HADMÉRNÖK

X. Évfolyam 3. szám - 2015. szeptember

JÉRI Tamás
jeri.tamas@bv.gov.hu

A WEB SZEREPE A KRITIKUS INTERNETES SZOLGÁLTATÁSOKBAN

Absztrakt

A kritikus internetes szolgáltatások egyik leggyakoribb előfordulási helye a Web, amely adottságai miatt optimális környezet az információk megjelenítésére, gyűjtésére, tárolására és továbbítására. Az állandó rendelkezésre állás a bizalmasság és a sértetlenség fenntartása mellett, a Weben is komoly kihívást jelent. Jelen írás azt foglalja össze, hogy a kritikus internetes szolgáltatások Web-es megjelenésekor milyen intézkedéseket kell, vagy lehet tenni a biztonság fenntartásához.

The Web is one of the most common occurrence place of critical internet services, which is optimum environment for information displaying, collecting, storing, transmitting, because its capability. The constant availability, confidentiality and integrity are serious challenges on the Web too. This paper summarizes, that what measures should or can be done for maintaining security, while critical internet services displayed on the Web.

Kulcsszavak: *Internet, Web, biztonság ~ Internet, Web, security*

BEVEZETÉS

"Tim Berners-Lee, brit tudós 1989-ben, a CERN-ben megalkotta a World Wide Web-et (a továbbiakban: Web). Az első honlapot a CERN-ben - és egyben a világon is - Berners-Lee NeXT számítógépe szolgáltatta és magához a World Wide Web projekthez jegyezték. A honlap a Web alapvető jellemzőit írta le[1]; hogyan lehet hozzáférni más emberek dokumentumaihoz és hogyan kell beállítani egy saját szervert. A CERN a World Wide Web programját 1994. április 30-án nyilvánosan elérhetővé tette, majd a következő kiadást nyílt licensz-el tette elérhetővé, biztosítva a terjesztés maximalizálását. A futtatáshoz szükséges szabadon elérhető webszerverrel, valamint egy alap böngészővel és egy (forrás)kód könyvtárral, a Web virágzása biztosított volt."[2]

Tim 1990 októberében három alapvető technológiát írt le[3], amelyek a mai Web alapjai is: HTML[4], URL[5], HTTP[6].

A Web fejlődése az első honlap megjelenése óta kétséget kizáróan töretlen, napjainkban a legtöbbet használt kommunikációs eszköze az internetnek és a kritikus internetes szolgáltatásokhoz (a továbbiakban: KRISZ [1]) is számos ponton kapcsolódik. A kezdeti honlapok nyilvánvalóan összehasonlíthatatlanok a mai modern, összetett, bonyolult, programozott weblapokkal, ezért a fejlődés menetében azokat a pontokat célszerű kiemelni, amelyek megalapozták a Web és a KRISZ közös jövőjét.

Az innováció követésében, a szerver-kliens modell analógiájára, érdemes különválasztani az információt nyújtó szolgáltatói-, és az azt igénybe vevő felhasználói oldalt azzal a tényszerű kiegészítéssel, hogy a két szegmens egymást a fejlődésben folyamatosan indukálta, erősítette. A Web evolúciójával foglalkozó kutatások, illetve a publikált számszerűsíthető eredmények jobbára a kliens oldalon bekövetkezett változásokat mutatják, ugyanakkor a KRISZ jellegéből adódóan a szerveroldali fejlődést is fontos bemutatni. A "National Center for Supercomputing Applications" (NCSA) "HTTP daemon" nevű programja volt a vezető webszerver 1995. februárjában, azonban a fejlesztő csapatból kilépett programozók 1995. december 1-én kiadták az "Apache" nevű webszerver hivatalos verzióját, amely rövid időn belül az elsődleges webkiszolgálóvá vált.[7] A Web szempontjából mérföldkőnek számító, első dinamikus weboldalak kialakulását nehéz meghatározni[8], ugyanakkor az Apache 1.1 béta 3 verziójának 1996. június 14-én megjelent leírásában[9] egyértelműen fellelhetők a CGI[10]-re vonatkozó utalások. A CGI mintegy csatlakozási felületként működik a külső programok és a webszerver között a dinamikus tartalmak előállításához; a leírásban utalást találhatunk[11] a másik nagyon fontos területre, az adatbázis-kezelésre is. Az Apache, az 1.3a1 verziótól - azaz 1997. őszétől - kezdve, a Unix után már Windows NT-n is futtatható vált, lefedve az operációs-rendszer platformok döntő többségét. A 2000-es évektől a webszerverekbe - nélkülözve a CGI-t - közvetlenül integrálhatókká váltak a dinamikus weboldalak előállításához és az adatbázisok kezeléséhez szükséges modulok, kiegészülve számos biztonsági, technológiai és kényelmi szolgáltatással. Az oldalak előállítását biztosító programozási nyelvek dinamikusan fejlődtek és az egyéb szerveroldali szolgáltatások Web-re történő implementálása is folyamatos volt.

---

[1] http://info.cern.ch/hypertext/WWW/TheProject.html
[2] http://home.web.cern.ch/topics/birth-web - The birth of the web (A web születése) [2]
[3] http://webfoundation.org/about/vision/history-of-the-web/
[4] HyperText Markup Language - hiperszöveges jelölőnyelv
[5] Uniform Resource Locator - egységes erőforrás-azonosító
[6] HyperText Transfer Protocol
[7] http://httpd.apache.org/ABOUT_APACHE.html
[8] https://en.wikipedia.org/wiki/Dynamic_web_page#History
[9] http://www.apacheweek.com/issues/96-06-14
[10] Common Gateway Interface
[11] "MSQL authentication module improvements,"

Napjainkban, köszönhetően a kialakult versenyhelyzetnek, a kiszolgálói oldalon számos - ingyenes - alternatíva közül választhatnak a webszervert üzemeltetők.

A felhasználók kegyeiért folytatott böngésző-verseny azonnal elkezdődött, amint a fejlesztők belátták a Web töretlen népszerűségéből húzható hasznot. A Különböző gyártó-, és fantázianevek folyamatosan bukkantak fel, vagy tűntek el, a legfőbb rendező elv pedig maga a felhasználói tömeg volt, amely saját igényeinek és lehetőségeinek megfelelően választotta ki a böngésző(ke)t. A fejlesztések és a felhasználói igények abba az irányba vezettek, hogy napjainkban szoftver óriások szállítják a legkorszerűbb böngésző programokat, amelyek

- felhasználóbarát, kényelmes GUI[12] felülettel rendelkeznek;
- törekednek a szabványokban rögzített feltételek teljesítésére;[13]
- támogatják a szkript-nyelvek és a beépülő programok futtatását;
- képesek az információ minden típusú megjelenítésére (szöveg, kép, hang, stb.);
- több platformon is elérhetők (a szállító üzleti szempontjainak függvényében);
- erős védelmi rendszerrel rendelkeznek, törekednek a biztonságra;
- ingyenesek.

A Web fejlődésébe a kiszolgáló és a felhasználói oldal mellett további, külső tényezők is szerepet játszottak, amelyek együttesen megteremtették a KRISZ, Web-es alkalmazási lehetőségét. Az infrastrukturális fejlődés növelte az internetelérés sávszélességét, nőtt az adattárolási kapacitás-, és biztonság, valamint meredeken emelkedett a mikroprocesszorok számítási teljesítménye.

A kereskedelem gyorsan felmérte a Web-ben rejlő lehetőségeket, hisz az első on-line vásárlás a feljegyzés szerint[14] - 1994-ben - már biztonságos SSL csatornán keresztül zajlott, majd 1995-ben megnyitottak a ma is legnagyobbnak számító webáruházak. A Web rövid időn belül a pénzügyi tranzakcióknak is teret adott, valamint a kereskedelmet mozgató reklámoknak is felületet biztosított. Egyirányúan erősödött és erősödik az a tendencia, hogy az emberek - megtakarítva rengeteg időt és energiát, - ügyeiket kényelmesen az Interneten, azon belül is legfőképpen a Web-en intézzék.

A Web alapjai a fejlődés során mindvégig megmaradtak, viszont létrejött egy komplex IT[15] környezet, amely megfelelő képességgel rendelkezik az információk széles spektrumú, nagy megbízhatóságú továbbítására, hordozására.

KRITIKUS INTERNETES SZOLGÁLTATÁS A WEB-EN

A Web, adottságaiból adódóan megfelelő környezet a KRISZ üzemeltetéséhez, ezért kedvelt információs platformja a szolgáltatóknak. Megvalósítása kétféle lehet; a KRISZ egyrészt működhet kizárólagosan a Web-en, ekkor a webszerver az egyedüli kiszolgáló, másrészt üzemelhet kiegészítő alkalmazásként, mely esetben a webszerver egy másik kiszolgálóhoz kapcsolódva rész-, vagy támogató szolgáltatást nyújt. A Web-re készített KRISZ igénybevétele vagy valamilyen böngészővel, vagy a HTTP(s) protokoll(oka)t ismerő egyéb programmal történik. A böngésző, vagy az alkalmazás típusa, alkalmazott operációs rendszere - a bevezetőben felsorolt funkciók rendelkezése állása esetén - nem releváns, tetszőleges.

Ahhoz, hogy a KRISZ Web-re kerüljön, teljesülnie kell a definíciókban megfogalmazott elvárásoknak: a szolgáltatást igénybe vevő felhasználók számától függetlenül, - feltételezve a megfelelő infrastrukturális és hardveres hátteret, - az állandó rendelkezésre állást a bizalmasság

[12] grafikus felhasználói felület
[13] https://hu.wikipedia.org/wiki/Acid-tesztek
[14] https://en.wikipedia.org/wiki/Online_shopping
[15] Information Technology - információtechnológia

és a sértetlenség megvalósulásával kell biztosítani. Ezen elvárások abszolválása összetett, számos biztonsági intézkedés foganatosításával érhető el, amelyeket operációs rendszer-, webszerver-, valamint alkalmazói program szintjén kell-, vagy lehet megvalósítani.

Az alább felsorolt ajánlások, a KRISZ biztonságos üzemeltetésével kapcsolatban megfogalmazott általános irányelvek mellett érvényesek. [3]

Stabilitás, hibatűrő képesség

Ahhoz, hogy a Web-en megjelenő KRISZ stabilitása megfelelő legyen, a közreműködő egységeknek is az elvárt hibatűrő képességgel kell rendelkezniük, hisz a KRISZ a leggyengébb láncszem szerint lesz ellenálló. Az ismert hibák és támadási pontok kiküszöböléséhez minden szinten kötelező a biztonsági frissítések és a szükséges karbantartások elvégzése.

A webszerver, a Web-en alkalmazott KRISZ nélkülözhetetlen komponense, az alkalmazás kiszolgáló és végrehajtó motorja, ezért kijelölése felelősségteljes döntés.

Webszervernek egy megbízható, már bizonyított, hatékony erőforrás-gazdálkodással bíró, kellő támogatással rendelkező programot célszerű választani. Figyelembe kell venni, hogy a tesztelés, vagy kipróbálás alatt álló programok még számos hibát tartalmazhatnak, ezért kiszolgálónak stabil kategóriába sorolt verziót ajánlott telepíteni. A HTTP(s), kérés-válasz alapú felépítésére tekintettel, a stabilitást elsősorban a kiszolgáló kérésekre adott reakcióinak szabályozásával lehet elérni, megőrizni.

A kérés(ek) feldolgozásában lényegi szerepet játszik a webszerver alá rendelt további kiszolgálók típusa és száma, ugyanis a - KRISZ-re jellemző - dinamikus weboldalak elő-feldolgozó programmal készülnek, tartalmuk pedig gyakran adatbázisból származik. A webszerver tűréshatárát - a rendelkezésre álló infrastruktúra és hardver figyelembevétele mellett, - az egységnyi idő alatt kiszolgálandó ügyfelek számának-, és a kiszolgálásra fordítandó időnek a szabályozásával szükséges korlátozni. Kiemelt figyelmet érdemel az alrendszeri kiszolgálók [4] pontos és körültekintő, erőforrás orientált konfigurációja, amellyel a túlterhelés megelőzhető és az áthárított feladatvégzés normalizálható.

Biztonság

A webszerver moduljait a biztonság és a szükségszerűség figyelembevételével kell be-, vagy kikapcsolni, amellyel elérhető, hogy csak a nélkülözhetetlen, vagy a biztonságot támogató programkódok kerülhessenek végrehajtásra, a feleslegesek, vagy kockázatosak pedig inaktívak maradjanak.

Minden webszerver konfigurációjában meg kell határozni a könyvtárstruktúra tetejét kijelölő, úgynevezett "DocumentRoot"[16] könyvtárat, amely egyben az összes URL[17] hivatkozás kiinduló pontja is. A könyvtárstruktúrában a KRISZ, mint alkalmazás is helyet kell, hogy kapjon, s mellette számos olyan állomány és adat is, amelynek illetéktelen kézbe kerülése veszélyt jelent. A webszervert úgy kell konfigurálni, hogy a legkevésbé támogassa a felhasználók által kért mappák, vagy állományok tartalmának megjelenítését, azokat tartsa rejtve.

Az összeköttetés biztonságának megvalósítása céljából, 1995-ben megalkották az SSL[18]-, majd annak továbbfejlesztéseként 1999-ben a TLS[19] nevű biztonsági programcsomagot [5], amellyel megbízható HTTP, azaz a HTTPS protokoll alkalmazására nyílt lehetőség. Napjainkban minden komolyabb webszerver támogatja a használatát, a beállítás és konfigurálás

[16] dokument-gyökér
[17] Uniform Resource Locator - egységes erőforrás azonosító
[18] Secure Sockets Layer
[19] Transport Layer Security

pedig a dokumentáció(k)[20] alapján könnyen kivitelezhető. A KRISZ jelenléte esetén, szinte kötelező a biztonságos átvitel webszerver általi támogatásának beállítása.

Az Interneten található, több milliónyi weboldalt prezentáló szervergépek száma töredéke a fellelhető oldalaknak, amelyből következik, hogy egy-egy számítógép - az úgynevezett "VirtualHost" technológiának köszönhetően - több weboldal egyidejű kiszolgálását biztosítja. Feltételezhető, hogy a KRISZ működéséért felelős webszerver adott esetben további Domain[21] nevekhez tartozó weboldalakat is kiszolgál, amely azonban elővigyázatosságra ad okot. A KRISZ-en kívüli, tetszőleges weboldalról bekövetkező támadás esetén, könnyen a KRISZ is áldozattá válhat. Amennyiben az illegális behatoló a webszerver adott könyvtárában írási jogosultsághoz jut és oda saját, szerver által végrehajtható programkódot képes feltölteni, úgy - a webszerver felhasználójának jogosultságával, - a birtokba vett könyvtárból elérhető mappaszerkezetre, rálátással bírhat és könnyen módosításokat is végezhet. Fontos kiemelni tehát, hogy a KRISZ programkódját szeparáltan kell elhelyezni a fájlrendszeren, majd kapcsolni a webszerverhez, hogy egyéb weboldalakról érkező támadás esetén is védett legyen.

A Web-en megjelenő KRISZ esszenciáját az a forrás(program) adja, amelyet a webszerver képes értelmezni, szükség esetén feldolgozni és a kliens részére a szabványok szerinti formában továbbítani. Ezen kódok a kezdetek óta nagy változáson mentek keresztül, hiszen amíg a Web-et először statikus oldalak alkották, addig napjainkra bonyolult programrendszerek állítják elő a világhálón megjelenő gigantikus információhalmazt. Temérdek előre megírt forráskód és több tucat ingyenes keretrendszer áll rendelkezésére az érdeklődőknek, akik költséghatékony eszközökkel is képesek -, akár kritikus internetes szolgáltatás nyújtására alkalmas, - saját weboldalt létrehozni. A KRISZ, Web-es rendszerbe állításától kezdődően azonban, elengedhetetlen néhány biztonsági intézkedés foganatosítása.

- Törekedni kell az információszegény URL-ek alkalmazására és a kockázattal járó elérési pontok elrejtésére. Biztonságosabb, ha a böngészés során a felhasználók nem látják, hogy a hivatkozásokban milyen paraméterek, változók és értékek kerülnek átadásra, hisz azok ismeretében számos - adatbázisra és működésre vonatkozó, - logikai következtetés vezethető le.
- Az adminisztráció a dinamikus weboldalak egyik megkerülhetetlen feladata. Fontos megjegyezni, hogy az adminisztrátor az adatokra és gyakran a programkódra vonatkozóan is írási jogosultsággal bír, ezért e lehetőség illetéktelen kézbe kerülése végzetes lehet. Elvárás tehát, hogy az adminisztrációra mutató hivatkozás a normál felhasználók számára ismeretlen, vagy elérhetetlen legyen. Tipikus hiba a "/admin", mint alapértelmezett adminisztrációs belépési pont nyitva hagyása, amely tálcán kínálja a belépés és a támadhatóság lehetőségét.
- A programozás íratlan szabályi szerint, a kód mennyiségével egyenes arányban az elkövetett hibák száma is növekszik. Egy KRISZ esetében, a feltárt hibá(ka)t a lehető legrövidebb időn belül javítani-, vagy kiadott biztonsági csomag esetén a programot frissíteni szükséges. A hibaüzenet kezelést éles üzemben ki kell kapcsolni és lehetőség szerint kerülni kell a bonyolult, átláthatatlan, erőforrásokat felemésztő adatbázis műveleteket. A nagyobb (keret)rendszerekre jellemző, gyártó és verziószámra utaló megjegyzéseket ki kell iktatni, elkerülve, hogy a rosszindulatú felhasználók annak ismeretében indítsanak támadást, vagy keressenek alapértelmezett, a programra jellemző beállításokat.
- A webszerver titkosított kapcsolat létesítésére vonatkozó alkalmassága esetén, a bizalmas információk továbbítására kötelező a HTTPS protokoll kikényszerítése. Ezek közé sorolandó a weboldal adminisztrálásával-, a pénzügyi tranzakciókkal-, a

[20] http://httpd.apache.org/docs/2.2/ssl/
[21] tartomány

felhasználók kezelésével-, és a felhasználók megszemélyesítésével kapcsolatos műveleteket. Titkosítás hiányában ezen kommunikációk lehallgathatók és a megszerzett információk könnyedén támadási eszközzé konvertálhatók.

– A nem várt események kezelése és a szolgáltatásban bekövetkező kimaradási idő minimalizálásának érdekében, a webprogramról és az adatvagyonról minél gyakoribb rendszerességgel mentéseket kell végezni és azt a kiszolgálótól eltérő helyen érdemes tárolni. A weboldalra érkező kéréseket lehetőség szerint naplózni kell, amit rendellenesség esetén ki lehet értékelni.

Társszolgáltatások kezelése

Függetlenül attól, hogy a KRISZ működésében résztvevő webszerver fő-, vagy támogató kiszolgáló, elengedhetetlen a funkcionáló szerverprogramok közötti megfelelő összhang. A levelező-, adatbázis-kezelő-, és cache kiszolgálók a Web gyakori -, általában függőségi viszonyban is álló - együttműködői, amelyből következik a precíz konfiguráció szükségessége. A webszerverrel mellé-, vagy alárendeltségben álló szolgáltatások együttesét úgy kell üzemeltetni, hogy a KRISZ rendelkezésre állása ne forogjon kockán, az mindig biztosított legyen.

ÖSSZEGZÉS, KÖVETKEZTETÉSEK

Dolgozatomban a kritikus internetes szolgáltatások Web-es megjelenését vizsgáltam. Megállapítható, hogy az elmúlt több mint két évtized dinamikus IT fejlődésének köszönhetően, a Web, az Internet meghatározó információs platformjává vált. A virtuális tér, mint lehetséges megjelenési pont, napjainkban társadalmi normaként, elvárásként van jelen és kihasználása standarddá vált. A hatalmas népszerűség tömegeket vonz a Web-kompatibilis eszközök (számítógép, telefon, tablet stb..) elé és oltja az állandóan fellobbanó információs éhséget. Az áldozatul esett weboldalak azonban jól mutatják, hogy egy hozzáértő(bb) kört kifejezetten szórakoztat a hibák, vagy figyelmetlenségek kihasználása és a károkat elszenvedők bosszantása. A kritikus internetes szolgáltatásokkal kapcsolatos elvárásokat figyelembe véve, személyes tapasztalataimból is merítve, kísérletet tettem a Web biztonságával kapcsolatos általános intézkedések összefoglalására. Természetesen tovább is lehetne boncolgatni a témával összefüggő lehetőségeket, hisz a tökéletes biztonságot elérni nem-, csak megközelíteni lehet. A jelen íráshoz szorosan kapcsolódó kliens oldali biztonság, terjedelmi okok miatt egy másik tanulmány részét fogja képezni.

Felhasznált irodalom

[1] Jéri Tamás - Kritikus Internetes Szolgáltatások. Hadmérnök, VIII. Évfolyam 1. szám 2013. március, NKE Budapest, ISSN 1788- 1919. http://hadmernok.hu/2013_1_jerit.pdf - letöltve 2014.01.25

[2] The birth of the web (A web születése). http://home.web.cern.ch/topics/birth-web - letöltve 2015.07.06

[3] Jéri Tamás - A kritikus internetes szolgáltatások biztonságos üzemeltetése. Hadmérnök, X. Évfolyam 1. szám 2015. március, NKE Budapest, ISSN 1788- 1919. http://hadmernok.hu/151_20_jerit_2.pdf - letöltve 2015.07.06

[4] Jéri Tamás - A kritikus internetes szolgáltatások alrendszerei. Társadalom és Honvédelem, 2013/3-4. szám, NKE Budapest, ISSN 1417-7293

[5] Andrew S. Tanenbaum: Számítógép-hálózatok, ISBN 963 545 384 1. Panem Könyvkiadó Kft., Budapest 2004.