A web 2.0 legfontosabb biztonsági fenyegetései

A dokumentumot kiadó *Secure Enterprise 2.0 Forum* célja, hogy a fogyasztói technológiák munkahelyi alkalmazásával kapcsolatos ismeretek terjedését elősegítse. Olyan biztonsági kockázatokról lesz szó az alábbiakban, amelyek kizárólag vagy jellemző módon a web 2.0-s alkalmazásoknál fordulnak elő. A webkettes alkalmazások megjelenése a munkahelyeken elkerülhetetlen. Nem félni kell azonban tőlük, hanem szakszerű és biztonságos működtetésükre kell törekedni. Ennek érdekében szólni kell sebezhetőségükről.

A web 2.0-s alkalmazások legfontosabb sebezhetőségi pontjait a következőkben lehet összefoglalni.

Elégtelen azonosítás

Az információs rendszerek általában feltételezik, hogy a rendszerben vannak kitüntetett (kiemelt/adminisztrátori jogokkal rendelkező) felhasználói fiókok. Ha nem ilyen felhasználók hajtanak végre változtatásokat, akkor azokat ellenőrizni kell, mielőtt a rendszerre hatással lennének. Mivel a web 2.0-s alkalmazások tartalmát a felhasználókra bízzák, a fenti állítás már nem érvényes, így fennáll annak a veszélye, hogy kevésbé tapasztalt felhasználók olyan változtatásokat vihetnek végbe, amelyek ártalmasak lehetnek az egész rendszerre.

Akik az online oldalakhoz jelentős mértékben hozzájárulnak, sokszor kapnak adminisztrátori jogokat ezeknek a webhelyeknek a működtetéséhez. Közülük sokan gyenge, könnyen visszafejthető jelszavakat használnak, amit még fokoz a jelszóemlékeztetők triviális jellege. A támadók ilyen módon hitelesített felhasználóként tudnak bejelentkezni, hamis információkat tudnak elhelyezni a rendszerekben és meg nem engedett adminisztrátori tevékenységeket végezhetnek.

Elégtelen védelem a brute force (nyers erő) támadásokkal szemben

Ez a hiányosság lehetővé teszi, hogy a támadók kitalálják a felhasználók vagy az adminisztrátor jelszavát. Még ha a fő bejelentkezési funkciók védve vannak is, az olyan kiegészítő hitelesítési funkciók, mint a „remember me” vagy a jelszóemlékeztetők, valamint a kijelentkezés védtelenné teszik a rendszert a nyers erővel szemben. (A brute force lényege a gyakori/valószínű azonosítókkal/jelszavakkal való kísérletezés, nagy tömegű próbálkozás.)

Nyílt szövegű (titkosítás nélküli) jelszavak

Az AJAX, a widgetek és a mashupok használatakor a jelszavakat titkosítás nélkül küldhetik és tárolhatják, úgy, hogy a hoszt ellenőrzési hatókörén kívül kerülnek. Ilyen módon az információ könnyen elérhető bárki számára, akinek hálózati forgalomelemző eszköze van.

Egyszeri bejelentkezés

A web 2.0-s, személyre szabott weboldalakon és widget környezetben igen kényelmetlen, hogy minden alkalmazásba külön-külön bejelentkezzünk. Ezért ezekben a környezetekben egyszeri bejelentkezési mechanizmusok vannak. A felhasználót azonosító, hitelesítési információk vagy az ő kliensgépén, vagy valahol a szerverek alkotta virtuális térben vannak.

Cross-site scripting (XSS) támadás

Az XSS esetében a támadók által küldött rosszindulatú kódot tárolja a rendszer, majd megjeleníti más felhasználóknak. Az olyan rendszerek, amelyek megengedik, hogy a felhasználók formattált

tartalmat, például HTML kódot vigyenek be, különösen érzékenyek az XSS-re, mivel rosszindulatú input könnyen előállítható, például szkriptek segítségével. Mivel a webkettes rendszerekre jellemző, hogy sok felhasználó hoz létre olyan tartalmakat, amelyeket mások is látnak, ezek különösen sebezhetők az XSS-típusú támadásokkal szemben.

Cross-site request forgery (CSRF)

A CSRF esetében az áldozat ártatlannak látszó, de rosszindulatú weboldalt látogat meg. Amíg az áldozat böngészője letölti az oldal tartalmát, a rosszindulatú oldal kódja egy kérést generál (kihasználva a hosszú ideig érvényes azonosítást tároló süti által okozott biztonsági rést) egy másik oldalhoz, amelynek hiteles felhasználója az áldozat. Az ilyen kérések az áldozat nevében hajthatnak végre műveleteket.

A web 2.0-s alkalmazások potenciálisan különösen sebezhetők a CSRF-fel szemben, mivel AJAX-ot használnak.

A hagyományos technológián alapuló, de üzleti értelemben még életképes alkalmazások (legacy applications) esetében a felhasználók által generált kérések vizuális effektust eredményeztek, ami könnyebbé tette a CSRF támadások felfedését. A web 2.0-s rendszerek viszont jellemzően olyan kiterjedt alkalmazásprogramozási felületeket (API) nyújtanak, amelyeket más rendszerek vagy a kliensoldali kódok vizuális hatás nélkül használhatnak. A funkciók nagy száma és a vizuális visszacsatolás hiánya az AJAX-alapú web 2.0-s alkalmazásokat érzékennyé teszik a CSRF támadásokra. (Arról van szó, hogy a hagyományos alkalmazásoknál a felhasználó látta, ha a kliense valami kérést küldött, mert újratöltődött az egész weboldal, az AJAX-alapú alkalmazásoknál ez nem feltétlenül van így, ezért tud egy szkript észrevétlenül kéréseket küldeni a felhasználó helyett.)

A CSRF támadásoknak határt szab a same origin policy (SOP), mert a rosszindulatú szkript ugyan tud kéréseket küldeni a szervernek, de nincs joga hozzáférni a visszakapott adatokhoz. Ugyanakkor egyes web 2.0-s környezetek, például a desktop widgetek vagy a személyre szabott honlapok nem mindig juttatják érvényre ezt a védelmet.

A CSRF-et elősegíti, hogy a kliensoldalon tárolják és minden kéréssel automatikusan elküldjék a hosszú ideig érvényes felhasználói hitelesítési információkat.

Amikor több nyitott session kombinálódik össze, amelyek különböző célalkalmazásokhoz tartoznak ugyanazon az asztalon (desktopon) vagy személyes honlapon, a CSRF számára nagyobb támadási felületet adunk.

A web 2.0-s, személyre szabott weboldalakon és a desktop widget környezetekben a felhasználók gyakran hosszú ideig kapcsolódnak a rendszerhez, ezért arra törekszenek, hogy minimalizálják az időtúllépések miatt szükséges újrabejelentkezések számát. Ennek eredményeként a webkettes környezetben a hitelesítések lejárati ideje viszonylag hosszú, növelve ezzel az olyan session-alapú támadásokat, mint a CSRF vagy a „munkafolyamat-eltérítés” (session hijacking).

Adathalászat (Phishing)

Az adathalász-támadások során az áldozat egy olyan tartalmú e-mailt kap, amelyben kérik, hogy töltsön ki egy online kérdőívet és így érzékeny személyes adatokhoz jutnak a csalók. Az online kérdőívet egy hamisított weboldalon helyezik el. Mivel elektronikus levélből irányítják az áldozatot a weblapra, az adathalászat nem valamilyen szoftverhiányosságot használ ki, hanem azt, hogy a felhasználó nem tud világosan különbséget tenni a valódi és a hamis weboldalak között.

A web 2.0-s mashupok, widgetek és az eltérő kliensszoftverek sokasága nagyon nehézzé teszi a felhasználók számára, hogy megkülönböztessék egymástól a valódi és a hamis oldalakat, ami hatékony adatgyűjtést tesz lehetővé.

Az adathalászat mérséklésében nagy szerepe van az oktatásnak. A felhasználókat megtaníthatjuk arra, hogy felismerjék a biztonságos oldalakat a doménnév vagy az SSL tanúsítvány alapján. Amikor egy oldal tartalma be van ágyazva egy másik oldalba egy webszolgáltatás segítségével, ez a védekezési lehetőség nincsen meg.

Azok a web 2.0-s alkalmazások, amelyek általában megengedik, hogy a felhasználók tartalmat töltsenek fel, megnövelik az adathalászat veszélyét, mivel a támadók csalárd információikat biztonságosnak tekintett webhelyeken helyezhetik el. Az XSS-t is rendszerint az adathalász-támadásokra használják, mivel lehetővé teszi, hogy a támadó

által létrehozott tartalom olyannak látszódjon, mintha megbízható oldalról jönne.

Az információ kiszivárgása

Mivel a web 2.0-s alkalmazások nemcsak a felhasználók által létrehozott tartalmat támogatják, hanem az otthoni és a munkahelyi tevékenység határainak elmosódásához is vezetnek, előfordulhat, hogy a felhasználók akaratlanul is a munkáltatójuk számára kényes természetű információkat hoznak nyilvánosságra. Még ha óvatosak is, és nem szivárogtatnak ki bizalmas információkat, a sok, nem bizalmas kis adat felhalmozódása is feltárhat érzékeny természetű információkat. Egy kis cég alkalmazottainak a közösségi hálózatokon megtalálható adatainak elemzése például felhasználható üzleti hírszerzés céljaira.

A webkettes alkalmazásokat gyakran ingyenes, nyilvános szolgáltatók, például közösségi hálózatok, blogszájtok, vagy információmegosztó oldalak nyújtják. Ezek egy része felhasználó-központú, ezért az itt közzétett információ a személyes és a céges információ keveréke, így a személyes és szakmai tapasztalatok között egyre inkább elmosódó határ is az információk kiszivárgásához vezet.

Másodlagos kockázatot jelent az, hogy még ha az online közzétett információ minden egyes darabja ártalmatlan is, ezek együttes hatása nem várt következményekkel járhat. Ennek legjobb példája az üzleti információ felhalmozódása a közösségi hálózatokon. A dolgozók száma, a korábbi alkalmazottak jegyzéke stb. felbecsülhetetlen értékű, főleg a versenytársak számára, és az adott cég elleni támadások indítására is alkalmas lehet, például a brute force, a social engineering, amelynek lényege az emberek manipulálása, vagy az adathalászat és a malware-ek (rosszindulatú számítógépes programok) felhasználásával.

Számos web 2.0-s szolgáltatás keveri a személyes védelmet kapott és a nyilvánosságnak szánt információkat, amelyeket csak egy gombnyomás választ el egymástól. Ennek következtében belső használatra szánt, kényes információk, például képek, térképek, kapcsolati adatok juthatnak nyilvánosságra. Ha nem megfelelően van kialakítva a könnyű és automatikus hozzáférés a webszolgálta-tásokhoz, akkor szintén szivároghatnak ki információk.

A felhasználóknak a személyes adatokkal kapcsolatos óvatlansága mellett az is problémát jelent, hogy a webkettes világ elősegíti a webszolgáltatások terjedését, amelyek túl sok információt tesznek könnyen elérhetővé.

A dokumentálatlan webszolgáltatások is elősegíthetik a kényes információk elérését, és az olyan WSDL (Web Services Description Language) nyelvű leíró (nyilvános) fájlok, amelyekkel az egyes webszolgáltatásokkal való kommunikáció módját dokumentálják, szintén értékes információt nyújtanak a támadóknak.

Injection rések; az injection technikával kihasználható hibák

A web 2.0 különösen érzékeny a beágyazott távoli parancsfuttatással megvalósított (command injection flaw) támadásokra, ideértve az XML, az XPath, a JavaScript és a JSON injection technikákat. Ráadásul a webkettes alkalmazások nagymértékben támaszkodnak a kliensoldali kódra, gyakran ott végzik az inputok validálását, amelyeket meg tud kerülni a támadó.

A webszolgáltatások és az AJAX kulcsfontosságúak a web 2.0-s alkalmazásokban. Mindkettő XML-t használ. Az injection olyan támadás, amelynél a felhasználói oldalon megadott információt megfelelő jóváhagyás nélkül XML rekordokba helyezik. Az injektált input aztán úgy módosítja az XML rekord szerkezetét, hogy nemcsak tartalmat, hanem címkéket is ad hozzá.

Az XML alkalmazások gyakran használnak Xpath relációs adatbázis-lekérdező nyelvet. Az Xpath injection olyan támadás, amelynek során az XPath lekérdezésbe beágyazott, specifikusan kialakított inputokkal érnek célhoz.

A JSON (JavaScript Object Notation) olyan protokoll, amelyet AJAX alkalmazások használnak a kliens és a szerver közötti információátvitelre. Mivel a JSON formátum érvényes JavaScript kód, egyes alkalmazások használják azt az adatok elérésére. Rosszindulatú JavaScript kódok injektálásával a JSON struktúrába a támadók kárt okozó program végrehajtását érhetik el.

Az információ integritása

Az adatok integritása vagy helyessége (correctness) egyike az adatbiztonság kulcselemeinek. Amíg az integritást gyakran sértik meg rosszindu-

latú hackertámadások, a szándékolatlan félreinformálás szintén ezt eredményezheti. Amikor *Kennedy* szenátor halálhíre (idő előtt) megjelent a Wikipédiában, a tévedés elég nagy volt, és az oldalt elég sokan használták ahhoz, hogy azonnal észrevegyék. A kisebb, kevésbé látványos tévedések kevésbé látogatott weboldalakon azonban lényegesen nagyobb problémát jelentenek. Még ha az elváltozások nem is rosszhiszeműek, az egyes pontatlanságok halmozódása az igazság jelentős torzulásához vezethet, ha egyetlen történetté gyúrják össze.

Mivel a web 2.0-s rendszerek sok felhasználónak engedik meg információk közzétételét, nehezen lehet elérni, hogy mindegyik megbízható legyen. A rosszindulatú felhasználók megtévesztő információkat helyezhetnek el, amelyek akár a tőzsdéket is befolyásolhatják. A web 2.0-s rendszerek hasonlóképpen jó eszközei pletykák, szóbeszédek terjesztésének. Sokszor az információ eredeti forrása nincs is tudatában annak, milyen intenzitással és sebességgel terjed az információ. Ilyen módon kisebb pontatlanságok nem szándékos és végzetes következményekkel járhatnak.

Elégtelen védelem az automatizált folyamatokkal szemben

A web 2.0-s alkalmazások programozható interfészei lehetővé teszik, hogy a károkozók automatizálják támadásaikat, például a brute force vagy a CSRF alkalmazása során. Nagy mennyiségű információ automatikus publikálása és a felhasználói fiókok automatikus megnyitása is megvalósítható adathalász támadások részeként. A web 2.0-s alkalmazásokba olyan, az automatizálást gátló mechanizmusokat kell beépíteni, amelyeket általában nem találunk meg a hagyományos alkalmazásokban.

Gyakori automatizálási visszaélés a nagy mennyiségű információ publikálása, kihasználva azt, hogy a webkettes alkalmazások elfogadják a felhasználók által létrehozott tartalmakat. Ezt gyakran web spamnek is hívják, amely legtöbbször linkek elhelyezését jelenti. Mivel a webes keresők az egy-egy helyre mutató linkek számával mérik az adott webhely népszerűségét, az ilyenfajta spam segít abban, hogy az adott oldal előkelő helyre kerüljön a találati listában.

Felhasználói fiókok automatikus létrehozásával a támadók saját alkalmazásuk részeként, visszaélhetnek a web 2.0 igény szerinti funkcionalitással. Jó példa erre az, amikor azért nyitnak egy webes e-mail postafiókot, hogy azt használják hitelesítéshez egy másik szolgáltatásnál.

A folyamatok automatizálása más felhasználókkal szembeni előnyöket jelenthet a támadóknak. Ennek egy példája, hogy a rosszindulatú felhasználók felvásárolják az online kínált jegyeket, majd újra eladják őket, de a játék- és az aukciós oldalakon is előnyt tudnak szerezni az automatikus eszközöket bevető támadók.

/Top Web 2.0 Security Threats. 2009 Industry Report. Secure Enterprise 2.0 Forum. = *http://www.secure-enterprise20.org/files/Top%20Web%202%200%20Security%20Threats.pdf*/

(Koltay Tibor és Jávorszky Ferenc)