Nemzeti Közszolgálati Egyetem
Katonai Műszaki Doktori Iskola

Mészáros Gergely

Létfontosságú Információs Rendszerelemekben alkalmazott nyílt forráskód és szabad szoftver rendszerszemléletű biztonsági analízise

című Doktori (PhD) értekezés szerzői ismertetője

Tudományos Témavezetők:

Prof. Dr. Haig Zsolt mk. ezredes **Dr. Muha Lajos mk. alezredes**

Budapest, 2020

Tudományos probléma megfogalmazása, időszerűsége

Korunk sokszorosan összetett, egymásba fonódó technológiákon alapuló információs társadalma egyre komolyabb kihívás elé állítja a biztonsági szakembereket. A rendszerek összetettsége, az alkalmazott eszközök és módszerek száma folyamatosan nő, lassan olyan mértéket érve el, amelyet az emberi elme már képtelen befogadni. Elég csak a képzésre gondolni. Míg alig néhány évtizede egyszerűen informatikusokat (sőt elektromérnököket) képeztek, ma már tengernyi szakirány létezik, külön szakmává válik a felhőtechnológia, adatelemzés, mesterséges intelligencia kutatás vagy a kiberbiztonság, de lassan azt is tovább bonthatnánk kriptográfiára, hálózatbiztonságra, üzemeltetésre, kockázatmenedzsmentre és sok egyéb területre. Az informatika exponenciális tudásnövekedése folytán ma már elképzelhetetlen, hogy valaki informatikai polihisztor legyen.

Az összetett világ összetett fenyegetéseket jelent, ha pedig mindezt összevetjük a mindenhová begyűrűző információs technológiával, meglehetősen veszélyes elegyet kapunk. Különösen veszélyeset, ha olyan területről van szó, ahol emberéletek foroghatnak kockán, ahol egy esetleges hiba vagy támadás jelentős kárt képes okozni, azaz a Kritikus Infrastruktúrák, más néven Létfontosságú Rendszerelemek területén (továbbiakban LRE).

A különféle Létfontosságú Rendszerelemek egymással összetett függőségi viszonyban állhatnak, így az sem feltétlen szükséges, hogy maga a rendszer kritikus legyen, elegendő, ha kiesésével más kritikus rendszer működése válik lehetetlenné, így a működési zavar térben és időben szétterjedve a lakosság ellátásában, a gazdaság vagy kormányzat működésében súlyos problémákat okozhat.

Az informatika fejlődésének egyik szembetűnő jelensége a nyílt forrású technológiák egyre erősödő terjedése. Néhány évtizede műkedvelők játékának, esetleg naiv ideológiának tartott fejlesztési modell üzleti támogatók seregét maga mellé állítva az ipar egyik legfontosabb tényezőjévé nőtte ki magát.

A Nyílt Forrás (továbbiakban FLOSS[1]) fogalom az utóbbi években jelentős utat járt be a kezdeti, első sorban ideológiai indíttatású Szabad Szoftver elképzeléstől számítva. Ez a piaci szereplők által támogatott, modern változat már egyértelműen mainstream technológiának számít, üzletileg elfogadható sőt kívánatos elemnek, amely gyakran alkalmazott megoldás mind a szoftverfejlesztés mind a terjesztés során. Használatának előnyei nyilvánvalóak: megengedi a hozzáférést a forráskódhoz, lehetővé teszi a származtatott termékek terjesztését és jelentős befektetés nélkül is elérhetővé teszi az élvonalbeli technológiákat. Ezekkel a potenciális előnyökkel nehéz versenyezni.

Szinte minden területen nyílt forrású fejlesztésekkel találjuk szembe magunkat, nyílt fejlesztést vezeti ma már előremutatónak, “trendinek” számít, olyan vezető tech cégek igyekeznek meggyőzni bennünket a nyílt forrással való szoros barátságukról mint a Google, Facebook vagy akár a Microsoft.

Nagyon erős nyílt forrás jelenléte az IoT[2] technológiák világában, a webszerverek és általában a szerver

[1] Free Libre and Open Source software.

[2] Internet of Things, a “dolgok internete”, egymással kommunikálni képes fizikai eszközök (épületek, járművek, háztartási és megfigyelő eszközök stb.) komplex hálózata.

üzemeltetés területén. A valamikor egyetlen – bár igen népszerű – webszerver köré szerveződő Apache™ alapítvány ma már közel kétszáz projektet gondoz számos területen. Sok közülük egyáltalán nem nevezhető jelentéktelennek, az Apache™ Hadoop® piaca például 2022-re az előrejelzések szerint meghaladja a 50 milliárd dollárt. A nyílt forrású motorra épülő böngészők pedig szinte teljes egészében uralják a piacot, ami jelentős változás a néhány évtizeddel korábbi állapothoz képest. A nyílt forrású technológiák egyértelműen elterjedtnek számítanak az üzleti világban és a kutatói munkában egyaránt.

A támogatott nyílt forrás elterjedése végül maga után húzta a klasszikus modellt is. A nyílt forrás mint használható alternatíva elfogadottsága nagy mértékben megnőtt és implicit vagy explicit módon begyűrűzik a korábban kizárólag üzleti termékek által uralt területekre is. A programfejlesztésben ugyanis napjainkban bevett szokás a nyílt forrásból származó komponensek extenzív használata – ezáltal a nyílt forrás – közvetett módon a technológia minden területére kihat.

Joggal merül fel tehát a kérdés, hogy vajon ez a megváltozott felállás milyen hatást gyakorol az információs rendszerek biztonságára, van-e jelentős eltérés az üzleti rendszerekhez és komponensekhez képest, valamint szükséges-e változtatni az alkalmazott védelmi eljárásokon.

A FLOSS fejlesztési módszertan vonzó lehetőség mind a technológiai óriások, mind a frissen induló startupok számára. Függetlensége, nyíltsága és átláthatósága révén használatának igénye egyre gyakrabban felmerül a közigazgatásban is.

A szoftver vagy komponens teljes átláthatósága komoly előnyöket is hordoz, sőt, a kormányzati szférában idővel akár követelmény is lehet. A könnyű elérhetőség és választék nagy mértékben lerövidíti a fejlesztési időt, amit egy adott piaci helyzetben egyszerűen nem lehet figyelmen kívül hagyni. Ilyenformán a FLOSS felhasználás stratégiai cél is lehet. Tapasztalataim szerint viszont a szervezetek legtöbbször nem rendelkeznek célzottan FLOSS specifikus szabályozással. A FLOSS fejlesztési környezete vagy felhasználásának körülményei ugyanakkor olyan mértékben eltérhet a megszokottól, hogy a meglévő szabályozás már nem alkalmas annak kezelésére. Ez szervezeti, állami vagy nemzetközi szinten egyaránt igaz lehet.

Természetesen az egész problémakört nagyon egyszerűen szőnyeg alá lehet söpörni annyival, hogy a szervezet nem használ semmilyen nyílt forrású terméket, tehát az üggyel nem kell foglalkozni. Ez a megközelítés azonban véleményem szerint ma már elégtelen, sőt veszélyes. A szervezet nem biztos, hogy tudatában van annak, ha FLOSS elemeket használ. Az informatikai fejlesztésekben rendkívül széles körben elterjedt FLOSS komponens használat következtében a szervezet beszállítói és fejlesztői nagy valószínűséggel használnak FLOSS komponenseket, továbbá az sem zárható ki, hogy a szervezet alkalmazottai rendszeren kívül használjanak ilyen terméket. Emiatt akkor is kell foglalkozni a FLOSS kérdésével ha a szervezet teljesen elhatárolódik a FLOSS felhasználástól, legalább annyiban, hogy ezt az elhatárolódást szabályozás révén a gyakorlatban is biztosítani lehessen.

Összefoglalva, a FLOSS biztonsági hatásait célzó rendszer szintű kutatás elvégzését három tényező indokolja:

- Az egyre erősödő kiberfenyegetés;

- FLOSS felhasználás növekvő mértéke és implicit jellege;
- meglévő FLOSS specifikus szabályozás hiánya vagy elégtelen volta.

A nyílt modell információs rendszerek biztonságára gyakorolt hatását nem lehet anélkül elemezni, hogy pontosan ismernénk azokat a hatáspontokat ahol és amilyen módon az információs rendszer kapcsolatba kerülhet FLOSS rendszerekkel. A kezdeti szabad szoftver fogalmától napjainkra egy összetett nyílt fejlesztési modellig jutottunk el, amely technológiánkat nyíltan vagy rejtetten számos helyen átszövi. Ennélfogva a célkitűzéseim között szerepelt, hogy a vizsgálatba ne pusztán az "ingyenesen felhasználható" Szabad Szoftverek köre kerüljön be, hanem minden olyan tény és információelem, amely a nyílt fejlesztési modell sajátosságaiból adódóan publikusan elérhető. A Szabad Szoftverek mellett ide értve FLOSS komponenseket, forrástárakat, szoftvertárolókat a nyílt fejlesztés során létrehozott és felhaszált minden metaadatot sőt, a vizsgálat tárgya kell legyen a fejlesztői közösség kommunikációja, szociális és gazdasági viszonyai és az egyes szereplők egymásra gyakorolt hatása is.

Tekintettel arra, hogy a FLOSS felhasználás több módon is megvalósulhat, valamennyi esetet érdemes vizsgálat alá vonni. A szervezet lehet közvetlen vagy – beszállítóin keresztül – közvetett felhasználó, lehet közösségi partner és a fejlesztési folyamat részeként saját vagy piaci célokra terméket előállító szereplő is. Fontos jellegzetesség a projektben való közvetlen részvétel vagy a saját belső fejlesztésben történő felhasználás.

Jelenleg nincs olyan általam ismert kutatás amely a nyílt fejlesztési modell módszertanának biztonsági hatásait komplex módon elemzi. Sok tanulmány foglalkozik a nyílt forrás minőségbiztosításának kérdésével, még több a vélt vagy valós előnyökkel és hátrányokkal, de nyílt forrású fejlesztési módszertan biztonsági hatásainak megértéséhez nem elegendő egyetlen oldalról megközelíteni a kérdést.

A FLOSS biztonságra gyakorolt hatása jelentős, sokrétű és rejtett lehet. Mind tudományos mind gyakorlati szempontból érdekes, hogy a jelenség milyen mértékű és horderejű hatást gyakorol a magas biztonsági követelményeket támasztó rendszerek, különösen a Létfontosságú Információs Infrastruktúrák biztonsági szintjére. Vajon elegendő-e a meglévő szabályozás, képesek-e a szervezetek megfelelni a változó körülményeknek és helyén tudják-e kezelni a FLOSS felhasználásból eredő esetleg szokatlan kockázatokat?

A kutatás tervezése során ezekre a kérdésekre igyekeztem választ találni.

Célkitűzések

A kutatás alapvető célkitűzése annak meghatározása volt, hogy a magas biztonsági követelményeket támasztó információs rendszerek – különösen és elsősorban a Létfontosságú Információs Rendszerelemek (továbbiakban LIRE) – biztonságára milyen hatást gyakorolhat a FLOSS technológiák napjainkban tapasztalható előretörése. A célkitűzésben megfogalmazott alapkérdésének megválaszolásához szükség volt a kérdés felbontására.

Egyrészt meg kell határozni, hogy milyen utakon kerülhet kapcsolatba a FLOSS és a Létfontosságú Rendszerelem információs rendszere és milyen szintű együttműködés szükséges ahhoz, hogy a biztonságra gyakorolt hatás már érezhető legyen. Másrészről, be kell azonosítania azokat a FLOSS sajátosságokat, amelyek konkrét biztonsági hatást képesek gyakorolni az információs rendszerre. A sajátosságok és a hatáspontok meghatározása után a következő lépés a konkrét biztonsági hatások és azok kiaknázását vagy elkerülését célzó műveletek feltérképezése és rendszerezése. Végül, véleményem szerint úgy lehet megítélni a FLOSS jelenség által kifejtett hatások jelentőségét, ha sikerül felderíteni a meglévő szabályozás által már lefedett és le nem fedett területeket, azaz be kell azonosítani azokat a pontokat ahol a FLOSS ténylegesen ki is tud fejteni pozitív vagy negatív hatásokat. Szükségesnek tűnt tehát a beazonosított biztonsági hatásokat a meglévő védelmi intézkedésekkel és szabályozással összevetve következtetéseket levonni a nyílt fejlesztési modell ténylegesen realizálódó biztonsági hatásait illetően.

A fentieken túlmenően a kutatás reprodukálhatósága érdekében célom volt jól definiált és dokumentált módszertant alkalmazni és az Open Science irányelvei mentén valamennyi kutatási anyagot és adatot publikusan elérhetővé tenni. Ezáltal remélhetőleg a kutatás minden lépése nyomon követhető és az összegyűjtött információ a kutatóközösség számára könnyen elérhetővé válik.

A fentiekkel összhangban a következő kutatási célkitűzéseket fogalmaztam meg:

- Meghatározni és rendszerezni a nyílt modell azon sajátosságait, amelyek a befolyásolhatják az informatikai biztonságot.
  - Felmérni, hogy a FLOSS egyes hatásai mennyira kutatottak a különféle területeken.
  - Létrehozni a FLOSS jellegzetességeit összefoglaló, minél átfogóbb rendszert.
  - Beazonosítani azokat a sajátosságokat, amelyek hatással vannak a biztonságra.
- Meghatározni azokat a pozitív és negatív hatásokat amelyek a Létfontosságú Rendszerelemek biztonságát befolyásolhatják és feltárni a lehetséges hatáspontokat.
- Intézkedések formájában javaslatot tenni a negatív hatások elkerülésére és a pozitív hatások kiaknázására.
- Az eredményeket az érvényes hazai szabályozással összevetve következtetéseket levonni a FLOSS LIRE-ben történő felhasználhatóságára vonatkozóan.

Kutatási hipotézisek

A kutatás céljainak megfelelően a következő kutatási hipotéziseket állítottam fel:

H1. Feltételezem, hogy a nyílt fejlesztési modell és az így előállított termék olyan egyedi tulajdonságokkal rendelkezik, amely sajátos módon befolyásolja az információs rendszerek biztonságát.

H2. Feltételezem, hogy bizonyos egyedi sajátosságokból adódó kockázatok közvetve vagy közvetlenül hatást gyakorolnak a Létfontosságú Rendszerelemek biztonságára.

H3. Feltételezem, hogy definiálhatóak olyan védelmi intézkedések, amelyek a FLOSS felhasználás sajátosságaiból eredő biztonsági problémák okozta kockázatot mérsékelni tudják vagy alternatív megoldást jelentenek általános problémákra.

H4. Feltételezem, hogy az eltérő minőségű és forrású FLOSS termékek bizonyos csoportját megfelelő védelmi intézkedések foganatosítása mellett a legmagasabb biztonsági elvárásokat támasztó Létfontosságú Rendszerelemek területén is fel lehet használni.

Kutatási módszer

A kutatás célkitűzéseivel összhangban olyan módszert igyekeztem alkalmazni, amely átfogó képet nyújt teljes kérdéskörről, ugyanakkor módszeres elemzést tesz lehetővé, hogy a szubjektivitás mértékét a lehetőségekhez képest alacsony szinten tartsam. A választott módszertan tehát legyen rendszerszemléletű, reprodukálható és szisztematikus.

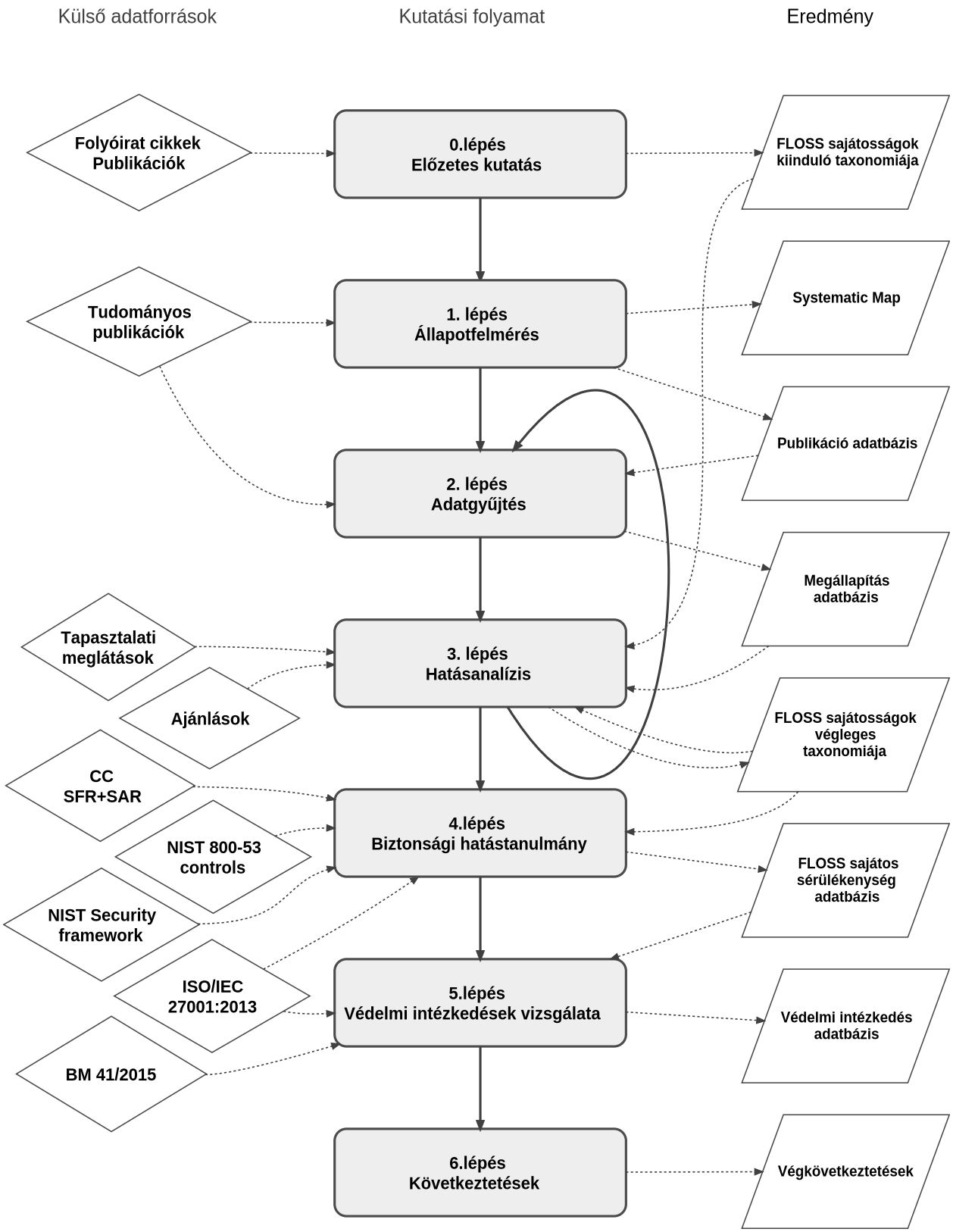
Az empirikus elemzések esetében az adatgyűjtés módszeréhez hierarchikus kérdésmegfogalmazás alapján juthatunk el. A szintek között dedukció és indukció módszerével haladva lehet a generalizált kérdésből kutatási kérdéseket alkotni majd az összegyűjtött eredményekből következtetéseket levonni. A kutatás tervezése során Punch, Leshem és Trafford által javasolt módszertan mentén haladva először a kutatási kérdéseket állítottam fel, majd a koncepcionális keretrendszert végül az alkalmazott módszereket és a kutatási szerkezeti felépítését határoztam meg.

A 1. ábrán a általam használt koncepcionális keretrendszer látható. Innen kiolvashatóak a kutatás szükséges lépései, a kutatás során feldolgozott információforrások és a várt eredmények.

Az ábra első oszlopában találhatóak a felhasznált információforrások. A kutatás elsődleges információforrása a feldolgozott cikkekből és tanulmányokból manuálisan kigyűjtött adatok. A hatásanalízis során meglévő ajánlásokat, illetve létező, aktívan használt keretrendszerek (ISO/IEC 27001, NIST 800-53, Common Criteria) javaslatait is figyelembe vettem. Ezáltal biztosított a kutatási célokban megfogalmazott teljeskörűség mind a sajátosságok, mind a biztonsági hatások tekintetében.

A második oszlopban lekerekített téglalpokkal jelölt kutatási folyamat lépéseiből látható, hogy a kutatás adatgyűjtő-elemző része ciklikus jellegű, azaz a hatásanalízis során finomított taxonomia és a biztonsági hatások bővülő kategóriái az adatgyűjtés során újra felhasználásra kerültek. A ciklikus megvalósításra azért volt szükség, mert a cimkézés során létrehozott új kategóriákat a már feldolgozott publikációk anyaga esetében is értelmezni kellett, valamint az időközben megjelent új publikációkat folyamatosan rendszerbe kellett illeszteni.

A harmadik oszlopban feltüntetett eredmények olyan dokumentumokat és adatbázisokat jelölnek, amelyek



1. ábra: A kutatás koncepcionális keretrendszere (szerkesztette a szerző)

a kutatás további fázisaiban, más kutatásokban vagy a gyakorlatban felhasználható információt és adatokat tartalmaznak.

Az első kutatási cél elérés érdekében Petersen által javasolt szisztematikus feltérképezés[3] módszerét alkalmaztam. A szisztematikus térképezés módszere széles körben alkalmazott praktikus eszköz a szoftvermérnöki területek osztályozási feladataira és a kutatás struktúrájának felmérésére. Ez az analízis a publikációk kategóriánkénti sűrűségére koncentrál, így meghatározható a terület kategóriánkénti becsült fedettsége. A klasszikus szisztematikus térképezés nem merül el a részletekben, a publikációkat nem elemzi részletesen. A javasolt módszerhez képest esetemben mélyebb elemzésre volt szükség – közeledve a klasszikus szisztematikus forráselemzéshez – ugyanis a publikációk alkalmazott módszertanát és eredményeinek típusát is meg akartam határozni.

A FLOSS sajátosságok kategóriáihoz az előzetes kutatások során előállított FLOSS sajátosság taxonomia vázlatot használtam fel amelyet a az összegyűjtött anyagok segítségével fokozatosan pontosítottam.

A szisztematikus feltérképezés alapját képező gyűjtőmunka kettős célt szolgált. Egyrészt a szisztematikus feltérképzés segítségével meghatározhatóvá vált, hogy mely területek milyen mértékben kutatottak, azaz az eredmények várható megbízhatósága és teljessége milyen szintű lesz az egyes területeken, másrészt az összegyűjtött és felcímkézett forrásanyag alapját képezhette a kutatás következő analitikus fázisának.

A feltérképezés és adatgyűjtés első fázisát 2016-ban végeztem. A teljes anyagmennyiség feldolgozása és analízise hosszú időt vett igénybe, ezért a meglévő publikáció-adatbázist folyamatosan frissítésekkel láttam el. A szisztematikus feltérképezés adatai azonban a korábbi állapotot tükrözték, így a kategorizálást azonos keresési metodika használata mellett 2020-ban megismételtem. Az irodalomkutatás így a 2020 elején fennálló állapotot tükrözi.

A második kutatási cél elérése érdekében az analízis-szintézis módszertanát alkalmaztam. Az első fázisban összegyűjtött és folyamatosan kiegészített dokumentumokból felépítettem a FLOSS sajátosságainak lehető legteljesebb modelljét, majd a modell alapján meghatároztam a biztonsági hatásokkal kapcsolatos jellemzőket illetve – amennyiben voltak ilyenek – a javasolt megoldásokat. Az első kutatási fázis analízis kimenetének szintézisével határoztam meg a második kutatási cél keretében megfogalmazott területeket és hatáspontokat.

A FLOSS sajátosságok és a hatáspontok modelljét, valamint a magas biztonsági rendszerekre vonatkozó ajánlásokat és előírásokat összevetve szintézis segítségével határoztam meg a harmadik cél alatt megfogalmazott lehetséges védelmi intézkedéseket. A védelmi intézkedések kialakítását a NIST 800-53 security overlay mechanizmusa inspirálta. Hasonló elképzelés mentén terveztem a nyílt forrást alkalmazó és magas biztonsági követelményeknek megfelelni kívánó szervezetek – különösképpen a Létfontosságú Rendszerelemek – számára olyan intézkedés-gyűjteményt képezni, amely a meglévő szabályozást kiegészítve felhívja a figyelmet a FLOSS esetében eltérően vagy különös figyelemmel kezelendő pontokra.

Végezetül, a sajátosságokból eredő hatásokat és a megállapított védelmi intézkedéseket összevetettem a

[3]Systematic Mapping Study

vonatkozó ajánlásokkal és hazai szabályozással az esetleges fehér foltok feltárása érdekében.

Az értekezés felépítése

Az értekezés hat fejezetből áll. Az első fejezet a kutatási célok és módszertan meghatározására koncentrál. A második fejezet a kutatás tárgyával és irodalmával foglalkozik. Itt definiálom a kutatás témájának kereteit és a vizsgált területeket. Ugyanitt találhatóak az irodalomkutatás összefoglaló eredményei amelyek a kutatási célkitűzésekkel összhangban az egyes biztonsági területek vizsgálatának alaposságát kívánják meghatározni, amely alapján az eredmények teljeskörűsége végül becsülhető. A második fejezet végén a Létfontosságú Rendszerelemek és a nyílt forrás kapcsolódási pontjait vizsgálom, amely alapul szolgál a következő fejezetekben tárgyalt biztonsági analízishez.

A nyílt forrás belső és külső egyedi jellegzetességeivel az fejezetek közötti egyensúly megtartása érdekében két külön fejezet foglalkozik. A harmadik fejezet a FLOSS belső, azaz külső tényezőktől független tulajdonságait vizsgálja, míg a negyedik fejezet a külső tényezők és a FLOSS kapcsolatrendszeréből adódó sajátosságokat elemzi. A fejezetek az első célkitűzésben megfogalmazott egyedi sajátosságokat foglalják össze, amelyek alapján a második célkitűzés szerinti hatások és javaslatok beazonosíthatók.

A harmadik és negyedik fejezet rendezőelveként szolgáló rendszertant a 2. ábra mutatja be. Az egyes fejezeteket a könnyű azonosíthatóság érdekében az itt meghatározott kóddal jelöltem FS-főkategória-alkategória alakban. A navigáció megkönnyítése érdekében az elektronikus verziót kattintható hiperlinkekkel láttam el.

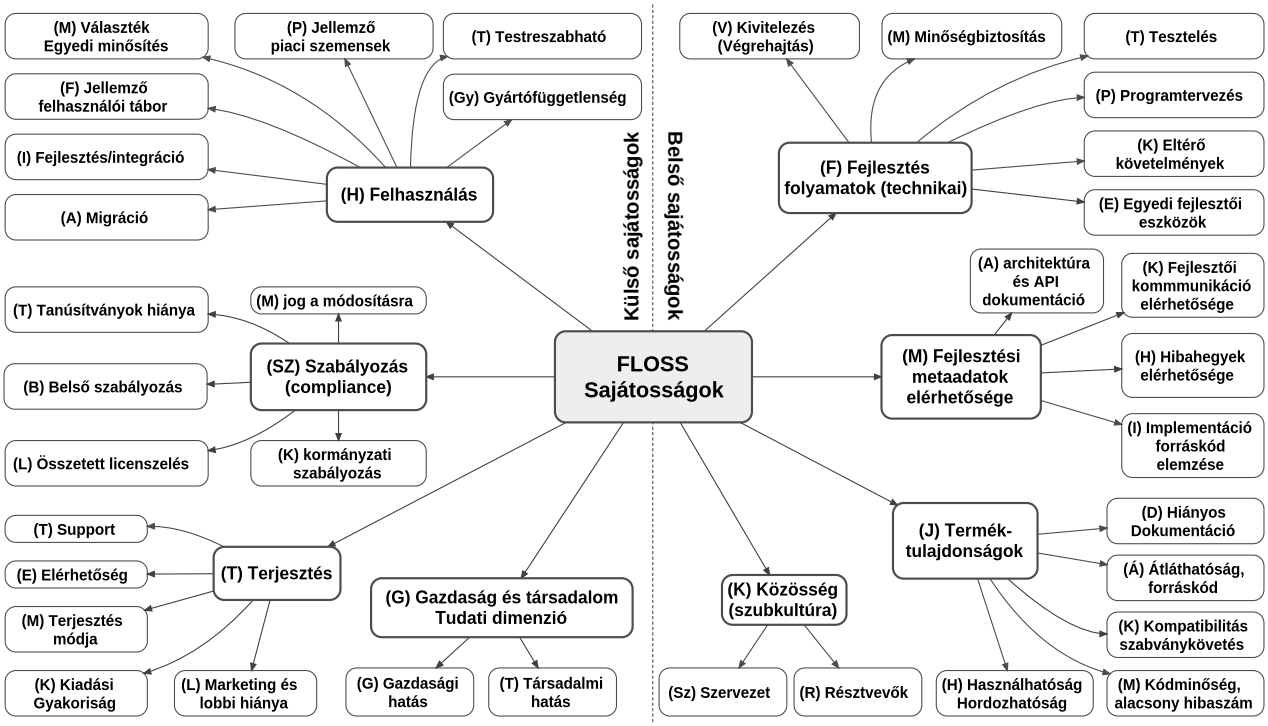
Az elemzett sajátosság kategóriák mindegyikénél meghatároztam a felmerülő sérülékenységeket valamint az azonosított javaslatokat, amelyeket az alfejezetek végén táblázatos formában foglaltam össze.

A további fejezetek az azonosított FLOSS sajátosságok hatását elemzik a Létfontosságú Rendszerelelemek területén. Ez két lépésben valósul meg. A negyedik fejezetben a NIST Cyber Security Framework által definiált információbiztonsági irányelvek mentén vizsgálom a nyílt forrás sajátosságainak lehetséges hatásait az egyes lépésekre, majd az ötödik fejezetben az azonosított feladatokat összevetem a Létfontosságú Rendszerelemek információbiztonsági követelményeit definiáló hazai szabályozással.

Az egyértelmű azonosíthatóság érdekében a sajátosságokat, intézkedéseket és biztonsági irányelveket egyértelmű kódolással láttam el, amelyet az egyszerű feldolgozhatóság érdekében adatbázisban foglaltam össze.

Összegzett következtetések

Értekezésemben a nyílt fejlesztési modellből származó szoftverek (továbbiakban FLOSS) és komponensek felhasználhatósági feltételeit vizsgáltam a Létfontosságú Rendszerelemek információs rendszereiben



2. ábra: A nyílt forrás sajátosságainak rendszertana (szerkesztette a szerző)

(továbbiakban LIRE).

A rendelkezésre álló források szisztematikus elemzésének segítségével megállapítottam, hogy a tudományos közösség jelentős túlsúllyal foglalkozik a FLOSS fejlesztési módszertan kérdéseivel, míg más aspektusok (biztonság, megfelelőség, életciklus menedzsment) kevésbé kutatottak. A biztonsági kérdéseket is érintő munkák döntő részben a nyílt forrású termékek szoftverminőség-vizsgálatára szorítkoznak. A könnyen hozzáférhető forráskód okán gyakran szerepelnek kutatások adatforrásaként, de a nyíltság konkrét hatásainak vizsgálatával a publikációk általában nem foglalkoznak.

A javasolt módszerek egyenletes eloszlást mutatnak a sajátosságok tekintetében, azaz a kutatói közösség valamennyit fontosnak tartja, ugyanakkor empirikus adatok és alapkutatási eredmények első sorban a nyílt forrású *fejlesztés* területén állnak rendelkezésre, ezért az eredmények döntő része is ide koncentrálódik. Megállapítottam, hogy a gazdasági hatással, tudati dimenzióval, terméktulajdonságokkal kapcsolatos jellemzők vizsgálatához szükséges adatot jelenleg kizárólag szük területre koncentrálódó (nem reprezentatív) felmérésekből lehet gyűjteni vagy egyáltalán nem áll rendelkezésre empirikus eredmény.

Megállapítottam, hogy a nyílt forráskód felhasználása nem mindig nyilvánvaló, ugyanis a felhasznált komponensek és alrendszerek révén akár többszörösen közvetett módon is megvalósulhat. A többszörösen indirekt felhasználás biztonságra gyakorolt egyértelmű negatív hatása, hogy a sérülékenységek javítása sokkal lassabban megy végbe, a zero-day sérülékenységek akár hosszú ideig is kihasználhatóak maradnak. *Következésképpen, a LIRE informatikai biztonsági szabályzatából akkor sem javasolt kihagyni a nyílt forrás hatásaival foglalkozó intézkedéseket, amennyiben ilyen rendszert a szervezet (látszólag) nem használ.*

Megmutattam, hogy LIRE és a hozzá kötődő LRE közvetlen FLOSS kapcsolódása alapvetően négyféle módon valósulhat meg: viszonteladótól beszerzett üzleti termékként vagy annak részeként, közvetlenül a közösségtől bináris vagy módosítás nélküli futtatható formában, belső elírások szerinti fordítás által, tehát módosított/ellenőrzött forráskód formában, végül a fejlesztői közösség részeként, a fejlesztésben való aktív részvétellel.

Az kapcsolat és együttműködés felsőbb szintjein a zárt forrás esetében ismeretlen védelmi lehetőségek nyílnak meg, de ezzel párhuzamosan a lehetséges sérülékenységek száma is növekszik.

A nyílt forrás sajátosságainak rendszerszemléletű elemzése során megállapítottam, hogy a FLOSS és fejlesztési metodikája bizonyos területeken olyan egyedi jellemzőkkel rendelkezik amelyek pozitív és negatív irányban egyaránt befolyásolhatják az informatikai biztonságot. Feltártam, hogy a felmerülő problémák egy részére a kutatóközösségnek már van valamilyen javasolt megoldása, de számos kérdés továbbra is nyitott és gyakran szabályozatlan marad.

Fejlesztési modelljét tekintve a FLOSS módszertan erősen eltérőnek mutatkozik, olyan jellemzőkkel bír, amelyek a zárt fejlesztés során kritikus hiányosságnak számítanának, ugyanakkor kiterjedt tesztelés és alternatív kódelfogadási módszerek révén végső soron mégis jó minőségű kódot képes előállítani. Megállapítottam, hogy a két modell az utóbbi években folyamatosan közeledik, mindkettő átvett a másikból bizonyos elemeket, de a különbségek még mindig elég jelentősek ahhoz, hogy a kooperáció során nehézségeket és így biztonsági kockázatot jelentsenek.

A FLOSS szoftverek és komponensek terméktulajdonságok tekintetében nem térnek el jelentősen az üzleti verzióktól, mindkét modellből származnak kimagaslóan jó minőségű és gyenge termékek. A különbség elsősorban a minőség tanúsításának módjában és a tanúsítás ellenőrizhetőségében mutatkozik meg. A FLOSS termékek átlagosan hordozhatóbbak de a hordozhatóság közvetlen biztonsági hatásait nem sikerült azonosítani.

A nyílt és zárt modell szervezeti felépítése és szociális struktúrája jelentősen eltér. A fejlesztőközösség szerkezete közösségi-hálózat szerű, irányítása magas technikai felkészültséget, jó szociális érzékenységet igényel. A nyílt közösség nehezen befolyásolható, működése viszont teljesen átlátható, így az esetleges kockázat könnyebben becsülhető. A közösség hatékony működése általában feltétele a hosszú távú működésnek, azaz a közösség segítése és irányítása, de legalább elemzése valamilyen módon szerepet kell kapjon az beszállítói lánc kockázatkezelésével kapcsolatos feladatokban. Stratégiai felhasználás esetén elengedhetetlen a szoros együttműködés, amelyet a közösségben való részvétel által érhető el.

Metaadatok tekintetében magától értetődő eltérés a nyíltság. Szabadon elérhető a forráskód, a rendszer API dokumentációja, a rendszerterv, a hibabejelentések, az azokra kapott válaszok, sőt sok esetben a hibák javítására tett kísérletek is. Ez az átláthatóság egyedülálló lehetőségeket biztosít de egyben biztonsági kockázatokat is hordoz. A metaadatok alapján kvalitatív vagy akár kvantitatív mérőszámokkal jellemezhető a projekt és a közösség minősége, készültségi foka, reakcióideje és általános állapota. Beazonosíthatóak a kulcsemberek, az egyes kódrészeket fejlesztők személye. Megfelelő keretrendszer esetén a teljes forrás-

kód működésének helyessége formálisan igazolható. Ezzel szemben a hibajegyeken vagy a forráskódon keresztül érzékeny információ válhat publikussá, a forrás és a hibajegyek elemzésével még javítás előtt felderíthetőek a sérülékenységek, a forrás manipulálásával az összes felhasználó támadható. A hibajavítás folyamatához másként kell hozzáállni mint az üzleti alkalmazások esetében, a hibajegyek nem megfelelő felvitele esetén azok javítása késhet vagy akár el is maradhat.

A nyílt forrásnak mint jelenségnek jelentős gazdasági és társadalmi haszna van, ugyanakkor a FLOSS hatásai és az nyílt modell módszerei ritkán szerepelnek az oktatásban, jóllehet az utóbbi években pozitív változás érzékelhető e téren. A megfelelő egyensúly megteremtése érdekében a FLOSS specifikus oktatásra kiemelt figyelmet kellene fordítani. Gazdasági és társadalmi hatás tekintetében nem tudtam egyértelmű biztonsági hatásokat azonosítani, azon túlmenően, hogy a jelenség ismerete előfeltétele a helyes védelmi eljárások alkalmazásának.

FLOSS esetén a megfelelő termék kiválasztása nem egyértelmű, a megszokott minősítő rendszerek nem használhatóak, egyedi adatbázisok és modellek szükségesek. Az igen széles körű FLOSS komponens integráció sajátos problémákkal küzd. A helyi változtatások kezelésére különös figyelmet kell fordítani. A vezetés és a kivitelezést végzők FLOSS-al kapcsolatos hozzáállásából adódó különbség oda vezethet, hogy a kivitelezők rejtett módon alkalmazzák a FLOSS komponenseket, ami szabályozás híján komoly biztonsági problémákhoz vezethet.

Megállapítottam, hogy a FLOSS szabályozása jelenleg nem teljesen kiforrott. Jogalkotási tekintetben megindultak a kezdeti lépések, de vállalati szinten általában nincs kifejezett FLOSS specifikus szabályzat, esetleg egyáltalán nem is foglalkoznak a kérdéssel, az elfogadott tanúsítási rendszerek pedig nem illeszkednek jól a nyílt modellhez. A FLOSS licencelése összetett, és bár ez közvetlen biztonsági problémákat nem okoz, a rendelkezésre állást veszélyeztetheti. A licencelés nem mindig egyértelmű, a jellemzően komponens alapú felépítés miatt helyzet felmérése is kihívás lehet. A feltárt hiányosságok, elsősorban a tanúsítások hiánya miatt nagyon fontos lenne a magasabb szintű összefogás, hiszen a gyártók – ez esetben a közösségek – nem képesek megszerezni a szükséges tanúsításokat és általában nem is érdekeltek ebben. A tanúsítás a felhasználó érdeke, amelyet viszont egyedileg rendkívül erőforrásigényes elvégezni, ami jelentősen hátráltatja a FLOSS termék kihasználhatóságát.

Üzleti támogatás hiányában a nyílt modell terjesztési metódusa egyértelmű eltérést mutat, amelyek a biztonságra is hatással lehetnek. A közösségi támogatás megbízhatatlan, használata időben nehezen tervezhető, szociális készségeket követel, ugyanakkor a piaci siker nem befolyásolja a támogatás minőségét. A bináris formában terjesztett termék vagy komponens sértetlensége elvben a forrás alapján tanúsítható, gyakorlatban viszont ez egyelőre nehézségekbe ütközik így a csomagolást végző fél iránti bizalom továbbra is követelmény. A belső támogatás alapját képező szervezett képzés gyakran hiányzik, így a megfelelő tudással rendelkező szakemberek felkutatása nehézségekbe ütközik. A formális képzési keretek hiányában a szaktudás megléte legfeljebb felvételi tesztekkel ellenőrizhető.

A fentiek alapján megállapítottam, hogy a nyílt forrású fejlesztési modellből származó termékek és komponensek olyan egyedi sajátosságokkal bírnak, amelyek befolyásolhatják a felhasználó szervezet informatikai

biztonsági szintjét. **Ennek alapján a H1 hipotézist elfogadtam.**

Megmutattam, hogy a nyílt forrású komponensek rejtett módon, többszörös áttételen keresztül is kifejthetik hatásukat a Létfontosságú Információs Rendszerelemekre, az azonosított sajátosságok okozta kockázatok között pedig vannak olyanok, amelyek ezeken a csatornákon keresztül közvetve vagy közvetlenül befolyásolják a LIRE biztonságát. **Ennek alapján a H2 hipotézist elfogadtam.**

A kifejezetten LRE elemzésekre szabott NIST Cybersecurity Framework módszertanát alapul véve meghatároztam azokat a védelmi intézkedéseket, amelyekkel a FLOSS specifikus sérülékenységek jelentette kockázat mérsékelhető. Valamennyi fő kategóriában azonosíthatóak voltak nyílt forrás specifikus elemek, következésképpen *a nyílt forrással kapcsolatos szervezeti szabályzatnak végig kell követnie a teljes folyamatot*.

Megállapítottam, hogy a COTS (dobozos) felhasználásnál magasabb szintű FLOSS felhasználási szint esetében *az üzleti termékek esetében bevált gyakorlattól jelentősen eltérő speciális lehetőségek és igények merülnek fel, amelyeket az azonosítás, védelem, felderítés, válaszlépések és helyreállítás tervezése és megvalósítása során egyaránt figyelembe kell venni*.

Az védelmi intézkedések egy része a nyílt fejlesztési modell sajátosságaiból adódó sérülékenységek kockázatának mérséklését célozza, néhány esetben azonban olyan új típusú védelmi intézkedéseket is sikerült azonosítani, amelyek alkalmazása zárt forrású termékek esetében elképzelhetetlen.

A definiált védelmi intézkedések alapján a H3 hipotézist elfogadtam.

A FLOSS sérülékenységek, javaslatok, védelmi intézkedések és a jogszabályi előírások összevetésével megállapítottam, hogy a FLOSS felhasználásnak a LIRE védelmére vonatkozóan egyértelmű előnyei azonosíthatóak, azonban a legmagasabb biztonsági elvárások által támasztott követelmények nem teljesíthetők maradéktalanul.

A szabályzat elvárja, hogy a szervezet szerződésben rögzítse az információbiztonságot érintő szerep és felelősségi köröket valamennyi külső partner esetén. A nyílt közösségek esetében ez nehézségekbe ütközik. A szerződő féllel szemben támasztott személyi biztonsági követelmények teljesítése problematikus, tekintve, hogy a nyílt közösség természeténél fogva anonim. A FLOSS fejlesztő közösség és általában maguk a szoftverek is elsősorban gráfszerkezetű WoT[4] alapú kriptográfiai tanúsítványokat alkalmaznak hierarchikus PKI helyett, amelyet a jelen jogi szabályozás nem kezel. Amennyiben a PKI tanúsítványokat kötelező érvényűnek tekintjük, azzal kizárjuk FLOSS rendszerek jelenetős körét valamint funkcionalitását (pl. automatizált frissítések) amelyek nélkül viszont más követelmények válnak teljesíthetetlenné. A biztonsági előírások a fejlesztőre is vonatkoznak, a tranzitivitás folytán végső soron a nyílt közösségnek is teljesítenie kéne valamennyi követelményt, amely néhány területen (pl. forrás megismerhetetlensége, adminisztratív intézkedések) egészen biztosan nem teljesül. A jogszabály előírja, hogy az információs rendszerre vonatkozó fejlesztői dokumentáció jogosulatlanok számára ne legyen megismerhető és módosítható, amely

[4]Web of Trust, az ismertebb PKI keretrendszer megoszló rendszerű, központi megbízható partner nélkül is használható alternatívája.

nyílt fejlesztési modellben való aktív részvétel mellett nem teljesíthető. A FLOSS projektek nagyon ritkán szereznek jogszabály által is elismert tanúsítást, így nem alkalmazhatók olyan esetekben ahol a tanúsítás megléte előkövetelmény. A fejlesztést végző szervezetnek biztonsági tervet kellene készítenie és dokumentálnia a tervben rögzített lépések elvégzését. A nyílt forrású projektek döntő részénél ez a lépés hiányzik.

A jelenlegi szabályozási környezetben a FLOSS felhasználása nehézkes, a felsőbb biztonsági szinteken a közösséggel együttműködve egyenesen kivitelezhetetlen. Az előnyök kiaknázásához és a védelmi intézkedések egy részéhez aktív közreműködés szükséges, ami gyakran túl nagy feladatot róna a szervezetre.

A fentiek alapján kijelenthető, hogy a nyílt projektek közvetlen felhasználása ütközik a jogszabályban foglaltakkal, a klasszikus FLOSS termékek nem felelnek meg maradéktalanul a legmagasabb biztonsági szint által definiált elvárásoknak, ezért a jelenlegi szabályozás alapján csak harmadik féltől származó termékekbe építve használhatóak. **A H4 hipotézist tehát el kell vetni.**

Tudományos eredmények

1. Szisztematikus és rendszerszemléletű feldolgozási módszer segítségével meghatároztam és egységes rendszerbe foglaltam a nyílt forrású fejlesztési modellből származó szoftverek és komponensek sajátosságait.

A Nyílt Forrás Létfontosságú Rendszerelemekre gyakorolt hatásainak megértéséhez mindenekelőtt tisztázni kell melyek a hagyományos szoftverfejlesztésből származó termékekhez képest mért elétérések és sajátosságok. A nyílt forrás sajátosságainak biztonsági hatása nem feltétlenül egyértelmű, nem elegendő tehát csak bizonyos, ad-hoc módon kiválasztott hatásokkal foglalkozni. Első lépésként tehát létre kellett hozni egy átfogó sajátosság-adatbázist, melynek alapján a szisztematikus analízis elvégezhető.

2. Az új osztályozási módszer mentén elvégzett analízis módszerével meghatároztam, hogy a nyílt forrású fejlesztési modell sajátosságai milyen konkrét hatást gyakorolhatnak a Létfontosságú Információs Rendszerelemek biztonsági szintjére, valamint azonosítottam a lehetséges hatásmechanizmusokat.

A konkrét biztonsági hatások azonosítása érdekében meg kellett határozni a nyílt forrású fejlesztési modell lehetséges hatáspontjait, amelyeken keresztül a vizsgált hatás egyáltalán kialakulhat. A hatásmechanizmus megértését követően a sajátosságok elemezése során feltárt hatások és a kutatóközösség által javasolt megoldások rendszerezésével egy olyan adatbázis jött létre, amely egységes, szűrhető és kereshető módon tárolja a biztonsággal kapcsolatos fellelhető információkat.

3. Az egyedi FLOSS sajátosságok és a Létfontosságú Rendszerelemek minősítésére alkalmas NIST Cybersecurity Framework kategóriáinak összevetésén alapuló deduktív stratégia alapján olyan lehetséges védelmi intézkedéseket definiáltam, amelyek mérsékelni képesek a FLOSS felhasználó szervezet által viselt kockázatokat.

A kutatóközösség által javasolt megoldások nem mindig fedik megfelelően az azonosított problémákat, to-

vábbá, még ha létezik is javasolt megoldás, az információ megtalálása rendkívül időigényes lehet. Logikus lépés tehát egy olyan egységes védelmi intézkedés adatbázis létrehozása, amely kereshető és rendszerezett formában rögzíti az ismert sérülékenységek hatásának csökkentése érdekében foganatosítható intézkedéseket. Az adatbázisnak figyelembe kell vennie a LIRE sajátosságait és az informatikai biztonság minnél nagyobb területét kell lefednie. Ennek érdekében a védelmi intézkedések kialakítása során a LIRE orientált NIST Cybersecurity Framework kategóriáit és javaslatait követtem.

4. A Létfontosságú Rendszerelemek biztonsági követelményeit szabályozó hazai jogszabályi környezet, a nyílt forrású modell egyedi sajátosságainak és a definiált védelmi intézkedések összevetésével feltártam, hogy a klasszikus FLOSS termékek nem felelnek meg maradéktalanul a legmagasabb biztonsági szint által definiált elvárásoknak, ezért a jelen szabályozás alapján csak harmadik féltől származó termékekbe építve használhatóak.

A FLOSS komponenseket felhasználó LIRE védelmének biztosításához a biztonsági hatások és védelmi intézkedések meghatározása önmagában még nem elegendő, hiszen bizonyos védelmi intézkedések kivitelezhetetlenek lehetnek, ütközhetnek az elsősorban zárt forrásra szabott elvárásokkal és irányelvekkel. A hatásokat és intézkedéseket a hazai elvárásokkal összevetve feltárhatóak azok a kritikus pontok, amelyek jelenleg akadályozzák vagy ellehetetlenítik a FLOSS felhasználását. Eredményeim alapján a kérdéses pontok szervezeti vagy globális szabályozási szinten tisztázhatók, egyértelműsíthetők.

Ajánlások

Az értekezésemben összefoglalt kutatási eredmények felhasználhatók minden olyan Létfontosságú Információs Rendszerelem védelmi intézkedéseinek tervezéséhez ahol nyílt forrású rendszereket használnak, használni fognak vagy azokkal valamilyen módon – például beszállítókon keresztül vagy SLA egyezmény keretében – kapcsolatba kerülnek. Véleményem szerint ez hosszabb távon elkerülhetetlen, ezért az itt bemutatott eredmények információs és képzési szinten hasznos segítséget nyújthatnak a szervezet biztonságért felelős szakemberei számára, valamint segíthetnek kiegészíteni a szervezet információ-biztonsági szabályzatait.

Nist Security Framework, a ISO 27000 sorozat és a magyar jogrend (41/2015. (VII. 15.) BM rendelet) egyaránt előírja a rendszeres biztonsági tudatossági képzést. Ezt a képzést csak szakképzett személy végezheti. Az értekezés számos szabványt és ajánlást használ fel, így közvetlen oktatási célú felhasználása nem javasolható, ugyanakkor ajánlom eredményeimet a szakképzést nyújtó szervezetek és szakemberek figyelmébe forrásanyagként, egyes könnyebben emészthető részeit pedig átdolgozott formában oktatási segédlet alapjaként.

Az elkövetkező kutatások célja lehet az egyes FLOSS specifikus problémák súlyának meghatározása és támadási valószínűségekhez rendelése. A javasolt védelmi intézkedések jelen formájukban csak ad-hoc módon alkalmazhatók, hiszen azok szükségességét a szabályozást megelőző kockázatbecslés során kelle-

ne megállapítani, amit támadási valószínűségek és súlyossági értékek hiányában nemigen lehet precízen kivitelezni. Természetesen a kockázatkezelés nehezen általánosítható, így egy olyan módszertanra lenne leginkább szükség, amellyel ezek az értékek az azonosított problémák esetében becsülhetők. A lehetséges kockázatok és valószínűségek ismeretében a kutatásban bemutatott rendszer alapját képezhetné egy nyílt forrásra szabott kockázatértékelő szakértői rendszernek, amelyre a nyílt forrás terjedésével párhuzamosan egyre nagyobb igény mutatkozik.

A témakörben készült publikációk

Lektorált, magyar nyelvű szakmai folyóiratcikkek

- Mészáros Gergely: Nyílt fejlesztői közösségek hatása az informatikai biztonságra, Hadmérnök XV. évfolyam: 3. szám (Megjelenés alatt) (2020)
- Mészáros, Gergely: Nyílt forráskódú rendszerek biztonsági kérdései, BOLYAI SZEMLE XXII : 1 pp. 63-76. , 14 p. (2013)
- Mészáros Gergely: Elosztott verziókezelés a közigazgatásban, Hadmérnök IX : 3 pp. 191-206. , 16 p. (2014)
- Mészáros Gergely: Információs rendszerek fenyegetéseinek képesség alapú osztályozása, Társadalom és Honvédelem XVII : 3-4. pp. 215-227. , 13 p. (2013)
- Mészáros, Gergely: Szun-Ce elvei a digitális világban, Hadmérnök VIII. : 2 pp. 377-388. , 12 p. (2013)
- Mészáros, Gergely ; Hufnagel, Levente: Filtering outliers in OMTK Data Set, ANNUAL NEWS OF THE SZENT ISTVÁN UNIVERSITY YBL MIKLÓS FACULTACY OF BUILDING SCIENCES 7 : 1 pp. 42-46. , 5 p. (2007)
- Mészáros, Gergely: Magyarországi szántóföldi tartamkísérletek adatainak mesterséges intelligencia alapú elemzési lehetőségei Tudományos Közlemények Szent István Egyetem Ybl Miklós Műszaki Főiskolai Kar 1 pp.32-34 (2004)

Lektorált, angol nyelvű szakmai folyóiratcikkek

- Mészáros, Gergely: Auditing Community Software Development, YBL Journal of Built Environment 3 : 1-2 pp. 26-33. , 8 p. (2015)
- Mészáros, Gergely: Yield Prediction Based on Neural Networks, Annual News of The Szent István Unniversity YBL Miklós Faculty of Building Sciences 3: pp. 90-95., 6 p. (2005)
- Gergely: Human-Computer Interaction through Hand Gestures, Annual News of the Szent István University Ybl Miklós Faculty of Building Sciences pp.38-41., 4 p. (2003)

Idegen nyelvű konferencia kiadványban megjelent cikkek

- Mészáros, Gergely: Lessons of Transparent Collaboration: Comparison of E-Government and Software Developer Communities, In: Balthasar, Alexander; Golob, Blaž; Hansen, Hendrik; Müller-Török, Robert; Nemeslaki, András; Pichler, Johannes; Prosser, Alexander (szerk.) Central and Eastern European e|Dem and e|Gov Days 2016 : Multi-Level (e)Governance : is ICT a means to enhance transparency and democracy? Vienna, Ausztria : Austrian Computer Society, (2016) pp. 383-392. , 9 p.
- Mészáros, Gergely: Security impacts of community based software development pp. 325-336. In: Alexander, Balthasar; Blaž, Golob; Hendrik, Hansen; Balázs, Kőnig; Robert, Müller-Török; Alexander, Prosser (szerk.) CEE e|Dem and e|Gov Days 2015 : Time for a European Internet? Wien, Ausztria : Austrian Computer Society, (2015) p. 629
- Mészáros, Gergely: Pattern Classification via Neural Networks, In: Bergmeister, K (szerk.) Proceedings of the 3rd International PhD Symposium in Civil Engineering : Vol.2 Wien, Ausztria : Fleck Druck Gmbh, (2000) pp. 475-478. , 4 p.

Lektorált, magyar nyelvű előadás

- Mészáros, Gergely: Katasztrófavédelem és nyílt forrás, In: Kiss, Dávid; Orbók, Ákos (szerk.) A haza szolgálatában 2014 konferencia rezümékötet, Budapest, Magyarország : Nemzeti Közszolgálati Egyetem, (2014) pp. 66-68. , 3 p.
- Mészáros, Gergely: Nyílt forráskód létjogosultsága a kormányzati rendszerekben, In: Keresztes, Gábor (szerk.) Tavaszi Szél, 2013 : Spring wind, 2013. 1-2. kötet, Budapest, Magyarország : Doktoranduszok Országos Szövetsége, (2013) pp. 46-54. , 9 p.
- Mészáros, Gergely: Kritikus infrastruktúrákban felhasznált nyílt forráskódú rendszerek auditálási kérdései, In: Szakál, Béla (szerk.) Intézeti tudományos konferencia, Budapest, Magyarország : Avernim, (2012) pp. 91-97. , 7 p.
- Mészáros Gergely: GIS rendszertervezés nyílt forráskódú alapokon, In: Márkus, Béla (szerk.) GISopen 2011 : Megfelelni az új kihívásoknak Székesfehérvár, Magyarország : Nyugat-magyarországi Egyetem Geoinformatikai Kar, (2011) pp. 55-64. , 10 p.

Lektorált, idegen nyelvű előadás

- Mészáros, Gergely: IoT security and education, In: Talata, István (szerk.) Matematikát, Fizikát és Informatikát Oktatók 41. Országos Konferenciája : MAFIOK 2017, Budapest, Magyarország : Szent István Egyetem Ybl Miklós Építéstudományi Kar, (2017) pp. 1-7., 7 p.

A doktorjelölt szakmai-tudományos életrajza

Név:	Mészáros Gergely Tibor
Szül. hely, idő:	Budapest, 1975 május 24.
Állampolgárság:	Magyar
Anyanyelv:	Magyar
Nyelvismeret:	Angol (B2), Német (B1)

Tanulmányok

2012-2015 PhD képzés, Nemzeti Közszolgálati Egyetem Katonai Műszaki Doktori Iskola, információs infrastruktúrák támadása és védelme

1993-1999 Okleveles Földmérő és Térinformatikus mérnök, Budapesti Műszaki Egyetem, Építőmérnöki kar;

1989-1993 Babits Mihály Gimnázium, matematika-fizika szak

1998 B kat. jogosítvány

Munkahelyek

2020- Óbudai Egyetem, Ybl Miklós kar, Építőmérnöki Intézet

2014-2020 SZIE YMK Építéstudományi intézet, Informatika szakcsoport

2007-2008 Soter Line, rendszerinformatikus képzés, oktató

2001-2014 SZIE-YMK Ábrázoló és Számítástechnika Tanszék, tanársegéd

1999-2000 BMKE Fotogrammetria és Térinformatikai Tanszék

2000- Geotronic Bt., ügyvezető (informatikai szaktanácsadás, ingatlanrendezés)

Oktatási tevékenység

- Műszaki Informatika I/II (SZIE/YMK, 2001-2020)
- Számítógépes épületábrázolás (SZIE/YMK, 2001-2020)
- Rendszerinformatikus képzés (Szamalk, 2002)
- Hálózatok (Soter Line 2004-2005)
- Számítástechnika (Ybl, 2000)
- Informatikai Alapismeretek (BME, 2000-2001)
- Alkalmazott Informatika (BME, 2000-2001)

Elismerések

1999 TDK konferencia I díj, Pro Progressio különdíj (BME)

2011 SZIE Babérkoszorú ezüst fokozat

Szakmai tapasztalat

- Szoftverfejlesztés:
  - neurális hálózatok, deep learning rendszerek;
  - adatbázis programozás;
  - szoftvertervezési elvek ismerete,
  - funkcionális programozás, algoritmusok, programtervezési minták ismerete;
  - webtervezés (JS, XHTML, CSS, PHP, CGI programozás);
  - python, Bash, JS/ECMAScript, SQL nyelvek magas szintű ismerete;
  - haskell, C, C++, x86 assembly, ruby programozási nyelvek ismerete;
- Rendszerfejlesztés, üzemeltetés:
  - magas szintű UNIX/Linux rendszergazdai ismeretek;
  - nyílt forrású technológiák felhasználása;
  - több éves gyakorlat spamszűrő/víruskereső rendszerek üzemeltetése, támogatása terén;
  - Linux rendszerek adminisztrációja vállalati környezetben;
  - TCP/IP hálózati ismeretek, útvonalválasztók, tűzfalak beállítása;
  - nyílt forrású konténer és virtualizációs technológiák gyakorlati alkalmazása;
- Számítógéppel segített tervezés, CAD rendszerek;
- Informatikai, földmérési és ingatlanrendezési feladatok, GIS rendszerek;
- Szakfordítói gyakorlat (volt Linux Journal magazin);
- alapszintű hobbielektronikai és áramköri ismeretek, Arduino programozás;