzeneteket és frissítse a weboldalt.

Azonban az ezen technológiák által használt információk szintén alkalmasak arra, hogy felmérjék a felhasználó szokásait, érdeklődési körét, azaz alkalmas profilok készítéséhez.

2.2.3.8 Kezdőlap olvasásának figyelése

Amikor elindítjuk a böngészőnket és az letölti a beállított kezdőlapot, akkor erről a letöltésről a honlapot kiszolgáló szerveren egy bejegyzés születik. Amennyiben ezeket a bejegyzéseket megfigyelik, az IP cím által beazonosíthatóvá válik például, hogy a számítógép ahonnét letöltik az oldalt otthoni vagy munkahelyi állomás-e. Ha belegondolunk ez egyszerűen megtehető, mivel a bejegyzés tartalmazza a letöltés időpontját, az IP címből pedig kideríthető, hogy hol, melyik időzónában található a felhasználó, így jó eséllyel megállapítható, hogy otthonról, vagy a munkahelyéről töltötte be a kezdőoldalt.

2.3 Védekezési lehetőségek

A következő fejezetben bemutatok olyan módszereket, eszközöket, kutatásokat, melyek alkalmasak a korábban bemutatott veszélyek elleni védekezésre, használatuk nagyobb védelmet biztosít adataink számára.

2.3.1 Anonim proxy

Az anonim proxy az Interneten böngésző felhasználó identitásának elrejtésére alkalmas proxy kiszolgálató. Működésének lényege, hogy a kiszolgáló és a böngésző fél közötti

kommunikációba harmadik félként beépül oly módon, hogy valójában az anonim proxy tölti le a kiszolgálóról a felhasználó által kért adatokat, amit utána továbbít az eredeti kérő részére. Mivel azonban így a kiszolgáló közvetlenül csakis a proxy szerverrel áll kapcsolatban, a tényleges böngésző identitása (IP-címe) a szerver elől rejtve maradhat.

Kétféle anonim proxy létezik:

- Az egyik fajta technikai értelemben is "proxy", tehát őt kell megadnunk a böngészőnkben az Internet-szolgáltató proxyja helyett. Ilyenkor automatikusan minden Web-hozzáférésünk névtelen lesz.
- A másik típusnál egy webes felületen kell beírnunk a letölteni kívánt oldal címét, az ahhoz tartozó oldalt a szerver ezután letölti, a linkeket pedig úgy módosítja benne, hogy szintén az anonimizálóra mutassanak. Ezáltal, amennyiben egy linket követünk, továbbra is névtelenek maradunk.

Problémát jelent az anonim proxyk használata esetén, hogy a felhasználó gépe és a proxy közötti hálózati szakasz lehallgatható az Internet szolgáltató által.

2.3.2 Anonim böngészők

Az anonim böngészők az anonim proxyk IP cím rejtésére vonatkozó szolgáltatásán túlmenően további lehetőségeket nyújt a felhasználók részére. Ilyen szolgáltatás lehet például az titkosítás, tartalomszűrés [12].

Egy anonim böngésző megfelelő esetben nem csak a kiszolgáló felé biztosítja az IP címünk takarását, hanem figyel arra is, hogy azon a számítógépen, amelyen böngészünk, se hagyjon nyomot utánunk, ne legyen visszakövethető, hogy milyen oldalakon jártunk. Továbbá védelmet nyújthat a már bemutatott eszközök, módszerek ellen, így lehetőséget biztosíthat a JavaScriptek, az URL Referer szűrésére, a Java, valamint Flash objektumok tiltására, a valamint a sütik osztályzására és kezelésére.

2.3.3 Anonim böngészők privátszféra-barát identitásmenedzsmenttel

Egy lehetséges válasz az ismertetett veszélyekre az anonim böngészők identitásmenedzsmenttel kiegészített alkalmazása. Az identitásmenedzsment olyan

rendszerek és folyamatok összessége, melyek segítségével szabályozható és irányítható, hogy a különböző felhasználók milyen körülmények között milyen erőforrásokhoz férhetnek hozzá [14].

Többféle identitásmenedzsment rendszerről beszélhetünk. A vállalati IDM rendszerek elsősorban a vállalati igényeket elégítik ki, ebben az esetben nem az egyén, hanem a vállalat gyakorolja az irányítást, ami ellentétes az ebben a diplomatervben megfogalmazott célokkal.

Ezzel szemben léteznek felhasználó központú IDM rendszerek, ebben az esetében olyan rendszerről beszélhetünk melynek keretien belül az egyén kezébe helyezik saját maga, saját adatinak sorsát. A diplomaterv célja is egy ilyen – anonim böngészőhöz csatolt – IDM alkalmazás megtervezése és implementálása.

Szövetséges IDM rendszerekről beszélünk, ha különböző rendszerek azonosítást végző felügyeletek megállapodnak abban, hogy amennyiben az egyik beenged valakit a rendszerbe, akkor a másiknak is kötelessége azt elfogadni.

2.3.3.1 Szerepalapú privátszféra-védelem

Az identitásmenedzsment egyik típusa a szerepalapú privátszféra-védelem (Role Based Privacy, RBP). Az RBP [15] nem új keletű dolog, már régóta és számos helyen alkalmazzák. A mindennapok során számos szerepet töltünk be, vagyunk családtagok, munkatársak, fogyasztók, és még hosszasan sorolhatnám. Szerepalapú privátszféra-védelem használata esetén ezekhez a különböző szerepekhez különböző identitásokat rendelhetünk hozzá, így külön-külön beállíthatjuk például, hogy egy csevegő-szolgáltatás használata során a munkaidő végéig láthatatlanok vagyunk a család, barátok részére, azonban, amint lejárt a műszak, a helyzet megfordul és a kollégák számára válunk láthatatlanná.

A szerepalapú privátszféra-védelem célja, hogy a felhasználó több profilt hozhasson létre saját maga számára a különböző szerepkörökhöz és ő maga eldönthesse, hogy adott helyzetben melyik szerepkörének megfelelő profilját mutassa más felhasználók, szolgáltatók felé.

2.3.3.2 PRIME Projekt

A fejezet elején megfogalmazott problémákat az Európai Unió döntéshozói is felismerték, ezért életre hívták a PRIME (Privacy and Identity Management for Europe) projektet. A PRIME egy olyan kutatási projekt, amely egy privátszféra-védelem erősítő identitás menedzsment rendszer felállítását célozza meg [17].

A felhasználó-központú identitásmenedzsment fő okait és céljait illusztrálja a következő példa. Mindenki a saját életében sok-sok különböző szerepet tölt be, mindnyájan vagyunk családtagok, barátok, kollegák, felhasználók és még sok minden más. Az új technológiák, az információs társadalom térnyerésével egyre többen intézzük ügyeinket interneten keresztül, egyre több helyen szerepelnek az adataink, sokszor akaratunktól függetlenül. Egy jól felépített IDM rendszer a korábban példaként felhozott szolgáltatásoknál természetesen jóval többre képes, mint azt a következőkben láthatjuk.

2. ábra A PRIME és különböző szerepek [17]

A PRIME egy olyan rendszer létrehozását célozza meg, melyet az informatikában kevésbé jártas emberek is egyszerűen bizalommal használhatnak. Ennek a bizalomnak alapja az egységes és egyszerű kezelőfelület, valamint az, hogy senkibe se kelljen bizalmat helyeznünk ahhoz, hogy igénybe vehessünk egy szolgáltatást. Célja, egy olyan keretrendszer létrehozása, amely integrálja az összes technikai, illetve nem technikai aspektusát az IDM alapú privátszféra-védelemnek.

Mint említettem, egy identitásmenedzsment rendszer számtalan területet magába foglalhat, a PRIME az alábbi területeken kíván biztonságos szolgáltatást nyújtani:

- egészségügy,
- helyi szintű szolgáltatások,
- privátszféra megőrző vásárlói adatbázisok,
- anonim hozzáférés biztosítása az infrastruktúrához mobil dolgozók részére.

Voltaképpen a PRIME az előre definiált szabályok és a felhasználó preferenciáit tükröző szabályok automatikus végrehajtása mellett szerepalapú privátszféra-védelmet valósít meg.

2.3.3.3 PrimeLife

A PrimeLife a PRIME projekt folytatása, célja, hogy fenntartható privacy- és identitásmenedzsmentet hozzon a jövő hálózatainak és webes szolgáltatásainak [18]. A két fő terület, amire választ kíván adni

- egyrészt az, hogy hogyan oldható meg a privátszféra védelme, amikor egyre jobban terjednek az együttműködő webes szolgáltatások és a virtuális társadalmak,
- másrészt pedig, hogy hosszútávon hogy lehet fenntartani a privátszféra épségét.

A PrimeLife igyekszik választ adni ezekre a kulcskérdésekre, amik a privátszférát és a bizalmat érintik. Ahhoz, hogy ezeket a problémákat meg lehessen oldani, kezelni lehessen a PrimeLife fejleszteni kívánja az ember-számítógép interfaceket, a konfigurálható politika nyelveket, webszolgáltatás szövetségeket és a privátszféra-védelmet növelő kriptográfiai eljárásokat.

Másrészről a PrimeLife projekt céljául lett kitűzve ezeknek a technológiáknak széles társadalmi körökben való elterjesztése. Ebből a célból a projekt részvevői együttműködnek a legjelentősebb nyílt forráskódú projektekkel, fejlesztőkkel.

2.3.4 Az azonosíthatóság behatárolása: Nymity slider

Érdekes kérdést vet fel, hogyan lehet mérni a biztonságot? Hogyan határozhatjuk meg, hogy a magunkról felfedett adatok mennyire veszélyeztetik anonimitásunkat?

A Goldberg disszertációjában [19] ismertet egy olyan nimity slidernek elnevezett skálát, amely alkalmas a biztonsági szintet mutatni egy négy fokozattal rendelkező skálán.

A skála különböző szintjein az általunk mások által ismert, pontosabban megismerhető adatok mennyisége és minősége alapján állnak a különböző fokozatok.

A négy fokozat:

- összeköthetetlen teljes anonimitás
- összeköthető anonimitás
- perzisztens pszeudonimitás
- az anonimitás teljes hiánya

Összeköthetetlen teljes anonimitás alatt nem csak azt értjük, hogy semmilyen személyes adatunkat nem adjuk meg, hanem azt is, hogy egy külső megfigyelő azt sem tudja megmondani, hogy egy korábbi tranzakció során mi

Összeköthető anonimitás alatt azt értjük, amikor továbbra sem adunk ki magunkról az identitásunkra vonatkozó adatokat, továbbra is anonimek maradunk, azonban azt már a figyelmes megfigyelő megállapíthatja, hogy két különböző tranzakciót ugyanazon személy hajtott végre.

Ezzel szemben a perzisztens pszeudonimitás esetében, már az identitásunkra vonatkozó információkat is felfedünk, azonban azt titkoljuk, hogy az általunk használt identitás mögött ki rejtőzik. Leginkább ahhoz hasonlíthatjuk ezt az esetet, amikor álnéven írunk, mindenki tudja, hogy az álnéven milyen cikkek kerültek megírásra, de azt senki sem tudja rajtunk kívül, hogy az álnév mögött a valóságban ki bújik meg.

Az anonimitás teljes hiánya alatt azt értjük, amikor olyan adatot adunk meg magunkról, ami kizárólag ránk vonatkozik, pontosan beazonosít minket. A hétköznapok során ilyen például a személyi vagy az adószámunk, az online világban pedig az IP címünk ilyen egyedi azonosító. De akár ennél „személytelenebb” adataink is beazonosíthatnak minket: egy amerikai kutatás arra a következtetésre jutott, hogy az Egyesült Államokban élő lakosok 87 százalékát meghatározza az irányítószáma, neme és születési dátuma [20].

Goldberg alapvetése, hogy ha több információt osztunk meg, és eggyel kevésbé biztonságos szintre süllyedtünk, akkor onnan már csak nagyon nehezen tudunk az előző magasabb szintre visszatérni. Ebből kiindulva a cél a lehető legkevesebb információ felfedése, a lehető legbiztonságosabb szint megőrzése, hogy innen mi dönthessük el, milyen adatokat fedünk fel, hogy mi határozhassuk meg a biztonsági szintünket, ne külső körülményekből kifolyólag süllyedjünk automatikusan lejjebb.

Az identitásmenedzsment alkalmazása során is érdemes kitűzni egy ilyen célt magunk elé. A felhasználó számára lehetővé kell tenni, hogy amikor elkezd böngészni garantált legyen számára a teljes anonimitás, majd innen kiindulva ő dönthesse el, hogy mennyi adatot kíván felfedni magáról, mekkora mértékben szeretne anonim maradni.

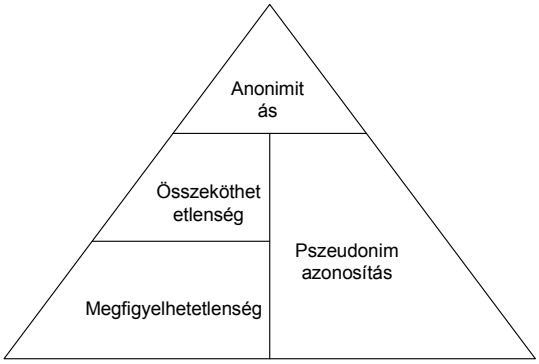
3 Egy IDM rendszerrel szemben állított kritériumrendszer

Az alábbiakban kifejtem, hogy milyen igények kerültek lefektetésre az alkalmazással szemben, mind az elméletei, mind a gyakorlati részeket tekintve.

3.1 Az anonimitás kérdése

A rendszernek egyik kiemelkedő fontosságú alapkövetelménye, hogy képes legyen az anonimitás gyakorlati biztosítására, hiszen ezen alapvető feltétel nélkül értelmét vesztené az identitásmenedzsment rendszer.

Azért használom a „képes legyen” kifejezést, mivel nem feltétlenül szükséges mindig teljesen anonimnek maradni. A rendszer a felhasználóra bízza az anonimitásról szóló döntést, azonban, ahol és amikor a felhasználó szeretne anonim maradni az alkalmazásnak ezt a lehetőséget garantálnia kell.



3. ábra Az anonimitás felépülése [21]

Anonimitás alatt a következőket értem [21]:

- a kommunikáció tartalmát csak a részvevők ismerhetik, más fél nem (megfigyelhetetlenség),
- egy külső megfigyelő nem képes megállapítani a felhasználó két üzenetéről, hogy mindkettő tőle származik (összeköthetetlenség),

- a felhasználó tevékenységei ha össze is köthetőek, legfeljebb egy fedőnévhez, vagy egyéb azonosítóhoz kapcsolhatóak, de ez alapján a valódi személyazonosság nem deríthető ki.(pszeudonim azonosítás),
- a felhasználó cselekedetei viszony szinten sem köthetőek össze, így üzenetenként független pszeudonim azonosítóval rendelkezik, ez adja összességében a rendszer teljes anonimitását.

A rendszert olyan funkcionalitásokkal kell ellátni, melyek lehetővé teszik az egyes honlapok anonimitást veszélyeztető elemeinek szűrését, korlátozását. Ezen szűrőknek a kritériumrendszerbe való illesztéséről a későbbi alfejezetekben részletesen írok.

3.2 Böngészővel szembeni követelmények

Legyen hordozható

Elvárás a használandó böngészőnk felé, hogy legyen hordozható. Ennek azt a módját választottam, hogy egy pendrive-on tároljuk, a böngésző fusson arról, így ha egy másik géphez illesztjük a pendrive-ot, akkor nem kell újra telepíteni, hanem egyből működésre készen áll a böngésző. Ennek az az előnye is megvan, hogy amennyiben a böngészés során letöltünk bizonyos sütiket, vagy LSO-kat, illetve vagy ha úgy állítjuk be a böngészőt, hogy tároljon el adatokat például a korábban meglátogatott oldalakról, vagy ha csak egyszerűen felfedjük magunkat a weben, akkor ezekről nem hagyunk nyomot, nem hagyunk adatot a használt gépen, mivel mindent a pendrive-on tárolunk.

Legyen kliensoldali

Ez a követelmény gyakorlatilag következik az előzőből, hogy a hordozható böngészőnk kliensoldali is lesz. Ez azért is előnyös többek között, mivel ellentétben azzal, amikor egy webes böngészőt használunk, ebben az esetben több beállítást kezelhetünk mi magunk, továbbá nem függünk attól, hogy mikor érhető el az adott oldal. Hátránya, hogy operációs rendszer-függő lesz az alkalmazás.

A szűrés menjen végbe az alkalmazási rétegben

Mivel anonim böngészésre alkalmas böngészőt szeretnénk létrehozni, ezért szükséges, hogy képes legyen egy anonimizáló hálózat alkalmazására. A böngésző a TOR (The Onion Routing) [22, 23] hálózatát fogja alkalmazni. A TOR használatával elérjük, hogy ne fedjük fel az IP címünket.

További fontos követelmény, hogy lehessen szűrni a tartalmi elemeket, információkat, így be tudjuk állítani, hogy engedélyezzük-e JavaScriptek, java alkalmazások letöltését, futását, engedélyezzük-e a flashek letöltését és megjelenését.

A biztonsági lehetőségeknek egy lényeges eleme, hogy az URL Refererrel kapcsolatosan is el tudjunk végezni beállításokat, hogy ne kelljen kiadnunk honnan jutottunk át egy adott oldalra.

Ezek mellett egyszerű lehetőséget kell biztosítanunk arra, hogy a böngészés során tárolt adatokat egyszerűen törölhessük. Ezzel kapcsolatosan mindenképpen szükséges egy sok beállítási lehetőséggel rendelkező sütimenedzsment modult megvalósítani.

Sütimenedzsment

Mint említettem, mindenképpen szükség van egy átfogó és sok lehetséges beállítást megengedő sütimenedzsmentre, annál is inkább, mivel ez az a pont, ahol rengeteg információt kiad magáról a gyanútlan felhasználó.

Ebben a pontban az alapvető követelményeket foglalom össze, a részletekkel a 4. fejezetben részletesebben foglalkozom.

Az alapkövetelmények a következők:

- az összes sütit bármikor törölhessük
- az adott session alatt letöltött sütik törülhetők legyenek a session végén
- amikor a másik fél sütit akar feltölteni nekünk, engedélyezhetjük, vagy elutasíthatjuk azt

- süti letöltésénél megadhatjuk, hogy szeretnénk-e az adott sütit a session végén törölni, vagy meg akarjuk tartani
- egy süti kezelőben megnézhetjük a sütik adatait illetve a hozzá tartozó beállításokat, azokat módosíthatjuk, illetve a sütiket bármikor törölhetjük
- beállíthatjuk, hogy míg bizonyos oldalakról nem fogadunk el sütiket, más oldalak kivételt jelentenek, onnan bármikor elfogadjuk a sütiket.

3.3 Identitásmenedzsment

Az elsődleges feladat az identitásmenedzsment megvalósítása, az anonim és hordozható böngésző ehhez csak szükséges kellék. Mivel a rendszer lényege már a korábbi fejezetek során összefoglalásra került, ezért itt csak röviden írok róla, majd utána rátérek a konkrét követelményekre.

Az IDM alkalmazás lényege, hogy szeretnénk egy olyan kiegészítést a böngészőhöz, amivel különböző weboldalakhoz különböző identitásokat tudunk létrehozni, és azokat kezelhetjük. Tehát létrehozhatok egy identitást például a levelezőoldalakhoz, és ebben a profilban tárolom azokat az adatokat, melyeket a regisztráció során meg kell adni, valamint a bejelentkezéshez szükséges adatokat.

A követelmények:

- Definiálhassunk identitásokat, akár egy, akár több laphoz. Meghatározzuk, hogy mely oldalak esetén, mit használjunk, milyen identitást, milyen adatokat adjunk meg.
- Támogassa a profilok közötti egyszerű váltásokat.
- Tegye lehetővé, hogy egy honlapon több profillal is jelen lehessünk.
- Biztosítsa az álcímek egyszerű generálást.
- Támogassa a profiladatok egyszerű karbantartását. Ezzel kapcsolatban megjegyzendő, hogy az nem elvárás, hogy ha beállítok egy jelszócserét, akkor azt csinálja is meg, de az igen, hogy figyelmeztessen a szükségességére.

- Tegye lehetővé az adatvédelmi beállítások identitásonként meghatározását. Minden egyes biztonsági beállítást meg kell tudnunk profilonként határozni.

Sütimenedzsment az identitásmenedzsment tükrében

Mint említettem a sütimenedzsmenttel egyfajta politikakialakítási lehetőséget is meg kell valósítanunk, tehát a sütikezelést az identitásokhoz is kell tudnunk kötni:

Be kell tudnunk állítani identitásonként, hogy hajlandóak vagyunk-e sütiket elfogadni az adott identitáshoz tartozó oldalaktól.

A fentebb említett sütikkel kapcsolatos követelmények összességét értelmeznünk kell identitásokkal kapcsolatosan is:

- identitásonként külön kell tudunk tárolni a sütiket
- az összes sütikre vonatkozó beállítást identitásonként kell tudnunk változtatni
- az összes az adott identitáshoz tartozó sütit bármikor törölnünk kell tudni
- az adott session alatt az adott identitás által letöltött sütiket törölnünk kell tudni a session végén
- egy süti letöltésénél megadhatjuk, hogy meddig kívánjuk azt megtartani: például session végéig, 1 napig, 1 hétig … vagy nem kívánjuk törölni
- be kell tudnunk állítani, hogy bizonyos oldalak kivételt jelentsenek, akár még az identitásokon belül is, onnan bármikor elfogadjuk a sütiket.

Definiáljunk sütikezelési alappolitikákat, amiket bármikor betölthetünk, és utána azokat módosíthatjuk is.

4 Egy IDM alkalmazás megtervezése

Tehát, amint azt a korábbi fejezetekben már kifejtésre került a cél egy olyan identitásmenedzsment alkalmazás megvalósítása a választott böngészőn, amellyel a lehető legmagasabb fokú anonimitást élvezhetjük, amikor csak akarjuk, egészen pontosan fogalmazva saját magunk állíthatjuk be, hogy mennyi és milyen adatot szeretnénk magunkról felfedni a böngészés során.

4.1 Identitásmenedzsment funkciók kialakítása

Az alábbi fejezetben a közvetlenül az identitások kezeléséhez kapcsolódó funkciók tervezését veszem sorra, úgy mint a profilok struktúráját, azok típusait, illetve a közöttük való váltások mikéntjét.

4.1.1 A profilhierarchia lehetséges megvalósításai

A korábbi kutatások során elkészült elméleti modellt kiindulási alapulnak véve, az identitásokat, profilokat egy fastruktúrára épülő profilhierarchiába szerveztem. Az alábbiakban bemutatom, hogy a fastruktúra kialakítása során milyen alternatívákat vizsgáltam meg, hogyan alakult ki az implementálandó modell.

4.1.1.1 Kiindulási modell

A fastruktúra gyökerében lévő identitásból első szinten származtathatjuk az úgynevezett sémákat, melyekkel azt határozzuk meg, hogy milyen célra fogjuk használni az identitást (fórumozás, levelezés, vásárlás stb…). Már ezekhez a profilokhoz is tárolhatunk oldalakat, amelyekhez az adott identitást hozzárendeljük, de ezekből az úgynevezett sémákból származtathatunk újabb identitásokat, melyeknél mindent megváltoztathatunk, kivéve a felhasználás célját, azaz a séma alapját.

A főprofilhoz nem lehet külön címeket megadni, amely oldalak esetén a profil automatikusan alkalmazásra kerülne, ehelyett a szerepe az, hogy azokon a helyeken, melyek címe egy identitáshoz sincs hozzárendelve, ez a főprofil fog érvénybe kerülni, ennek a profilnak a biztonsági beállításai kerülnek használatra.

A főprofilban viszonylag kevés adatot adhatunk meg, legfőképpen itt a biztonsági beállításokat állíthatjuk be, illetve, amennyiben szeretnénk automatikusan nevet, becenevet, e-mail címet rendelni a profilhoz, akkor ezen adatokat is hozzárendelhetjük a profilhoz.

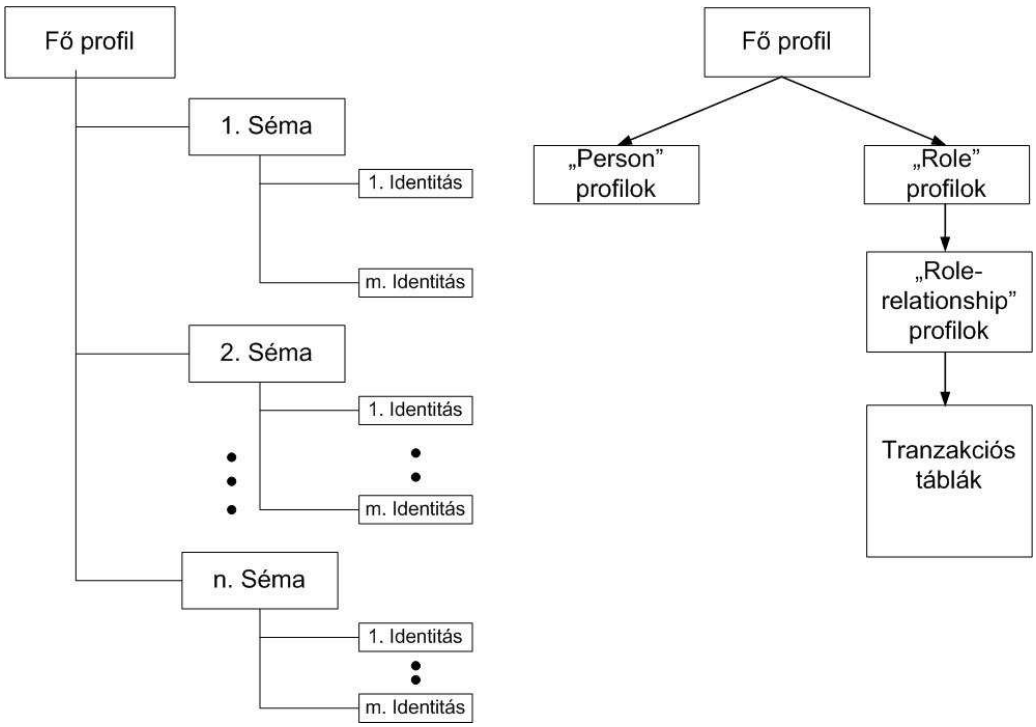
Ezzel szemben a következő szinten (a séma szintjén) már megadhatjuk, hogy milyen honlapok esetén kívánjuk használni az adott identitást, illetve jóval szélesebb körű adatokat adhatunk meg.

Az alsóbb szinteken található identitásokat mindig az eggyel feljebb lévő szülőidentitásból származtatjuk. Minden adatot megváltoztathatunk ezekben az identitásokban a szülőadatokhoz képest, kivéve azt az egyet, ami a sémának az alapja, mégpedig, hogy mire használjuk (fórum, levelezés, vásárlás, stb…).

A rendszerben lesz pár alapséma, de akármennyit felvehetünk melléjük. Az eltárolandó adatokat is tudjuk bővíteni, létrehozhatunk új mezőket. Hogy mennyire lesznek használhatóak az új mezők, az a felhasználó felelőssége.

4.1.1.2 Más modellekkel való összehasonlítás

A következő lépésben egy másik kutatás [14] által használt modellel hasonlítottam össze a fent vázolt modellt, melynek segítségével sikerült azt továbbfejleszteni.



4. ábra Vizsgált profilhierarchiák

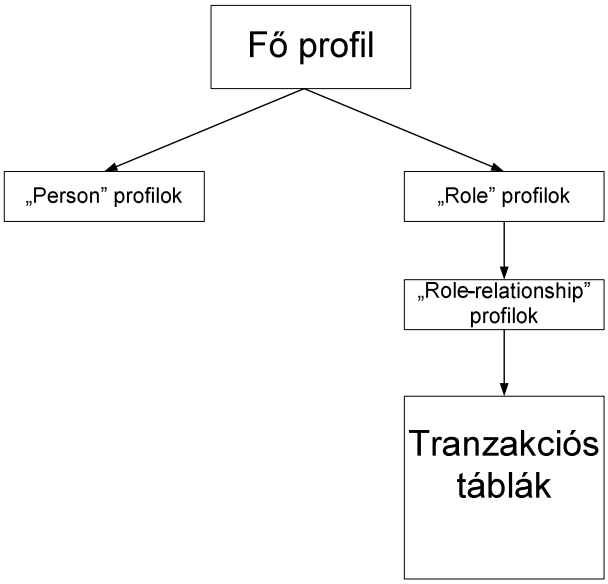
A cikk személyi pszeudonimjének az alapbeállítások profilja felel meg. A szerep pszeudonimnek a sémák felelnek meg. A kapcsolati pszeudonimnek megfelelő szint nem szerepel a modellben, illetve kicsit összeolvad a szerep pszeudonimmel, mivel nincsenek külön profilok arra, hogy kivel kommunikálunk, hiszen ez a rendszer nem tartalmaz például egy csevegő-szolgáltatást, vagy nem nézi azt, hogy kinek írunk levelet.

A cikk szerep-kapcsolati pszeudonimjének gyakorlatilag a hierarchia legalján szereplő azonosító felel meg, a szerep pszeudonimből származtatva van egy újabb, amiben pontosan meg van adva, hogy mely honlap(ok) esetén lehet használni.

A tranzakciós pszeudonimnek is van megfelelő profil. Gyakorlatilag ilyennel is rendelkezünk, ugyanis amikor folyamatosan új pszeudonimet generálunk egy identitáshoz (például fórumozásnál), akkor biztosítani kell az ideiglenes e-mail címek, ideiglenes profilok használatát a felhasználó számára.[2]

[2] Eldobható e-mail címek: http://eldobhato.mpp.hu/

Ezek alapján a végleges profilhierarchia a 5. ábrának megfelelően került kialakításra.



5. ábra Végső profilhierarchia

4.1.2 Szerepsémák

Amit eleinte sémáknak neveztem el, azok a következő fejlesztési szakaszban a szerep pszeudonimek nevet kapták, melyek köre bővíthető a felhasználó által. Az alábbiakban felsorolom, hogy az alapvető szerepekhez milyen mezőket javaslok létrehozni.

Levelezőrendszerhez:

- e-mail cím
- jelszó
- jelszó megerősítése
- vezetéknév
- keresztnév
- nem
- születési adatok
- cím (irányítószám, város, utca)
- biztonsági kérdés (kigyűjtendő tipikus kérdéseket)
- biztonsági kérdésre adandó válasz

Fórumokhoz, ahol kell valós e-mail

- e-mail cím
- jelszó
- jelszó megerősítése
- név
- nick

Fórumokhoz, ahol nem kell valós e-mail

- e-mail cím (generált is)
- név
- nick

Elektronikus üzletekhez

- e-mail cím
- jelszó
- jelszó megerősítése
- vezetéknév
- keresztnév
- nem
- születési adatok
- cím (irányítószám, város, utca)
- biztonsági kérdés (kigyűjtendő tipikus kérdéseket)
- biztonsági kérdésre adandó válasz
- bankszámlaszám
- telefonszám

Természetesen minden egyes adatot letilthatunk, attól még, hogy szerepel az identitásban, nem feltétlenül kerül automatikus kiadásra, csak akkor, ha engedélyezzük is az adat használatát.

4.1.3 Identitások közötti váltások

Az identitások váltásának teljes automatizálhatósága kérdéses. Fontos, hogy változtathassunk bármikor identitást manuálisan is, azonban gondolni kell a kényelmi szempontokra is.

Arra az esetre, ha egy adott oldalon jelen vagyunk (pl: www.bme.hu/.....) és átnavigálunk egy másik oldalra (www.xyz.com/....), amelyik nem rendelkezik külön identitással, a böngészőnek több lehetőséggel kell rendelkeznie:

- egyrészt alkalmazhassuk a régi profilt egészen addig, amíg nem jutunk egy olyan oldalra, amelyhez van saját profil definiálva
- másrészt pedig választhassunk, hogy kívánjuk-e az adott identitáshoz kapcsolódó biztonsági (és egyéb) beállításokat használni a továbbiakban, vagy pedig a főprofilt, ami az alapbeállításokat tartalmazza kívánjuk alkalmazni.

Az identitásmenedzsmentnek figyelnie kell arra, hogy milyen oldalra navigálunk és ha az adott oldal url-jét megtalálja az identitásokat tartalmazó adatbázisban, akkor arra kel váltania. Ha több helyen is megtalálja, akkor fel kell ajánlania a választási lehetőségeket.

Elképzelhetőnek tartom olyan lehetőség leprogramozását is, hogy amennyiben az adott oldalhoz nincs profil, de az url-ben megtalál bizonyos kulcsszavakat (mail, forum, shop, stb), akkor az adott kulcsszóhoz tartozó sémát, illetve a sémából való profil létrehozását felajánlja.

4.1.4 Kényelmi funkciók kialakítása

Az alábbiakban azok a kényelmi funkciók kerülnek bemutatásra, melyek nem szükségesek az identitásmenedzsment funkciók megvalósításához, azonban azokat jól kiegészítik és az implementálandó rendszer általuk, sokkal inkább felhasználóbarát lehet.

4.1.4.1 Regisztráció segítése

Ennek a funkciónak az a szerepe, hogy az új regisztrációkat, illetve a hozzájuk tartozó új identitásokat összekösse, egyrészt kényelmesebbé tegye a regisztrációt, másrészt pedig segítse az identitás és a regisztráció adatainak konzisztenciáját.

Ha egy új oldalra lépünk, és létre akarunk hozni egy új identitást hozzá, például egy regisztráció céljából, akkor a következő a teendő. Az új profilunkat a megfelelő sémából létrehozzuk, tehát ha egy levelezőoldalon kívánunk regisztrálni, akkor a levelezőoldalakhoz készített sémából hozunk létre új alprofilt. A felhasználó joga eldönteni, hogy az űrlap melyik mezőjébe a profil mely adatát másolja be. Tehát miután létrehoztuk a profilt a megfelelő adatokkal, utána tudjuk felhasználni a regisztrációhoz, illetve az oldalon való további böngészés során a biztonsági beállításokat.

Ezeket az alprofilokat is választhatjuk regisztrációkor, vagy amikor egy új identitást szeretnénk létrehozni, azonban ekkor csak egy új honlapcímet adhatunk hozzá, nem módosíthatjuk a beállításokat. Ha mégis megpróbáljuk, akkor az alkalmazás felajánlja, hogy szeretnénk-e egy új alprofilt, identitást létrehozni, az eredeti alprofilhoz tartozó szülősémából.

4.1.4.2 E-mail címek generálása

Az e-mail címek generálásának az elsődleges célja az eldobható e-mail címek létrehozásának könnyítése. Erre fórumozásnál lehet leginkább szükség. Mi szokott történni egyébként, ha foglalt egy cím? Általában felajánl a regisztrációs oldal pár lehetőséget. Ebből kézzel kell választanunk. Egyébként fórumozásra szolgáló sémáknál lesz olyan lehetőség, hogy megadhatunk pár kifejezést amiből generálhatunk e-mail címeket véletlenszerűen. Például megadunk 3 szót (sanyi, sanya, sanyo) 3 kiszolgálót (sanyi.hu, sanyo.com, sanya.net) ezeket kombinálja úgy, hogy véletlenszerűen rak az első rész elé vagy után egy számot 0 és 9999 között. Ha ez foglalt, akkor újra próbálkozik. Ha 10 (állítható legyen) próbálkozás után sem sikerült megfelelő e-mail címet találni, akkor kézzel kell megadnunk a címet. Beállíthatjuk úgy is például, hogy a kiszolgálót ne generálja, csak azt az azonosítót, ami előtte van (ez jól jön a levelezőrendszerek esetén például).

Az egy oldalhoz használt címeket el kell tárolni egy táblázatban. Ez teszi lehetővé, hogy azt is beállíthassuk, hogy egy e-mail címet felhasználhatunk-e többször egy fórum oldalon. Természetesen bármikor saját magunk is megadhatunk címet. A táblázatban annak is szerepelnie kell (azon kívül, hogy milyen e-mail címet hol használtunk fel), hogy mikor használtuk az adott címet.

4.1.4.3 Karbantartásban segítség

Ennek a funkciónak a lényege, hogy segítséget nyújtson a felhasználóknak az egyes identitások, illetve a hozzájuk tartozó regisztrációkhoz tartozó adatok menedzselésében.

A profilokhoz alapvetően kétféle adat tartozik. Egyrészt az oldalnak adható adatok, a regisztrációhoz szükséges adatok, másrészt pedig a böngésző biztonsági beállításait vezérlő adatok.

Ha a biztonsági beállításokat frissítjük és a frissítéseket jóváhagyjuk, akkor annak azonnal látható hatása lesz.

Ezzel ellentétben, ha például egy levelezőrendszerben regisztráció során megadott adatokat kívánjuk módosítani, akkor annak nem lesz azonnal effektív hatása. Ezt az alkalmazás nem tudja maga megtenni. Azt lehet csinálni, hogy minden egyes megnyitásakor a böngészőnek kapunk egy figyelmeztetést, hogy mely oldalakon kell elvégeznünk változtatásokat. Ha ráklikkelünk az oldal nevére, akkor az bejön, másrészt pedig felugrik egy ablak, hogy miket is kell módosítanunk.

Ha nem élünk ezzel, akkor abban az esetben fogunk figyelmeztetést kapni, amikor ellátogatunk az adott oldalra. Ha elvégeztük az adott karbantartást, akkor rákattintunk, hogy elvégeztük és többet nem jelenik meg a figyelmeztetés. Ezen túlmenően a megfelelő oldalon a beállításoknál kérhetjük az elmentett adatok bemásolását (mint amikor regisztráltunk), hogy megőrizzük az adatok konzisztenciáját.

4.2 Biztonsági beállítások

4.2.1 A sütimenedzsment lehetőségei

A sütimenedzsment modulon keresztül tudjuk az adott profilhoz beállítani, hogy milyen biztonsági beállításokkal kívánunk élni a sütiket illetően.

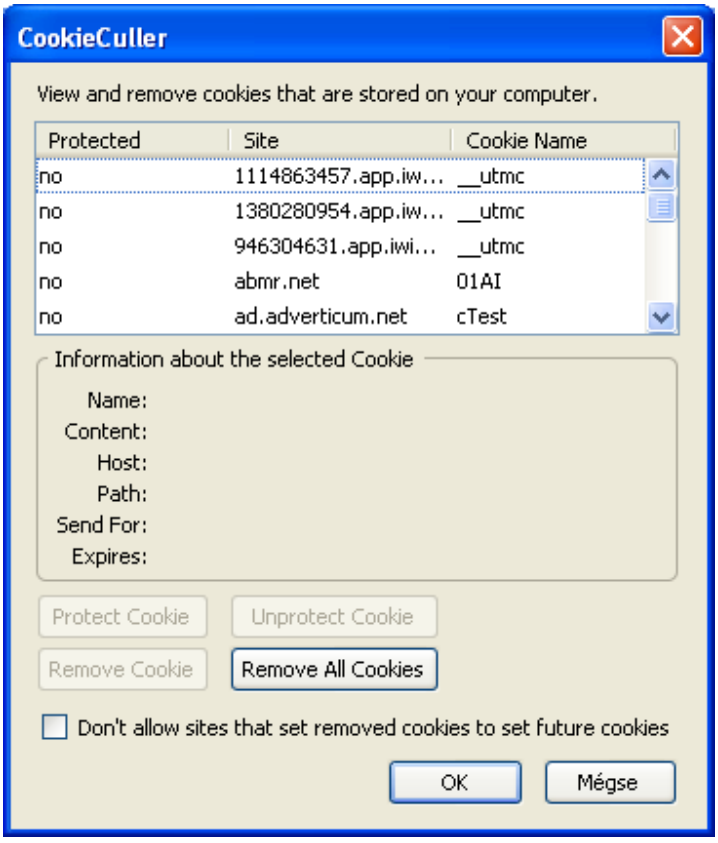
A tervezés során figyelembe vettem az előző fejezetben meghatározott követelményeket, illetve a más megvizsgált Firefox kiterjesztések által kínált lehetőségeket, más kiterjesztések kialakítását.

Az implementálásra kerülő modulon belül lehetőséget biztosítunk az összes süti közötti keresésre. Ezt úgy tehetjük meg, hogy begépelünk egy keresőszót, ami alapján az

alkalmazás kilistázza azon sütiket melyek nevében, tartalmában, domainjében szerepel a keresett szó.

Lehetőséget ad a modul a listázott sütik eltávolítására, akár egyenként, akár egyszerre, de ugyanakkor sütik manuális hozzáadása is lehetséges a rendszerhez.

A kilistázott sütik adatait megtekinthetjük és szerkeszthetjük is. Az implementálás során úgy kell megvalósítani a modult, hogy a szerkesztés során a leghangsúlyosabb szerepe a lejárati idő módosíthatóságának kell lennie. Ebből a célból megadhatunk konkrét időpontot, megadhatjuk, hogy a pillanatnyi időponttól kezdve meddig legyen érvényes a süti, illetve megadhatjuk hogy az adott session végén, vagy a böngésző becsukásakor kerüljön törlésre.



6. ábra CookieCuller kezelőfelülete

4.2.2 Veszélyforrások blokkolása

A különböző veszélyforrások blokkolásának megtervezésekor fontos szempontként szerepelt, hogy a meghatározott kritériumokat úgy teljesítsék az implementálásra kerülő modulok, hogy azok felépítése a lehető leghasonlóbb legyen. Ez két okból is igen fontos, egyrészt hasonló felépítésű felületek által könnyebben kezelhetővé válik a rendszer, másrészt pedig az implementálása és a későbbi fejlesztések is egyszerűbbé válnak

Ebből és egy másik tervezési szempontból kifolyólag a különböző blokkolók tervezését egy fejezetben tárgyalom. Ez a másik ok pedig az, hogy a veszélyforrások szűrése is nagyon hasonló módon történik, a programozás is hasonló módon zajlik.

De először tekintsük is át újra, hogy milyen elemeket szeretnénk szűrni. Egyrészt célunk, hogy az URL referer módosítható legyen, ez kivétel a programozás szempontjából, azonban a következők esetén ugyanazzal a módszerrel élhetünk:

- Java alkalmazások szűrése
- JavaScriptek szűrése
- Flashek szűrése

A módszerünk pedig az lesz, hogy a betöltött oldal HTML kódjában a megfelelő tagekre szűrünk és ezeken hajtunk végre akciókat.

Ezeket az akciókat pedig a kezelőfelületen keresztül állíthatjuk be. Minden egyes identitáshoz beállíthatjuk, hogy mely oldalhoz tartozó elemeket blokkolja, vagy pedig éppenséggel engedélyezze a böngészőnk letöltődni a számítógépünkre.

Emellett nem csak konkrét oldalakhoz adhatjuk meg a kívánt beállítást, hanem profilonként beállíthatunk egy default értéket, hogy a listánkon nem szereplő honlapok esetén milyen eljárást kövessen a böngésző.

4.2.3 Beállítások profilonkénti mentése

A böngésző mindenféle módosítás, kiterjesztés nélkül rendelkezik bizonyos biztonsági beállításokkal. Az identitásmenedzsmentet kezelő alkalmazást úgy kell megvalósítani,

hogy a biztonsági beállításokat személyre lehessen szabni, minden egyes paramétert profilonként lehessen szabályozni és tárolni.

4.3 Az anonimitás megőrzése

Ez a legnagyobb kérdés. Mitől maradunk anonimek [12]? Egyrészt azért leszünk azok, mert használhatunk anonimizáló hálózatot [23]. De ez mit sem ér, ha felfedjük az adatainkat. Az anonimitást az alkalmazás nem tudja folyamatosan garantálni, csak azt, hogy megfelelő használat esetén anonimek maradhatunk. Mivel az IP-címünket nem fedjük fel, a többi tőlünk függ. Tőlünk függ, hogy a sütikezelés során milyen beállításokkal élünk, tőlünk függ, hogy a különböző adatainkat megadjuk-e vagy sem, illetve mennyit adunk meg közülük. Az álcímek, illetve a pszeudonimek használata garantálja, hogy nem kell kiadni feleslegesen adatainkat, még akkor sem, ha mindenképpen meg kell adnunk például egy e-mail címet. Mindenhol lehetővé kell tenni az ideiglenes címek megadását, és bár azt nem fogjuk elérni, hogy például egy elektronikus bolt esetében automatikusan létrehozzunk egy új profilt, de az új profilhoz tarozó ideiglenes cím létrehozásában tudunk segíteni.

A biztonsági beállítások profilonként való menthetősége, illetve ezen kényelminek tűnő lehetőségek nagyban javítják az esélyt arra, hogy megőrizzük anonimitásunkat. Nem véletlenül fogalmaztam úgy, hogy kényelminek tűnő lehetőség. Ugyanis bár létrehozhatjuk manuálisan is az eldobható címeket, hiszen az alkalmazásunk ezt a folyamatot csak automatizálni próbálja, valószínűleg kevesen vennék a fáradtságot arra, hogy kézileg hozzanak létre folyamatosan eldobható postafiókokat. Ezáltal végső soron ez a kényelmi lehetőség is fontos biztonsági funkcióval bír.

5 Az implementált IDM rendszer bemutatása

A következő fejezetben részterületenként bemutatásra kerül az implementált rendszer, megmutatom milyen problémákba ütköztem a megvalósítás során, milyen alternatívák kerültek elő a megoldásra, mit választottam ezek közül és miért. Minden egyes megvalósított részterülethez, végül pedig általánosságban megfogalmazom továbbfejlesztési javaslataimat.

Az első döntést akkor kellett meghoznom, amikor eldöntöttem, hogy a Firefox egy kiegészítéseként készítem el az alkalmazást. Emellett szólt, hogy a Firefox kiegészítések megfelelő fájlkezelési struktúrát választva – melyet a későbbiek során bemutatok – platform-független működést garantálnak, valamint a fejlesztésre használt nyelv egyszerűsége. A Firefox kiegészítések készítésére egy saját programnyelvet (XUL) fejlesztettek ki, amely az XML-en alapul, és a megírt kódok funkcionalitásának bővítése érdekében ezekbe JavaScript kódok lehet ágyazni.

Azonban ami miatt egy a Firefoxtól elkülönülő alkalmazás implementálása reális alternatívaként fel sem vetődött, az volt, hogy egy Firefox kiterjesztés magában hordozza a böngészővel való egyszerű kommunikáció lehetőségét, nem kell irracionális energiákat fektetni eme kapcsolat megteremtésére.

5.1 Kezelőfelület

Elsőre azt szükséges tisztázni, hogy jelen esetben mit értek kezelőfelület alatt. A kérdés azért jogos, mert logikusan akár az összes menü, összes beállítási lehetőség kezelőfelületét érthetném ez alatt. Azonban jelenleg csak azt a részt kívánom bemutatni, csak azt a felületet, ahonnan elérem az egyes menüket, egyes beállítási lehetőségeket tartalmazó ablakokat, mivel azon felületek bemutatása, az azokhoz tartozó fejlesztési javaslatok jelentős mértékben összefüggenek az általuk megvalósított funkcionalitással.

5.1.1 Tervezési szempontok, szóba jött változatok

A kezelőfelület megtervezése során a fő szempontok a gyors elérhetőség, a praktikum és az esztétikum voltak. Bár több lehetőség is felvetődött, mégis a választás a végén ezen

szempontok alapján egyértelmű volt. Az alábbiakban röviden bemutatásra kerülnek a szóba jött lehetőségek.

IDM Eszköztár

Első lehetőségként adódott egy saját eszköztár létrehozása az IDM kiterjesztés számára, melyen megtalálható lehetne minden fontosabb menühöz egy saját gomb, egy kattintással elérhető lenne minden blokkoló, illetve lehetőség lenne azonnali identitásváltásra, és lenne hely visszajelzések számára is (7. ábra IDM eszköztár)**.** A saját eszköztár egy előnye, hogy be lehet állítani, mely oldalán helyezkedjen el a böngészőnek.

Előnyök	**Hátrányok**
Mindig jól látható	Nagy felületet foglal el
Állítható az elhelyezése	Nem kellően esztétikus

7. ábra IDM eszköztár

Gomb a Firefox eszköztárán

A következő lehetőség szintén az eszköztárakra hagyatkozik, azonban ebben az esetben nem tartozna saját eszköztár hozzá, mindösszesen a Firefox eszköztátán helyeznénk el egy ikont a számára.

Ebben az esetben bal egérgombbal az ikonra kattintva egy úgynevezett plain popup [24] menüből választhatnánk a lehetőségek közül.

Ennek a megoldásnak az előnye az esztétikus megjelenés, mindamellett, egy helyen jól össze vannak fogva a különböző lehetőségek.

Hátránya a saját eszköztárhoz képest, hogy nem elegendő egy kattintás a funkciók előhívásához, mivel azok egy popup menübe vannak összekötve. További probléma, hogy a Firefox eszköztárának zsúfoltsága miatt nincsen olyan szabad felületünk, ahol állandó visszajelzést adhatnánk az aktuális állapotokról.

Előnyök	**Hátrányok**
Esztétikus megjelenés	Nincs lehetőség visszajelzésre
Állítható az elhelyezése	A funkciók nem elérhetők egy gombnyomással

Elhelyezés a context menüben

Az alábbi lehetőség a context menüt[3] használja fel az IDM funkciók elérhetőségének biztosítására. A context menü alján kerülne elhelyezésre egy IDM menü, azon belülre lennének elhelyezve a szükséges funkciókhoz tartozó elemek.

Ennek a lehetőségnek az önmagában való használata több hátrányt jelente, mint előnyt.

Egyrészt ez a lehetőség egyáltalán nem biztosít a program számára visszajelzési lehetőséget. Másrész, ami még komolyabb probléma, mivel a context menü megjelenése függ a HTML, illetve XUL elemek programozott beállításaitól, ezért az állandó elérhetőség is csak nehézkesen lehetne megoldható.

Előnyök	**Hátrányok**
	Nincs lehetőség visszajelzésre
	A funkciók nem elérhetők egy gombnyomással
	Nem, vagy csak nehezen biztosított az állandó elérhetőség

Elhelyezés a statusbaron

Az utolsó számba vett lehetőség a statusbaron való elhelyezése a funkcióknak. Az előnyök és hátrányok tekintetében ez a lehetőség nagyon hasonlatos a második esethez, amikor a Firefox eszköztárán helyeztük el az IDM ikonját.

[3] A context menü a jobb egérgomb hatására felbukkanó popup menü

Ebben az esetben is az ikonra kattintva egy plain popup menüből választhatnánk ki a lehetőségeket.

Előnye a második megoldáshoz képest, hogy mivel a statusbar kevésbé zsúfolt és jellegéből adódóan alkalmasabb szöveges visszajelzésekre, ezért az aktuális állapotról is egyszerűen visszajelzést adhatunk a felhasználó számára.

Előnyök	**Hátrányok**
Esztétikus megjelenés	A funkciók nem elérhetők egy gombnyomással
Egyszerű visszajelzési lehetőség	

8. ábra Ikon és visszacsatolás a statusbaron

5.1.2 A lehetőségek kombinációja

A megvalósítás során a felsorolt lehetőségek közül több is kipróbálásra került, kihasználva a biztosított előnyöket. A fejlesztés során szinte végig az első lehetőséget használtam, azaz a képernyő alján elhelyeztem egy saját eszköztárat, ami az (7. ábra IDM eszköztár) képen látható. Ez a megoldás bár nem volt esztétikus, a tesztelés során nagy segítséget nyújtott mivel a böngésző indítása után azonnal elértem egy gombnyomással a kívánt funkciót.

A fejlesztés során tovább haladva, amikor már az esztétikai szempontok, a felhasználóbarát megjelenés is soron következtek, megfontolásra került a második, valamint a harmadik előző pontban bemutatott lehetőség is, azonban ezeket elvetettem.

A Firefox eszköztárán való elhelyezést azért vetettem el, mert a statusbaron való elhelyezéshez nagyon hasonló, viszont nem nyújt olyan kézenfekvő lehetőséget a visszajelzésre, mint a statusbar.

A context menüből való elérhetőség csak félig került elutasításra, jelenleg is használom, mint azt az 5.3.3. fejezetben bemutatom, azonban nem az IDM-hez tartozó menük, ablakok elérésére, abból az okból kifolyólag, hogy nem garantált az állandó elérése a context menünek.

A fentiekből következően végül a statusbar kínálta lehetőség mellett döntöttem, mivel ez biztosította számomra a lehető legtöbb előnyt.

A böngésző jobb alsó sarkában folyamatosan elérhető az alkalmazás ikonja, melyre kattintva választhatok a jelenleg elérhető funkciók közül

- Anonim mód
- Flash Blokkoló beálltásai
- Profilbeállítások

Az IDM ikonjától balra található címke pedig az aktuális állapot visszajelzésére szolgál, itt látható, hogy aktuálisan mely profil használjuk.

5.1.3 Értékelés

Az implementált kezelőfelület teljesíti az elvárt feltételeket, hiszen egyszerűen elérhetőek a különböző menüpontok, nem vesznek el nagy teret a rendelkezésre álló térből, valamint megfelelő a szükséges méretű visszajelzési felület is biztosított a szoftver számára.

5.1.4 Továbbfejlesztési javaslatok

Meglátásom szerint a kiterjesztés ezen része nem tartalmaz jelentős továbbfejlesztési lehetőségeket.

Természetesen a menü bővülni fog az implementált részegységekkel párhuzamosan, azonban ez nem tekinthető a kezelőfelület továbbfejlesztésének.

A tapasztalatotok tükrében el kell majd azon gondolkodni, hogy milyen adatokról lehet visszajelzésre szüksége a felhasználó számára, azonban a felületen várhatóan ez is cak mennyiségi, nem pedig minőségi értelemben fog módosítani

5.2 Egy kiválasztott modul: a Flash blokkoló beállításai

A következő fejezetben a flash animációk blokkolásáért felelős programrész kerül bemutatásra. Az előző fejezet mintájára megmutatom, hogy melyek voltak a fő tervezési szempontok, milyen problémákba ütköztem a megvalósítás során, illetve, hogy azokra milyen megoldások születtek. Végül megvizsgálom, hogy milyen irányba lehet továbbfejleszteni a blokkolót.

Amiért a címben is „egy kiválasztott modulról” írok, annak az az oka, hogy az implementálásra fordított idő racionális lekorlátozása végett a hasonló működésű blokkolók közül egyet valósítottam meg, ez pedig a Flash blokkoló.

5.2.1 Követelmények a blokkolóval szemben

A flash blokkoló helyes megtervezése, praktikus felhasználói felülettel való ellátása azért is kiemelt fontosságú, mert a jövőben a többi részegységnek is a lehetőségekhez mérten ugyanilyen küldőt szánok, illetve a fájlkezelést is úgy ismertem meg, hogy az egységesen alkalmas legyen a további bővítések kezelésére is.

Az egyszerű használhatóság alapkövetelménye mellett a következők voltak az alapvető szempontjaim a tervezés során. Egyrészt lehetőséget kell biztosítani a felhasználó számára, hogy minden egyes profilhoz külön tudja tárolni a beállításait, azok ne vesszenek el, mindig be tudja újra tölteni. Másrészt emellett mindenképpen szükségesnek tartottam, hogy az egyes profilokhoz létre lehessen hozni úgynevezett whitelisteket, a még nagyobb mérték testreszabhatóság érdekében.

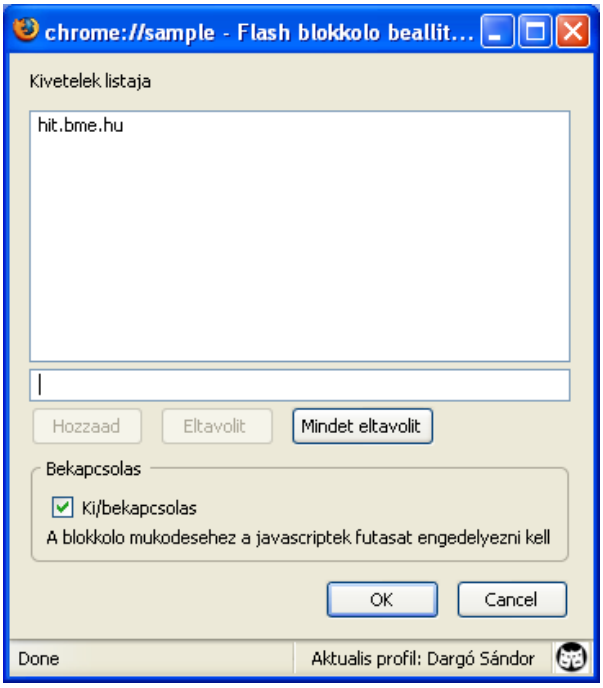
A whitelistek segítségével lehet azt beállítani, hogy bár a flash blokkoló aktív, bizonyos oldalakról az aktív blokkolás esetén is automatikusan letöltődjenek az animációk

5.2.2 A blokkolóhoz tartozó kezelőfelület

Amint az a 9. ábra Flash blokkoló is látszik a blokkoló beállításait tartalmazó, a beállítások módosítását lehetővé tevő ablak két fő részre bontható. Az ablak felső kétharmadában található részen található a whitelist, azaz a kivételek listája, illetve a hozzájuk tartozó kezelőgombok. Itt lehetőségünk nyílik a kivételeket felvenni, illetve azokat egyesével, vagy egyetlen gombnyomással mindet törölni.

Az ablak alsó harmadában pedig beállíthatjuk, hogy az éppen aktuális profil alkalmazása során akarjuk-e, hogy működjön a blokkoló. Fontos, hogy amennyiben a JavaScriptek futása nem engedélyezett, abban az esetben a blokkoló sem fog működni, hiszen az animációk szűrése is JavaScript kódok segítségével történik az adott oldal betöltése során.

Ez felveti azt az érdekes kérdést, hogy mi történik akkor, ha szeretném a JavaScript kódok és a flash animációk futását is blokkolni. Mivel jelenleg a JavaScript blokkoló még nincs implementálva, ezért ezzel a kérdéssel az 5.5.1 fejezetben foglalkozom bővebben.



9. ábra Flash blokkoló kezelőfelülete

5.2.3 Fájlkezelés

A megvalósítás egyik legkritikusabb eleme annak a megoldása, hogy a különböző profilokhoz egyenként lehessen a whitelisteket, illetve a ki/bekapcsolás tényét tárolni.

Ahhoz, hogy erre a felhasználónak lehetősége legyen, szükséges mindenképpen egyfajta fájlkezelést megvalósítani.

Amikor az implementálás kérdésig jutottam két lehetőséget vizsgáltam meg. Egyrészt lehetőségem van saját fájlok létrehozására, melyeknek én határozom meg nevét, kiterjesztését, helyét, másrészt pedig lehetőség nyílik a Firefoxhoz tartozó úgynevezett preferences (a továbbiakban PREF) fájlok létrehozására [25].

Mivel nekem egyszerű szövegfájlokra volt szükségem, amelyekben egymástól elválasztva eltárolhatom az kivételekhez tartozó címeket, ezért mindkét lehetőség alkalmas volt a számomra. Ami miatt végül a PREF fájlok használata mellett döntöttem, az volt, hogy ez a módszer biztosítja a platform függetlenséget [26].

Minden egyes profilhoz létrehozok egy PREF fájlt, melynek a neve két összetevőből áll össze:

- A név első fele: „flashblock”
- A második fele pedig az elsőtől ponttal (.) elválasztva a profil neve, amelyhez a kivételek listája tartozik

A név két részre bontása azért praktikus, mert minden egyes részegységnél ezt a fájlkezelési technikákat alkalmazom, és így egyszerűen tudok keresni az egyes beállításokra.

Az egyes PREF fájlokon belül egymástól pontosvesszővel (;) elválasztva tárolom az kivétellista egyes elemeit. Ez az elválasztás lehetővé teszi, hogy a fájl tartalmát egy string változóba beolvasva egyszerűen tömbösíteni tudjam.

5.2.4 A blokkoló működése

Amint azt már említettem a blokkoló csak akkor működőképes, amennyiben a JavaScriptek futása engedélyezve van a böngészőben.

Amennyiben ez a feltétel teljesül, a blokkoló működése a következő lépésekből áll össze:

- új oldal betöltésekor betöltődik a memóriába az adott profilhoz tartozó flash blokkoló beállítások, a kivételek listája egy tömbbe kerül
- amennyiben a flash blokkoló be van kapcsolva, az lekérdezi az éppen betöltés alatt álló oldal URL-jét
- megvizsgálásra kerül, hogy a kivétel lista egyes elemeit tartalmazza-e az adott oldal URL-je
    - amennyiben igen, az esetben az oldalhoz tartozó flashek nem töltődnek be
    - negatív keresési eredmény között betöltődik a flash

  Az 5.2.1 fejezetben meghatározott követelményekhez képest annyival tud többet a blokkoló, hogy amennyiben egy flash blokkolásra kerül, a helyére rákattintva betölthetjük az animációt.

5.2.5 Értékelés

Az 5.2.1 fejezetben támasztott követelményeknek maradéktalanul eleget tesz a blokkoló.

Az ismertetett fájlkezelés biztosítja a felhasználó számára, hogy minden egyes profilhoz eltárolhassa különböző beállításait, valamint a használt megoldás lehetővé teszi, hogy továbbfejlesztés esetén bonyolultabb módosítások nélkül lehessen eltárolni a meglévő szerkezetben az új opciókhoz tartozó értékeket.

A kivételek kezelése is megoldásra került, lehetősége nyílik a felhasználónak profilonként olyan listák megadására, amelyek tartalmazzák azon honlapokat, melyeken nem kell blokkolni a flash animációkat.

5.2.6 Továbbfejlesztési javaslatok

Továbbfejlesztési javaslatként elsősorban a felhasználó kényelmét segítő funkciókat tudok meghatározni, de mielőtt ezekre rátérnék, egy más jellegű javaslattal is élnék.

Jelenleg a bekapcsolt flash blokkolóhoz tartozik egy kivételeket tartalmazó whitelist, amikor úgy kell viselkedni, mintha kikapcsolt állapotban lenne. Ennek mintájára érdemes lehet létrehozni egy olyan listát, amire azokat az oldalakat vehetjük fel, amelyeket akkor is blokkolni szeretnénk, amennyiben a blokkolónk egyébként kikapcsolt állapotban van.

Egyéb a felhasználó kényelmét javító javaslatok:

- A blokkolt flashek feltűntetésének javítása. Ezalatt arra gondolok, hogy érdemes a blokkolt elemek helyén megjeleníteni egy feliratot, képet, ikont, hogy feltűnően jelezve legyen, hova kell kattintanunk, amennyiben mégis a flash letöltése mellett döntünk.
- A context menü bővítése olyan funkcióval, hogy egy nem blokkolt oldalt törölhessünk a whitelistről, vagy a későbbiekben egy blokkolt oldalt törölhessünk a mindenképpen blokkolni kívánt oldalak listájáról

5.3 Profilbeállítások

A következő fejezetben a profilbeállítások menedzselésére hivatott programrészt mutatom be. Az előző fejezetekhez hasonlóan megmutatom, hogy melyek voltak a fő tervezési szempontok, milyen problémákba ütköztem a megvalósítás során, illetve, hogy azokra milyen megoldások születtek. Végül megvizsgálom, hogy milyen irányba lehet továbbfejleszteni a profilmenedzsert.

5.3.1 Tervezés

A profilmenedzsmenttel foglalkozó programrész a kiterjesztés központi eleme, mondhatni lelke, jó kezelhetősége és működése kritikus az egész rendszer működésére nézve. Éppen ezért rendkívül fontos volt, hogy jól átgondolt legyenek azok a funkciók, az a felület, amelyek a felhasználó rendelkezésére fognak állni.

A profilmenedzsmentet 3 jól elkülöníthető részre osztottam:

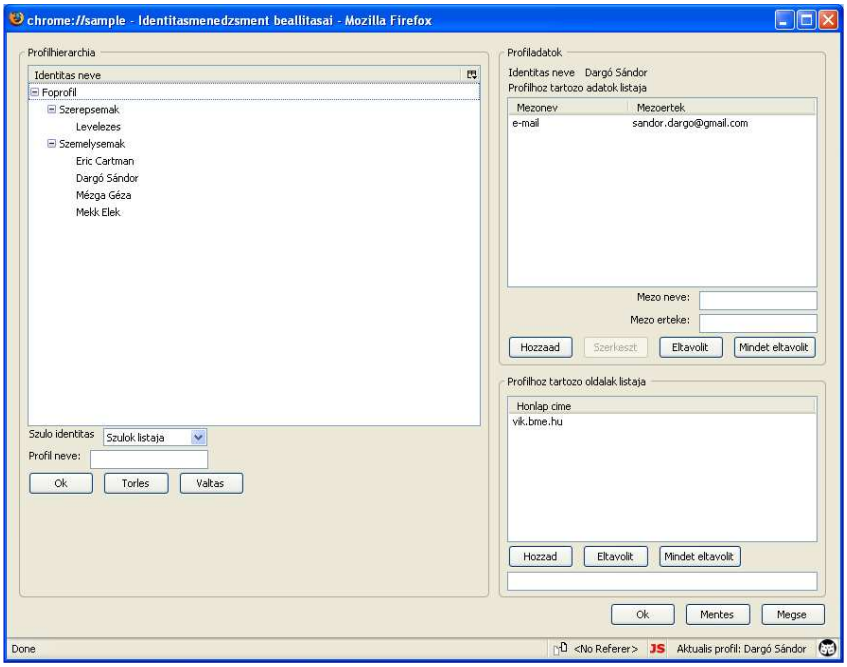
Egyrészt szükséges egy rész, amelyen keresztül magukat a profilokat hozhatjuk létre, azok struktúráját tekinthetjük meg.

Másrészt szükség van egy olyan felületre, ahol a profilokhoz különböző mezőket vehetünk el, és azokhoz a mezőkhöz értékeket rendelhetünk.

Harmadrészt szükséges egy olyan felület is, ahol az egyes profilokhoz honlapokat rendelhetünk hozzá. Ezeknek a hozzárendeléseknek az identitásváltásoknál lesz meghatározó szerepe, erről részletesebben az 5.3.4 fejezetben szólok.

5.3.2 Profilhierarchia

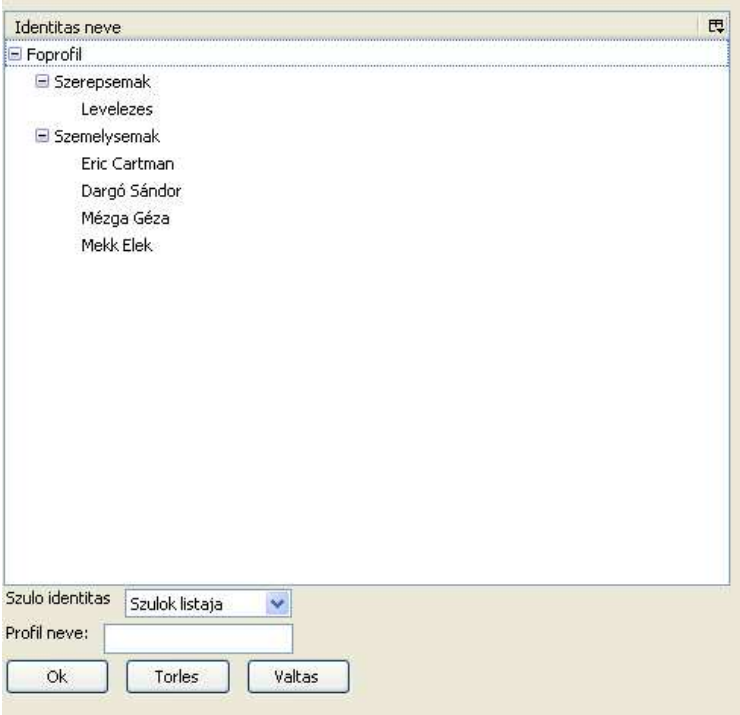
Az előzőekben meghatározott 3 rész alapján a kezelőfelület is 3 jól elkülöníthető részre bonthatjuk, mint az a 10**.** képen látható. Baloldalon haladva láthatjuk a profilhierarchiáért, jobb-felül a profiladatokért, míg a jobb-alsó részen a profilhoz tartozó adatokért felelős részeket. Ezek közül most az első, azaz a profilhierarchiát ábrázoló rész bemutatása következik.



10. ábra Profilmenedzser kezelőfület

Kezelőfelület

A 11. képen jól látható, hogy a profilhierarchiáért felelős rész két részre bontható. A felső részen a profilok kerülnek ábrázolásra fastruktúrában, míg alatta új profilok felvételére, meglévők törlésére, illetve a közöttük történő váltásokra nyílik lehetősége a felhasználónak.



11. ábra Profilhierarchia ábrázolása

Működés

Az 4.1.14.1 fejezetben leírtak kerültek követésre, amikor az 11. képen is látható profilhierarchia kialakításra került. Van egy főprofil, ebből származnak szerepsémák és személysémák, melyek alá profilokat hozhatunk létre.

Jelenleg a profilok létrehozása viszonylag kötetlen, mivel csak egy kikötés van, hogy kötelező szülő identitást megadnunk új profil létrehozása során, azaz mindenképpen csak a meglévő Főprofil alá hozhatunk létre újakat, azonban már az első szinten létrehozhatunk új profilokat a szerep- és a személysémák mellé, illetve végtelen mély lehet a fa mélysége, azonban ezt korlátozni, könnyű feladat.

A profilok törölhetősége ennél jobban korlátozva van, a Főprofilt, Személysémákat, Szerepsémákat nem törölhetjük ki, ha mégis megpróbáljuk, hibaüzenetet kapunk.

A szülő identitások listája minden profil felvételekor, vagy törlésekor automatikusan frissítésre kerül.

Fontos megjegyezni, hogy bár a fastruktúrából ha kiválasztunk egy profilt és a váltásra kattintunk, akkor a Profilmenedzser kezelőfelületének másik jobb oldalán látható másik két szövegdobozban betöltésre és megjelenítésre kerülnek a kiválasztott profilhoz tartozó adatok, azonban a tényleges profilváltáshoz a jobb alsó sarokban található Ok, vagy Mentés gombra kell előbb kattintanunk.

Fájlkezelés

A fájlkezelés a már az 5.2.3 fejezetben megismertekhez hasonlóan működik. A hierarchia mentéséhez a Firefox PREF fájljait használom. A használt PREF fájl neve `profiles.profilelist`. Mivel a hierarchia eltárolásához összesen egy fájlra van szükségünk, ezért itt nem kell több fájlt nyilvántartani, a két részből összeálló nevet azért választottam, hogy minden programrésznél azonos legyen a kezelt fájlok nevének felépítése.

A lista egyes elemei itt is pontosvesszővel vannak egymástól elkülönítve, azonban itt még egy bontás bevezetésre került. A pontosvesszővel határolt minden egyes elem – az elsőt kivéve, amiben az utolsó kilépéskor aktív profil neve van elmentve – két részre van osztva, melyek vesszővel (,) elkülönítve egymástól. Az első rész tartalmazza a profil nevét, a második pedig a szülő profil nevét, ezen információk alapján pedig már felépíthető a szükséges struktúra

Értékelés

A megvalósított profilhierarchia teljesíti azon elvárásokat, hogy a profilokat egy hierarchikus struktúrába szervezzük és ábrázoljuk, illetve azt az elvárást is, hogy a felhasználó létrehozhasson új profilokat és törölhesse a meglévőket.

Egy elvárás van azonban, melyet jelen pillanatban nem teljesít a jelenlegi hierarchia. Nem ad lehetőséget tranzakciós táblák létrehozására és kezelésére, amiben az 4.1.1

fejezetben esett szó először. Ennek a hiányosságnak a kijavítására a továbbfejlesztési javaslatok között teszek kínálok megoldást.

Továbbfejlesztési javaslatok

Az első és legfontosabb lépésnek a tranzakciós táblák elkészítésének kell lennie. Ennek segítségével megoldhatóvá válik majd, hogy például fórumozás során egy adott szobában ne használjuk kétszer ugyanazt a becenevet.

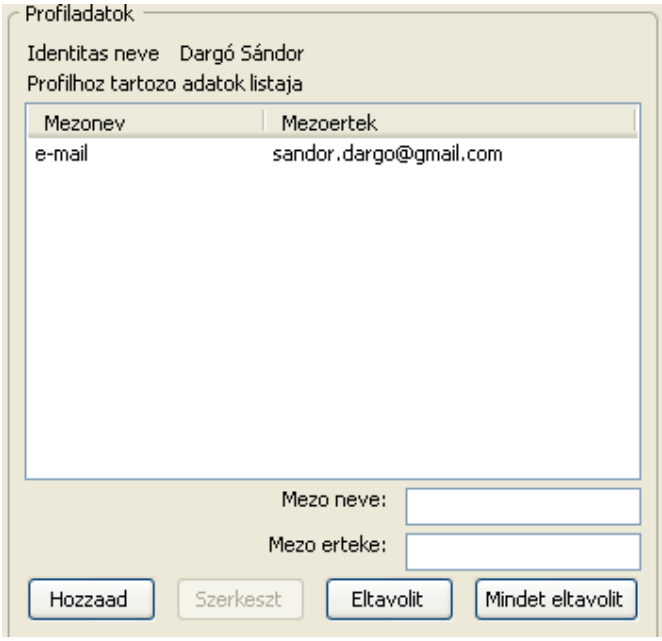
Amikor a tranzakciós táblák létrehozásra kerültek, akkor érdemes lesz a profilok létrehozását korlátozni, hogy ne lehessen akármilyen mélységig bővíteni a profilokat tartalmazó fát.

A felhasználó kényelmét segítő továbbfejlesztési lépés lehet, hogy ha a profilfa egy elemét kiválasztjuk, akkor a szülő identitás lista aktív eleme is eszerint frissüljön. Így azt új profil felvételekor nem kell külön kiválasztania a felhasználónak, hanem elég lenne, ha a hierarchia megfelelő elemére kattint.

5.3.3 Profiladatok

Kezelőfelület

A profiladatok menedzselésre szánt felület, mint látható 3 részre osztható. Legfelül láthatjuk annak a profilnak a nevét, amelynek az adatai listázásra kerülnek lejjebb, mégpedig két oszlopban. Az első oszlopban láthatjuk a mező nevét, a másodikban pedig a hozzátartozó értéket. A lista alatt találhatjuk azt a részt, amelynek a segítségével lehetőségünk nyílik új adatok, új mezők felvételére, illetve meglévők törlésére.

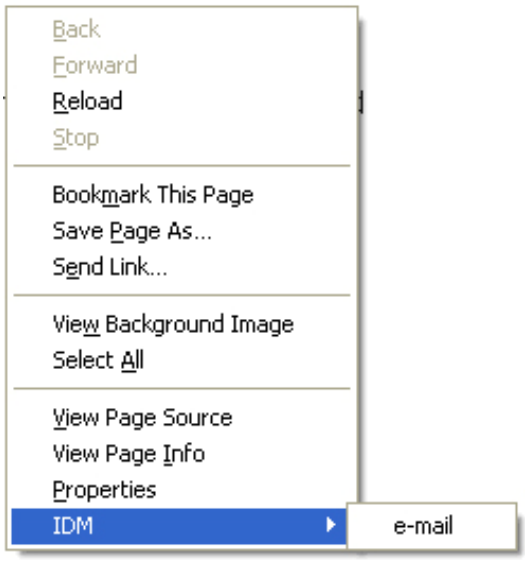


12. ábra Profiladatok felvétele

Működés

Ami a működés kérdését illeti, a kérdés, amire válaszolni kell, az az, hogy hol használhatók fel ezek az adatok. Amint azt a 4.1.4.1 fejezetben korábban leírtam, e profilokhoz tartozó adatok segítségével szeretném segíteni a regisztrációs folyamatokat, melynek egy kezdetleges változata már most is működik.

Jelenleg ezek az adatok a context menü segítségével érhetők el. Legutolsó menüpontként létrehoztam egy IDM menüt, ezen belül mindig az aktuálisan aktív profilhoz tartozó adatmezők kerülnek felsorolásra, amint az a 13. képen is látható. Amennyiben kiválasztom valamelyik adatmezőt, úgy az ahhoz tartozó érték a vágólapra kerül, ahonnét azt egyszerűen be tudom másolni a szükséges mezőbe.



13. ábra Profiladatok és a context-menü

Az egyes mezők és hozzájuk tartozó értékek öröklődnek a profilok között. Amikor egy új profilt hozunk létre, akkor a létrehozott profil megkapja a közvetlen szülőjéhez tartozó összes adatot. Ez a módszer garantálja, hogy a hierarchia létrehozása során minden egyes leszármazott rendelkezni fog az ősei összes adatmezőjével, ugyanakkor nem okoz problémát, hogy az egyes mezőkhöz milyen érték öröklődjön, mert az mindig a közvetlen szülő által kerül átadásra. A módszer hátránya, hogy amennyiben a közvetlen szülő profilból törlünk egy mezőt, akkor azt nem örökölheti meg a leszármazott, hibába szerepel az továbbra is a szülő szülőjénél. Véleményem szerint, ez nem okoz komoly problémát, ugyanis feltételezhetjük, szándékosan került a mező törlésre a szülőprofilnál, így annak nem öröklődése lehet akár kívánatos is.

Fájlkezelés

A fájlkezelés a már az 5.2.3 fejezetben megismertekhez hasonlóan működik. A hierarchia mentéséhez a Firefox PREF fájljait használom. A használt PREF fájlok nevei két részből épülnek fel:

- A név első fele: „`fieldlist`”
- A második fele pedig az elsőtől ponttal (.) elválasztva a profil neve, amelyhez az adatok listája tartozik

A lista egyes elemei itt is pontosvesszővel vannak egymástól elkülönítve, azonban a profilhierarchiát tároló fájlhoz hasonlóan itt is még egy bontás bevezetésre került. A pontosvesszővel határolt minden egyes elem – az elsőt kivéve, amiben az utolsó kilépéskor aktív profil neve van elmentve – két részre van osztva, melyek vesszővel (,) elkülönítve egymástól. Az első rész tartalmazza az adatmező nevét, a második pedig a második pedig az adatmezőhöz tartozó értéket (például: e-mail,sandordargo@gmail.com).

Értékelés

Az implementáltak teljesítik az elvárásokat, hiszen lehetősége nyílik a felhasználónak, hogy minden egyes profilhoz felvehessen korlátlan számú mezőt, melyekhez megadhatja az értékeket. Ezeket a mezőket, mint láthattuk, a későbbiekben törölheti, újakat hozhat létre.

Mint az bemutatásra került, a megadott adatokat a böngészésünk során, egyszerűen pár kattintással, bármikor felhasználhatjuk, a további egyszerűsítésre, pedig a továbbfejlesztési javaslatok között kínálok megoldást.

Továbbfejlesztési javaslatok

Jelenleg a mezőket nem lehet szerkeszteni, bár az ehhez szükséges gomb inaktív állapotban már megtalálható a kezelőfelületen, így ezt tervezem következő lépésként megvalósítani.

A további javaslatok inkább az adatok felhasználásához kapcsolódnak.

Lehetővé kívánom tenni, hogy az egyes mezőkhöz tartozó adatok ne csak a vágólapra kerülnek rá, hanem a fókuszált szövegdoboz tartalmát is írják felül. Azonban a vágólapos megoldást is szükségesnek tartom, mivel vannak esetek, hogy a context menü le van tiltva bizonyos helyekről, azonban a vágólap segítségével az adatokat így is elérhetjük.

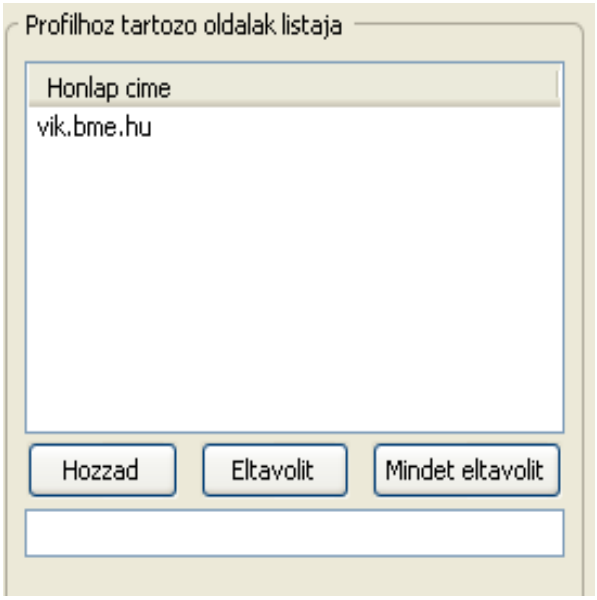
További lehetőségként felvetődik az űrlapok kitöltésének automatizálása, az azonosítható szövegdobozok tartalma egy gombnyomással kitölthető lenne, amennyiben tartoznak megfelelő adatok az aktuális profilhoz.

Hasznos volna, hogy ha éppen manuálisan töltünk ki egy űrlapot, és úgy gondoljuk, hogy egy adat hasznos lenne, ha a profilhoz tartozóan eltárolódna, akkor azt egy kattintással felvehessük.

5.3.4 Profilhoz tartozó oldalak

Kezelőfelület

A 14. képen látható, a profilhoz tartozó honlapok karbantartására felület két részre különül. Felül láthatjuk listázva azon honlapokat, melyek az aktuális profilhoz lettek hozzárendelve, alatta pedig lehetőségünk nyílik felvenni további, illetve törölni meglévő honlapokat.



14. ábra A profilhoz tartozó honlapok

Működés

A profilhoz tartozó adatokhoz hasonlóan (5.3.3) itt is azt a kérdést kell először megválaszolni, hogy mikor van szerepük ezen adatoknak.

A honlapcímeknek az profilváltások során van szerepük, mely váltások jelenleg manuálisan is működhetnek, azonban emellett automatikusan is történhetnek váltások.

Bármely oldal betöltésekor meghívásra kerül egy olyan függvény, amely megvizsgálja, hogy valamelyik profilhoz tartozó listában megtalálható-e a betöltendő oldal URL-je.

Pontosabban azt vizsgálja, hogy van-e olyan bejegyzése az egyes listáknak, mely bejegyzést tartalmaz a betöltendő oldal URL-je. Ez a gyakorlatban azt jelenti, hogy amennyiben van egy „index.hu” bejegyzésünk, de mi a http://index.hu/tech/ oldalt töltjük be, a keresés eredménye ekkor is pozitív lesz.

Amennyiben a keresés eredménye pozitív és az egyezés nem aktuális profilban található meg, erről a felhasználó értesítést kap és az identitásváltás megtörténik.

Felmerül az a kérdés, hogy mi történik akkor, ha egy oldal több listához is fel van véve.

Hogy erre választ kapjunk, meg kell nézni, hogy milyen sorrendben történik a honlapok a listákban való keresés. A keresés a profiles.profilelist PREF fájl tartalma szerint történik, azaz, először az aktuális profilhoz tartozó listában keresünk, majd a listában található többi profil következik. Jelenleg az első találathoz tartozó profil kerül betöltésre, nincs lehetőség választásra többszörös találat esetén. Ennek a megoldására a továbbfejlesztési javaslatok között teszek javaslatot.

Fájlkezelés

Az adatok tárolása a már korábban megismertekhez hasonlóan működik. Minden egyes profilhoz létrehozok egy PREF fájlt, melynek a neve két összetevőből áll össze:

- A név első fele: „whitelist”
- A második fele pedig az elsőtől ponttal (.) elválasztva a profil neve, amelyhez a kivételek listája tartozik

Az egyes PREF fájlokon belül egymástól pontosvesszővel (;) elválasztva tárolom a profilhoz tartozó egyes honlapokat.

Értékelés

A megvalósított profilváltási lehetőségek megvalósítják a meghatározott alapvető igényeket, miszerint lehetősége nyílik a felhasználónak manuálisan történő profilváltásokra, illetve lehetőség nyílik arra is, hogy megadott honlap címeket figyeljen a kiterjesztés, és amennyiben azokat nyitjuk meg a böngészés során, automatikus profilváltás történik.

Tulajdonképpen az automatikus profilváltás ennél többet valósít meg. Mivel a váltás szükségességének megállapítása úgy történik, hogy vizsgálatra kerül, hogy az aktuálisan betöltendő honlap URL-je tartalmazza-e valamelyik bejegyzést, ezért illeszkedésvizsgálat valósul meg a gyakorlatban, azaz, lehetővé válik, azon honlapok hozzárendelése egy profilhoz, melyek URL-je tartalmazza a „forum” kifejezést. A továbbfejlesztési lehetőségekről a következő bekezdésekben ejtek szór.

Továbbfejlesztési javaslatok

A továbbfejlesztési javaslatim az identitások közötti váltásokhoz köthetőek.

Első lépésben azt kívánom megvalósítani, hogy az automatikus identitásváltást ki/be lehessen kapcsolni.

A következő lépés annak a kezelése, amikor egy oldal több profilhoz is tartozik. Először is be kell tudni állítani, hogy többszörös találat esetén szeretnénk-e az eredeti gyakorlattal élni, miszerint az első találat esetén megtörténik a váltás, avagy mi szeretnénk eldönteni, hogy az összes találat közül melyik legyen az új profil.

Ebben az esetben oldalváltáskor egy listából kell tudnunk kiválasztani, hogy melyik profilt is választjuk.

Érdemes elgondolkodni azon a lehetőségen, hogy amennyiben egy oldal több listához is tartozik, beállíthassuk, hogy melyik legyen az alapértelmezett profil. Tehát nem nekünk kéne egy listából kiválasztani a következő profilt, nem is az első találat döntene, hanem az, hogy melyik profil az alapértelmezett egy adott honlaphoz.

Ezen kívül érdemes fontolóra venni reguláris kifejezések szerinti váltások fontolóra vételét. Ennek kapcsán két lehetőséget tudok elképzelni, az egyik jobb konfigurálhatóságot, a másik viszont inkább felhasználóbarát működést eredményez.

Mivel a kiterjesztés célja, hogy a privátszféra védelmét közelebb hozza az átlagos felhasználókhoz, ezért én itt csak a második lehetőségre térnék ki, amely számukra kedvez. Ez alapján egy a Mozilla Thunderbirdben és egyéb levelezőrendszerekben már korábban kialakított felület kerülne implementálsára, amely segítségével a felhasználó egyszerűen megadhatná preferenciát, például „Olyan honlapok esetén történjen váltás,

amelyek tartalmazzák X-t, de nem tartalmazzák Y-t. A megvalósítás részleteivel az a rendelkezésre álló időkeret racionális lekorlátozása miatt nem foglalkoztam.

5.4 Anonim mód

Az „anonim mód” bevezetése kettős célt szolgál a jelenlegi tesztrendszerben. Egyrészt megmutatom általa, hogy bár a saját biztonsági modul implementálása nem teljes, az elkészített kiterjesztés mégis tudja garantálni a szükséges anonimitást, másfelől pedig célja volt annak a bemutatása is, hogy lehetőség van más letöltött kiterjesztések vezérlésére a diplomaterv tárgyát képező identitásmenedzsment alkalmazás által.

5.4.1 Az „anonim mód” felépítése

Az előző mondatokkal már utaltam részben az anonim mód felépítésére, azaz a minél nagyobb fokú anonimitás megőrzésének érdekében alkalmaz más – nem saját fejlesztésű – kiterjesztéseket.

Így alkalmaz egy JavaScript blokkolót, egy URL Referer manipulátort, valamint egy a TOR használatát lehetővé tevő kiterjesztést.

5.4.2 Az „anonim mód” működése

Az anonim mód aktiválására a context menüpontból van lehetőségünk. Ekkor nem a felhasználó által definiált profilhierarchiában mozgunk, hanem abból, mintegy kilépve, egy úgynevezett „anonim profillal” dolgozunk.

Ennek a profilnak az a jelentősége, hogy ekkor a legmagasabb biztonsági beállításokkal dolgozunk. Tehát a Flash blokkoló tiltja az összes flash letöltését és nem használ whitelistet, valamint ugyanígy a JavaScript blokkoló is tiltja a kódok futását minden kivételre tekintet nélkül. Emellett, az URL Referer küldése törlésre kerül, így az előző általunk látogatott oldalra vonatkozó információt is titkoljuk.

Valamit emellett az anonim mód aktiválásakor az aktuális oldal újra betöltődik a TOR segítségével, azaz IP címünket is fedjük. Ezek után – kivéve ha minden honlap

betöltésekor újra aktiváljuk az anonim módot – a megnyitandó honlap címét a külön erre szolgáló TOR eszköztár szövegmezőjébe kell beírnunk.

Azért választott a saját eszköztárral rendelkező kiterjesztést, mivel több más kiterjesztéssel összehasonlítva ez kínálta a legmegbízhatóbb működést, és úgy döntöttem, hogy az egy címsort jelentő kényelemnél fontosabb a minél megbízhatóbb anonimitás.

5.5 Továbbfejlesztési javaslatok

Az alábbi fejezet tartalmazza a tesztrendszerre vonatkozó továbbfejlesztési javaslataimat. Először a privátszférát erősítő javaslatimat veszem sorra, utána pedig azokat, melyek az alkalmazás felhasználóbarát funkció számát növelhetik.

5.5.1 A privátszférát erősítő továbbfejlesztési javaslatok

Blokkolók

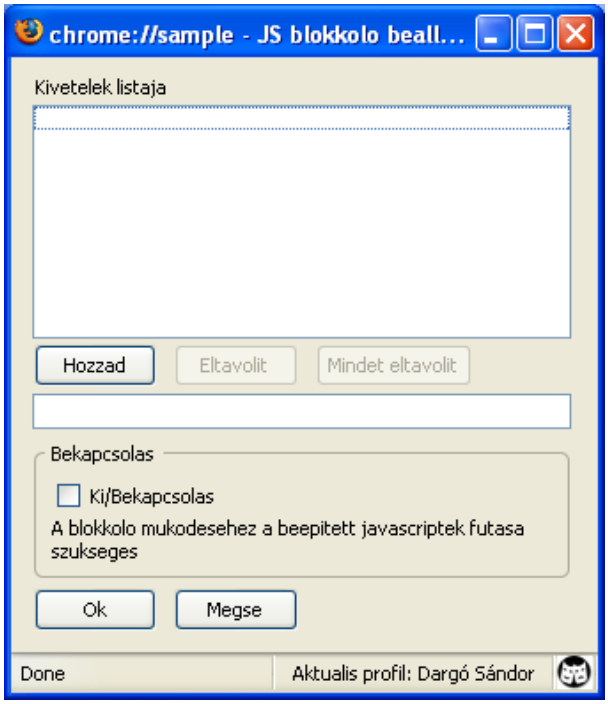
A többi blokkolót érintő felvetésemre talán nem is a továbbfejlesztési javaslat a legmegfelelőbb megfogalmazás. Ezen részelemekre mindenképpen szükség van, amennyiben a jövőben használni, fejleszteni kívánom a kiterjesztést, ezek a blokkolók elengedhetetlenek a teljes funkcionalitás biztosításához.

A következő blokkolókat tervezem a kiterjesztéshez csatolni:

- JavaScript kódoló
- JAVA alkalmazások blokkolója

Ezen blokkolók implementálása nagyon hasonló feladatot jelentenek a Flash blokkolóéhoz. Amint azt már a Flash blokkoló esetén az 5.2.2 fejezetben említettem, annak kezelőfelületét úgy terveztem meg, hogy alkalmas legyen más blokkoló modulok megjelenítésére is.

Ennek illusztrálásaként az 15. képen látható a JS blokkoló kezelőfelületének terve. Ha összehasonlítjuk a 9. képen látható Flash blokkolóval, akkor láthatjuk, hogy mindösszesen az ablak fejlécében olvasható feliratban található különbség.



15. ábra A JavaScript blokkoló kezelőfelületének terve

Ami a JAVA és JS blokkolók működését illeti, ugyanúgy fognak működni, mint a FLASH blokkoló, azzal a különbséggel, hogy más típusú elemeket fognak szűrni.

A fájlkezelésük is teljesen megegyező lesz a korábban ismertetekkel, a Firefox PREF fájljait fogják használni és két részből fognak összetevődni a fájlnevek:

- A név első fele: „java”, illetve „JavaScript”
- A második fele pedig mindkét esetben az elsőtől ponttal (.) elválasztva a profil neve, amelyhez a kivételek listája tartozik

Sütimenedzsment

A blokkolókhoz hasonlóan megvalósítandó egy a sütik menedzselésére szánt modul is. Ezen keresztül a felhasználónak alkalma nyílik profilokhoz kötni a letöltött sütiket, azaz identitás váltásakor a betöltött sütik is megváltoznak.

A sütimenedzser első változatának elkészítettem a kezelőfelületét, ami a 16. képen látható. A képről leolvashatjuk, hogy először lehetősége lesz a felhasználónak

- név szerint szűrni a sütiket,

- megtekinteni az egyes sütikhez tartozó adatok,
- a meglévő sütikhez újakat hozzáadni,
- a meglévő sütiket törölni,
- illetve szerkeszteni.

16. ábra A sütimenedzser kezelőfelületének terve

A beállítások tárolására a már korábban ismertetett fájlkezelési technikát kívánom alkalmazni, a Firefox PREF fájljaiban fogom tárolni a sütikhez tartozó adatokat és két részből fognak összetevődni a fájlnevek:

- A név első fele: „cookies”
- A második fele pedig mindkét esetben az elsőtől ponttal (.) elválasztva a profil neve, amelyhez a kivételek listája tartozik

A továbbfejlesztési javaslat továbbfejlesztési javaslataként olyan funkciót fogalmazok meg, amely egy új süti letöltődése előtt a felhasználó kezébe adja a döntés jogát, hogy akarja az adott sütit, természetesen a süti tulajdonságainak vizualizálása mellett.

Ezt a funkciót tudni kell ki és bekapcsolni, nem szabad elvenni a felhasználótól az automatizmus kényelmét, még akkor sem, ha ez kisebb biztonsággal jár bizonyos esetekben.

URL Manipulátor

Anonimitási szempontból indokolt az URL Referer kezelése, annak manipulálása, mivel az fontos információkat árulhat el böngészési szokásainkról, erről részletesebben már beszéltem az **Hiba! A hivatkozási forrás nem található.** fejezetben.

Alapvetően az URL Manipulátornak is meg kell tudnia valósítani a már korábban tárgyalt blokkolók feladatait, lehetőséget el adni az URL Referer engedélyezésére és tiltására profilonként, illetve kivételeket tartalmazó listák kezelésére.

Emellett azonban más lehetőségeket is kell adni a felhasználónak, ezen lehetőségek meghatározásakor felhasználtam az mozdev.org oldalon található hasonló funkcióval rendelkező kiterjesztések által felkínált funkciókat.

Ezek alapján biztosítani kell a felhasználók számára annak a lehetőségét, hogy az URL-nek mindig csak egy részét küldje el a böngészőnk, például, a Műegyetem bármely oldaláról is látogatunk egy másik (al)oldalra, csak a www.bme.hu URL kerüljön megküldésre.

Továbbá lehetőséget kell adni arra is, hogy a felhasználó „becsaphassa” a következő oldalt, amelyet meglátogat azzal, hogy egy a valóságostól eltérő Referert mutat a külvilág felé.
Ezeket a lehetőségeket nem elég a whitelist elemeinél szabályozni, hanem definiálni kell tudni az úgynevezett default beállításokat, melyek a whitelistben nem szereplő oldalakra lesznek igazak.

Természetesen ezen funkciókat egyenként kell tudni beállítani az egyes profilokhoz, egyenként kell tárolni a beállításokat a már ismertetett PREF fájlok segítségével

WEB poloskák szűrése

A WEB poloskák már bemutatásra kerültek az **Hiba! A hivatkozási forrás nem található.** fejezetben. Ezen elemek szűrésére az addons.mozdev.org honlapján megtalálható kiterjesztések közül kívánok választani körültekintő tesztelés után. Ennek működésébe mindössze annyira kívánok beleszólni, hogy alkalmazása ki- és bekapcsolható legyen, és ennek állapota profilonként kerüljön tárolásra.

Anonimizáló hálózat használata

Az 3.1 fejezetben ismertetett okokból kifolyólag anonimizáló hálózat használatát lehetővé tevő kiterjesztés csatolását javaslom a kiterjesztéshez. A webbugok szűréséhe hasonlóan ebben az esetben is már meglévő kiterjesztés használatát javaslom, aminek a működésére vonatkozó állapotát kívánom profilonként tárolhatóvá és szabályozhatóvá tenni.

Jelenleg több ezt a funkciót megvalósító kiterjesztés is letölthető az internetről, mielőtt a továbbfejlesztés során döntés születik valamelyik használatáról, kiterjedt tesztelésnek kell alávetni a lehetséges jelölteket, hogy a legmegbízhatóbb változat kerüljön beépítésre.

HTML Tagek szűrése

A HTML Tagek szűrése leginkább a blokkolók működéséhez hasonlít, azonban annál némileg bonyolultabb, nem is annyira a működése, mint inkább a konfigurációja.

Ezt arra alapozom, hogy véleményem szerint meggondolatlan dolog lenne egy honlap összes tagjét tiltani, – bár erre is lehetőséget kell adni a felhasználónak, de – nem elegendő ha csak azt lehet beállítani, hogy mely honlapok esetén menjen végbe a szűrés, hanem azt is meg kell tudni határozni, hogy milyen tagek, illetve a tegeken túl a HTML kódban milyen minták kerüljenek szűrésre. Tehát nem csak a tageket lehetne tiltani és szűrni, hanem az oldal HTML kódjában szereplő bármilyen kifejezést.

Emellett lehetőséget kell biztosítani alapbeállítások kezelésére is, tehát beállíthatónak kell lennie, hogy a kivételeket tartalmazó lista elemeit leszámítva mely tagekre menjen végbe automatikusan szűrés.

Ennél a modulnál egy az egyben nem alkalmazható a blokkolóknál megismert felület, implementálás esetén az átalakításra, kibővítésre szorul.

5.5.2 A felhasználó-barát működést segítő javaslatok

Közös honlap-menedzsment

A jövőben az egyszerűbb használhatóság érdekében érdemes kialakítani egy olyan funkciót, egy olyan elemet elhelyezni a context menüben, melynek segítségével egy adott oldal URL-je egyszerre hozzáadásra kerül az összes blokkol kivétellistájához.

Ennek a funkciónak akkor lesz komolyabb szerepe, amikor az 5.5.1 fejezetben leírt blokkolók is implementálásra kerülnek. Amennyiben ez a javaslat megvalósul megkímélhető a felhasználó attól, hogy egy honlaphoz tartozó URL-t egyesével kelljen felvennie, minden egyes blokkoló modulhoz.

„Anonim” mód

Az előző alfejezetben leírt biztonsági fejlesztések megvalósítása után szükségesnek látom egy olyan lehetőség implementálását, egy olyan elem elhelyezését a kiterjesztéshez kapcsolódó plain popup menüben, amelyre kattintva minden modul a lehető legnagyobb biztonsági szintre vált azonnal.

Az anonim mód némiképpen kilógna a jelenlegi profilhierarchiából, az „anonim profil” a fastruktúrában sem jelenne meg külön, illetve ahhoz mezőket, hozzá tartozó honlapokat sem lehetne hozzárendelni.

Az anonim mód kikapcsolása után böngészőnk az aktiválás előtti utoljára használt profilra váltással folytatná tovább a működését.

Nymity slider

Amennyiben a korábban felsorolt biztonsági és kényelmi funkciók már megvalósításra kerültek, érdemes végiggondolni egy olyan felület kialakításának lehetőségét, melyről a felhasználó visszajelzést kaphatna arról a biztonsági, anonimitási szintről mellyel az aktuális profiljával való böngészés során rendelkezik.

Ezen felület kialakításakor az 2.3.4 fejezetben ismertetett Goldberg-féle nyimty slidert, annak fokozatait javaslom alapul venni.

Az anonimitási szint meghatározásának folyamata túlmutat ezen munka keretein.

6 Az implementált tesztrendszer működése

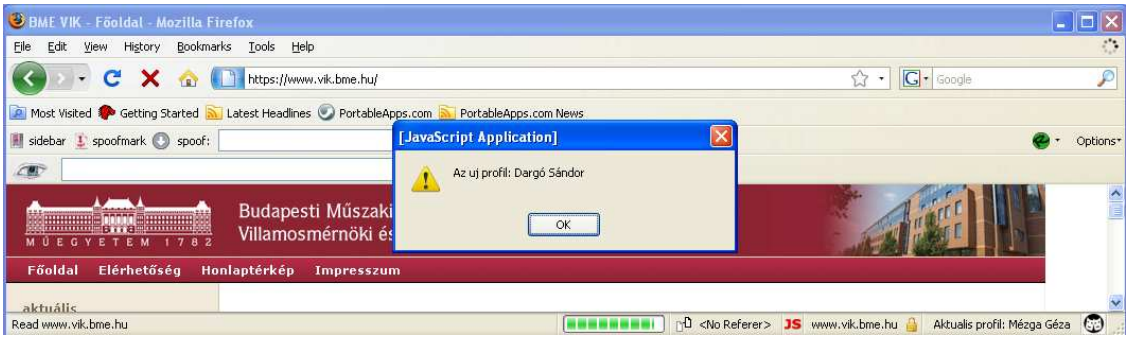
A tesztelés során kitértem az egyes biztonsági, illetve identitásmenedzsment funkciók egyenkénti kipróbálására, majd miután megbizonyosodtam ezek helyes működéséről, összetettebb eseteket is megvizsgáltam.

6.1 Profilváltások vizsgálata

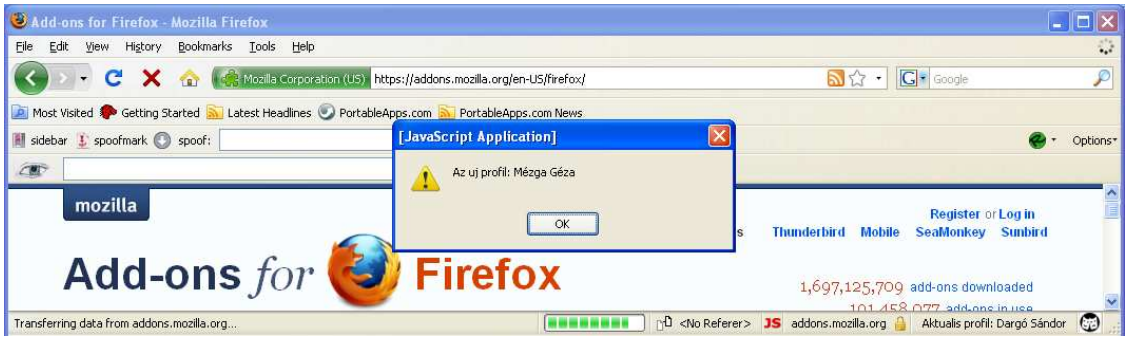
Először a profilváltások folyamatát teszteltem. A teszteléshez létrehoztam profilokat, melyekhez egy-egy különböző honlapot rendeltem hozzá első körben.

Profilnév	**Hozzárendelt honlap**
Dargó Sándor	vik.bme.hu
Mézga Géza	addons.mozilla.org
Mekk Elek	kth.bme.hu

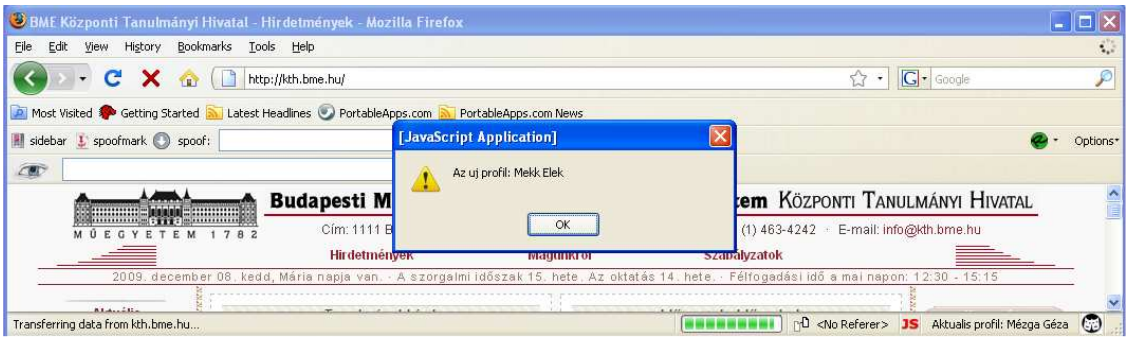
Ezek után kipróbáltam, hogy mi történik, amikor sorra látogatom az egyes oldalakat.



17. ábra Profilváltás 1.



18. ábra Profilváltás 2.

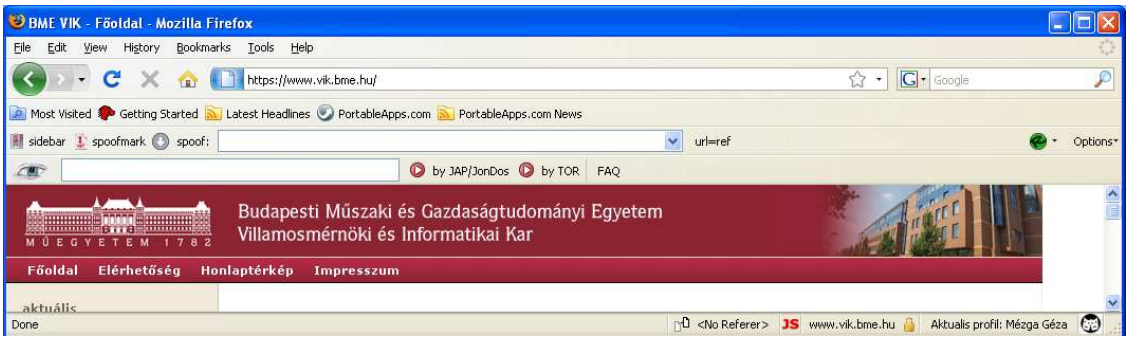


19. ábra Profilváltás 3.

A 17. ábra Profilváltás 1.17-19. képeken látható, hogy a profilhoz hozzárendelt oldalak betöltésekor, a böngésző jelezte, hogy új – a megadottak alapján szükséges – profilra vált.

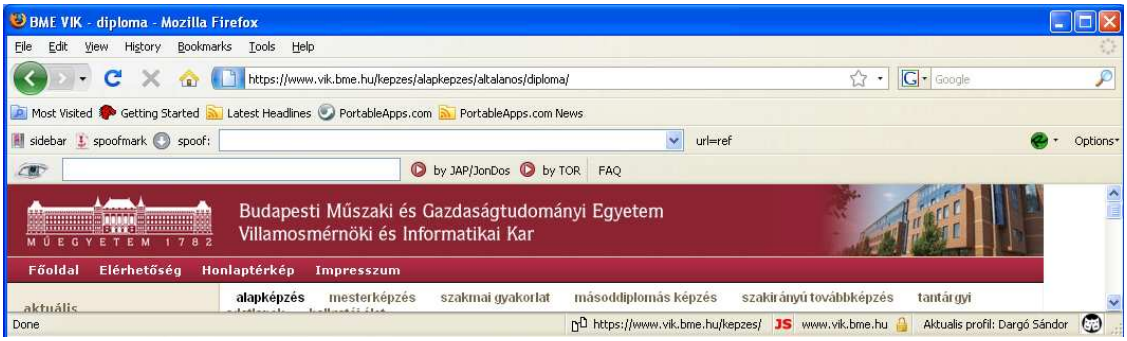
A következő tesztesetnek azt választottam, amikor a profilokhoz tartozó oldalak átfedésben vannak egymással.

Profilnév	**Hozzárendelt honlap**
Dargó Sándor	vik.bme.hu
Mézga Géza	bme.hu



20. ábra Profilváltások 4.

21. ábra Profilváltások 5.



22. ábra Profilváltások 6.

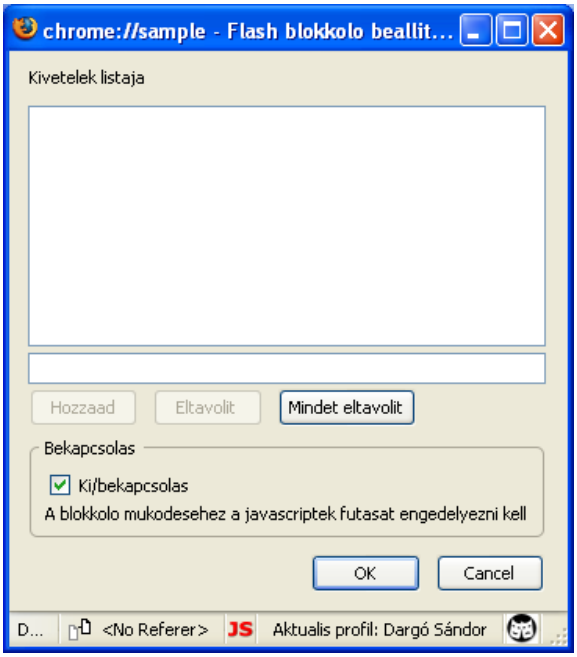
A 20-22. képek igazolják azon tapasztalatimat, melyek rávilágítottak a jelenlegi honlapváltási mechanizmus egyik hibájára.

Jelenleg az aktuális profilhoz rendelt címek ellenőrzésével kezdődik az a procedúra, aminek a végén eldöntésre kerül a profilváltás szükségessége. Ez a gyakorlatban azt jelenti, hogy amennyiben Mézga Gézaként a felhasználó a Műegyetem bármely olyan aloldalát tölti be, melynek URL-jében szerepel a bme.hu rész, úgy nem történik profilváltás, akkor sem, ha esetleg a teljes URL-re van jobban illeszkedő profil. Ez a továbbfejlesztések során javításra szorul. Továbbá amint azt már a 5.3.4 fejezetben megfogalmaztam, szükségesnek látom, hogy amennyiben több profil is hivatkozik egy oldalra, úgy kiválaszthassunk alapértelmezett profilt.

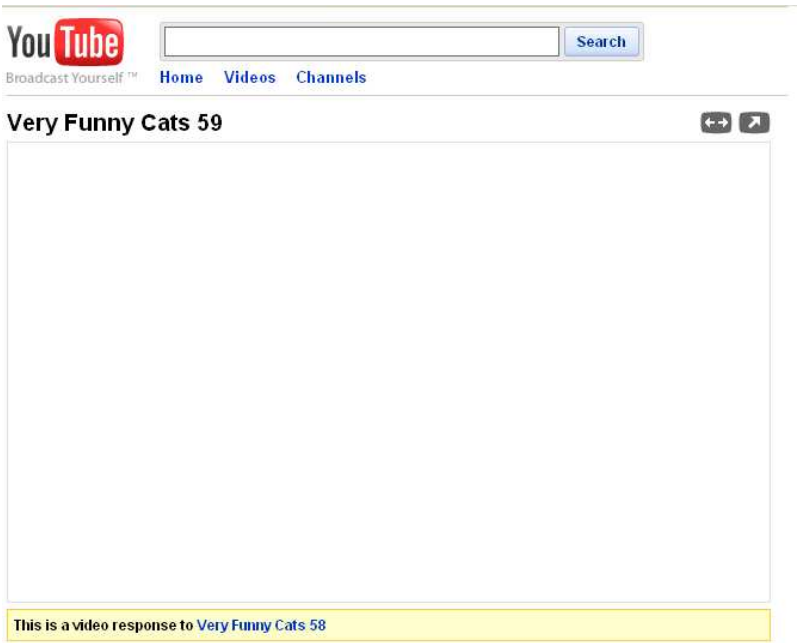
6.2 Flash blokkoló tesztelése

A Flash blokkoló tesztelése során, először megvizsgáltam, hogy az az elvártaknak megfelelően működik-e. Először kipróbáltam, hogy whitelist alkalmazása nélkül ki-

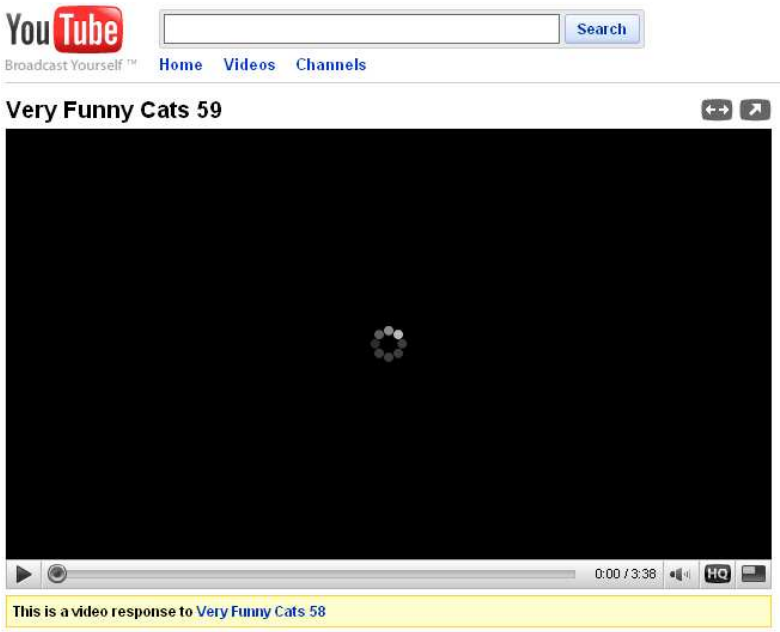
valamint bekapcsolt állapotban is az elvárt eredmények születnek-e, majd whitelist alkalmazásával is megvizsgáltam ezt a kérdést.



23. ábra Flash blokkoló bekapcsolva kivételek nélkül

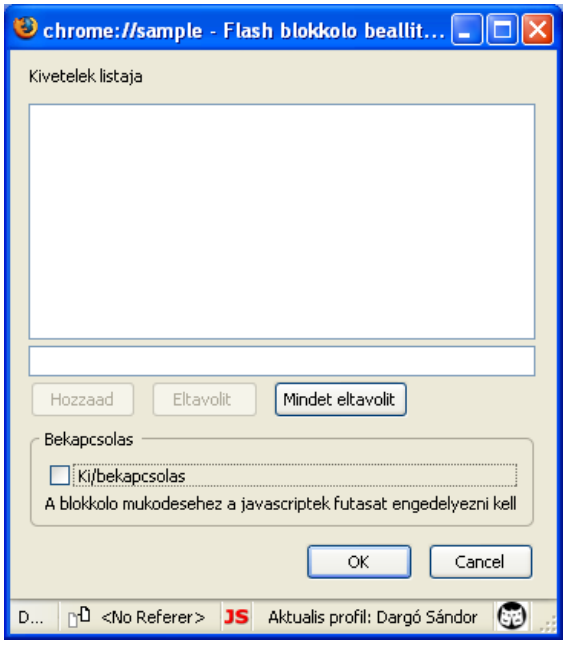


24. ábra Flash blokkoló működése whitelist nélkül



25. ábra Az eredeti flash betöltése

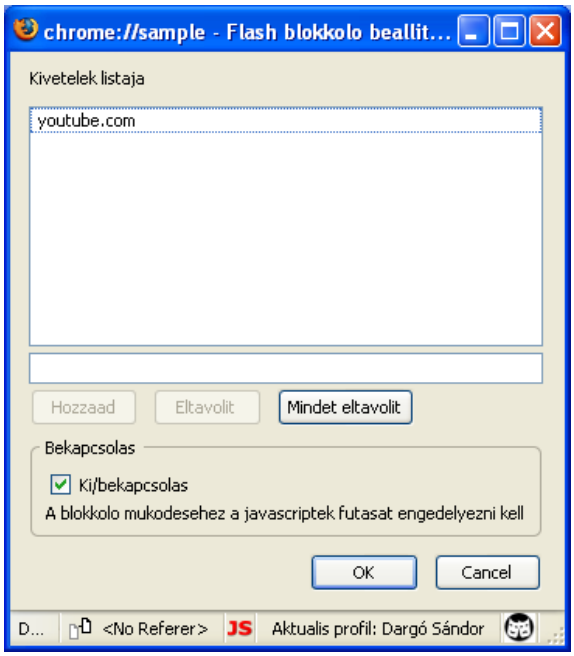
A 23-25. képeken látható, hogy a flash blokkoló bekapcsolt állapotában valóban blokkolja az oldalon található flash elemet, azonban annak a helyére kattintva a felhasználó manuálisan betöltheti azt.



26. ábra Flash blokkoló kikapcsolt állapotban

27. ábra A flash blokkolása nem történik meg

A 26-27. képek azt az esetet mutatják, amikor blokkoló kikapcsolt állapotban. Látható, hogy ekkor a flash eleme a megnyitott oldalnak automatikusan betöltődik.



28. ábra Flash blokkoló whitelisttel

29. ábra A kivételezett oldalon található flash betöltődése

30. ábra A whitelisten nem található honlap blokkolása

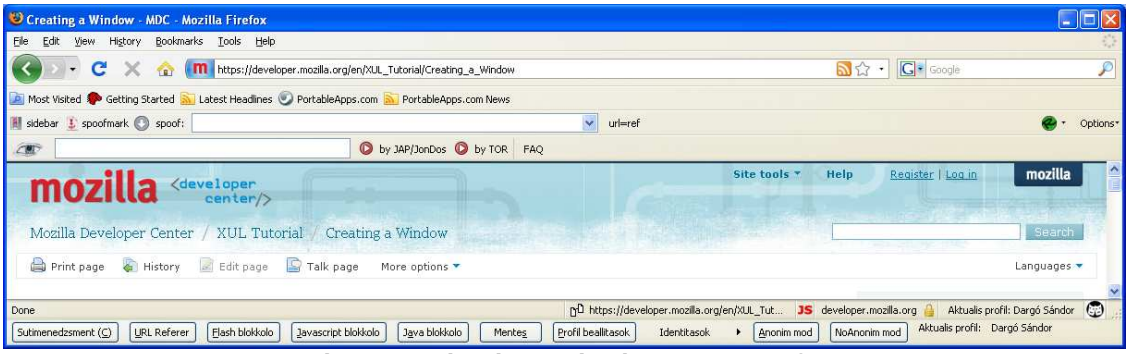
A 28-30. képeken látható, hogy a blokkolóhoz tartozó whitelisten megtalálható oldalon található flash automatikusan betöltődik, viszont a listán nem megtalálható honlap

sörgyűjtő játékát blokkolja a böngésző, tehát a Flash blokkoló az elvártaknak megfelelően működik.

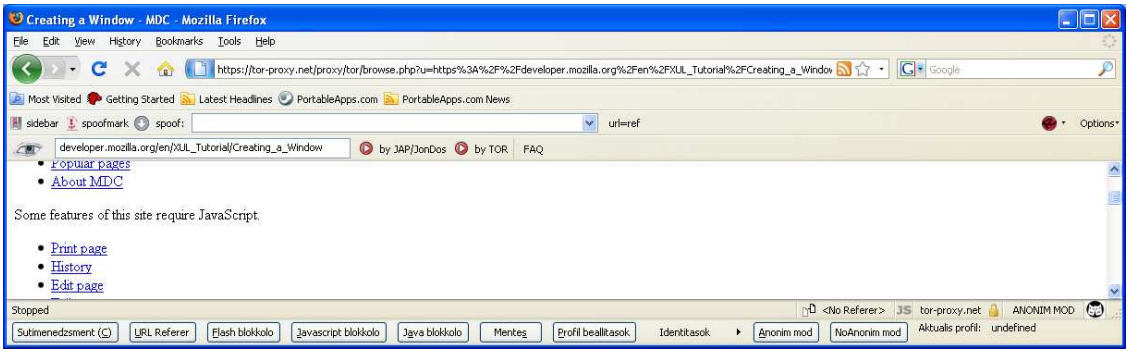
6.3 Anonim mód tesztelése

Az anonim mód tesztelése során az 6.2 fejezetben vizsgált eseteket nem teszteltem újra, hanem arra voltam kíváncsi, hogy a 5.4 fejezetben bemutatott anonim mód az elvártaknak megfelelően működik-e, garantálja a felhasználó számára az anonimitást.

Első esetben csak azt kívántam ellenőrizni, hogy a böngésző blokkolja-e az URL Referert. Amennyiben az 31. és 32. képeket összehasonlítjuk, a böngésző statusbarján láthatjuk, hogy az URL Referer tényleg blokkolásra került, sőt emellett azt is láthatjuk, hogy a weboldlal felhívta a figyelmünket, hogy bizonyos elemek működéséhez a JavaScriptek bekapcsolása szükséges, tehát megállapíthatjuk, hogy anonim módban az URL Referer és a JavaScriptek is blokkolásra kerültek.



31. ábra Normál módban látszódik az URL Referer



32. ábra Anonim módban az URL Referer és a JavaScriptek blokkolásra kerülnek

Azonban ezen elemek blokkolása nem elegendő célom – az anonimitás – eléréséhez, ezért következő lépésben megvizsgáltam közelebbről, hogy milyen adatokat is fed fel magáról a böngésző. Amint arra az 2.2.3.2 fejezetben rávilágítottam, már önmagában az IP cím alkalmas arra, hogy rengeteg információ ki lehessen deríteni a felhasználóról, ezért felkerestem egy olyan honlapot(hiv), ami megosztja az érdeklődő felhasználóval, hogy milyen adatokat fed fel róla az IP címe, illetve a böngészője.

Az 33.-36. képek jobb oldalán láthatók a felfedett információk anonim mód esetén, a bal oldalon pedig a normál mód esetén megosztott adatok. Amint az leolvasható, anonim mód esetén ténylegesen sikerült blokkolni eredeti IP címünket, illetve minden lényeges adatunkat, amely az ellenőrző honlap alapján kideríthető volt rólunk.

33. ábra Valós IP cím felfedése

34. ábra Hamis IP cím "felfedése"

IP Information

IP Address [?] :	91.120.149.112
Hostname [?] :	p9p9yg9rll.adsl.datanet.hu [>> Whols]
Remote Port:	2501
Protocol:	HTTP/1.1
Connection:	keep-alive
Keep Alive:	300

IP Location

Country of this IP:	Hungary (HU)
Region of this IP:	Budapest [>> IP MAP]
City of this IP:	Budapest [>> IP MAP]
ISP [?]:	GTS - DataNet Telecommunication Ltd
Latitude:	47.5000
Longitude:	19.0833

Proxy

Proxy Type:	None / Highly Anonymous [>> Proxy Checker]

IP Information

IP Address [?] :	141.76.45.35
Hostname [?] :	proxy2.anon-online.org [>> Whols]
Remote Port:	34920
Protocol:	HTTP/1.1
Connection:	keep-alive
Keep Alive:	

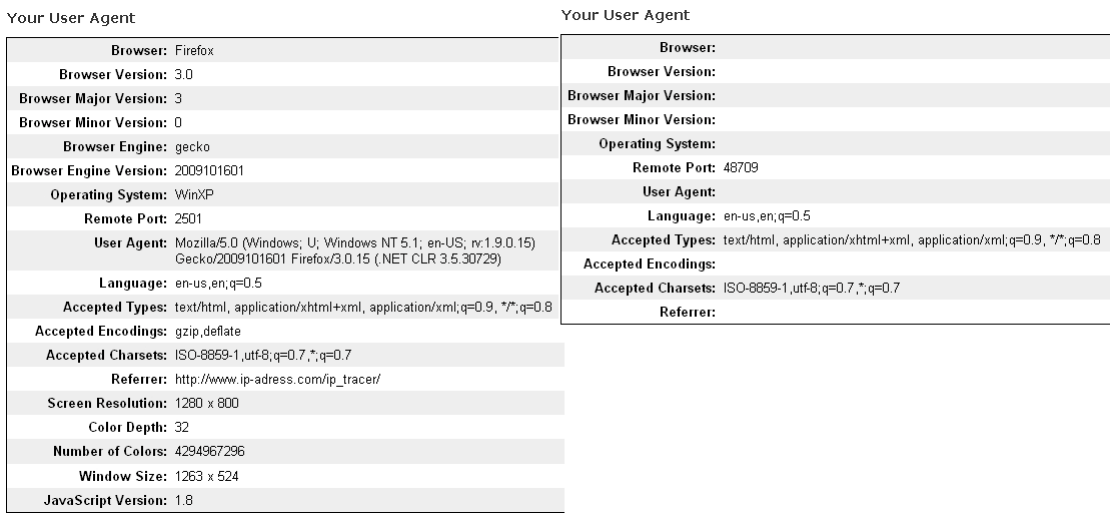
IP Location

Country of this IP:	Germany (DE)
Region of this IP:	Sachsen [>> IP MAP]
City of this IP:	Dresden [>> IP MAP]
ISP [?]:	Technische Universitaet Dresden
Latitude:	51.0500
Longitude:	13.7500

Proxy

Proxy Type:	None / Highly Anonymous [>> Proxy Checker]

35. ábra Nem anonim és anonim módokban felfedett adatok 1.



Your User Agent

Browser:	Firefox
Browser Version:	3.0
Browser Major Version:	3
Browser Minor Version:	0
Browser Engine:	gecko
Browser Engine Version:	2009101601
Operating System:	WinXP
Remote Port:	2501
User Agent:	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.15) Gecko/2009101601 Firefox/3.0.15 (.NET CLR 3.5.30729)
Language:	en-us,en;q=0.5
Accepted Types:	text/html, application/xhtml+xml, application/xml;q=0.9, */*;q=0.8
Accepted Encodings:	gzip,deflate
Accepted Charsets:	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referrer:	http://www.ip-adress.com/ip_tracer/
Screen Resolution:	1280 x 800
Color Depth:	32
Number of Colors:	4294967296
Window Size:	1263 x 524
JavaScript Version:	1.8

Your User Agent

Browser:	
Browser Version:	
Browser Major Version:	
Browser Minor Version:	
Operating System:	
Remote Port:	48709
User Agent:	
Language:	en-us,en;q=0.5
Accepted Types:	text/html, application/xhtml+xml, application/xml;q=0.9, */*;q=0.8
Accepted Encodings:	
Accepted Charsets:	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Referrer:	

36. ábra Nem anonim és anonim módokban felfedett adatok 2.

6.4 A tesztrendszer értékelése

Az implementáció és a tesztelés során szerzet tapasztalatok alapján arra a következtetésre jutottam, hogy bár a rendelkezésre álló idő rövidsége miatt, kénytelen voltam a megvalósítandó funkciókat racionalizálni a tesztrendszer teljesíti az anonimitás és az identitásmenedzsment alapvető követelményeit.

A rendszer lehetővé teszi a felhasználó részére, hogy a különböző identitásait, azok jellege, felhasználási típusa szerint fastruktúrába szervezze, és bármikor újakat hozhasson létre, vagy meglévőket törölhessen. A megvalósított felületen keresztül bármikor lehetősége nyílik gyorsan áttekinteni profiljai hierarchikus viszonyait. A profilhierarchiát illetően az eredeti tervekből az idő hiánya miatt egyelőre nem valósultak meg a tranzakciós táblák, melyek lehetővé teszik a felhasználó számára, hogy egy profilon belül egy-egy azonosítót csak egyszer használjon – például fórumozás esetén egy névvel csak egyszer szólhasson hozzá –, azonban azok implementálása a legfontosabb továbbfejlesztési javaslatok között szerepelnek.

Megvalósítottam, hogy az egyes profilokhoz tetszőleges mezőket és hozzájuk tartozó adatokat rögzíthessen a felhasználó, illetve, hogy ezeket törölhesse. Ezek az adatok bármikor felhasználhatók, az egyes mezőkhöz tartozó adatokat a context menü segítségével bármikor a vágólapra lehet helyezni néhány kattintással. Az adatok tárolása kapcsán a továbbfejlesztés során ki kell térni az adatok profilhierarchián belüli öröklődésére, illetve az adatok felhasználásának automatizálhatóságára.

A tesztrendszer lehetőséget ad a profilok közötti egyszerű váltásokra két különböző módon. Egyrészt a felhasználó manuálisan is válthat az egyes profilok között, másrészről lehetősége nyílik arra, hogy az egyes identitásokhoz honlapokat rendeljen hozzá. Amennyiben egy valamely identitáshoz hozzárendelt honlapra látogat el, úgy az alkalmazás jelzi, hogy az ahhoz definiált identitásra váltott. A későbbiekben mindenképpen javasolt finomítani az identitásváltások automatizmusának mechanizmusán, lehetővé kell tenni, hogy amennyiben egy honlap több profilhoz is tartozik, a felhasználó dönthessen, hogy egy alapértelmezett, vagy bármely más, általa választott profilt aktiválva folytatódjon a böngészés.

Egy kiválasztott modulon keresztül megmutattam, hogy a rendszer alkalmas arra, hogy a biztonsági beállításokat profilonként tárolja a tesztrendszer, a felhasználó számára lehetővé téve, hogy a különböző profiljai számára a Flash blokkolóhoz különböző beállításokat használjon.

Bár a tervezet funkciók még nem kerültek teljes körűen megvalósításra, látható, hogy jelen állapotában is megvalósítja azon alapvető követelményt, hogy a felhasználó

szabályozhassa, hogy hol milyen identitás mögött kíván megjelenni, alkalmas arra, hogy a magáról felfedett adatok minőségét szabályozza.

Az anonim mód által bemutattam, hogy a tesztrendszer jelenlegi struktúrája alkalmas arra, hogy további biztonsági korlátozásokat, beállításokat kezeljen, illetve alkalmas anonimizáló hálózat kezelésére. Bár a JavaScriptek, az URL Referer és az IP cím blokkolása nem beállítható profilonként, azonban az anonim mód létrehozásával mégis szabályozható ezen funkciók működése. Ez a tény és a Flash blokkoló működése garancia arra, hogy az anonim mód kiegészítéseinek kezelése is alkalmas a tesztrendszer. A tesztelés során megmutattam, hogy az anonim mód bekapcsolásával sikerült elérni a böngésző, illetve a felhasználó adatainak rejtését.

A leírtak tükrében ki merem jelenteni, hogy az alapvető célomat sikerült teljesítenem, azaz implementáltam egy olyan tesztrendszert, ami identitásmenedzsment által biztosítja a felhasználó számára adatainak védelmét és a felfedett adatainak kontrollálását, valamint lehetőséget ad a böngészés során anonimitásba burkolózásra. Bár a feladatok racionalizálása végett még vannak hiányosságok, de megvalósított tesztrendszerre építve, a megfogalmazott továbbfejlesztési javaslatok által létrehozható egy olyan alkalmazás, mely az elméleti modell megalkotása, valamint a tervezés során megfogalmazott összes követelményt kielégíti és lehetőséget biztosít a felhasználó számára, hogy a teljes anonimitás állapotából kiindulva önmaga dönthesse el, hogy milyen minőségű és mennyiségű információt kíván felfedni magáról a böngészés során.

7 Összefoglalás

Diplomatervem célja egy olyan alkalmazás megtervezése volt, ami lehetővé teszi a felhasználó számára az anonim böngészést és személyes adatainak privátszféra barát identitásmenedzsment általi védését, valamint egy az ezen tervekre alapuló tesztrendszer megalkotása, melyen keresztül megmutathatom a tervek életképességét.

Az ismertetett célokhoz vezető út első lépéseként sorra vettem, hogy kik által milyen veszélyek fenyegetik a felhasználók privátszférájába tartozó adatokat a böngészés során, majd kitértem az ezekre a fenyegetésekre adható lehetséges válaszokra.

A lehetséges válaszok között megvizsgáltam az identitásmenedzsment, és ismertettem az annak részterületét képező szerepalapú privátszféra-védelem fogalmát, majd bemutattam a terület technikai mellett szociológiai aspektusával is foglalkozó – az EU által finanszírozott – PRIME és Primelife projekteket.

A bemutatott veszélyek és lehetőségek alapján megalkottam egy elméleti modellt, amely meghatározza egy ideális identitásmenedzsment szolgáltatással szemben támasztott követelményeket, kitérve külön az anonimitás és külön a privátszféra barát identitásmenedzsment követelményeire.

Az elméleti modellt alapján meghatároztam az alkalmazást érintő konkrét követelményeket. A követelményeken belül kiemelten foglalkoztam az identitásmenedzsment egyik legfontosabb sarokpontjával a profilok megfelelő struktúrába szervezésével, valamint a felhasználói profiljai közötti váltások mikéntjével választ keresve arra a kérdésre, hogy hogyan lehet megoldani az egyszerű kezelhetőség, valamint a megfelelő konfigurálhatóság dilemmáját. Úgy gondolom ez a kérdés meghatározó fontossággal bír azon alkalmazások körében, amelyek a laikus felhasználók számára készülnek, mivel olyan beállítási lehetőségeket kell kínálni számukra, melyek már garantálják az alkalmazás céljának elérését, azonban még nem riasztják el a felhasználókat bonyolultságuk által.

Ennek a kérdéskörnek is köszönhetően a tesztrendszer az implementálás során a lefektetett tervekhez képest számos ponton finomodott, idomult felhasználóbarát használat és az implementálhatóság követelményeihez.

A tesztelés során megmutattam, hogy már a tesztrendszer által biztosított lehetőségek kihasználásával is elérhető az anonimitás, szabályozhatóvá válik annak az adatnak a mennyisége és minősége, melyet a felhasználó fel kíván fedni magáról a böngészés során.

Kitértem azokra a részben tesztrendszerből szándékosan kihagyott funkciókra, részben a megvalósítás és a tesztrendszer kipróbálása során felmerült továbbfejleszti javaslatokra, melyek implementálásával – melyekhez a tesztrendszer eredményei alapján a keretek adottak – reményeim szerint a kiterjesztés a jövőben a felhasználók hasznára válhat.

Ábrajegyzék

1. ábra Webpoloskák működési mechanizmusa [8] 10
2. ábra A PRIME és különböző szerepek [17] 17
3. ábra Az anonimitás felépülése [21] 21
4. ábra Vizsgált profilhierarchiák 28
5. ábra Végső profilhierarchia 29
6. ábra CookieCuller kezelőfelülete 27
7. ábra IDM eszköztár 31
8. ábra Ikon és visszacsatolás a statusbaron 33
9. ábra Flash blokkoló kezelőfelülete 36
10. ábra Profilmenedzser kezelőfület 40
11. ábra Profilhierarchia ábrázolása 41
12. ábra Profiladatok felvétele 44
13. ábra Profiladatok és a context-menü 45
14. ábra A profilhoz tartozó honlapok 47
15. ábra A JavaScript blokkoló kezelőfelületének terve 52
16. ábra A sütimenedzser kezelőfelületének terve 53
17. ábra Profilváltás 1. 57
18. ábra Profilváltás 2. 58
19. ábra Profilváltás 3. 58
20. ábra Profilváltások 4. 58
21. ábra Profilváltások 5. 59
22. ábra Profilváltások 6. 59
23. ábra Flash blokkoló bekapcsolva kivételek nélkül 60
24. ábra Flash blokkoló működése whitelist nélkül 60
25. ábra Az eredeti flash betöltése 61
26. ábra Flash blokkoló kikapcsolt állapotban 61
27. ábra A flash blokkolása nem történik meg 62
28. ábra Flash blokkoló whitelisttel 62
29. ábra A kivételezett oldalon található flash betöltődése 63
30. ábra A whitelisten nem található honlap blokkolása 63
31. ábra Normál módban látszódik az URL Referer 64
32. ábra Anonim módban az URL Referer és a JavaScriptek blokkolásra kerülnek 64
33. ábra Valós IP cím felfedése 65
34. ábra Hamis IP cím "felfedése" 65
35. ábra Nem anonim és anonim módokban felfedett adatok 1. 66
36. ábra Nem anonim és anonim módokban felfedett adatok 2. 66

Irodalomjegyzék

[1] Soraya Roberts, Nathalie Blanchard loses benefits over Facebook beach photos, Daily News, 2009.11.22.,
http://www.nydailynews.com/news/world/2009/11/22/2009-11-22_nathalie_blanchard_loses_benefits_over_facebook_beach_photos.html

[2] Identitáslopás elleni biztosítás a British Telecomnál, SG.hu – Informatikai és Tudományos Hírmagazin, 2006.02.24
http://www.sg.hu/cikkek/42922/identitaslopas_elleni_biztositas_a_british_telecomnal

[3] Kovács Áron, Nagyvonalú vírusölők adják ki titkainkat az államnak, [origo], 2008.10.12.
http://www.origo.hu/nagyvilag/20081009-internet-megfigyeles-hatosagi-kemprogramok-kutakodnak-a-vilag-szamitogepein.html

[4] Cookies, NetLingo
http://www.netlingo.com/word/cookies.php

[5] Cookie-k vs. anonimitás, IT Security Blog, 2009.12.02
http://secblog.stratis.hu/2009/12/cookie-k-vs-anonimitas.html

[6] AboutCookies.org
http://www.aboutcookies.org/default.aspx

[7] What are local shared objects, Adobe.com
http://www.adobe.com/products/flashplayer/articles/lso/

[8] Hullám Gábor: A web bug technológia — barát vagy ellenség?, Szabad adatok, védett adatok, BME ITM, 2005. március

[9] Ashlee Vance, Times Web Ads Shoe Security Breach, NY Times, 2009.09.15
http://www.nytimes.com/2009/09/15/technology/internet/15adco.html?_r=1

[10] Web beacons, Yahoo! Privacy
http://info.yahoo.com/privacy/us/yahoo/webbeacons/

[11] Refspoof Firefox Extension, mozdev.org
http://refspoof.mozdev.org/

[12] Gulyás Gábor: Anonim-e az anonim böngésző? Technológiák és szolgáltatások elemzése, Alma Mater, 6. évfolyam, 1. szám

[13] XSS kicsiknek és nagyoknak, Webmánia
http://webmania.cc/xss-kicsiknek-es-nagyoknak/

[14] M. Hansen, A. Schwartz, A. Cooper, - "Privacy and Identity Management", IEEE Security and Privacy, VI. évfolyam, 2. szám, 2008 márc./ápr.

[15] J. Hakkila, and I. Kansala. Role based privacy applied to context-aware mobile applications. In Proc. of IEEE Conference of System, Man and Cybernetics, 2004.

[16] R. Leenes, J. Schallaböck, and M. Hansen. PRIME white paper (V3). May 2008.

[17] PRIME – Privacy and Identity Management for Europe

https://www.prime-project.eu/

[18] PrimeLife – Privacy and Identity Management in Europe for Life

http://www.primelife.eu/

[19] Ian Avrum Goldberg – A Pseudonymos Communications Infrastructure for the Internet, 2000, II.4. fejezet

[20] Ari Juels, How 10 digits will end privacy as we know it, CNET News, 2009.08.17

http://news.cnet.com/8301-1009_3-10310446-83.html

[21] Gulyás Gábor, Schulcz Róbert, Imre Sándor - Comprehensive Analysis of Web Privacy and Anonymous Web Browsers: Are Next Generation Services Based on Collaborative Filtering?, Proceedings of the Joint SPACE (Sustaining Privacy in Autonomous Collaborative Environments, TIME (Trust in Mobile Enviroments) Workshops, Trondheim, Norvégia, 2008. június 17.

[22] Szili Dávid – Anonimizáló rendszerek elméletben és gyakorlatban, Hacktivity 2008, 2008.09.21

http://pet-portal.eu/files/blogs/pet/2008/09/hacktivity/4_szili.pdf

[23] Gyöngyösi László – Tor és Torpark: az új generációs anonim böngészők funkcionális és teljesítményelemzése, Tanulmányok az információ- és tudásfolyamatokról 11., BME GTK ITM, Budapest, 2006. október

[24] Popup Menus, Mozilla Developer Center

https://developer.mozilla.org/en/XUL_Tutorial/Popup_Menus

[25] A Brief Guide for Mozilla Preferences, Mozilla Developer Center

https://developer.mozilla.org/En/A_Brief_Guide_to_Mozilla_Preferences

[26] Customizing Mozilla, Mozilla.org

http://www.mozilla.org/unix/customizing.html