Dunakavics

A Dunaújvárosi Főiskola online folyóirata 2015. III. évfolyam VIII. szám

Műszaki-, Informatikai és Társadalomtudományok

Dunakavics

A Dunaújvárosi Főiskola online folyóirata 2015. III. évfolyam VIII. szám

Műszaki-, Informatikai és Társadalomtudományok

Megjelelenik évente 12 alkalommal

Felelős szerkesztő Németh István
Tördelés Duma Attila

Szerkesztőség és a kiadó címe 2400 Dunaújváros, Táncsics M. u. 1/a.

Kiadja DUF Press, a Dunaújvárosi Főiskola kiadója
Felelős kiadó András István, rektor

A lap megjelenését támogatta TÁMOP-4.2.3-12/1/KONV-2012-0051
„Tudományos eredmények elismerése és disszeminációja
a Dunaújvárosi Főiskolán".
http://dunakavics.duf.hu

ISSN 2064-5007

Tartalom

Dunakavics - 2015 / 8.

OROSZI ESZTER DIÁNA *

Kártékony programok terjedése social engineer szemmel

Összefoglalás: Az információbiztonság leggyengébb láncszeme, az emberi tényező. Jelen dokumentum célja annak bemutatása, hogy egy potenciális támadó milyen Social Engineering technikákat alkalmazhat kártékony programok terjesztése során. A terjesztési lehetőségeken kívül ismertetésre kerülnek a kártékony kód lefutásának módszerei, valamint a fájlok álcázásának lehetsőségei is.
Kulcsszavak: Social Engineering, emberi tényező, információbiztonság, biztonságtudatosság, kártékony program, fertőzött fájl.

Abstract: Human factor is the weakest link of the chain of security. The goal of this paper is to present how can an attacker use Social Engineering tricks to malware spreading. Besides these methodologies I will show the techniques of running malicious scripts and the opportunities of masking infected files, too.
Keywords: Social Engineering, human factor, information security, security awareness, spreading malware, infected files.

* *GRID CEE Tanácsadó Zrt.*
E-mail: oroszi.eszter@grid.co.hu

A Social Engineering rövid bemutatása

A Social Engineering az emberi tényező kihasználható tulajdonságaira építő támadási forma, tulajdonképpen olyan technikák gyűjteménye, mely az emberek befolyásolására, manipulálására alapozva teszi lehetővé bizalmas információk megszerzését, vagy éppen egy kártékony program terjedését és működését.

Legmegfelelőbb definíciója a következő:
„A social engineering a befolyásolás és a rábeszélés eszközével megtéveszti az embereket, manipulálja, vagy meggyőzi őket, hogy a social engineer tényleg az, akinek mondja magát. Ennek eredményeként a social engineer – technológia használatával vagy anélkül – képes az embereket információszerzés érdekében kihasználni.” [1]

Az ilyen jellegű támadások veszélye abban rejlik, hogy az emberi tényező a legtöbb védendő értékhez (adat, alkalmazás, hardver, stb.) közvetlenül hozzáférhet, ezáltal mint a biztonság leggyengébb láncszeme, vonzó célponttá válhat egy támadó szemében. Ezen kívül elmondható, hogy az emberek számtalan olyan tulajdonsággal rendelkeznek, melyeket egy támadó könnyen ki tud használni. Ezek közé tartozik a segítőkészség, mely az egyik legalapvetőbb emberi tulajdonság, amit a social engineerek gyakran ki is használnak, de hasonló kategóriát képez a kíváncsiság, hiszékenység, naivság, melyre különösen adathalász támadások és kártékony programok beküldése során lehet építeni. A támadások során építeni lehet továbbá a felhasználók figyelmetlenségére, hanyagságára, illetve bizonyos értelemben tudatlanságára is.

A Social Engineering módszerek két nagy csoportba sorolhatók: egyik kategóriát képezi a humán alapú módszerek gyűjteménye, másikat pedig a számítógép alapú technikák. [2]

Az alábbi ábra összefoglalja az egyes technikák kategóriákba sorolását:

Humánalapú támadások	*Számítógépalapú támadások*
– Segítség kérése	– Spam
– Segítség nyújtása	– Ál-weboldalak
– „Valamit valamiért”	– Phishing (klasszikus adathalászat)
– Megszemélyesítéses támadások személyes megjelenés során	– Vishing (telefonos adathalászat)
– Megszemélyesítéses támadások telefonon keresztül	– Smishing (SMS-en,csevegőprogramon keresztüli adathalászat)
– Shoulder surfing	– Pharming (eltérítéses adathalászat)
– Tailgating	– Whaling (vezetők ellen irányuló adathalászat)
– Piggybacking	– Baiting (adathordozó szétszórás)
– Dumpster diving	

[1] Mitnick K. D.–Simon W. L. (2003): *A legendás hacker – A megtévesztés művészete.* Budapest: Perfact.

[2] Guenther, M. (2001): Social Engineering – Security Awareness Series előadásanyag.

Írásom a továbbiakban a számítógépen keresztüli támadásokkal foglalkozik, azon belül is azokat a módszereket vizsgálja, melyek lehetővé teszik a kártékony programok terjesztését az emberi tényező megtévesztésével. 4 főbb, Social Engineering technikákra alapozó módszert különböztethetünk meg:
– Kártékony program letöltése valamilyen weboldalról, tárhelyről.
– Kártékony program megnyitása elektronikus levél csatolmányaként.
– Kártékony program futtatása valamilyen adathordozóról.
– Kártékony program elindítása más személy általi ráhatás során.

Kártékony program letöltése

Az egyik leggyakoribb eset, amikor a kártékony programot különböző kétes eredetű, különféle letöltéseket biztosító oldalakról töltik le a gyanútlan internetezők. Ezek az oldalak általában ingyen kínálnak csábító képeket, videókat, zenét, egyéb népszerű tartalmat. Azonban, amikor ezeket az oldalakat a mit sem sejtő áldozat megtekinti, vagy tartalmát letölti, egy kártékony program is letöltődik, illetve feltelepül a gépére.

A kártékony programokat terjesztő weboldalak forrása lehet valamilyen keresőszolgáltatás, banner, hirdetés, de érkezhet a linkjük elektronikus levélben, fórumon, közösségi portálon (pl. Facebook post, üzenet, stb.), de akár azonnali üzenetküldő programon keresztül (pl. MSN-en való csevegés közben) is.

Weboldal látogatása során észrevétlenül letöltődik és lefut

A legrosszabb esetben a kártékony kód lefutásához elegendő az azt terjesztő weboldal meglátogatása – a felhasználónak nem kell semmilyen bővítményt engedélyeznie, semmilyen fájlt letöltenie, egyszerűen csak meglátogatni az adott oldalt, a kártékony program automatikusan és észrevétlenül megkezdi működését, a felhasználónak gyakorlatilag nincsen lehetősége előzetesen észlelni és megakadályozni a támadást. Erre az egyik legjobb példa a 2008. augusztusában terjedő Antivirus XP 2008 nevű ál-vírusirtó volt.

Weboldal látogatása során engedélyezés után letöltődik és lefut

Egy fokkal szerencsésebb annak az esete, amikor a kártékony kód nem tud automatikusan lefutni az oldal meglátogatásakor, hanem valamilyen módon a felhasználó jóváhagyása szükséges a lefutáshoz, például az áldozatnak engedélyeznie kell a felugró ablakban a kód futását (pl. Java-alkalmazások esetében).

[3] Norton (2012): *Kapja el a kémprogramokat, mielőtt azok kárt tehetnének*. http://hu.norton.com/catch-spyware-before/article.

Fertőzött fájl letöltése

Előfordulhat az a szituáció is, amikor a kártékony kód az oldalról letöltött más fájlhoz kapcsolódik, és azzal együtt kerül észrevétlenül letöltésre, a felhasználó tudta nélkül. Ilyen elven működnek a klasszikus vírusok, melyek terjedéséhez valamilyen „gazdafájl" szükséges. Social Engineering szempontjából ennek a fájlnak kell valami vonzó tulajdonsággal rendelkeznie, mely arra buzdítja a felhasználót, hogy töltse le, például valamilyen új film, slágerlistát vezető zene, érdekes program vagy játék, stb. formájában.

Letöltés után a telepítés során a felhasználó beleegyezésével feltelepül

Elsősorban reklámprogramok, vagy más kiegészítő alkalmazások (pl. toolbar) esetében jellemző, hogy a felhasználónak lehetősége nyílik egy, egyébként fizetős szoftver ingyenes használatára, amennyiben hozzájárul, hogy az alkalmazás mellett egy, legális esetben csak reklámprogram települjön, amely pl. a felhasználó internetezési szokásairól küld adatokat a kereskedőnek vagy fejlesztő cégnek, annak érdekében, hogy minél célzottabb hirdetésekkel keressék meg az áldozatot. [3]

Legtöbb esetben ezen kiegészítő programok települését egy checkbox kipipálásával maga a felhasználó döntheti el, előfordul azonban, hogy a kéretlen program installálása az eredeti szoftver licencszerződésnek apróbetűs részében kerül rögzítésre, melyet a felhasználók többsége nem olvas el.

Kártékony kód terjesztése elektronikus levél csatolmányaként

A másik kedvelt kártékony program terjesztési módszer az e-mail mellékletként való küldés. Ezeknek a leveleknek a tárgya általában valamilyen csábító téma, lehet például játék, sport vagy szexuális tartalom, stb. – hogy a célszemély biztosan megnézze a melléklet tartalmát. Ily módon terjedő ismertebb kártevők voltak az I Love You- és az Anna Kournikova-vírusok.

Napjainkban, mivel az emberek többsége hallott már a különféle szenzációkra hivatkozó, fertőzött melléklettel rendelkező e-mail-ekről, az igazán kifinomult támadók leginkább teljesen hétköznapi levélnek és csatolmányának álcázzák a küldött kártékony programjukat, mint tették azt például a „MyParty" nevű féreg készítői is 2005-ben.

Reklám vagy figyelemfelkeltő levél

A legegyszerűbb módszer a kártékony programok terjesztésére, ha a támadó valamilyen egyszerű, mindenkit érdeklő reklámlevélben célozza meg áldozatait. Mint ahogyan a reklámlevelek általában, ezen levelek sem egy konkrét személytől, hanem valamilyen kitalált szervezettől, vagy valós cég nevében érkeznek. A módszer előnye, hogy a támadás előkészítése viszonylag kevés időt és erőfeszítést igényel, ugyanakkor elég széles körű tömeget lehet megcélozni vele. Témája lehet valamilyen szenzáció (pl. csábító képek, ingyenes zene és filmletöltés kínálata), vagy akár teljesen hétköznapi dolog is, például egy új pizzéria akciója.

Eltévedt levél

Előfordulhat azon eset is, amikor a támadó egy ismeretlen személytől érkező, rossz címre küldött levélnek álcázza megkeresését. Ebben az esetben megpróbálja felkelteni az áldozata kíváncsiságát a másnak szóló bizalmas levéllel kapcsolatban. Küldhet pl. képet, vagy akár egy megígért bizalmas munkadokumentumot, de akár jelszót és elérési információkat egy ingyenes letöltést biztosító oldalhoz is.

Idegen, de hihető személytől érkező levél

A harmadik esetben a támadó egy valós, hiteles személynek vagy szervezetnek tűnő feladó nevében küldi el megkeresését, melynek célja lehet első lépésben akár csak a kapcsolat kiépítése. Ezen kategória inkább a célzottabb támadások közé sorolható, hiszen a megkeresés tartalma elsősorban az áldozat személyéhez, munkájához, vagy érdeklődési köréhez kapcsolódik. A támadás során előfordulhat, hogy több levélváltás is történik a felek között, és a kártékony kódot tartalmazó csatolmány vagy link csak egy későbbi megkeresésben jelenik meg.

Ismerőstől érkező levél

A leghihetőbb forgatókönyv, ha egy ismerőstől érkezik a kártékony programot tartalmazó levél. Ez több módon valósulhat meg:

Cím hamisítás. A támadó egy ismerős személy nevében hamisít elektronikus levelet, és úgy küldi el a kártékony kódot tartalmazó mellékletet. A támadás megtervezéséhez remek alapot adnak a különféle közösségi portálok, ahonnan lehet tudakozódni lehetséges ismerősökről, illetve lehetséges példákról.

Új emailcím készítése. Természetesen nem feltétlenül szükséges emailcím-hamisítással bajlódni, egy ingyenes tárhelyen készített fiók is tökéletesen megfelelő lehet a fájlok beküldésére – hiszen bárkinek bármennyi e-mail fiókja lehet különböző ingyenes szolgáltatónál, lehetetlen lenyomozni, hogy az adott fiók tényleg az illetőhöz tartozik-e.

Automatikus továbbküldés kontaktoknak. Sok esetben maga a kártékony program is tartalmazza azon funkciót, hogy automatikusan, észrevétlenül továbbítsa magát az áldozat címtárában található ismerősöknek – ezesetben széles körben, kevésbé célzottan hajtható végre a károkozás. Ilyen támadásra példák az I love you-, illetve az Anna Kournikova-vírus.

Felhasználó meggyőzése a továbbküldésről. Lehetőség van arra is, ha magát a felhasználót győzte meg a támadó arról, hogy továbbítsa ismerőseinek a kapott fájlt – a lánclevelekhez, hoax-okhoz hasonlóan, például nyereményt ígér, amennyiben elküldi a csatolt képet 10 ismerősének.

Kártékony kód terjesztése adathordozón

A kártékony programok terjesztésének harmadik nagy kategóriája, amikor valamilyen adathordozóra írja ki a támadó a kártékony kódot tartalmazó fájlt. Ez lehet CD/DVD, vagy pendrive, sőt memóriakártya, MP3-lejátszó vagy akár egy fényképezőgép is. Az, hogy a támadás során milyen adathordozók kerülnek alkalmazásra elsősorban a célszemélyek köre, illetve az alkalmazott forgatókönyv határozza meg. A CD/DVD előnye elsősorban az ára, hiszen egy optikai adathordozó lényegesen olcsóbban szerezhető be mint egy pendrive. Észlelés szempontjából szintén pozitív tulajdonsága, hogy az adathordozó felületén fel lehet tüntetni a tartalmára vonatkozó leírást, képet, mely ösztönzi a felhasználót a tartalom megtekintésére. A pendrive-ok elszórása ugyan költségesebb megoldás, előnyösebb lehet azonban olyan helyzetben, amikor a kártékony kód lefutásához elegendő az eszköz csatlakoztatása a számítógéphez. Becsületesebb felhasználó már csak abból a célból is szeretné megtekinteni a pendrive tartalmát, hogy esetleg abban rábukkanhat az adathordozó jogos tulajdonosára, de motiválhatja a megtalálót az is, ha valamilyen feltűnőbb, nagy tároló kapacitású eszközt talál, melyet később saját célra használhat.

Elhagyott adathordozó

A kártékony programok terjesztésének egyik nemrégiben megjelent módszere a „Road Apple"-nek vagy baiting-nek nevezett technika. Ekkor a támadó egy fertőzött adathordozót (CD, DVD, pendrive, MP3 lejátszó, memóriakártya, sőt akár egy fényképezőgép) egy nyilvános (és persze a célszemélyhez közeli)

helyen „elveszít". A csalit valaki előbb-utóbb megtalálja és megnézi a tartalmát – mivel CD/DVD esetén a támadó valamilyen érdekes címmel címkézi fel az adathordozót (pl. bizalmas információra vagy szexuális tartalomra célozva), pendrive esetén pedig csábító lehet, hogy megtarthatjuk, így garantált, hogy behelyezik a gépbe. Amint ez megtörténik, a program megkezdi működését.

A trükk természetesen pendrive-okkal is tökéletesen működik. A HVG 2008. április 19.-i számában egy konkrét esetről is olvashattunk, amikor biztonsági audit során egy magyar cég informatikai rendszerébe hasonló módszerrel törtek be. A cég parkolójában szétszórt pendrive-okat a legtöbb munkatárs gyanútlanul bedugta a számítógépébe, elindítva ezzel az eszközön található kémprogramot. [4]

Akár pendrive, akár CD/DVD kerül elszórásra a támadás során, általában forgalmas helyeken kerülnek elhelyezésre: parkoló, mosdó, fénymásoló, tárgyaló, büfé, stb., de célirányosan hagyható irodában is.

4] Márk Edina (2008): *Adatvédelem: az emberi tényező.* HVG XXX. évfolyam, 16. szám.

Postán beküldött adathordozó

Postai beküldésre általában CD vagy DVD kerülhet, valamilyen érdekesnek tűnő tartalommal és kísérőlevéllel. Ezen forgatókönyv során a célszemélyek száma korlátozott és előre meghatározott.

A levél tárgya lehet valamilyen nyereményjáték, ismertető anyag, vagy éppen munkával kapcsolatos, hivatalosnak tűnő dokumentum is. Szintén hatásos, ha a csomag valamilyen eseményre, ünnepre hivatkozva kerül beküldésre (pl. karácsony, névnap, stb.)

Ajándék adathordozó

Gyakori módszer, hogy a vállalatok reprezentációs anyagaikat, esetleg demó programjaikat különböző konferenciákon, rendezvényeken valamilyen adathordozón osztogatják a résztvevők számára. Egy támadó számára ötletes próbálkozás lehet fertőzött állományt tartalmazó adathordozót ilyen jellegű eseményeken eljuttatni a célszemélyeknek. Az adathordozó lehet CD vagy DVD (amennyiben a rajta levő tartalom elég érdekesen van összeállítva), vagy pendrive is (amennyiben a támadó célja, hogy a résztvevők biztosan elvegyék további használatra). Hasonló forgatókönyv képzelhető el valamilyen újság vagy könyv mellékleteként történő terjesztésre is.

Hiteles személy, ismerős saját adathordozója

A kártékony programok adathordozón történő terjesztésének egyik leggyakoribb módszere, ha a kártékony kód a felhasználó saját hordozható adattárolóját is automatikusan megfertőzi. Ezután az áldozat bármilyen okból kifolyólag, bármely más számítógéphez csatlakoztatja az eszközt, a tudtán kívül továbbfertőzi azt. Mivel egy valós személyben, ismerősben az ember jobban megbízik, a legtöbben nyugodt szívvel csatlakoztatják annak pendrive-ját a számítógépükhöz, úgy hogy fel sem merül bennük a véletlen károkozás gyanúja. Sokakban akkor sem merül fel annak gyanúja, hogy „összeszedhetnek" valamilyen kártékony programot, amikor saját pendrive-jukon adják át pl. egy póló vagy más ajándéktárgy nyomtatására szánt képet, vagy a kinyomtatandó dokumentumokat a nyomtatást vállaló szolgáltatónak.

Kártékony kód lefuttatása személyes ráhatással

A támadás célja, hogy a támadó maga győzze meg a gyanútlan felhasználót a kártékony kód lefuttatásának szükségességéről. Ennek érdekében a social engineer egy rendszergazdát megszemélyesítve hivatkozhat valamilyen szükséges biztonsági frissítés telepítésére, valamilyen gyakran használt programmal kapcsolatban észlelt hiba miatt indokolt javítócsomag lefuttatásra, egyéb különös eseményre. A meggyőzés történhet személyesen, vagy telefonon keresztül is, ezek az alábbi pontokban kerülnek bemutatásra.
A meggyőzés során a támadó élhet a segítségnyújtás technikájával (azaz arról győzi meg a felhasználót, hogy későbbi problémáit előzi meg tevékenységével), valamint akár a segítségkérés módszerével is (tehát magának a támadónak van szüksége a felhasználó segítségére, pl. hogy helyette futtat le egy biztonsági frissítést).

A módszer előnye, hogy a támadás során a célszemély egy valós, hús-vér emberrel kommunikál, mely a legtöbb ember gyanakvását elaltatja és arra ösztönzi, hogy bízzon meg a segítséget nyújtó vagy épp kérő félben. A megkeresés történhet személyesen, illetve telefonon keresztül is, utóbbi esetben kisebb a leleplező dés kockázata.

A kártékony fájl és működése

Legyen szó akár egy letöltött állományról, akár elektronikus levél csatolmányaként érkező kártevőről, vagy éppen egy fertőzött fájlt tartalmazó adathordozóról, a kártékony kódnak valamilyen módon le kell futnia a sikeres támadás végrehajtásához. Ez történhet automatikusan, vagy a felhasználó közreműködését igényelve is.

Automatikus lefutás

Támadás szempontjából a legszerencsésebb eset, ha az automatikus futtatás engedélyezve van az operációs rendszeren, és ekkor pl. az elhagyott CD/DVD-re vagy pendrive-ra (sőt, esetleg MP3-lejárszóra, memóriakártyára, vagy fényképezőgépre!) írt autorun.inf fájl az eszköz csatlakoztatásakor automatikusan, a felhasználó közreműködése nélkül lefut. Biztonságtudatosabb felhaszálók ezt a funkciót tiltják, így a támadás nem minden esetben sikeres.

Futtatható állomány

Amennyiben a kártékony program automatikus lefutása nem lehetséges, a támadónak lehetősége van egy jól megszerkesztett forgatókönyvvel rávenni a felhasználót, hogy ő indítsa el az általa az alábbi módszerek valamelyikével bejuttatott kártékony kódot. Ehhez szükség esetén valamilyen módon álcáznia kell a futtatható állományt, illetve annak kiterjesztését.

A következő lehetséges forgatókönyvek jöhetnek szóba:

Megkeresés forgatókönyvében indokolttá tenni a futtatható állományt. A legegyszerűbb módszer, amikor a támadó nem törekszik a kiterjesztés elrejtésére, hanem magát a támadást alakítja úgy, hogy a célszemélynek egy programot kelljen lefuttatnia, pl. a kísérőlevélben vagy a személyes ráhatás során egy szükséges biztonsági frissítés telepítésére hivatkozik.

Ikoncsere rejtett kiterjesztés mellett. Az egyik lehetőség a fájl álcázására, ha a kártékony kódot tartalmazó állomány ikonját valamilyen ikon cserélő programmal (pl. IconChanger) lecseréljük pl. egy Word dokumentuméra.

Plusz kiterjesztés alkalmazása rejtett kiterjesztés mellett. Amennyiben az ismert kiterjesztések rejtése engedélyezve van, de nem szeretnénk ikoncserélő programmal bajlódni, választhatjuk a plusz kiterjesztés lehetőségét is, megzavarva ezzel a felhasználót – azaz a fájlnév mögé először beírja azt a kiterjesztést, amit mutatni szeretne, amögé pedig a valósat, mely úgysem látszik rejtett kiterjesztések esetén.

Kiterjesztés kitolása space-ekkel. Ötletes megoldás lehet a fájl kiterjesztésének space-ekkel való elrejtése is, ez abban az esetben jó megoldás, ha a támadó tisztában van vele, hogy a fájlok kiterjesztése látszódni fog még ikoncsere ellenére is. Ilyenkor hasznos megoldás lehet, ha a fájlnév és a kiterjesztés közé annyi space-t szúr be, hogy a valós kiterjesztés már ne látszódjon az ablakban.

Makró

Munkahelyi környezetben sokan használnak makrókat, elsősorban Excel állományokban (de természetesen a többi Office fájlban is előfordulhat). A makrók alkalmazása ugyan nagyon hasznos lehet, és nagyban

megkönnyíti a felhasználók munkáját, azonban sokan megfeledkeznek róla, hogy egy ártó szándékú támadó akár kártékony részt is beszúrhat a kódba. Az ily módon módosított fájlt a támadó beküldheti e-mail-ben, vagy bejuttathatja valamilyen adathordozón. A fájl nevét célszerű valami érdekesre kitalálni, projektek során a bérlista.xls a „legsikeresebb", de szóba kerülhet ajánlat, vagy valamilyen szerződés is.

A fájl tartalmazhat fiktív adatokat, de az sem baj, ha nem bajlódunk ezek kitalálásával: amennyiben a makrók futtatása engedélyezve van, és lefut a kártékony kód, onnantól kezdve a felhasználó egy üres, „hibás" fájlt is találhat, amennyiben a makrófuttatás tiltva van, vagy a felhasználó engedélyezése szükséges, felhívhatjuk a figyelmét a küldött dokumentum elején a makrók engedélyezésére. Természetesen amennyiben a makrók futtatása automatikusan engedélyezve van, ebben az esetben a makróba bele lehet írni, hogy ez a figyelmeztető szöveg ne jelenjen meg, és vagy üres fájl, vagy fiktív adatok, esetleg egy újabb hibaüzenet bukkanjon elő.

Egyéb fertőzött fájl (PDF, MP3, stb.)

Abban az esetben, ha sem az autorun, sem a makrók nincsenek engedélyezve, a támadó készíthet más típusú fertőzött fájlt is, vállalati környezetben például egy fertőzött PDF-et vagy Word dokumentumot, magáncélra valamilyen videó vagy zenefájlt (pl. MP3). Ebben az esetben szükséges, hogy a támadó tudja, a célszemély milyen szoftvert, és a szoftvernek milyen verzióját használja pontosan az adott fájl megnyitására, lejátszására. Amennyiben a felhasználó által használt – elsősorban régebbi – verzió rendelkezik valamilyen sérülékenységgel, a kártékony kód írója könnyedén ki tudja használni azt egyáltala beküldött fájlban. Projektek során például előszeretettel alkalmazunk PDF fájlokat, melyek az Adobe Reader bizonyos verzióinak sebezhetőségeit használják ki. [5]

[5] PTA CERT-Hungary (2012): *Adobe Reader és Acrobat többszörös sérülékenység.* http://tech.cert-hungary.hu/vulnerabilities/CH-6216.

SZAKONYI LAJOS *

Hőtechnikai modellezés és identifikáció a lakótérben

Összefoglalás: Intelligens lakótér megtervezéséhez, ellenőrző és irányítási rendszerének kifejlesztéséhez szükséges ismernünk az irányított beavatkozásoknak és az esetleges nemkívánatos zavaroknak a céljellemzők (a beállítandó termodinamikai állapotjellemzők) tényleges értékeire gyakorolt hatását. Az *a priori* modellezési technikát követve, megmaradási törvényeket felállítva – élve a munkaponti linearizálással – határozhatjuk meg az egyes működtetési alternatíváknál az operátor-tartománybeli átviteli függvényeket. A folyamatdinamikai modellek paraméterezéséhez szükség volt kísérleti identifikációra is, mely hőszigeteléssel, hőmérsékletszabályozással ellátott lakótérben lefolytatott méréseket jelentett. A definiált folyamatdinamikai modellek (radiátoros fűtés a konvekciós és sugárzásos hőhasznosítás, a légtér és falazat közötti hőtranszport, stb. figyelembevételével) harmadrendű időkésleltetéses jelátviteli tagokat eredményeztek, melyek tartó- és mintavételező szervekkel kiegészítve impulzus-átviteli függvényekkel jellemezhető tagcsoportot szolgáltattak. Az egyes átviteli függvények az adott üzemállapotnál egyértelműen jellemzik a folyamatdinamikát, melyhez kiválasztható a megfelelő irányítási rendszer, megtervezhető az intelligens irányítási stratégia.
Kulcsszavak: Matematikai modellezés, folyamatirányítás, intelligens rendszerek.

Abstract: In order to develop a control system for an intelligent accommodation, one has to know the impact of the control signals and other undesired noises on the goal parameters (thermodynamic attributes). Following the apriori modeling technique the transfer functions corresponding to various inputs, noises and operating modes can be given on the bases of a linearized system of conservation equations. The parameters of the dynamic model had been estimated by measurement. The dynamic representation of the accom-

* *PTE, Pollack Mihály Műszaki és Informatikai Kar*
E-mail: szakonyi.lajos@pmmik.pte.hu

[1] Mohilla, R.–Ferencz, B. (1982): *Chemical process dynamics.* Budapest: Akadémiai.

[2] Szakonyi, L. (1994): Energetic model of an elementary pipe-segment of a steam-water network. *Pollack Periodica, An International Journal for Engineering and Information Sciences.*

[3] Szakonyi, L.–Jancskar, I. A.–Sari, Z. (1994): Energetic model for an elementary unit of a steamnetwork. *Pollack Periodica, An International Journal for Engineering and Information Sciences.*

[4] Szakonyi L. (2005): Infokommunikációs technológia kidolgozása és regionális hasznosítása az energiaelosztás területén. *Informatika a felsőoktatásban Konferencia.* Debrecen: Konferenciakiadvány.

[5] Szakonyi L. (2009): Városi vízgőzhálózat modellezése és identifikációja. *Doktori (PhD) értekezés.* Pannon Egyetem, Veszprém. 155 p.

[6] Szakonyi L. (2002): *Jelek és rendszerek.* Pécs: PTE PMMK jegyzet.

modation taking into account the heating, heat losses, filtration, and the heat capacity of walls, resulted in third order transfer functions, which can be directily applied for simulations. Each of the transfer functions describes the effect of one particular input, and based of this kind of separation, the linear approximation of the behavior of the accommodation can be modeled accurately around any operating point. This representation can enable the construction of the energetic model of a whole building, following the topology of the building, and allows the establishment of an intelligent control strategy.
Keywords: Mathematical modeling, process control, intelligent systems.

Bevezetés

A cél olyan modellek felállítása volt, melyek egy lakótér leggyakoribb „üzemállapotainál" (a fűtési rendszerben történt technológiai, ill. üzemviteli módosítások, a hőenergia-szolgáltatás zavarai, s a környezeti hőmérséklet ingadozó jellege esetén) a zavarok céljellemzőre (léghőmérsékletre) gyakorolt hatását (ezeket egymástól függetlenítve is) képesek jellemezni. Rögzítésre kerültek egy hőszigeteléssel ellátott, gázkazánnal, hőmérséklet-szabályozással (helyiség-hőmérséklet és külső hőmérséklet alapján) üzemelő társasházi lakótérnél a hőtechnikai viszonyait jellemző működési egyenletek.

A definiált folyamatdinamikai modellek (radiátoros fűtés a konvekciós és sugárzásos hőhasznosítás, a légtér és falazat közötti hőtranszport, a filtráció stb. figyelembevételével) struktúrájának meghatározása a deduktív modellezési technikát követve indult, de a modellparaméterek tisztázásához szükséges volt a működő objektumon kísérleteket, méréseket elvégezni, melyek megbízható támpontokat szolgáltattak a modell-paraméterek finomításánál.

Irányítási rendszert megtervezni az instacionárius működés ismeretében lehetséges, ezért is követelmény az átmeneti jelenségeket, az állandósult állapotok közötti időszakot leíró modellek létrehozása, a folyamatdinamika ismerete. [1]

A műszaki-szolgáltató rendszerre megmaradási törvényeket felállítva [2, 3, 4, 5] – élve a munkaponti linearizálással – határozhatjuk meg az egyes üzemviteli alternatíváknál az operátor-tartománybeli átviteli függvényeket. [6] Utóbbiak inverz Laplace-transzformálásával a tranziensek lefutása szemléltethető.

Az egyes átviteli függvények az adott üzemállapotnál egyértelműen jellemzik a folyamatdinamikát, melyhez kiválasztható a megfelelő irányítási rendszer, megtervezhető az irányítási stratégia. [7, 8] E közlemény egy kiválasztott modellstruktúra esetén – eltekintve a modellparaméterek meghatározásának menetétől – a vizsgált lakótér hőtechnikai modelljének szimulációjával nyert tranzienseket kívánja bemutatni.

A mellékelt *1. ábrához (130. old.)* kapcsolódva különböző üzemállapot-alternatíváknál vizsgálhatjuk a lakótér dinamikáját, figyelembevéve:
– a környezeti veszteségeket;
– a szellőztetés hatását (friss levegő beszívást);
– az előbbiek mellett a falazat hőkapacitásának szerepét.

A modellstruktúra rögzítése

A fűtőközegre a fűtött tér belépési és kilépési csőszelvényénél értelmezhető konvekciót, a fűtött légtér felé a közvetett hőátadást, s a fűtőközeg hőáramának időbeli változását tekintetbe vevő modell a következő energiamegmaradási egyenlettel jellemezhető:

$$q_{vf}\,\rho_f\,c_{pf}\,\vartheta_{fb} - q_{vf}\,\rho_f\,c_{pf}\,\vartheta_{fk} - \alpha A(\vartheta_{fk} - \vartheta_{lk}) = V_f\,\rho_f\,c_{pf}\,\frac{d\,\vartheta_{fk}}{dt} \quad . \qquad (1)$$

A lakótér légterének hőmérlegét írjuk le a szellőztetésnél értelmezhető konvekción túlmenően a nyílászárókon, a falazaton, illetve a födémen át a környezet felé irányuló, továbbá a légtér és a belső válaszfalak közötti hőtranszportot jellemző, ugyancsak átadásos árammal:

$$q_{vl}\,\rho_l\,c_{pl}\,\vartheta_{lb} - q_{vl}\,\rho_l\,c_{pl}\,\vartheta_{lk} + \alpha A(\vartheta_{fk} - \vartheta_{lk}) - \alpha_t\,A_t\,(\vartheta_{lk} - \vartheta_t) -$$

$$-\alpha_k\,A_k\,(\vartheta_{lk} - \vartheta_k) = V_l\,\rho_l\,c_{pl}\,\frac{d\,\vartheta_{lk}}{dt} \quad . \qquad (2)$$

A belső falazat és a légtér közötti hőátszármaztatásra vonatkozó mérleg átmeneti állapotban az alábbi:

[7] Szakonyi L.–Jancskárné A. I. (2002): *Szabályozások.* Pécs: PTE PMMK jegyzet.

[8] Szakonyi L.–Jancskárné A. I.–Sári Z. (2010): Regionális anyagáramhálózat modellezése és identifikálása infokommunikációs rendszer kiépítésével. *Informatika korszerű technikái konferencia.*

$$\alpha_t A_t(\vartheta_{lk} - \vartheta_t) = V_t \rho_t c_t \frac{d\vartheta_t}{dt} \quad . \tag{3}$$

Az előbbi nemlineáris differenciálegyenletek a munkaponti linearizálás alkalmazásával átalakíthatók lineáris egyenletekké, ha az egyes időfüggő változókat úgy értelmezzük, mint munkaponti értékük (felülvonással jelezve), s ettől való eltérésük (Δ jelölést alkalmazva) összege. Így nyerjük a következő egyenleteket a fűtőközegre:

$$(\overline{q_{vf}} + \Delta q_{vf})\rho_f c_{pf}(\overline{\vartheta_{fb}} + \Delta\vartheta_{fb}) - (\overline{q_{vf}} + \Delta q_{vf})\rho_f c_{pf}(\overline{\vartheta_{fk}} + \Delta\vartheta_{fk}) - \alpha A(\overline{\vartheta_{fk}} + \Delta\vartheta_{fk}) +$$

$$+ \alpha A(\overline{\vartheta_{lk}} + \Delta\vartheta_{lk}) = V_f \rho_f c_{pf} \frac{d(\overline{\vartheta_{fk}} + \Delta\vartheta_{fk})}{dt} \quad , \tag{4}$$

a légtérre:

$$(\overline{q_{vl}} + \Delta q_{vl})\rho_l c_{pl}(\overline{\vartheta_{lb}} + \Delta\vartheta_{lb}) - (\overline{q_{vl}} + \Delta q_{vl})\rho_l c_{pl}(\overline{\vartheta_{lk}} + \Delta\vartheta_{lk}) + \alpha A(\overline{\vartheta_{fk}} + \Delta\vartheta_{fk}) -$$

$$- \alpha A(\overline{\vartheta_{lk}} + \Delta\vartheta_{lk}) - \alpha_t A_t(\overline{\vartheta_{lk}} + \Delta\vartheta_{lk}) + \alpha_t A_t(\overline{\vartheta_t} + \Delta\vartheta_t) - \alpha_k A_k(\overline{\vartheta_{lk}} + \Delta\vartheta_{lk}) +$$

$$+ \alpha_k A_k(\overline{\vartheta_k} + \Delta\vartheta_k) = V_l \rho_l c_{pl} \frac{d(\overline{\vartheta_{lk}} + \Delta\vartheta_{lk})}{dt} \quad , \tag{5}$$

s végül a légtér és a falazat közötti hőátszármaztatásra:

$$\alpha_t A_t(\overline{\vartheta_{lk}} + \Delta\vartheta_{lk}) - \alpha_t A_t(\overline{\vartheta_t} + \Delta\vartheta_t) = V_t \rho_t c_t \frac{d(\overline{\vartheta_t} + \Delta\vartheta_t)}{dt} \quad . \tag{6}$$

A munkaponti értékek alapján az állandósult állapotra vonatkozó hőmérleg a fűtőközeg:

$$\overline{q_{vf}}\ \rho_f c_{pf}\overline{\vartheta_{fb}} - \overline{q_{vf}}\rho_f c_{pf}\overline{\vartheta_{fk}} - \alpha A\overline{\vartheta_{fk}} + \alpha A\overline{\vartheta_{lk}} = 0, \tag{7}$$

a légtér:

$$\overline{q_{vl}}\rho_l c_{pl}\overline{\vartheta_{lb}} - \overline{q_{vl}}\rho_l c_{pl}\overline{\vartheta_{lk}} + \alpha A\overline{\vartheta_{fk}} - \alpha A\overline{\vartheta_{lk}} - \alpha_t A_t\overline{\vartheta_{lk}} + \alpha_t A_t\overline{\vartheta_t} - \alpha_k A_k\overline{\vartheta_{lk}} + \alpha_k A_k\overline{\vartheta_k} = 0 \ , \tag{8}$$

s a beltéri falazat esetén:

$$\alpha_t A_t \overline{\vartheta_{lk}} - \alpha_t A_t \overline{\vartheta_t} = 0 \cdot \tag{9}$$

Képezve a kéttagú kifejezések szorzatát, eltekintve a másodrendűen kicsiny mennyiségektől ($\Delta * \Delta$), s levonva előbbiekből az állandósult állapotokra vonatkozó egyenleteket, lineáris differenciálegyenleteket nyerünk, melyekre alkalmazhatjuk a Laplace-transzformációt. A transzformált algebrai egyenletek így az alábbiak a fűtőközegre:

$$\overline{q_{vf}} \rho_f c_{pf} \vartheta_{fb}(s) + \overline{\vartheta_{fb}} \rho_f c_{pf} q_{vf}(s) - \overline{q_{vf}} \rho_f c_{pf} \vartheta_{fk}(s) - \overline{\vartheta_{fk}} \rho_f c_{pf} q_{vf}(s) - \alpha A \vartheta_{fk}(s) +$$

$$+ \alpha A \vartheta_{lk}(s) = V_f \rho_f c_{pf} s \vartheta_{fk}(s) , \tag{10}$$

a légtérre: $(\overline{\vartheta_{lb}} = \overline{\vartheta_k} = 0)$:

$$\overline{q_{vl}} \rho_l c_{pl} \vartheta_{lb}(s) + \overline{\vartheta_{lb}} \rho_l c_{pl} q_{vl}(s) - \overline{q_{vl}} \rho_l c_{pl} \vartheta_{lk}(s) - \overline{\vartheta_{lk}} \rho_l c_{pl} q_{vl}(s) + \alpha A \vartheta_{fk}(s) -$$

$$\alpha A \vartheta_{fk}(s) - \alpha_t A_t \vartheta_{lk}(s) + \alpha_t A_t \vartheta_t(s) - \alpha_k A_k \vartheta_{lk}(s) + \alpha_k A_k \vartheta_k(s) = s V_l \rho_l c_{pl} \vartheta_{lk}(s) \tag{11}$$

valamint a belső falazatra:

$$\alpha_t A_t \vartheta_{lk}(s) - \alpha_t A_t \vartheta_t(s) = s V_t \rho_t c_t \vartheta_t(s) . \tag{12}$$

Bevezetve a *Stanton*-számként értelmezhető konstansokat és időállandókat: $a_1 = \dfrac{\alpha A}{\rho_f c_{pf} \overline{\overline{q_{vf}}}}$,

$a_2 = \dfrac{\alpha A}{\rho_l c_{pl} \overline{\overline{q_{vl}}}}$, $a_4 = \dfrac{\alpha_k A_k}{\rho_l c_{pl} \overline{\overline{q_{vl}}}}$, $a_6 = \dfrac{\alpha_t A_t}{\rho_l c_{pl} \overline{\overline{q_{vl}}}}$, $a_7 = \dfrac{\alpha_t A_t}{\rho_t c_t V_t}$, $\overline{t_f} = \dfrac{V_f}{\overline{\overline{q_{vf}}}}$ ill. $\overline{t_l} = \dfrac{V_l}{\overline{\overline{q_{vl}}}}$

(az áramló fűtőközeg, ill. levegő átlagos tartózkodási ideje), nyerjük az előbbi egyenletekből a fűtőközeg kilépési hőmérsékletére (ϑ_{fk}) rendezett formák alapján az alábbi kapcsolatokat (bevezetett jelölés):

$$\Delta\overline{\vartheta_f} = \overline{\vartheta_{fb}} - \overline{\vartheta_{fk}}):$$

$$\frac{a_1}{1+a_1+s\overline{t_f}}\vartheta_{lk}(s) + \frac{1}{1+a_1+s\overline{t_f}}\vartheta_{fb}(s) + \frac{\Delta\overline{\vartheta_f}}{\overline{q_{vf}}}\frac{1}{1+a_1+s\overline{t_f}}q_{vf}(s) = \frac{1+a_2+a_4+a_6+s\overline{t_l}}{a_2}\vartheta_{lk}(s) -$$

$$-\frac{1}{a_2}\vartheta_{lb}(s) + \frac{1}{a_2}\frac{\overline{\vartheta_{lk}}}{\overline{q_{vl}}}q_{vl}(s) - \frac{a_6}{a_2}\vartheta_t - \frac{a_4}{a_2}\vartheta_k(s). \qquad (13)$$

Az előbbi egyenletből kiemelve a légtér-hőmérséklet Laplace-transzformáltját, s elkülönítetten szerepeltetve az irányítandó objektum (lakótér) bemeneti-, valamint állapotjellemzőit, az alapjel-követésre vonatkozó és az egyes zavarátviteli függvényeket leszármaztatására alkalmas, alábbi forma áll elő, tehát $\vartheta_{lk}(s) =$

$$= \frac{(a_2 s + a_2 a_7)(1+a_1+s\overline{t_f})\left[\vartheta_{lb}(s) - \frac{\overline{\vartheta_{lk}}}{\overline{q_{vl}}}q_{vl}(s) + a_4\vartheta_k(s)\right]}{(1+a_2+a_4+a_6+s\overline{t_l})(a_2 s + a_2 a_7)(1+a_1+s\overline{t_f}) - a_2 a_6 a_7(1+a_1+s\overline{t_f}) - a_1 a_2(a_2 s + a_2 a_7)} +$$

$$+ \frac{a_2(a_2 s + a_2 a_7)\left[\vartheta_{fb}(s) + \frac{\Delta\overline{\vartheta_f}}{\overline{q_{vf}}}q_{vf}(s)\right]}{(1+a_2+a_4+a_6+s\overline{t_l})(a_2 s + a_2 a_7)(1+a_1+s\overline{t_f}) - a_2 a_6 a_7(1+a_1+s\overline{t_f}) - a_1 a_2(a_2 s + a_2 a_7)}$$

(14)

A hőtechnikai folyamatok számítógépes szimulációja

A (14) összefüggéssel jelzett modellstruktúrába, mint operátor-tartománybeli eredő átviteli függvénybe helyettesítsük be a meghatározott modell-paramétereket. Így e tört számlálójában elkülönítetten szereplő öt bemeneti jellemzőre külön-külön is meghatározhatók a zavarátviteli, illetve az alapjel-követésre vonatkozó átviteli függvények (eközben mindig a másik négy bemenet időbeli állandóságát feltételezzük, tehát transzformáltjaikat zérussal helyettesítjük). Az alábbiak szerint számszerűsített zavarátviteli függvények kifejezik a légtér hőmérsékletére kifejtett hatását
– a belépő levegő hőmérséklet-változásának:

$$Y_1(s) = \left[\frac{0{,}000982194 \cdot (s + 0{,}0000103042) \cdot (s + 0{,}000765495)}{s^3 + 0{,}00337456 \cdot s^2 + 1{,}58921 \cdot 10^{-6} s + 1{,}17788 \cdot 10^{-11}} \right], \qquad (15)$$

– a fűtővíz (előremenő) belépési hőmérséklet-változásának:

$$Y_2(s) = \left[\frac{1{,}61476 \cdot 10^{-7} \cdot s + 1{,}66387 \cdot 10^{-12}}{s^3 + 0{,}00337456 \cdot s^2 + 1{,}58921 \cdot 10^{-6} s + 1{,}17788 \cdot 10^{-11}} \right], \qquad (16)$$

– a környezet hőmérséklet-ingadozásának:

$$Y_3(s) = \left[\frac{0{,}000300159 \cdot (s + 0{,}0000103042) \cdot (s + 0{,}000765495)}{s^3 + 0{,}00337456 \cdot s^2 + 1{,}58921 \cdot 10^{-6} s + 1{,}17788 \cdot 10^{-11}} \right], \text{ s} \qquad (17)$$

– a friss-levegő áram bevezetés, a szellőztetés igénybevételének:

$$Y_4(s) = \left[\frac{0{,}124639 \cdot (s + 0{,}0000103042) \cdot (s + 0{,}000765495)}{s^3 + 0{,}00337456 \cdot s^2 + 1{,}58921 \cdot 10^{-6} s + 1{,}17788 \cdot 10^{-11}} \right]. \qquad (18)$$

A fűtővíz árama – mint módosított jellemző – változtatásának a céljellemzőre (szabályozott jellemzőre) gyakorolt hatása pedig a következő, alapjel-követésre vonatkozó átviteli függvény alapján értelmezhető:

$$Y_5(s) = \left[\frac{0{,}236865 \cdot s + 2{,}44069 \cdot 10^{-6}}{s^3 + 0{,}00337456 \cdot s^2 + 1{,}58921 \cdot 10^{-6} s + 1{,}17788 \cdot 10^{-11}} \right]. \qquad (19)$$

Az előbbi átviteli függvényekben a számítási műveletek elvégzése az s-operátorra nézve harmadrendű egyenletek megoldását, a zérushelyek és a pólusok kiszámítását tette szükségessé, s így nyerhető az átviteli függvényeknek a 3 részlettörtre bontott, gyöktényezős, ill. időállandós alakja. Digitális (mintavételes) irányítási rendszer megtervezésénél, intelligens rendszerek szimulálásánál a folyamatos elemet – esetünkben az átviteli függvényével jellemzett lakótéri objektumot – szükséges kiegészíteni tartó-, s mintavételező szervvel, majd meghatározni az így nyert tagcsoport impulzusátviteli függvényét. [9] A lakóteret jellemző „szakasz-dinamikák" alapján jellemeztük a végbemenő hőtechnikai változásokat, továbbá ezekhez illesztettük az irányítási rendszer algoritmusát, azok egyes alternatíváit. A struktúrájával és modell-paramétereivel rögzített folyamatdinamikai modellek vizsgálata – a szimulációs feladatokra kiválóan alkalmas, grafikus felhasználói felülettel rendelkező – *MATLAB* fejlesztő-környezetben valósult meg.

A modell matematikai leírását jelentő eredeti, nemlineáris differenciálegyenletek (nemlineáris modell), illetve a megelőző fejezetben leírt munkaponti linearizálással linearizált egyenletek (lineáris modell) adott feltételekre történő megoldásán, a különböző üzemállapotok, beavatkozások, zavarások szemléltetésén túlmenően a modell-megoldások összevetésére, a tranziens lefutások közös koordináta-rendszerben történő ábrázolására is sor került. Utóbbi, lineáris modell ugyan nem számol az időfüggő változók szorzatával (ezeket „másodrendűen kicsiny" mennyiségként tekinti), de a lakótér fűtési időszakára definiált paraméter-beállítások, dinamikák, kisfrekvenciás időbeli változások esetén, s ezek szimulációja során nem mutatott számottevő eltérést a nemlineáris modelltől, az állapotteres leírással nyert megoldásoktól. [10]

A *2. ábrán* (131. old.) látható, hogy a lakótéri klímát alapvetően meghatározó bemenetek időbeli jellegét, alakulását a működtetett rendszerben tapasztalható tényleges változtatásaikhoz igazítva rögzítettük. Így a környezeti hőmérséklet napi, periodikus jelleggel történő ingadozását $\pm 4 \cdot C^{\circ}$ amplitúdójú szinuszfüggvénnyel közelítettük, a véges időtartamig tartó szellőztetés és a lakás egyéb melegvíz-fogyasztóinak esetenkénti belépésével jelentkező fűtővíz-hőmérséklet csökkenés ($3 \cdot C^{o}$) jellegét négyszöglökésként definiáltuk.

A *3. ábrán* (132. old.) adott szabályozó-paraméter beállítással és szabályozó algoritmussal működtetett lakótéri objektum – alapjel-ugrásra (a beállítandó lakótéri hőmérséklet $3 \cdot C^{o}$-os ugrásszerű növelésére) és zavarójel-bemenetre vonatkozó – együttes válasz-függvényei szerepelnek a tényleges lakótéri hőmérséklet és a módosított jellemző tranzienseinek bemutatásával. A szabályozó algoritmus beállított paraméterei nem az optimális beállítást (pl. minimális túllendülés, ill. beállási idő) jelentették, hogy jobban láthatók legyenek a lineáris, ill. a nemlineáris modell-megoldások között mutatkozó esetleges eltérések.

Konklúzió

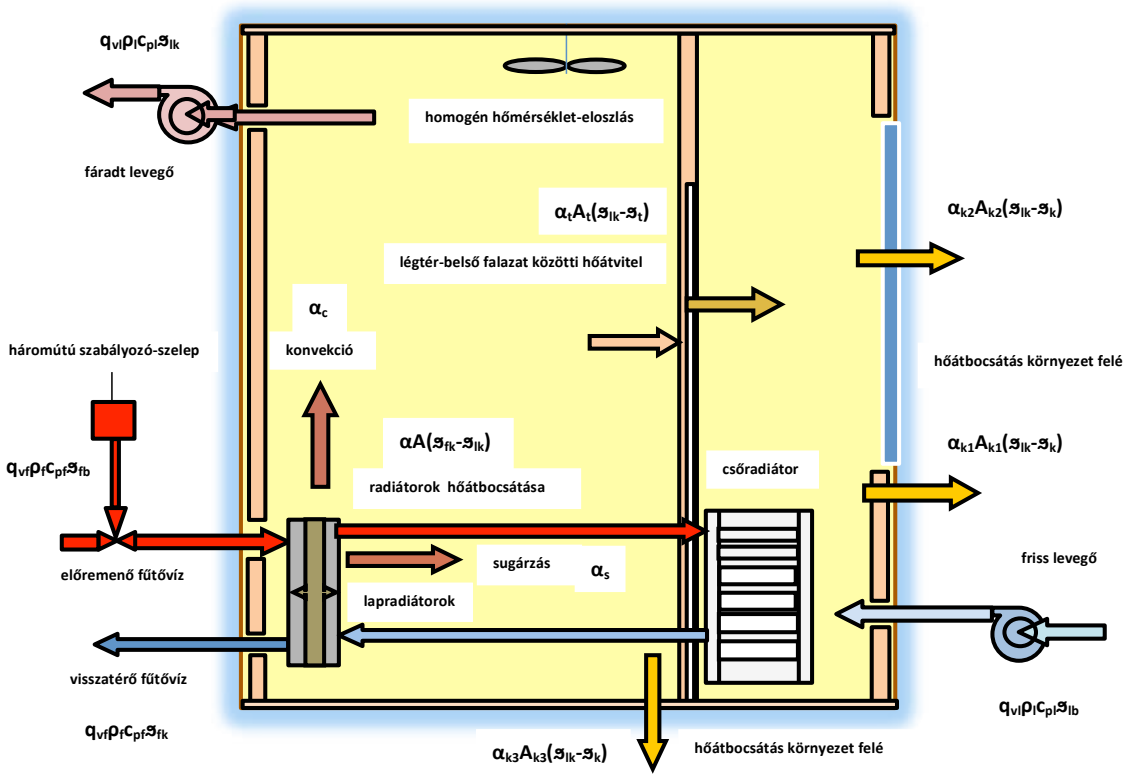
A modellparaméterek ismeretében a működési egyenletekből munkaponti linearizálással nyert transzformált algebrai egyenletek, s az ezekből adódó átviteli függvények igen előnyösen, egymástól elkülöníthető módon jellemezhetik az energiaellátásban jelentkező zavarok hatását az üzemvitelre.

A szimulációs futtatások igazolták a lineáris és a nemlineáris modellmegoldások közötti hasonlóságot. Így a lakótér egyes üzemállapotainak modellekkel történő leképezése, az irányítási stratégiák felállítása, s az irányított rendszer szimulációja viszonylag egyszerű matematikai módszerekkel és számítógépes apparátussal volt elvégezhető.

Célunk továbbra is olyan, intelligens épületirányítási rendszer megtervezése, mely a lakótéri egyedi, sajátos dinamikák alapján építi fel a többlakásos, többszintes lakóépület topológiáját, s képes alkalmazkodni a helyi, lakótéri (lakásonkénti) igényekhez.

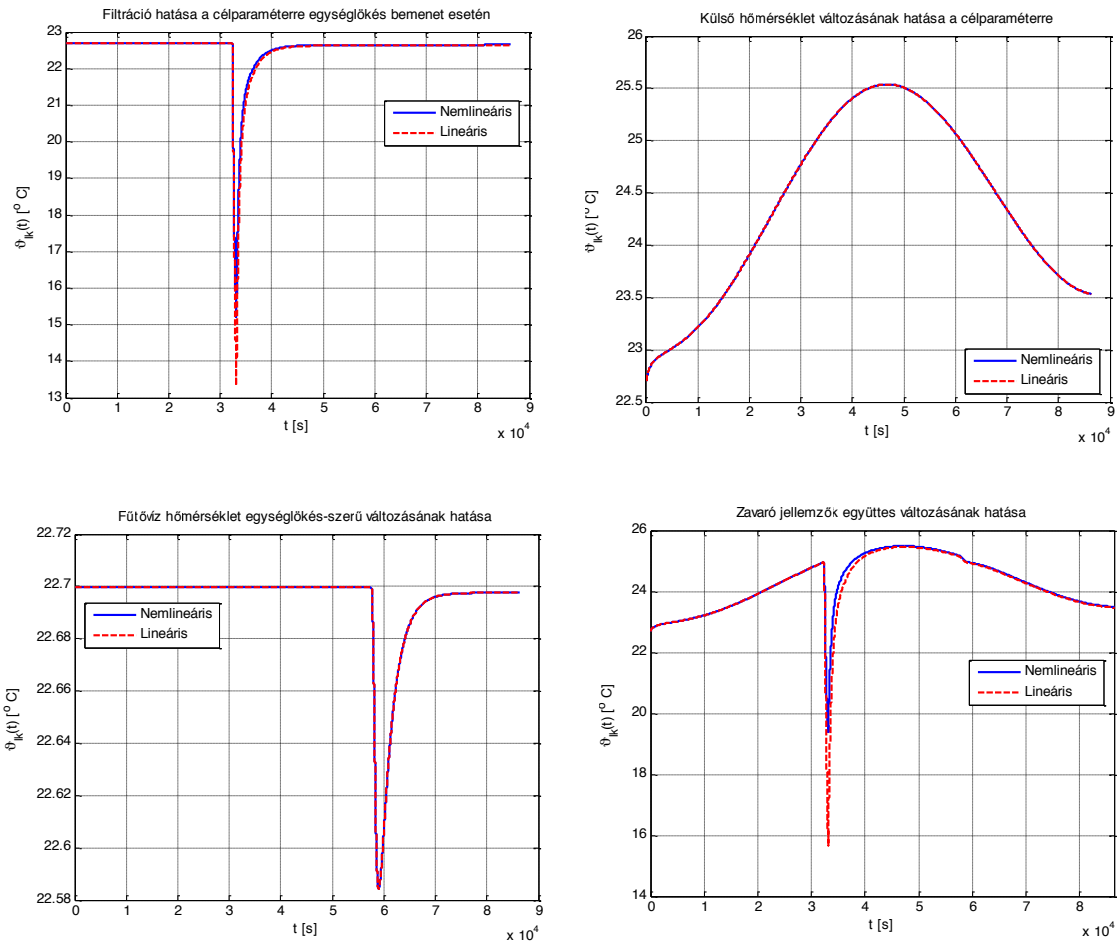
Ábrák

1. ábra. A lakótéri transzportfolyamatok összetevői.

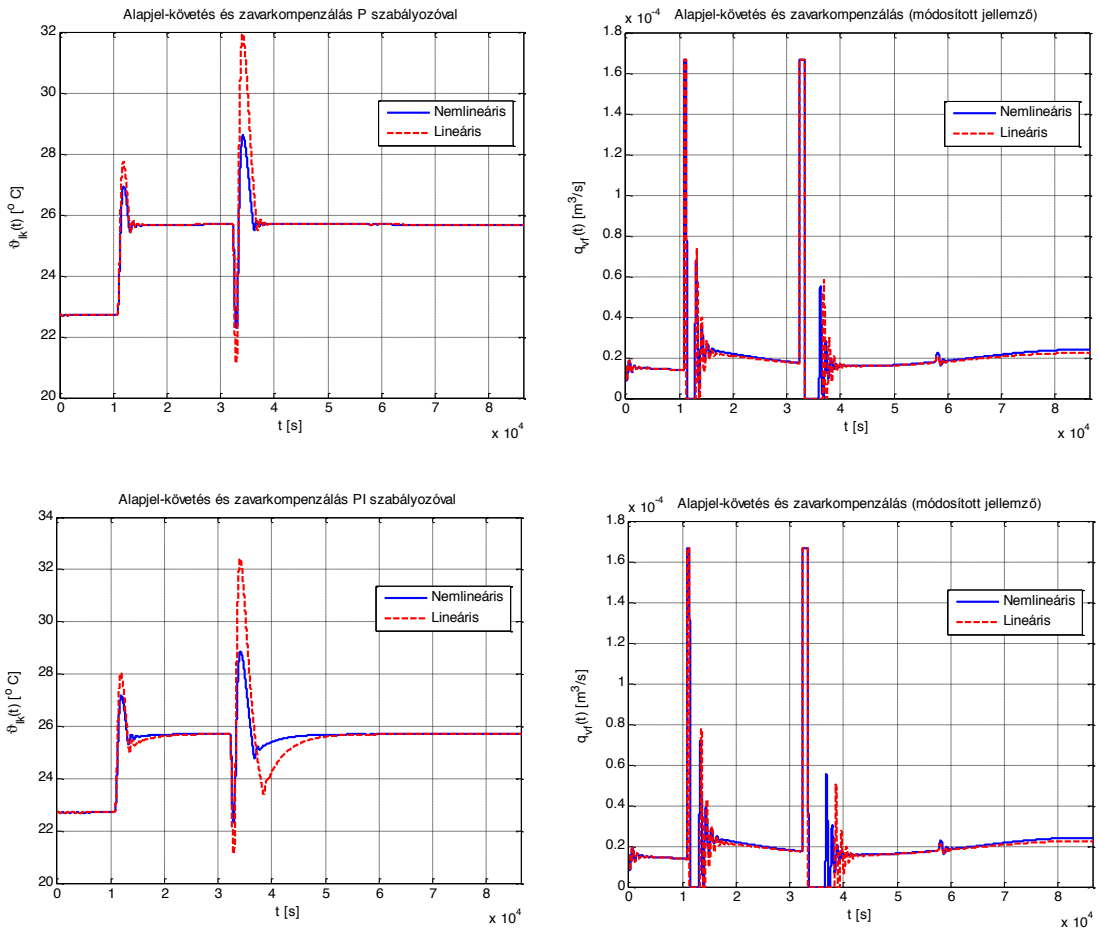


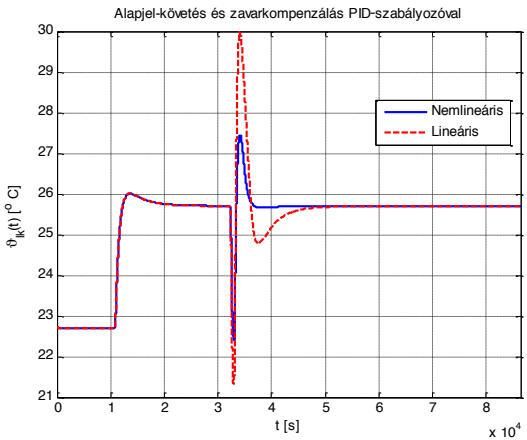
(q_{vf} – fűtőközeg térfogatsebesség; ρ_f - fűtőközeg sűrűség; c_{pf} - fűtőközeg fajhő; ϑ_{fb} , ϑ_{fk} – a fűtőközeg belépési, kilépési hőmérséklete; ϑ_{lk} – légtér hőmérséklet; α, A – a radiátorok hőátbocsátási tényezője, hőátadásra hasznosítható felülete; V_f fűtővíz térfogat; α_k, A_k – a környezet felé értelmezett hőátbocsátási tényező, hőátadó felület; ϑ_k, ϑ_{lb} – a környezeti hőmérséklet, a levegő belépési hőmérséklete; ρ_l, c_{pl} – levegő sűrűség, állandó nyomáson mért fajhő; V_l – légtér térfogat; α_t – a belső válaszfalak hőátbocsátási tényezője; ϑ_t, ρ_t, c_t - a beltéri falak hőmérséklete, sűrűsége, fajhője; A_t, V_t - a válaszfalak felülete, térfogata; t – időkoordináta)

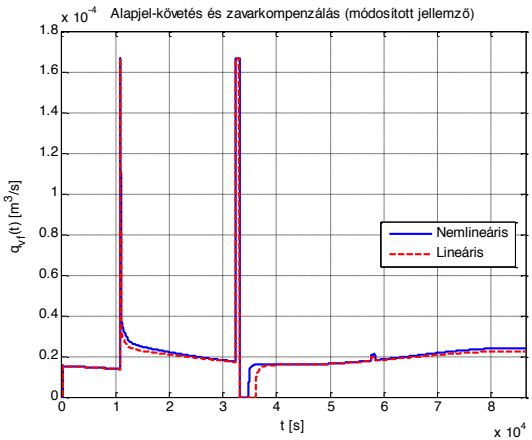
2. ábra. A különböző zavarások tipikus jelszerű változtásának hatása a légtér hőmérsékletére a lakótér szabályozás-nélküli üzemeltetése során (a környezeti hőmérséklet napi periodikus változása szinusz-függvénnyel közelítve).



3. ábra. Az alapjel-követés (légtér-hőmérséklet alapértékének ugrásszerű megváltoztatására adott válasz) és az egymástól függetlenül fellépő zavarok kompenzálása különböző szabályozó algoritmusokkal (a környezeti hőmérséklet napi periodikus változása szinusz-függvénnyel közelítve; szabályozó paraméterek: AR=2*10- 4m3/sK; AI=5,4*10-8 s-1; AD=0,117 s; T*=0,3145 s).




Alapjel-követés és zavarkompenzálás PID-szabályozóval
Nemlineáris
Lineáris
t [s]
x 10^4


x 10^-4
Alapjel-követés és zavarkompenzálás (módosított jellemző)
Nemlineáris
Lineáris
t [s]
x 10^4

ROSTÁS ISTVÁN *–KŐVÁRI ATTILA **

Vakok közlekedését támogató rendszerek áttekintése

Összefoglalás: A 2014-es, WHO által közzétett adatok alapján, földünkön 39 millió vak és 246 millió gyengén látó ember él. Számukra – a közlekedés során – különböző tereptárgyak, objektumok nehezen, vagy egyáltalán nem észlelhetőek, mégis az informatika korában a hagyományos eszközökön (fehérbot, vakvezető kutya) kívül nincs más elterjedt eszköz, amely a látássérültek tájékozódását hivatott segíteni. Erre alapozva igen hasznos lehet egy olyan kompakt eszköz, mely elérhető áron képessé teszi felhasználóját ezen tereptárgyak lokalizációjára, ezáltal az önálló gyalogos közlekedésre. A cikkben bemutatásra és összehasonlításra kerülnek a témával kapcsolatosan végzett eddigi fejlesztések, azok előnyei és hátrányai, valamint egy olyan rendszer kerül leírásra, mely az eddigi rendszerek hiányosságait hivatott megszüntetni és egy újfajta összetett hangjelzés formájában tájékoztatni viselőjét az akadályok távolságáról és irányáról.
Kulcsszavak: Akadályérzékelés, gyalogos közlekedés, gyengén látók.

Abstract: Based on the data that has been released by WHO in 2014, there are 39 million blind and 246 million sighted people, who lives on our earth. While they are walking, some objects can't or slightly can be recognized by them. However, in the age of Information Technology besides the usual tools like the long white cane, or the dog guide, there is no other popular tool for them to use. Based on this can be very useful a compact device which can make its user to recognize the objects while they are walking with no help, at a low price. The article shows and compares the developments in connection with this subject, it's advantages and disadvantages and also describes a system, which bypasses the previous system's deficiencies . Informs the user of the distance and direction of the objects around him, with a new and complex tone.
Keywords: Obstacle detection, walking, partially sighted.

** Dunújvárosi Főiskola, Informatikai Intézet*
E-mail: rosti.istvan@gmail.com

*** Dunújvárosi Főiskola, Informatikai Intézet*
E-mail: kovari@mail.duf.hu

Bevezető

A földünkön közel 290 millió látássérült ember él, esetükben a hétköznapi önálló gyalogos közlekedés sok problémát jelent. A tereptárgyak, akadályok felderítésére elsődlegesen botot alkalmaznak, de más önálló gyalogos közlekedést segítő eszközök, elgondolások is fejlesztés alatt vannak.

Ezen eszközök általában túl drágák így állami támogatás hiányában egy fogyatékkal élő ember számára ezek megfizethetetlenek. Persze több eszköz található a megfizethető kategóriában is, viszont ezek a technológiák nem elég kiforrottak, nem elég pontosak ahhoz, hogy az önálló gyalogos közlekedést támogassák, csupán kiegészítik és biztonságosabbá teszik a fehérbottal való közlekedést.

Az előzőkben levont következtetések alapján egy olyan kompakt eszköz lenne célszerű, ami képessé teszi a felhasználóját a környezetében lévő akadályok helyének és távolságának a meghatározására.

A rendszerben használt eszközök és technológiák kiválasztásánál fontos szempont, hogy a kész termék forgalombahozása esetén megfizethető áron legyen elérhető.

A következőkben néhány látássérültek számára kidolgozott rendszer kerül bemutatásra és értékelésre.

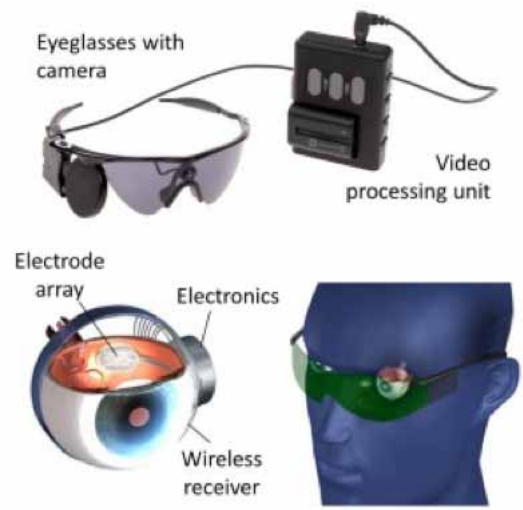
Látássérültek számára kidolgozott rendszerek

Bionikus szem implantátum

A bioinformatika (a biológia és az informatika határterülete) fejlődésével olyan eszközök kialakítása vált lehetővé, mint a Duke Eye Center által kifejlesztett Argus II. (*1. ábra.*) [1] névre keresztelt bionikus szem.

[1] Kotecha, A.–Zhong, J.–Stewart, D.–Da Cruz, L. (2014): *The Argus II prosthesis facilitates reaching and grasping tasks: a case series.* BioMed Central Ltd. (http://www.biomedcentral.com/1471-2415/14/71)%88MC4_255_250_poly%EF%BC%89_EN.pdf

1. ábra. Argus II.



forrás: http://scienceinclusive.com/

Az eszközt egy kameraegység és egy az agyba ültetett implantátum alkotja. A sérült vagy beteg látószerv kikerülésével a kamera által érzékelt fényt továbbítja az implantátumnak, ezáltal olyan betegségek gyógymódjának szánják, mint például a szemideghártya gyulladás (retinitilis pigmentosa).

Az eszköz nem adja vissza a látást, jelenleg a felhasználó a rendszer segítségével megtudja különböztetni a sötétet a világostól. Bár már ez is óriási előre lépés a vakság és a látássérülés orvoslásában, de ezzel a jelenlegi működési elvvel nem megvalósítható az önálló gyalogos közlekedés, mivel a tereptárgyak és a szintkülönbségek nem érzékelhetők.

Mivel a bionikus szemek és implantátumok jelenlegi fejlettségi szintje nem éri el azt a szintet, hogy a felhasználó képes legyen olyan vizuális képet alkotni, aminek a segítségével meghatározhatja a közelében lévő akadályok távolságát és irányát, valamint a szintkülönbségeket, ezért más jellegű kutatási eredmények vizsgálata következett.

[2] Saksham, SmartCane Device. 2012. (http://smartcane.saksham.org/)

[3] Magyar találmány válthatja a vakok kezében a fehér botot, 2014. (http://richpoi.com/cikkek/tudo-many/magyar-talalmany-valthatja-a-vakok-kezeben-a-feher-botot.html)

Ultrahangos távolságmérésen alapuló tereptárgy-érzékelés

Számos olyan eszköz létezik, amik arra lettek tervezve, hogy rezgés- vagy hangalapú visszajelzés formájában tájékoztassák a látássérült felhasználót az észlelt tereptárgy távolságáról és irányáról. Ezek közös jellemzője az, hogy valamilyen szenzor segítségével távolságot mérnek, majd egy jelfeldolgozó egység segítségével azt kiértékelik. Az alábbi rendszerek mindegyike ultrahangos szenzort használ.

A SmartCane Device (*2. ábra.*) egy fehérbotra szerelt ultrahangos szenzorból és jelfeldolgozóból áll. Az eszköz működése közben ultrahangot bocsájt ki, majd a visszhangokból határozza meg a távolság értékét. [2]

2. ábra. SmartCane Device.

forrás: http://www.dailymail.co.uk/

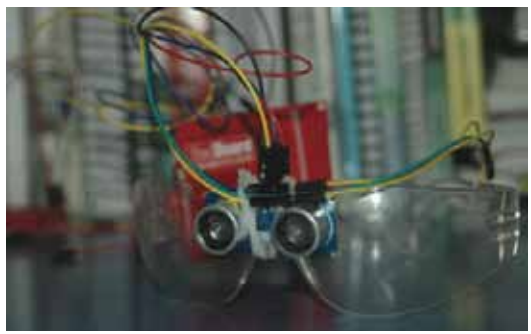
A rendszer rezgésintenzitás-változtatásokkal jelez vissza a felhasználónak a távolság függvényében hasonlóképp, mint a magyar fejlesztésű Aurora (*3. ábra.*). [3]

3. ábra. Aurora.

forrás: http://richpoi.com/

Angelos Getsis által kifejlesztett Smart Glasses egy szemüvegen elhelyezett ultrahangos távolságmérőt tartalmaz (*4. ábra.*). A fejlesztés első helyen végzett a 2014-es Google's Science Fair versenyen [4]

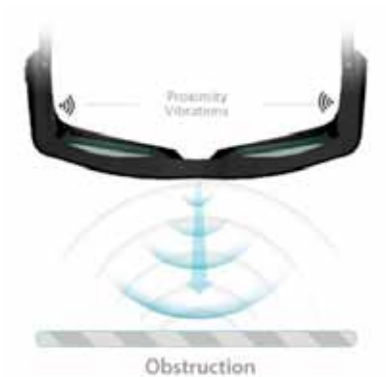
4. ábra. Smart Glasses.



http://www.vice.com/

Hasonló fejlesztés a AmbuTech és RNIB által kifejlesztett iGlasses™ (*5. ábra.*) szemüvegen a szenzorok egy szemüvegen helyezkednek el [5].

5. ábra. iGlasses™.



forrás: http://ambutech.com

[4] Greek teen wins first prize at Google's Science Fair for smart glasses for the blind, 2014. (http://en.protothema.gr/greek-teen-invents-smart-glasses-for-the-blind/)

[5] Ambutech: *Finally an innovative mobilty aid at an affordable price!* 2012. (http://ambutech.com/iglasses)

Az University of Auckland egyetemen Claire Davies által fejlesztett AUDEO [6] és a Yorkshire Company Sound Foresight Technology által forgalmazott UltraCane [7] esetén is a szenzorok egy fehérboton helyezkednek el (*6. ábra.)*.

6. ábra. UltraCane.

forrás: http://ultracane.com

A www.grathio.com weboldalon található „Meet The Tacit Project. It's Sonar For The Blind." cikk egy olyan útmutatót tartalmaz, ami alapján otthon is összeállítható egy a felsoroltakhoz hasonló eszköz (*7. ábra.*). [8]

7. ábra. Tacit Project.

forrás: www.grathio.com

[7] Paul Fanning (2013): *Ultrasonic sensors give additional mobility to visually impaired.* (http://www.eurekamagazine.co.uk/design-engineering-features/technology/ultrasonic-sensors-give-additional-mobility-to-visually-impaired/48431/)

[8] Steve Hoefer (2011): *Meet The Tacit Project. It's Sonar For The Blind.* (http://grathio.com/2011/08/meet-the-tacit-project-its-sonar-for-the-blind/)

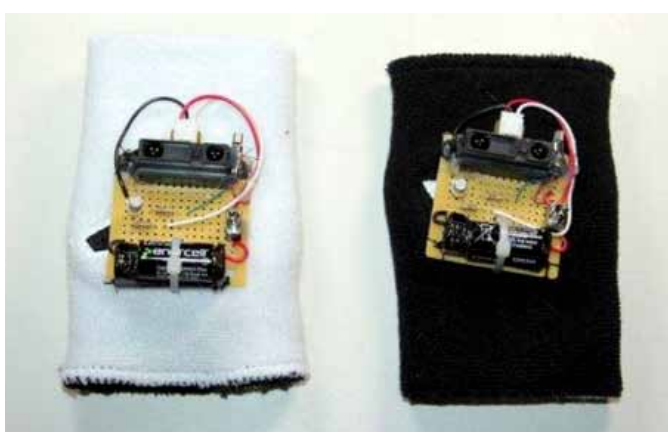
Ezekkel a rendszerekkel már meglehet határozni egyes tereptárgyak távolságát önálló gyalogos közlekedéskor, viszont nagy betekintési szögben található akadály esetén, a visszaverődés törvénye miatt nem tér vissza a méréshez szükséges visszhang, ezért előfordulhatnak olyan veszélyes akadályok, amiket nem jelez a rendszer. [9] Emellett a közlekedés közben, a felhasználó előtt lévő szintkülönbségek észlelése is megoldatlan fehér bot hiányában.

Infravörös távolságmérésen alapuló tereptárgy-érzékelés

A robotikában használt legnépszerűbb távolságérzékelő szenzorok infravörös fényt használnak a tereptárgy távolságának a meghatározására. Ezekből a szenzorokból többféle létezik különböző méréstartományokkal.

Ilyen elven működő szenzorok találhatók a Hannah Prutchi által kifejlesztett SharkVision névre keresztelt rendszeren. A csuklókra erősített eszköz (*8. ábra.*) fő részét egy IR-szenzor és egy vibrátor képezi. A szenzor által érzékelt objektum távolságának függvényében rezgéssel jelez vissza a rendszer a felhasználónak. [10]

8. ábra. SharkVision.



forrás: http://www.prutchi.com/

[9] Tóth András (2003): *Hullámtani összefoglaló.* (http://www.muszer-oldal.hu/measurenotes/hullamtan.pdf)

[10] Prutchi, D. (2012): (SharkVision) – A Sen-sing Suit for the Blind by Hannah Prutchi, d.i.y. *Medical Devices.*

A University of Cincinnati hallgatói, Luis Favela, Tom Froese és Adam Spiers által kifejlesztett Enactive Torch nevű eszköz (*9. ábra.*) is IR-szenzort használ a távolság mérésre, viszont a közelségről a felhasználót nem rezgésimpulzus változtatás segítségével, hanem berregéssel tájékoztatja. [11]

9. ábra. Enactive Torch.

forrás: http://techcrunch.com/

Az Amjed S. Al-Fahoum, Heba B. Al-Hmoud, és Ausaila A. Al-Fraihat álat írt „A Smart Infrared Microcontroller-Based Blind Guidance System" [12] kutatási cikkben egy olyan rendszer olvasható, ami bár hasonló elven működik, mint az eddig felsoroltak, de a visszajelzés összetettebb formában történik.

A rendszer két különálló hangszórójának segítségével abban a fülben történik a folyamatos hangjelzés, amelyik irányban található az észlelt tereptárgy. Az objektum távolságáról pedig rezgés-intenzitással jelez vissza a felhasználója számára.

A szenzorok elhelyezése miatt csak kis sávban történik az akadályészlelés, fej vagy törzs magasságban az ilyen kialakítással a szintkülönbségek észlelése szintén megoldatlan. A kézre szerelt IR-szenzorokkal észlelhetőek a szintkülönbségek, viszont a keskeny észlelési sáv itt is problémát jelent.

[11] Biggs (2014): *The Enactive Torch Is a Sennsor That Helps The Blind „See"*. (http://techcrunch.

[12]Al-Fahoum, A. S.–Al-Hmoud, H. B.–Al-Fraihat, A. A. (2013): A Smart Infrared Microcont-roller-Based Blind Guidance System. *Active and Passive Electronic Components*. Article ID 726480, 2013 (http://www.hindawi.com/journals/apec/2013/726480/)

Gyalogos közlekedését segítő fejlesztések értékelése

A következő *1. táblázat*ban összehasonlításra kerülnek a gyengén látók gyalogos közlekedését segítő, az előzőkben bemutatott fejlesztések.

1. táblázat. Gyengén látók gyalogos közlekedését segítő fejlesztések értékelése.

Implanátum

Előnyök	**Hátrányok**
Vizuális képet szolgáltatnak Nincs szükség a visszajelzések megértésére, magtanulására Nem csak a gyalogos közlekedést segítik Látás mesterséges visszaadása Továbbfejlesztési lehetőségek	Fejlesztési fázisban van Szűk körben elérhető Drága

Ultrahangon alapuló fejlesztések

Előnyök	**Hátrányok**
Olcsó Széles körben elérhető Akadály távolság pontos meghatározása	Kevés szenzor használat miatti szűk érzékelési tartomány Nagyobb betekintési szögben pontatlan mérést eredményez Járófelületben lévő szintkülönbség érzékelése megoldatlan

IR szenzoron alapuló fejlesztések

Előnyök	**Hátrányok**
Megfizethető Széles körben elérhető Akadály távolság pontos meghatározása Szintkülönbségek érzékelése Nagy betekintési szög	Kevés szenzor használat miatti szűk érzékelési tartomány Az IR szenzor jelentősen drágább, mint az ultrahangos

Az értékelésből látható, hogy mind a szűk érzékelési tartomány, mind pedig a járófelület szintkülönbségeinek érzékelése nem valósítható meg megfelelően.

Továbbfejlesztési lehetőségek

Egy új rendszer tervezésekor az előző fejezetben kapott eredményeket célszerű figyelembe venni, a meglévő eszközök előnyeinek megtartásával minél több hátrányként jelentkező tényező kiküszöbölésével. Egy ilyen rendszer az alábbi követelményeknek kell megfeleljen:
– A fej és törzsmagasságban, a gyalogos közlekedés főbb irányaiban, tehát balra, jobbra és előre, meg kell oldani az akadályérzékelést.
– Legalább 3 db távolságérzékelő szenzor legyen elhelyezve a főbb irányokon, így a fej mozgatásával széles sávban lehetséges az akadályok felderítése.
– A távolságérzékelő szenzorok egy könnyű viseletű szemüvegen kerüljenek elhelyezésre.
– Törzs alatti akadályok és a járófelület-magasság érzékelését is tartalmazza, legyen képes a lépcsők, egyéb járófelület-magasságbeli változások érzékelésére, észlelése egy külön szenzor segítségével, mely csípő magasságban helyezkedik el (pl. övre rögzített). Szenzor alkalmas legyen az érzékelés irányára nem csak merőleges akadályok érézékelésére.
– Szenzorok jeleit feldolgozó egység kis fogyasztású és méretű legyen, övre rögzíthető kisméretű dobozban.
– Hang- vagy vibrációs jelzés segítségével tájékoztassa a viselőjét az akadályok távolságáról és irányáról.
– Olcsó legyen.

Hallásalapú lokalizáció során a hangforrás helyének meghatározásában két fő paraméter vesz részt:
– a két fül közötti időeltérés, az interaurális időkülönbség (ITD – Interaural Time Difference),
– illetve a két fül közötti interaurális szinteltérés (ILD – Interaural Level Difference).

A hangforráshoz közelebbi fülkagylóba előbb és hangosabban érkeznek be a hanghullámok, így annak a hozzánk viszonyított iránya meghatározható. A megfelelő hangvisszajelzés működéséhez sztereóhangszóró használata szükséges. Először az észlelési irányra vonatkozó hangjelzést kell megvalósítani oly módon, hogy egy sztereóhangszóró segítségével az észlelési iránnyal megegyező oldali fülkagylóban szólal meg a hangjelzés. Ezzel a módszerrel azonnal kialakul a felhasználóban egy egocentrikus téri kép, amiben ő és az észlelt objektum található. Az akadályok feltérképezéséhez nem elegendő az észlelés irányát megállapítani, azok a felhasználótól viszonyított pontos távolságát is meg kell határozni. A pontos távolságérték hangalapú visszajelzésére több darab egyenként állítható kiterjedésű, észlelési sávot kell létrehozni. Mindegyik észlelési sávhoz más-más egymástól jól elkülöníthető hangmagasságú hangjelzés tartozik.

A hangmagasságok közötti távolság meghatározásánál figyelni kell arra, hogy a sávok között mozgás során a hangmagasság-változások ne legyenek bántóak a fülnek, ugyanakkor egyből kihallható különbségnek kell lenni közöttük.

10. ábra. észlelési sáv.

Forrás: saját ábra.

A fejlett képernyő-felolvasó programoknak köszönhetően az okostelefonok a vakok és a gyengénlátók körében is elterjedt eszközök. A rendszer jelkiértékelését és az összetett hangjelzés kialakítását ezért célszerű okostelefonon végezni, mivel így a rendszer továbbfejlesztésénél, plusz funkciók bevezetésénél nincs szükség járulékos hardware hozzáadására.

A két egység közötti kommunikációt más eszközöknél is alkalmazott USB soros kapcsolaton [13] vagy Bluetooth-kapcsolat segítségével célszerű megoldani, az okostelefonokkal történő minél nagyobb kompatibilitás miatt a Bluetooth-kapcsolat előnyösebb, melyet például az okostelefonokkal támogatott tanulás területén is alkalmaznak [14].

[13] J Katona, J. (2014): *Examination and comparison of the EEG based Attention Test with CPT and T.O.V.A.* Proceedings of 15th IEEE International Symposium on Computational Intelligence and Informatics: CINTI 2014, Budapest, Pp. 117–120.

[14] Benedek A.–Molnár G. (2014): *Supporting the m-learning based knowledge transfer in university education and corporate sector.* Proceedings of the10th Int. Conf. on Mobile Learning 2014. Madrid, Spanyolország. Pp. 339–343.

Összefoglalás

A cikk röviden összefoglalta a gyengén látók számára fejlesztett eszközöket és egy összehasonlítást adott az egyes eszközök előnyeiről és hátrányairól. Megállapítható, hogy az eszközök egyes előnyös tulajdonságaik mellett számos hátránnyal is rendelkeznek. Ezek közül az egyik legfontosabb a járófelület-szintkülönbségek (pl. lépcső) érzékelésének hiánya. Specifikálásra került egy olyan rendszer, mely egy kiegészítő szenzor alkalmazásával járófelület-szintkülönbség mérésének lehetőségét is tartalmazza. Továbbá leírásra került egy olyan újfajta hangjelzési módszer, mely az akadályok távolságáról és irányáról összetett hangjelzés formájában tájékoztatja viselőjét.

HADARICS KÁLMÁN *

Nyílt forráskódú szoftverek sebezhetőségeinek vizsgálata

Összefoglaló: Az utóbbi hónapokban több súlyos biztonsági hiba látott napvilágot különféle nyílt forráskódú szoftverekkel kapcsolatban. Ezek egy része gyakran használt alkalmazásokhoz, mások a nyílt forráskódú operációs rendszerek kerneléhez vagy parancsértelmezőjéhez köthetők. Napjainkban a Linux a legelterjedtebb a nyílt forráskódú operációs rendszerek között. Rengeteg helyen alkalmaznak valamilyen Linux, vagy ráépülő operációs rendszert (pl. Android). Éppen ezért e biztonsági hibák nagyon sok használatban lévő informatikai, infokommunikációs rendszert és eszközt érintenek. Manapság a biztonsági hibák aktív kihasználására iparág épült. Ezek révén lehetővé válhat illegális adatok eltulajdonítása, vagy más informatikai bűncselekmények elkövetése. Írásomban részletes elemzésre kerülnek a legsúlyosabbnak minősített biztonsági hibák. A legtöbb ilyen jellegű hiba valamilyen emberi figyelmetlenségre, nem kellően elvégzett adat vagy paraméter-ellenőrzésre vezethető vissza. Ezen túl azt is bemutatom, hogy milyen körülmények között és milyen módon lehet „visszaélni" az adott sebezhetőséggel. Bármely szoftver esetében létezhetnek biztonsági problémák. Minél elterjedtebb egy szoftver, ennek hatásai annál markánsabban jelentkezhetnek. Cikkemben összegzem azt is, hogy milyen védekezési lehetőségek vannak. A jövőben is előfordulhatnak hasonló vagy még súlyosabb biztonsági hibák. Ezek egy része megfelelő konfigurációs beállításokkal, speciális biztonsághoz kapcsolódó szoftver-komponensekkel csökkenthető.
Kulcsszavak: Sebezhetőség, biztonság, nyílt forráskód.

Abstract: In the last month more serious security bugs were discovered related to open-source software. One part of the vulnerabilities is related to frequently used applications, others belong to open-source operating system kernel or shell. These days Linux is the most widely used open-source

* *Dunaújvárosi Főiskola, Informatikai Intézet*
E-mail: hadarics@mail.duf.hu

operating system. Linux and related operating systems (e.g. Android) are used in a lot of different ways. Therefore these vulnerabilities have impact to many IT and information and communication systems and devices. Security flaws are actively exploited. With the help of vulnerabilities, illegal data acquisition is possible or executing other IT crimes. In this publication I will analyze the most serious security bugs discovered in open-source software. The most of these bugs are based on human behavior, missed data or parameter checking. I also will introduce how to abuse with vulnerabilities. Most software may have vulnerabilities. As a software is widely used the effects of vulnerabilities became bigger and bigger. In this publication I also will summarize how to defend our IT infrastructure. In the future it is possible that new serious security bugs will be discovered. One part of these can be reduced by configuration or with the usage of security related applications.
Keywords: Vulnerability, security, open-source.

Bevezetés

Az utóbbi években egyre szélesebb körben alkalmazunk nyílt forráskódú (open-source) operációs rendszereket, alkalmazásokat. Ezen szoftverek bizonyították, hogy stabilitásukban, használhatóságukban lényegében nem különböznek a kereskedelmi szoftverektől. Érdekes kérdés, hogy biztonsági szempontból ezekre a szoftverekre milyen megállapítás tehető?

Az okostelefonok, és vele együtt az Android operációs rendszer elterjedése markáns változást hozott a Linux operációs rendszer és a rajta futó, zömmel nyílt forráskódú, szoftverek kapcsán. Egyre több biztonsággal foglalkozó cég ismerte fel, hogy e szoftverek használata elért egy kritikus tömeget, és meglehetősen nagy felhasználói táborral rendelkeznek. Ennek következtében megnőtt az érdeklődés az operációs rendszer és a rajta futó szoftverek biztonsága iránt is.

A nyílt forráskódú szoftverek jellemzői

A nyílt forráskódú szoftverek (Open-source software, OSS) olyan számítógépes programok, amelyek használata bárki számára megengedett. Az Open-source definition [1] tartalmazza azokat a kritériumokat, amelyeket teljesülése esetén hívhatunk adott szoftvert nyílt forráskódúnak. Ezek közül a legfontosabbak:

[1] Open-source definition, http://opensource.org/osd

– szabadon használhatók, másolhatók, terjeszthetők,
– a program forráskódja szabadon tanulmányozható és módosítható,
– speciális megengedő ún. copyleft licenc vonatkozik rájuk.

A nyílt forráskódú szoftverek szerepe egyre növekszik napjaink informatikai rendszerei kialakítása során. Mind a kormányzati, mind pedig a vállalati szféra egy valódi alternatívának tekinti a zárt forráskódú programokkal szemben. A zárt forrású programok tipikusan licencdíj-köteles alkalmazások, ahol az alkalmazásokat használó személyek/szervezetek az adott szoftverek használatáért fizetnek. A zárt forrású szoftverek profitorientált cégek szellemi termékei, akik bevételeiket részben vagy egészben a licence és egyéb díjakból szerzik. Ezzel szemben a szabad szoftverek fejlesztése közösségi alapon történik. Bárki, aki kedvet érez és rendelkezik programozói ismeretekkel, bekapcsolódhat egy-egy szoftver fejlesztésébe, tesztelésébe.

A téma aktualitása

Az előző bekezdésben felsoroltak alapján a nyílt forráskódú szoftverek használata és jelentősége folyamatosan növekszik. Ez segíti, hogy mind a magyar kormány, mind pedig az EU részéről támogatás és igény mutatkozik a nyílt forráskódú szoftverek és szabványok használatára. [2]

Ezen túl bizonyos területeken kifejezetten releváns a nyílt forráskód használata. Ilyen például:
– szuperszámítógépek, ahol a legnagyobb teljesítményű ötszáz szuperszámítógép 97%-a valamilyen Linux-verziót használ, [3]
– az okostelefon operációs rendszerek esetében az Android közel 80%-os részesedéssel rendelkezik, [4]
– Web kiszolgálók és webes alkalmazás-fejlesztés, (Apache + PHP)
– Hálózati alkalmazások és hálózati eszközök (pl.: otthoni routerek, hálózati forgalom megfigyelésére alkalmas szoftverek (pl.: Wireshark)

A nyílt forráskódú szoftverek elterjedése túlhaladta azt a kritikus szintet, amely miatt érdemes ezek biztonsági hibáival foglalkozni.

A 2014-es évben több kritikus besorolású biztonsági hiba is napvilágot látott, amely nagyon sok nyílt forráskódú szoftvert, operációs rendszert érint. Ezek közül a legnagyobb publicitást kapta:

[2] *E-közigazgatási Szabad Szoftver Kompetencia Központ tanulmányai*. http://szabadszoftver.kormany.hu/sajat-tanulmanyok/

[3] *Top 500*. http://www.top500.org/statistics/list/

[4] *Android*. http://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

– Az ún. „Heartbleed” bug, amelyet 2014 áprilisában fedeztek fel az OpenSSL-ben
– Az ún. „Shellshock” bug, amelyet 2014 szeptemberében fedeztek fel az egyik legygyakrabban használt parancsértelmezőben a GNU bash-ben

A nyílt forráskódú szoftverek fejlesztési modellje nagyban különbözik a zárt forráskódú szoftverek modelljétől. Nincsen erős kontrol a szoftvertervezés és a -megvalósítás során. Sokszor a fejlesztők közti informális kommunikáció dönt adott szoftver kérdéssel kapcsolatos feladat megoldásáról. Eric S. Raymond szerint a nyílt forráskódú szoftverek esetében igaz az alábbi állítás: „Given enough eyeballs, all bugs are shallow.”

A valóság az, hogy különféle képességű programozók dolgoznak nyílt forráskódú projektekben. Ezen programozók képességeitől nagyban függ az előállított forráskódok minősége.

Azáltal, hogy az adott szoftverek forráskódja bárki számára elérhető, kétség kívül egy támadó számára könnyebbséget jelenthet a potenciális biztonsági hibák, sérülékenységek keresése. Ezzel szemben azonban azt se felejtsük el, hogy a nyíltság bárki számára biztosíthatja azt a lehetőséget, hogy saját maga ellenőrizze/ellenőriztesse a forráskódok minőségét, mielőtt az adott alkalmazást bármilyen módon felhasználná. Manapság nagy szoftverelőállító cégek (pl.: IBM, Red Hat, Oracle) szakemberei vesznek részt a nyílt forráskódú szoftverek fejlesztésében és ellenőrzésében. Ez természetesen javítja az előállított forráskódok minőségét, de mindig előfordul olyan eset, amikor súlyos biztonsági hibák maradnak különböző szoftverekben.

Zárt forráskódú szoftverek esetében a biztonsági hibák ugyanúgy jelen vannak. A forráskód zártsága nem jelenti szükségszerűen azt, hogy e kódok minősége jobb lenne. A sebezhetőségek felkutatása itt más módszereket igényel, hiszen közvetlenül a forráskódot nem lehet felhasználni.

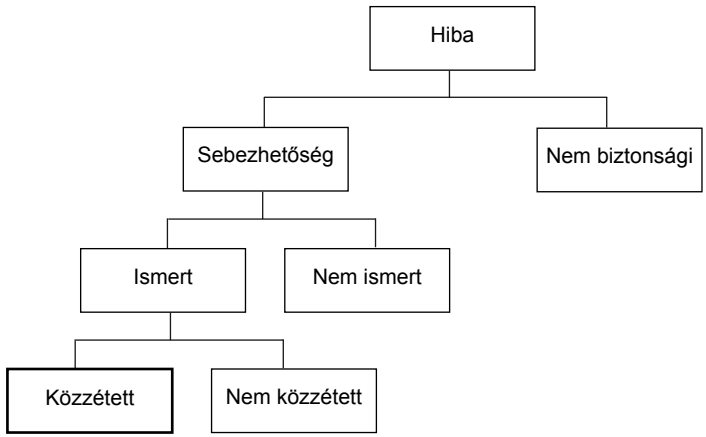
Szoftver hibák, sebezhetőségek és javításuk

A szoftvereket bizonyos feladatok elvégzésére terveznek meg és készítenek el. Előfordul azonban, hogy más módon történik meg a végrehajtásuk, mint azt a készítő tervezte, ezért bizonyos feltételek teljesülése esetében a szoftver egyáltalán nem, vagy csak korlátozott mértékben működőképes. Anderson szerint [5] szoros kapcsolatban van a szoftver hibák és a forráskód sorainak a száma között. A kettő közti arány 1:35.

[5] Anderson, R. (2001): *Why information security is hard - An economic perspective.* In: Proceedings of the 17 th Computer Security Applications Conference.

A szoftverek hibái és sebezhetőségei az alábbiak szerint jellemezhetőek:

1. ábra. Szoftverek hibáinak és sebezhetőségeinek osztályozása.



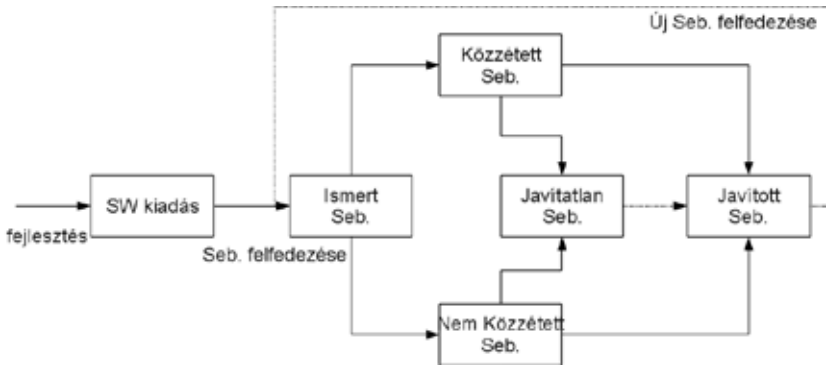
CVE (Common Vulnerability and Exposures) [6] az alábbiak szerint definiálja a sebezhetőség fogalmát:
„A sebezhetőség (vulnerability) egy olyan hiba adott szoftverben, amelyet közvetlenül kihasználhat egy támadó, és ezáltal hozzáféréshez jut adott rendszerhez vagy hálózathoz."

A sebezhetőség egy állapot az informatikai rendszerben, ahol valamelyik teljesül az alábbiak közül:
– A támadó képes parancsok végrehajtására másik felhasználó nevében.
– A támadó képes olyan adatokat elérni, ami ellentmond a hozzáférés szabályozásnak.
– A támadó kiadja magát más entitásnak.
– Lehetővé teszi, hogy a támadó szolgáltatás-megtagadást indítson.

A „kitettség" (exposure) egy állapot az informatikai rendszerben, ami nem egy sebezhetőség, hanem valamelyik teljesül az alábbiak közül:
– Futnak olyan szolgáltatások, amelyeket „információ begyűjtésére" lehet használni.
– Nem megfelelő biztonsági szabályok/beállítások alkalmazása.
– Támadási felületet adó szolgáltatások futtatása (pl.: HTTP, FTP, SMTP, …).
– „Gyenge védelemmel" rendelkező alkalmazások futtatása (könnyen törhetők „brute force"-szal).
A sebezhetőségek felfedezéséhez és javításához tartozó folyamatmodellt láthatunk a *2. ábrán.*

[6] CVE https://cve.mitre.org

2. ábra. Sebezhetőségek életciklusa.



– Egy sebezhetőség/sérülékenység a szoftverfejlesztő „kódolási hibájából", vagy a fejlesztőeszköz által, vagy egyéb speciális módon kerülhet bele egy kiadott szoftverbe.
– A sebezhetőségek felfedezése önkéntesek és informatikai biztonsággal foglalkozó szakemberek vizsgálatai révén kerülhetnek napvilágra. Az etikus (white hat) hackerek általában értesítik az érintett szoftver gyártóját a sebezhetőség létezéséről és körülményeiről. A „rossz fiúk" (black hat hackers) pedig egymás között terjesztik és megpróbálnak a sebezhetőségek révén támadásokat végrehajtani megbízóik számára és ezzel pénzt keresni. Nekik nem érdekük, hogy a sebezhetőség „napfényre kerüljön", és a védelmi szoftverek képesek legyenek az érzékelésére. Amennyiben egy sebezhetőség publikálásra kerül, azonosítót rendelnek hozzá, és belekerül egy bárki által hozzáférhető adatbázisba. pl.: CVE [6]
– Javítás (patching). A publikálást követően a szoftvergyártó elemi érdeke, hogy a sebezhetőség a lehető leggyorsabban javításra kerüljön.

Nyílt forráskódú szoftverek a CVE adatbázisában

A CVE adatbázis 1999 óta tartalmaz rekordokat az adatbázisában. A cikk írásokor több mint 75000 adatbázis-rekordot tartalmazott különböző operációs rendszerekre és szoftverekre vonatkozóan.

A vizsgálatom során az alábbi szoftverekkel foglalkoztam:
– GNU bash (Bourne-Again SHell) – népszerű nyílt forráskódú parancsértelmező,
– OpenSSL – az SSL és TLS protokollok nyílt forráskódú implementációja,
– OpenSSH – az SSH (Secure Shell) nyílt forráskódú implementációja,

– Apache – Nyílt forráskódú webkiszolgáló alkalmazás,
– MySQL – Nyílt forráskódú adatbázis-kiszolgáló alkalmazás,
– Linux – Nyílt forráskódú kernel és operációs rendszer.

Az *1. táblázat*beli értékek reguláris kifejezések alkalmazásával történő szűrések eredményeit mutatják.

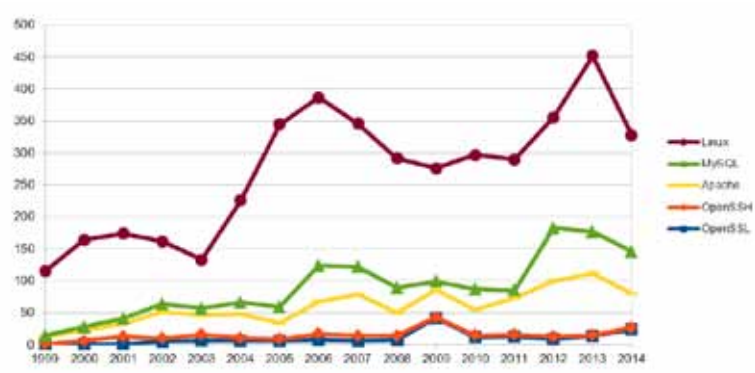
1. táblázat. Népszerű nyílt forráskódú szoftverek sebezhetőségeinek darabszáma.

Program	Sebezhetőségek darabszáma 1999 - 2014 nov.
GNU bash	7
OpenSSL	150
OpenSSH	76
Apache	711
MySQL	489
Linux	2891

A táblázat eredményeiből kitűnik, hogy a GNU bash esetében meglehetősen kis számú sebezhetőség látott napvilágot. Pedig köztudottan a bash jelenléte domináns a vizsgált intervallumban Linux és egyéb Unix-szerű operációs rendszerek körében.

Felettébb különös, hogy a 7 GNU bash biztonsági hiba közül 6 felfedezése 2014-ben történt.

3. ábra. Sebezhetőségek számának változása évenkénti bontásban.



[7] TLS – DTLS Heartbeat Extension - IETF RFC 6347 http://tools.ietf.org/html/rfc6520

A *3. ábrán* látható, hogy többi vizsgált nyílt forráskódú szoftver sebezhetőségei emelkedő számot mutatnak az utóbbi években.

Véleményem szerint ez nem mutat extrém eltéréseket a korábbi években tapasztaltakhoz képest.

A tendencia azonban emelkedést mutat, ami lehet annak a következménye, hogy fokozódik a nyílt forráskódú szoftverek biztonságára fordított figyelem.

A „Heartbleed Bug"

A sebezhetőség a CVE-2014-0160-es azonosítót kapta. Az itt lévő leírása az alábbiakat tartalmazza: „The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug."

A TLS és DTLS protokollok lehetővé teszik, hogy két különböző számítógépen futó folyamat biztonságos csatornán keresztül tudjon kommunikálni egymással. A gyakorlatban elsősorban a biztonságos web és e-mail megvalósítások esetében használatosak.

A sebezhetőség révén a lehetséges érzékeny információkhoz való hozzáférés. Ezek az információk lehetnek felhasználónevek, jelszavak, egyéb hitelesítési adatok, e-mailek, vagy weben keresztül megjelenő vagy bevitt adatok.

Az RFC 6520-nak megfelelően [7], a heartbeat kiterjesztés egy új protokoll a TLS/DTLS-hez fejlesztve, amely lehetővé teszi a kapcsolatfenntartás (keep-alive) funkcióját anélkül, hogy újra kellene a két félnek egyeztetni és felépítenie a kapcsolatot.

```
tls1_process_heartbeat(SSL *s)
	{
	unsigned char *p = &s->s3->rrec.data[0], *pl;
	unsigned short hbtype;
	unsigned int payload;
	unsigned int padding = 16; /* Use minimum padding */

	/* Read type and payload length first */
	hbtype = *p++;
	n2s(p, payload);
	pl = p;

	if (s->msg_callback)
		s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
			&s->s3->rrec.data[0], s->s3->rrec.length,
			s, s->msg_callback_arg);

	if (hbtype == TLS1_HB_REQUEST)
		{
		unsigned char *buffer, *bp;
		int r;

		/* Allocate memory for the response, size is 1 bytes
		 * message type, plus 2 bytes payload length, plus
		 * payload, plus padding
		 */
		buffer = OPENSSL_malloc(1 + 2 + payload + padding);
		bp = buffer;

		/* Enter response type, length and copy payload */
		*bp++ = TLS1_HB_RESPONSE;
		s2n(payload, bp);
		memcpy(bp, pl, payload);
		bp += payload;
		/* Random padding */
		RAND_pseudo_bytes(bp, padding);

		r = ssl3_write_bytes(s, TLS1_RT_HEARTBEAT, buffer, 3 + payload + padding);

		if (r >= 0 && s->msg_callback)
			s->msg_callback(1, s->version, TLS1_RT_HEARTBEAT,
				buffer, 3 + payload + padding,
				s, s->msg_callback_arg);

		OPENSSL_free(buffer);
```

A heartbeat protokollt arra tervezték, hogy válaszként küldje vissza a kapott adatokat, amelyeket a kérésben kapott. Itt az egyik paraméter, a hossz-paraméter (s->s3->rrec.length). A probléma hátterében egy elmulasztott ellenőrzés áll. Mivel a hossz-paraméter értéke nincs ellenőrizve, ezért egy nem megfelelő hossz-paraméterrel lehetséges más memóriatartalom kiolvasása is, aminek a révén a korábbiakban felsorolt adatok hozzáférhetővé válhatnak.

[8] Shellshock bug https://shellshocker.net/

A bug megjelenése: 2012. március 14. OpenSSL 1.0.1-es verzióban.
Javításra került: 2014. április 7. OpenSSL 1.0.1g verzióban.

A legtöbb Linux disztribúció érintett benne. Különösen az alábbi népszerű alkalmazások: Apache, OpenSSH, VPN, SMTP(S) kiszolgálók.

A „Shellshock Bug" (Bashdoor)

A 2014 szeptemberében a GNU bash szoftverben több biztonsági hiba került feltárásra. Ezek a sebezhetőségek „Shellshock bug"-ként, vagy „Bashdoor"-ként kerültek be a köztudatba. [8] A sebezhetőségek az alábbi azonosítókat kapták:
– CVE-2014-6271,
– CVE-2014-6277,
– CVE-2014-6278,
– CVE-2014-7169.

Ezen sebezhetőségek leírása az alábbiakat tartalmazza: „GNU Bash through 4.3 processes trailing strings after function definitions in the values of environment variables, which allows remote attackers to execute arbitrary code via a crafted environment, as demonstrated by vectors involving the ForceCommand feature in OpenSSH sshd, the mod_cgi and mod_cgid modules in the Apache HTTP Server, scripts executed by unspecified DHCP clients, and other situations in which setting the environment occurs across a privilege boundary from Bash execution, aka „ShellShock."

A bash-képes függvények értelmezésére és végrehajtására nagyon sok alkalmazás környezeti változókat használ, amelyeken keresztül lehetséges az alkalmazások működését befolyásolni. A sebezhetőségnek köszönhetően azonban speciális függvény-definíció hatására nem környezeti változók értékadása történik meg, hanem tetszőleges alkalmazás futtatása válik lehetővé.

A sebezhetőség tesztelésére használható az alábbi egyszerű parancs:

```
env x='() { :;}; echo vulnerable' bash -c "echo this is a
test"
```

A parancs definiálja az x környezeti változó értékét, majd a bash segítségével végrehajt egy echo parancsot, amellyel a „this is a test” szöveg kerül kiírásra. Amennyiben olyan bash-verziót használunk, ahol ez még nem került javításra, akkor az x környezeti változó definiálásakor beillesztett echo vulnerable parancs is végrehajtásra kerül.

A parancs végrehajtásának eredménye sebezhető operációs rendszerben:

```
user@unpatched:~$ env x='() { :;}; echo vulnerable' bash -c "echo this is a test"
vulnerable
this is a test
user@unpatched:~$
```

Amennyiben a sebezhetőség már javításra került, akkor ugyanazon parancs kimenetében már nem látszik a *vulnerable* szó:

```
user@patched:~$ env x='() { :;}; echo vulnerable' bash -c "echo this is a test"
this is a test
user@patched:~$
```

A sebezhetőség felhasználásával a bash révén befolyásolható minden olyan alkalmazás, amely környezeti változók értékét veszi figyelembe a végrehajtás folyamán. Ezek közül a legnépszerűbbek:
– Apache és egyéb CGI-t (Common Gateway Interface) használó web kiszolgálók
– OpenSSH szerver,
– DCHP kliens alkalmazások,
– Qmail levelező szerver.

A sebezhetőségen túl további kettő (CVE-2014-7186, CVE-2014-7187) bash-sebezhetőség is napvilágot látott. Ezek mindegyike a puffer-túlcsordulásos hibák közé sorolható. A bash nem megfelelően ellenőrzi a használt memória határait, ezáltal kártékony kód (malicious code) beszúrására és futtatására alkalmas.

A detektálás és a védekezés lehetőségei

A korábbiakban bemutatott sérülékenységekkel kapcsolatban a legnagyobb gond, hogy a jelenlegi szoftver-sebezhetőségek keresésére használt eszközök nem képesek ezek detektálására. Ennél fogva csakis programozók/biztonsági teszttel foglalkozó személyek által ismerhetőek fel és vizsgálhatók.

Ez azt vonja maga után, hogy várhatóak a közeljövőben hasonló jellegű sebezhetőségek eseti jelleggel fognak megjelenni, amelyek előfordulása statisztikai módszerekkel nem becsülhető meg és jelezhető előre. [9]

A védekezés területén konkrét intézkedéseket nehéz megfogalmazni. Általánosságban azonban elmondható:

– Bármely alkalmazás/hálózati szolgáltatás esetében figyeljünk oda a használt környezeti változókra. Amennyiben adott változókra nincsen szükségünk, abban az esetben ne engedélyezzük ezek használatát. Ha ezek feltétlenül szükségesek a megfelelő működéshez, akkor pedig figyeljük ezek értékeit, és lehetőség szerint naplózzuk az ezekben bekövetkezett változásokat.

– Sokszor a biztonságosnak gondolt hálózati szolgáltatások révén kerülhet adott informatikai rendszer kompromittált állapotba. Ezen szolgáltatások iránt nagyobb az „üzemeltetői bizalom", holott ezekre ugyanolyan figyelmet kell fordítani, mint a köztudottan kevésbé biztonságosnak ítélt szolgáltatások esetében.

– Az ismeretlen sebezhetőségek előzetes érzékelése nem lehetséges. Ami segítheti egy „éles rendszer" védelmét, hogy „honeypot"-szerűen futtatunk és megfigyelünk a saját rendszerünkhöz hasonlókat. Ha ezekben sikerül elkapni, azonosítani adott fenyegetést, támadási módszert, akkor esetleg még időben felkészülhetünk a saját „éles rendszerünk" védelmére.

Összegzés

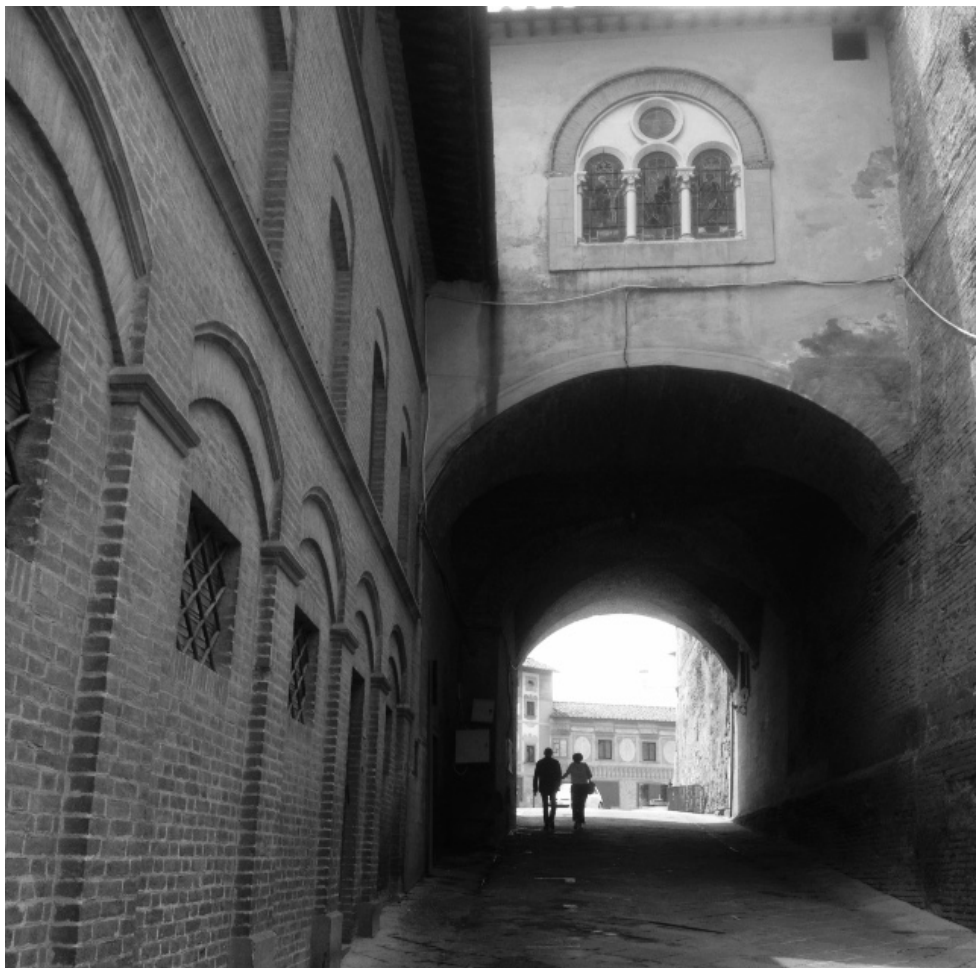
A nyílt és zárt forráskódú rendszerekkel kapcsolatban megállapíthatjuk, hogy mindkettő esetében előfordulhatnak biztonsági hibák. [10] A nyílt forráskódú szoftverfejlesztési modellben az egyes sérülékenységek általában hamarabb publikálásra kerülnek. Az emberi tényező szerepe egyre fontosabbá válik a szoftverfejlesztésben. Fokozottan jelentkezik az igény az oktatás irányába, hogy ne csak „kódolni", hanem „biztonságosan kódolni" kell a jövő programozóit megtanítani. Olyan súlyos és nagy hatású biztonsági hibák, mint amilyen a „Heartbleed" és a „Shellshock bug", előfordulása eseti jelleggel a közeljövőben is várható. Ezen sebezhetőségek megjelenése rávilágított, hogy nyílt forráskódú alkalmazásokat is kellő körültekintéssel kell használni és üzemeltetni.

[9] Kupsch, J. A.– Miller, B. P. (2014. April 22): *Why Do Software Assurance Tools Have Problems Finding Bugs Like Heartbleed?* [PDF file]. Continuous Software Assurance Marketplace. Retrieved from https://continuousassurance.org/swamp/SWAMP-Heartbleed.pdf

[10] Schryen, G. (2011): *Is Open Source Security a Myth?* Communications of the ACM.

Galéria

Németh István fotói




32