A szkriptek diadalútja

Halászat új vizeken

Sorozatunk harmincharmadik részében csokorba gyűjtjük a szkriptes kártevők krémjét. A szkriptnyelveket sokan lenézik, de igazság szerint szinte bármit meg lehet velük valósítani a számítógépben – és ez jót és rosszat egyaránt jelenthet.

A szkriptek néhány fajtája komolyabb hatáskörrel a gép fájlrendszeréhez, mélyebb rétegeihez is hozzáférhet, ugyanúgy, mint a VBS a Windows, vagy a Bash a Linux alatt. Cikkünkben szó esik a szkriptalapú kártevőkről, de bemutatunk néhány olyan szimpla parancssori utasítást is, amelyek akár komolyabb kárt is tehetnek rendszerünkben. Mindenesetre nemcsak az látható, hogy ezek a szkriptek milyen sokszor lehetnek veszélyesek, hanem az is, hogy segítségükkel olyan új támadási módszerek jelentek meg, amelyek az újszerűségükkel az antivírus szoftverek gyártóinak is rendre feladják a leckét.

Ez egyáltalán nem vicces

Egy újszülöttnek minden vicc új. Ugyanígy, minden újonc számítógép-használónak végig kellene járnia legalább egy alapszintű tanulási folyamatot – legyen ez akár önszorgalomból végzett autodidakta ismerkedés a géppel, vagy egy külön tanfolyam –, hogy apróbb-nagyobb problémáit képes legyen saját maga is megoldani. Ha nagy bajban lesz, úgyis segítséget kér a barátjától, szomszédjától, családtagjaitól vagy a munkahelyén a rendszergazdától, esetleg netes fórumokban. A fórumok lakóinak többsége olyan jó szándékú veterán, aki már sok mindent megtapasztalt, és egy-egy jól irányzott kérdésre adott válaszuk nagyon meg tudja könnyíteni a problémájára megoldást kereső életét. Azonban – ha ritkán is – köztük is akad olyan rossz szándékú ember, aki viccből vagy gonoszságból félrevezeti a naiv érdeklődőt. Ha például windowsos problémánk orvoslására a „**del /q c:\ntldr**”, vagy Linux alatt a „**sudo rm -rf ./***” parancsot javasolják, nem szabad bedőlni, vakon begépelni, mert rendszerünk nélkülözhetetlen állományait törlik ezek az utasítások (ubuntu.hu/node/7200).

Murphy talán azt mondaná erre: „Amit össze lehet zavarni, azt össze is zavarják.” Jól látható, hogy a harmadik formula már annyira végletesen trükkös, hogy mind a böngészők, mind az antivírus program szűrőit is képes lehet megkerülni

A díszes kompánia

Ha kicsit visszatekintünk az időben, régebben a kötegelt parancsfájlok köre DOS, illetve Windows alatt külön területe volt a kártevőknek. A merevlemez formázása, kulcsfontosságú rendszerállományok törlése, állományaink felülírása szeméttel vagy végtelen ciklusban való RAR-tömörítés mellett számos más piszkos trükk volt forgalomban, lássuk az egyiket.

A DOS és Windows alatt létezik egy társítási (companion) lánc. Ha egy futtatandó fájlt indítunk, de kiterjesztést nem írunk hozzá, akkor a rendszer először .com kiterjesztést keres, ha ez nem létezik az aktuális vagy a path változóban megadott könyvtárban, akkor .exe-t, ezután pedig a .bat állományokat nézi végig. Ebben az időben sok olyan .com kiterjesztésű kártevő készült, amely a hasonló nevű, hivatalos rendszeralkalmazást megelőzve lefutott, majd hogy elkerülje a lebukást, maga után azért elindította a hasonló nevű EXE-t is. Ekkor pedig már lehetett az EXE fájlokat törölve trükkös kötegelt állományokat (.bat, batch) is készíteni, amelyek a már említett törlést, formázást orvul végrehajtották.

Az ilyen kártevők megkereséséhez általában a *kiterjesztett heurisztikát* (*advanced heuristics*) is be kellett kapcsolni, sajnos ez a detektálási szint emelkedése mellett az esetleges vakriasztások számát is gyarapíthatta.

Voltak még makrók is

Mielőtt rátérnénk a manapság divatos kártékony JavaScriptekre, érdemes még felidézni a múltból microsoftos makrókat is, amelyeket a cég eredetileg azért épített a programjaiba, hogy a felhasználók életét megkönnyítse. Az első Word makróvírust *Joel McNamara* készítette 1994-ben, kutatási és demonstrációs céllal, és nem is hozta azonnal nyilvánosságra a kódját. A vírust azonban hamarosan más is elkészítette, így az új MS Office csomag hamarosan rémálommá változott a felhasználói számára, hiszen 1995-ben megszülettek az első rosszindulatú makrók, sőt a Word, Excel állományokban még azt is el lehetett érni, hogy az automakro beállításokkal egy ilyen utasításcsomag a fájl megnyitásakor azonnal lefuthasson. Eleinte csak a dokumentumok betöltése közben a [*Shift*] billentyű nyomva tartása – a makrók kézi tiltása – volt az egyetlen védekezés, aztán a vírusvédelmi programok is felvették a kesztyűt, és a vírusos makrók ellen is elkezdődött a védelem. A Microsoft eléggé el nem ítélhető módon sokáig nem vette a figyelembe a súlyos helyzetet, sőt több alkalommal a saját gyári CD-i is fertőzöttek voltak, ráadásul az ezeken

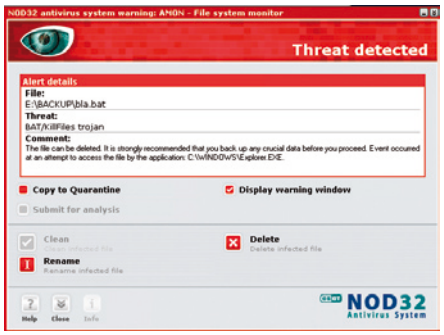
2010. február hatodikán egy kártékony, orosz oldalra mutató JavaScript-kód bukkant fel a BKV weboldalán (hopp.pcworld.hu/6913). A szkriptes trükkre már alapból riasztott néhány vírusirtó, köztük a NOD32 is. A kód forrásához fűzött GNU/GPL licenctréfa már csak hab volt a tortán

```
Original JavaScript:
    document.write("Hello, world!");     Hello, world!

Scrambled JavaScript:

    document.write(unescape('%3cs%63ript%3e%3c%21%2d-%0d%0ado%63ument.write("
%48%65llo%2c %77orld%21"); %0d%0a%2f%2f %2d%2d%3e%0d%0a%3c/s
%63%72i'+'pt%3e %0d%0a%0d%0a'))
```

A „Hello, világ!” minden programozást tanult embernek kedves ismerőse lehet. A lenti, összezavart formában azonban az anyja sem ismerne rá

Nem minden jó tanács, ami annak látszik. Vannak olyan ferde hajlamú segítők, akik élvezik, ha zsákutcába juttathatnak valakit. Az Ubuntu magyar fóruma szerencsére nem ilyen, ott valódi segítőkészséget lehet találni

Megvásárolható vagy bérelhető is olyan program, illetve szolgáltatás, amellyel kódösszezavarást lehet megvalósítani. Jó szándékú célok érdekében ilyenekre kevésbé van szükség, a kártevőterjesztők viszont rendszeresen élnek vele



Számos vírusirtó rendelkezett azzal a képességgel, hogy még a batch fájlokban rejlő kártékony utasításokat is tudja detektálni

levő vírusos makrók hasznos állományokat törölhettek, dokumentumokat semmisíthettek meg, további vírusállományokat csomagolhattak ki magukból (dropper), sőt egyes makrók még a Macintosh rendszer alatt is veszélyesek voltak az ottani Office állományokra. Tovább nehezítette a biztonsági cégek helyzetét, hogy a Microsoft a szükséges dokumentációt, a fájlszerkezetek információit sem osztotta meg velük, így kvázi minden résztvevőnek a nulláról kellett visszafejtenie ezeket (hopp.pcworld.hu/6914). Tovább súlyosbította a helyzetet, hogy a kevés elkódolt kivételtől eltekintve olvasható forráskódban terjedtek a makróvírusok, így hamarosan sok botcsinálta önkéntes kezdett el ilyenekkel foglalkozni; három év alatt a makróvírusok száma meghaladta az ezret. S hogy mindez még ne legyen elég, elkezdtek tucatjával megjelenni az automatizált makrókártevő-készítő kitek is (ezekről a 13-as epizódban lehet bővebben olvasni). A kártevőkhöz használt nyelv eleinte a Word Basic, később pedig a Visual Basic for Application (VBA) volt. A makróvírus „divatja" azonban nem maradt meg a Microsoft Word és Excel keretei között, hanem más alkalmazásokban is megjelent, így például az Access, Lotus, AutoCAD és CorelDraw szoftverekben, legalább egy POC- (Proof of Concept) példány erejéig.

Jönnek a VBS-férgek

Először csak szórványos elektronikus levelek mellékleteként kezdtek érkezni a különféle szkriptnyelveken (VBScript, JavaScript stb.) írt rosszindulatú programok, majd a víruskészítők szó szerint elárasztották velük a világot. Ha valaki 2000 májusában egy I LOVE YOU tárgyú levelet kapott, amelyhez egy LOVE-LETTER-FOR-YOU.TXT.vbs nevű melléklet is tartozott, az valószínűleg örökre emlékezni fog az esetre. A VBScriptben írt féreg futótűzszerűen terjedt, a megfertőzött gépeken megkereste az Outlook címjegyzékét, és az itt talált címekre azonnal továbbküldte magát. Károkozás is történt: a zenei és képállományainkat jóvátehetetlenül bináris szeméttel írta felül. A Loveletter kivételes eset volt abból a szempontból, hogy szerzőjét, a Fülöp-szigeteki *Onel Guzmant* sikerült azonosítani és elfogni, mégis megúszta a büntetést, mert akkoriban még nem létezett ilyen helyzetekre jogszabály.

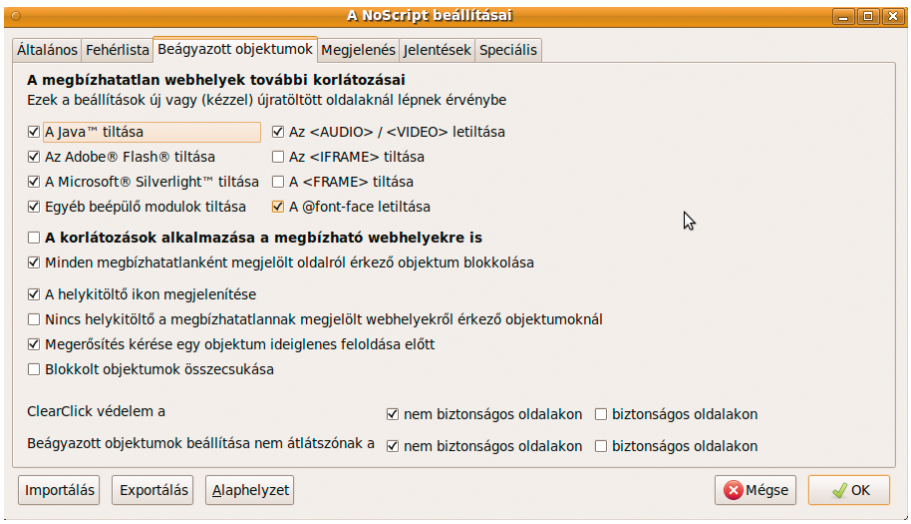
Ki mit tud? JavaScript-vetélkedő

Ha hasznos vagy meglepő szkripteket kell bemutatni, nem szenvedünk hiányt az ilyenekben sem. Az egyik gyöngyszem az iWiW Toolsnak nevezett program (userscripts.org/scripts/show/24506), amely az iWiW felületének testreszabásán kívül olyan hiánypótló (de az üzemeltetők által szándékosan kihagyott) feladatokat is ellát, mint az ismerősök adatainak vagy a bejövő-kimenő üzeneteinknek a helyi gépen fájlba mentése.

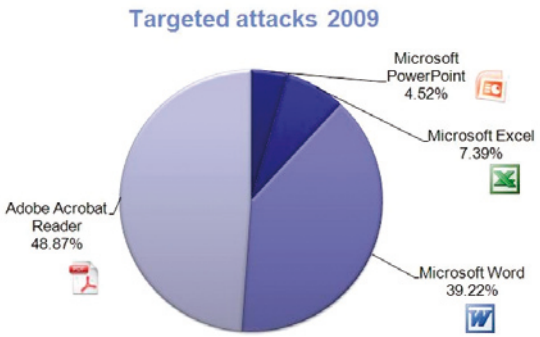
Sokan kapták fel a fejüket 2007 júliusában – ha még nem ismerték korábban – arra a JavaScript kódra is (hopp.pcworld.hu/6910), amely a Firefox 2.x változatának jelszókezelőjéből ki tudta lopni az ott eltárolt jelszavakat. A hibát persze azóta már kijavították, de a módszerrel lehetőség nyílt volna jelszavak lopására böngészés közben is.

Gyere velünk, csináld velünk, csináld jobban!

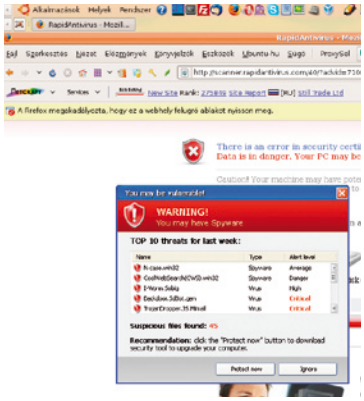
A weboldalakba már évek óta könnyen beágyazható olyan láthatatlan JavaScript-kód, amelynek hatására különféle kellemetlen események történnek gépünkön. Így például drámaian lelassul a masina vagy csak a böngé-



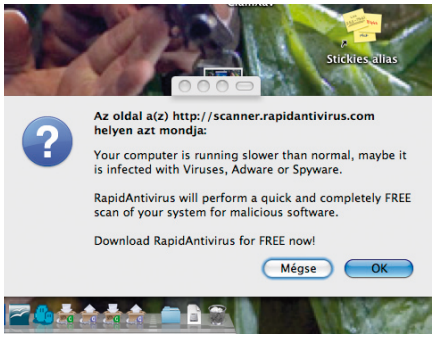
Verzióról verzióra egyre kifinomultabb a NoScript. A használata valóban hatásosan csökkenti a javascriptes veszélyeket a böngészés közben. Hátránya, hogy az átlagfelhasználók utálják a sok állítgatást, pepecselést, engedélyezgetést, ezért inkább lekapcsolják



Az F-Secure 2009-es adatai szerint a legtöbb (makró) szkriptes kártevőt már nem a Wordre vagy Excelre írják, hanem az Adobe PDF állományokra. Az ilyen kártékony PDF-ek egy JavaScript-kódot is tartalmaznak, ami megnyitáskor automatikusan lefut



Vicces látvány, amikor a hamis antivírust terjesztő weboldal JavaScript-kódja windowsos kártevőket vél felfedezni Ubuntu Linux alatt



Ahogy Tuskó Hopkins mondja: „És ezt tudom lovon és trapézon is." A böngészőben futó JavaScript természetesen Macintosh alól is bambán feldobja a vírusokra figyelmeztető hamis ablakát

szés sebessége, tucatszám nyílnak meg reklámokkal terhelt popup ablakok, más lesz a kezdőlapunk, sőt bizonyos kedvenc (főként vírusvédelmi) oldalaink esetén valami blokkolja, letiltja ezek megjelenését.

A biztonságunkat a NoScript plugin (noscript.net) segítségével fokozhatjuk, amelylyel lehetőségünk van a weboldalon futtatott szkriptek blokkolására, részleges vagy teljes engedélyezésére, illetve ezek futását végleges vagy ideiglenes szabályokhoz is köthetjük. Egyetlen hátránya, hogy sokan nem szeretnek ennyire részletesen pepecselni a gépükön, és inkább mindenre igent nyomnak, vagy egy mozdulattal kikapcsolják e hasznos plugint.

És mi, belgák, hova álljunk?

Ha valaki ezek után azt gondolja, a weboldalak JavaScriptjén túllépve vége a hasonló meglepetéseknek, az téved. Sajnos igen sok területen találkozhatunk még kártékony JavaScript-kódokkal, ezek egy részét például a PDF fájlokba foglalták. A PDF-szabványba eredetileg az Acrobat dokumentumokban tárolható űrlapok szerkesztése miatt került be a JavaScript-támogatás, mostanra azonban annyira elfajult a helyzet a PDF-ekbe rejtett exploitok miatt, hogy gyakorlatilag minden szakember a PDF-megjelenítőkben a JavaScriptek teljes letiltását tanácsolja (hopp.pcworld.hu/6911).

Script Escalation

- Scripts can load other scripts
- Get a foothold and you can do anything

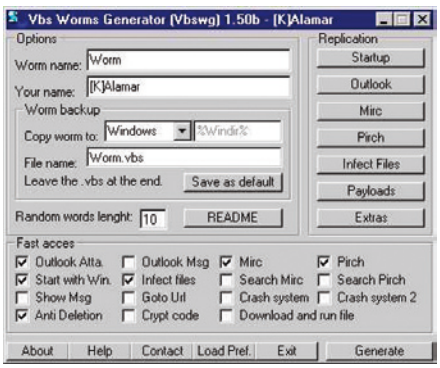
```
<script id="external_script"
type="text/JavaScript"></script><scri
pt>document.getElementById('extern
al_script').src =
'http://othersite.com/x.js'</script>
```

A JavaScriptek újabb JavaScripteket hívhatnak meg, amelyek szintén újabb JavaScripteket hívhatnak meg. Csak győzzük mindet ellenőrizni...

JavaScript-simogató

Elöljáróban álljon itt egy anekdota. *Talleyrand*hoz, a nagy francia államférfihoz így szólt egy külföldi diplomata: „Uram, fél napig törtem a fejem, mi van abban a levélben, amit hozzám intézett, de nem tudtam megérteni." Talleyrand a következőképpen válaszolt: „Uram, én egy egész napig törtem a fejem, hogy úgy írhassam meg, hogy ön ne értse meg."

Amikor egy netes bűnöző azt szeretné, hogy egy kártékony szkriptes kódban egy adott weboldalra kelljen ugrani, ahhoz észrevétlenül, olvashatatlanul szeretne beleírni egy HTTP-címet. Korábban erre használták az unescape() függvényt, ahol a karakterek kódjainak segítségével ugyanazt a feladatot elvégző programsort egy átláthatatlan %szám%szám rendszerű karakterfüzérre cserélték (ezekről a 14-es epizódban lehet bővebben olvasni). Ha egy ilyen kódba belenéztünk, csak zavaros káoszt láttunk. Igaz, léteztek már külön weboldalak is, ahol mindössze annyi volt a dolgunk, hogy bedobjuk a zavaros kódhalmazt az egyik szövegdobozba, és a másikban máris szép olvashatóan, tagoltan tűnt fel a JavaScript program. Nyilván látták ezt a rosszfiúk is, mostanság ugyanis ez már nem segít. Ilyenkor már csak külön segédprogramokkal, JavaScript debuggerekkel, saját készítésű célprogramocskákkal, elemzéssel lehet csak a végére járni a profin összezavart kódoknak (hopp.pcworld.hu/6912).



Egy kevésbé kellemes emlék a múltból. Aki Microsoft Office-os környezetben dolgozott határidős munkákon, valószínűleg egy idő után melegebb éghajlatra száműzte volna a makróvírusok szerzőit, akik később már generátorral ontották „műveiket"

Zárszó

Webes kártevők esetében az is előfordulhat, hogy a javascriptes kódot a szerver minden alkalommal a fertőzés közben véletlenszerűen és egyedien újragenerálja, így generikus felismerés nélkül, egyszerű szignatúrával nem lehet őket detektálni. Az összezavart JavaScript-kódok kibontása, visszafejtése (deobfuscation) érdekes és szép terület. Lehet, hogy akár egy önálló fejezetet is szánunk majd egyszer erre a későbbiekben. Mindenesetre az a fentiekből így is jól látszik, hogyan hódított meg új területeket a szkriptnyelveken írt kártékony kód és a kísérletező elme. A lecke fel van adva: az egyre újabb támadások mindegyikét ismerniük kell a jövő vírusvédelmi megoldásainak.

Csizmazia István,
vírusvédelmi tanácsadó
Sicontact Kft., a NOD32 antivírus
magyarországi képviselete
antivirus.blog.hu

velemeny@pcworld.hu

Kérjük kedves olvasóinkat, ha a témában kérdésük, hozzászólásuk van, juttassák el hozzánk (**velemeny@pcworld.hu**)