SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

**Ing. Juraj Varga**

**Autoreferát dizertačnej práce**

# Odhaľovanie malvéru v OS Android

na získanie vedecko-akademickej hodnosti
*philosophiae doctor, PhD.*

| | |
|---|---|
| **v doktorandskom študijnom programe:** | Aplikovaná informatika |
| **v študijnom odbore:** | 9.2.9 aplikovaná informatika |

**Bratislava 2018**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

---

**Ing. Juraj Varga**

**Autoreferát dizertačnej práce**

# Odhaľovanie malvéru v OS Android

na získanie vedecko-akademickej hodnosti
*philosophiae doctor, PhD.*

| | |
|---|---|
| **v doktorandskom študijnom programe:** | Aplikovaná informatika |
| **v študijnom odbore:** | 9.2.9 aplikovaná informatika |

**Bratislava 2018**

**Dizertačná práca bola vypracovaná:** v dennej forme doktorandského štúdia na Ústave informatiky a matematiky FEI STU v Bratislave.

**Predkladateľ:** Ing. Juraj Varga
FEI STU v Bratislave
Ilkovičova 3, 812 19 Bratislava

**Školiteľ:** prof. Ing. Pavol Zajac, PhD.
FEI STU v Bratislave
Ilkovičova 3, 812 19 Bratislava

**Autoreferát bol rozoslaný dňa:** .....

**Obhajoba dizertačnej práce sa koná:** ..... o ..... hod.

**Na:** Fakulte elektrotechniky a informatiky STU,
Ilkovičova 3, 812 19 Bratislava
v .....

prof. Dr. Ing. Miloš Oravec
dekan FEI STU v Bratislave

# Obsah

# Úvod

Operačný systém Android je najrýchlejšie sa rozvíjajúci OS pre mobilné platformy. Pod pojmom mobilné platformy môžeme v súčasnosti chápať malé zariadenia s vysokým výpočtovým výkonom - inteligentné telefóny či tablety.

S počtom dostupných zariadení ruka v ruke rastie aj dopad a počet rôznych bezpečnostných hrozieb. Podobne ako sa kedysi pri rozmachu klasických počítačov pravidelne vyskytovali rôzne nové vírusy a chyby, tak podobný trend môžeme sledovať v súčasnosti na mobilných zariadeniach. Bezpečnostné hrozby sa plynule presúvajú na mobilné zariadenia hlavne z dôvodu veľkého množstva osobných a citlivých informácií (prihlasovacie mená a heslá, čísla účtov a platobných kariet), ktoré sa na nich nachádzajú, nízkeho povedomia používateľov o týchto hrozbách, či nedostatočnej kontrole aplikácií na úložiskách (marketoch). Napriek snahe odborníkov o ich elimináciu, počet bezpečnostných chýb a hrozieb neklesá, ale minimálne sa drží na rovnakej úrovni.

# 1 Ciele dizertačnej práce

Teoretická časť predkladanej práce obsahuje zhrnutie aktuálnych poznatkov z troch kľúčových oblastí:

- Operačný systém Android a implementované bezpečnostné mechanizmy (kapitola 1 dizertačnej práce).
- Systém povolení v OS Android a možnosti ich zneužitia (časti kapitoly 1 a 2 dizertačnej práce).
- Malvér v OS Android a možnosti jeho detekcie (taktiež v kapitole 2 dizertačnej práce).

Na základe zistených nedostatkov súčasného stavu problematiky sme si stanovili nasledovné ciele dizertačnej práce:

- Vytvorenie modelu povolení založeného na rolách.
- Zhodnotenie možnosti efektívneho využitia upraveného modelu povolení na detekciu malvéru.
- Zavedenie nových bezpečnostných mechanizmov súvisiacich s vykonanými zmenami v modeli povolení.

# 2 Teória a metódy

## 2.1 Systém povolení

Bezpečnosť operačného systému Android je postavená na upravenom linuxovom jadre [1]. Jednotlivé linuxové komponenty a bezpečnostné mechanizmy boli upravené tak, aby vyhovovali potrebám mobilného OS a zariadeniam s limitovanými hardvérovými možnosťami. Patria sem napríklad povinný sandboxing všetkých aplikácií alebo limitácia prístupu k systémovým zdrojom pomocou povolení schvaľovaných používateľom pre každú aplikáciu zvlášť [2]. Aj týmito prostriedkami sa autori OS snažia dosiahnuť ochranu používateľských dát pred krádežou alebo iným únikom a ochranu systémových zdrojov pred vyčerpaním, resp. znemožnením používania.

Aplikácie bežia v aplikačnom sandboxe a majú prístup k obmedzeným systémovým zdrojom. Systém spravuje prístup aplikácií k zdrojom a pokiaľ je použitý nesprávne alebo zlomyseľne, môže závažne ovplyvniť celkové fungovanie zariadenia alebo ohroziť dáta. Tieto obmedzenia sú implementované rôznymi spôsobmi. Niektoré možnosti sú obmedzené nedostatkom príslušných API pre konkrétnu funkcionalitu, ďalšie napr. separáciou rolí. Citlivé API sú používané len dôveryhodnými aplikáciami a chránené systémom povolení. Povolenia sú rozdelené do štyroch skupín podľa úrovne ochrany:

- Normal - povolenia aplikačnej úrovne, nepredstavujúce vážne riziko pokiaľ ich aplikácia využíva.
- Dangerous - nebezpečné povolenia, ktoré môžu spôsobiť únik a manipuláciu s citlivými dátami alebo využívať potenciálne nebezpečné zdroje zariadenia. Musia byť explicitne potvrdené používateľom pri inštalácii aplikácie. Patria sem napr.: lokačné dáta z GPS (ACCESS_FINE_LOCATION) alebo funkcie SMS/MMS (WRITE_SMS).
- Signature - povolenia, ktoré je možné prideliť len aplikáciám podpísaným privátnym kľúčom zodpovedajúcim certifikátu ako má aplikácia, ktorá ich volá. Sú využívané vývojármi na zdieľanie informácií medzi ich aplikáciami.
- Signature-or-system - zvláštny typ povolenia, ktoré je možné prideliť len aplikáciám inštalovaným v systémovom obraze alebo aplikáciám alebo sú podpísané rovnakým certifikátom ako systémový obraz.

Tieto zdroje sú prístupné len z OS. Aplikácie musia mať v manifeste povoleniami špecifikované aké majú požiadavky na tieto zdroje. Do vydania verzie 6.0

pri inštalácii aplikácie sa tieto povolenia zobrazili a používateľ ich mohol prijať alebo odmietnuť. Po prijatí sa pokračovalo v inštalácii a systém tieto povolenia akceptoval. Nebolo možné vyberať, ktoré povolenia chcel používateľ povoliť, museli byť povolené ako celok, čo mohlo viesť k bezpečnostným incidentom. Povolenia boli aplikácii pridelené počas celej doby čo bola nainštalovaná v zariadení a neboli dodatočné pýtané od používateľa. Odstránené boli v momente odinštalovania aplikácie. Dali sa pozrieť v nastaveniach aplikácií a mohli byť obmedzené vypnutím globálnej funkcionality, napr. vypnutím wi-fi alebo GPS. V prípade, že sa aplikácia pokúšala dostať k zdrojom, na ktoré nemala oprávnenie, tak vyvolala bezpečnostnú výnimku a chybové hlásenie v aplikácii. Kontroly povolení pre chránené API sú vykonávané na čo najnižšej úrovni, aby sa zabránilo ich obchádzaniu. Niektoré možnosti zariadenia nie sú dostupné pre aplikácie tretích strán, ale môžu byť používané predinštalovanými aplikáciami [3], [4].

Vo verzii 6.0 boli do Androidu zabudované tzv. "runtime" povolenia [5]. Pokiaľ chce aplikácia pristúpiť ku konkrétnemu systémovému zdroju, musí o to požiadať používateľa v danom momente. Ten sa môže rozhodnúť prístup povoliť alebo zamietnuť, pokiaľ sa mu zdá, že požiadavka nie je oprávnená. Vzhľadom na fakt, že väčšina zariadení stéle beží na starších verziách OS Android, je tento model plne funkčný len pre aplikácie vyvýjané priamo na verziu 6.0 (API level 23) a vyššie [6]. Staršie aplikácie sú z tohto modelu vynechané, fungujú normálne a aj ich inštalácia je rovnaká, ako pri predošlom modeli. Pokiaľ sa však používateľ rozhodne blokovať im prístup k nejakým zdrojom, aplikácia spadne, pokiaľ nebola aktualizovaná pre API level 23 a vyššie.

Najväčším problémom systému povolení v Androide je, že tvorcovia aplikácií dávajú svojim aplikáciám aj množstvo povolení, ktoré v skutočnosti na svoj chod vôbec nepotrebujú. Väčšina aplikácií ako napr. rôzne kalkulačky, kancelárske aplikácie a utility na správu zariadenia nepotrebujú prístup na Internet, ale aplikácia toto povolenia vyžaduje, lebo sa pomocou neho sťahujú reklamy, z ktorých autori primárne získavajú peniaze na ďalší vývoj. Väčšina používateľov tieto reklamy toleruje, ale použité knižnice môžu predstavovať bezpečnostné riziko [7]. Pokiaľ má aplikácia pridelených viac povolení ako reálne potrebuje môže byť zneužitá na útok tzv. privilege escalation.

## 2.2 Mobilný malvér

Systém povolení je z dôvodu svojej funkcie často zneužívaný pri útokoch mobilným malvérom na vykonávanie rôznej podvratnej činnosti. Podobne ako pri desktopovom malvéri, aj mobilný prešiel viacerými štádiami vývoja. Podrobný

prehľad a charakteristiku existujúceho malvéru publikovali v roku 2012 Zhou a Jiang [8], ktorých databázu vzoriek sme neskôr využili v našej práci. Dynamický vývoj malvéru je najväčším problémom pri jeho detekcii - väčšinou je totiž založená na analýze príbuzných vzoriek malvéru, čo je len do istej miery efektívny spôsob. Pokiaľ sa však objaví nový druh malvéru, tak trvá pomerne dlho, kým sa odchytia a zanalyzujú nejaké vzorky. Aktuálne sa využíva niekoľko spôsobov detekcie malvéru, napr. značkovanie dátových tokov [9], statická analýza [10], dynamická analýza (behaviorálna) [11] alebo ich kombinácie.

Vzhľadom na veľmi rýchly vývoj v oblasti bezpečnosti OS Android bolo nutné priebežne meniť ciele dizertačnej práce. Ako sa postupne vyvíjala situácia bezpečnosti OS Android, rozhodli sme sa rozvinúť tézu detekcie malvéru resp. potenciálne nebezpečných aplikácií na základe nie modelu rolí, ale statickej analýzy požadovaných povolení a analýzy zdrojového kódu aplikácie.

Aktuálny výskum ukazuje, že žiadna metóda na detekciu škodlivých aplikácií nie je stopercentne úspešná. Pri navrhovaní čo najefektívnejšieho spôsobu detekcie sme sa inšpirovali predošlým výskumom vykonanom medzi študentami [12]. Rozhodli sme sa zapojiť do procesu detekcie malvéru aj používateľov, keďže detekčné systémy často vyhodnotia škodlivé aplikácie ako bezpečné a naopak. Za cieľ sme si stanovili prepojiť detekčný systém s klientskou aplikáciou, kde si budú môcť používatelia svoje nainštalované aplikácie skontrolovať a pridať aj vlastné hodnotenie, ktoré poslúži nám na zlepšenie fungovania a presnosti detekcie celého systému. Riešenie sme pomenovali Distribuovaná detekcia malvéru so sociálnymi aspektami.

V dizertačnej práci predstavené riešenie je postavené na klient-server architektúre. Klientská aplikácia slúži na posielanie požiadaviek na kontrolu aplikácií na server a následné zobrazenie výsledku analýzy používateľovi. Serverová časť obsahuje analytickú logiku. Tá je postavená modulárne, aktuálne obsahuje štyri analytické moduly. Tieto moduly hodnotia aplikácie separátne. Čiastkové výsledky sa následne spracujú v tzv. agregátore, keďže každý modul má inú váhu hodnotenia. Výstup z agregátora sa následne odošle pomocou zabezpečenej webovej služby do klientskej aplikácie a zobrazí sa používateľovi. Ten môže následne využiť ďalšiu funkcionalitu aplikácie, a to zadať vlastné hodnotenie pre konkrétnu aplikáciu. Používateľské hodnotenia slúžia pre nás, aby sme vedeli korigovať váhy jednotlivých modulov a dosiahli tak vyššiu presnosť analýzy.

# 3 Dosiahnuté výsledky dizertačnej práce

Pri tvorbe nášho systému sme sa zamerali ako na presnosť riešenia, tak aj na používateľský komfort. Okrem toho sme nadviazali na existujúce publikácie, napr. [13], [14] či [15] a preskúmali distribúciu povolení v dostupných vzorkách aplikácií. Preskúmali sme aj používateľské povedomie o bezpečnosti mobilných zariadení formou možnosti zadania vlastného hodnotenia k zvolenej aplikácii:

- **Rýchlosť analýzy:** čo sa týka rýchlosti vyhodnotenia rizika zvolenej aplikácie, používateľ je limitovaný rýchlosťou svojho pripojenia k Internetu, ale len v prípade, že musí posielať inštalačný *.apk* súbor na server. Pri testovaní sme dosahovali priemerné rýchlosti analýzy od 5 do 17 sekúnd - v závislosti od veľkosti a zložitosti aplikácie. Výhodou veľkých úložísk aplikácií ako napr. Google Play je, že analýzu veľkého počtu aplikácií vedia rozložiť medzi veľké množstvo svojich používateľov.

- **Presnosť analýzy:** podrobným testovaním sme našli nastavenie váh čiastkových hodnotiacich algoritmov, ktorého výstupy majú najnižšie chyby prvého a druhého druhu. Napriek postaveniu nášho riešenia na jednoduchom princípe sme dosiahli veľmi kvalitné výsledky na dostupnej testovacej vzorke. Viac ako 60% vzoriek malvéru sa nášmu systému podarilo korektne označiť ako malvér, ak zoberieme do úvahy aj všetky vzorky so skóre z intervalu $< 0.4; 0.6)$, tak toto číslo narastie na 75.66%. Tieto výsledky by sme vedeli zlepšiť zmenou váh jednotlivých analytických modulov, čo by však malo za následok zhoršenie úspešnosti správneho posúdenia legitímnych aplikácií. Ďalšou možnosťou by bolo napr. pridanie modulu na dynamickú analýzu aplikácií, kde by mohli byť sledované ďalšie charakteristiky aplikácií ako napr. sieťová komunikácia. Pri legitímnych aplikáciách naše riešenie dosahuje takmer 90%-nú úspešnosť správneho posúdenia legitímnych aplikácií. Pri niekoľkých aplikáciách sa vyskytli odchýlky, tieto však boli spôsobené veľkým počtom vyžadovaných povolení, čo však pri aktuálnom nastavení algoritmu nevieme lepšie odfiltrovať

- **Distribúcia povolení:** rozhodli sme sa nadviazať na vyššie spomenuté články a vykonať podobne zamerané testy. Výskyt povolení sme zmapovali na všetkých dostupných vzorkách a do veľkej miery korešpondujú s výsledkami publikovanými vo vyššie spomenutých článkoch. Takisto podporujú správnosť našej metodiky testovania nainštalovaných aplikácií a následnej detekcie (aj potenciálneho) nebezpečenstva. Jednoduchým porovnaním výskytu jednotlivých povolení a ich skupín vieme vo veľkej miere určiť, či sa jedná

o podozrivú aplikáciu alebo nie. Navyše je tento spôsob jednoduchý na pochopenie aj pre len minimálne technicky zdatného používateľa.

- **Používateľské hodnotenia:** analýzou výsledkov z testovania presnosti analýzy aplikácií sme zistili, že aj pri najlepšom možnom nastavení hodnotiaceho algoritmu sa objavia nebezpečné aplikácie označené ako bezpečné a naopak. Na základe týchto zistení sme sa rozhodli dať používateľom možnosť zadania vlastného hodnotenia na testované aplikácie. Používatelia tak prispievajú svojím hodnotení ku korekciám v hodnotiacom algoritme, a tým k zvýšenej presnosti analýzy. Vo viacerých prípadoch používatelia odhalili, že overená aplikácia bola považovaná za nebezpečnú a jej hodnotenie bolo korigované. Problémom však je, že nie všetci používatelia so systémom spolupracujú, naopak, niektorí zadávajú falošné hodnotenia. Pri dostatočne veľkej vzorke používateľov by však takéto správanie malo byť štatisticky odhaliteľné.

V práci sa nám podarilo zlepšiť v minulosti publikované výsledky z predošlých [16], resp. čiastkových [17] verzií tohoto projektu. Výsledné riešenie je modulárne, takže je možné ho ľubovoľne rozširovať o ďalšie moduly na analýzu aplikácií. Klientská aplikácia má jednoduchý dizajn, je používateľsky priateľská, takže práca s ňou je jednoduchá, rýchla a intuitívna. Analytické moduly sa dajú dopĺňať a postupne tak umožniť vybudovať väčší znalostný systém na detekciu škodlivých aplikácií. Plný potenciál podobného riešenia by však bolo možné uplatniť len v kontexte veľkého množstva aplikácií a používateľov, napr. integráciou s platformou Google Play.

# 4 Literatúra


[1] BRAY, T.: What Android Is, Dostupné na internete: `http://www.tbray.org/ongoing/When/201x/2010/11/14/What-Android-Is`, 2010

[2] Android Security Overview, Dostupné na internete: `http://source.android.com/tech/security/index.html`

[3] SHABTAI, A. et. al.: Google Android: A Comprehensive Security Assessment. *Security & Privacy, IEEE, Volume: 8 Issue: 2*, 2010, s. 35 - 44.

[4] ENCK, W.: Defending Users Against Smartphone Apps: Techniques and Future Directions, *ICISS'11 Proceedings of the 7th international conference on Information Systems Security*, 2011, s. 49 - 70

[5] Android: Security Enhancements. Dostupné na internete: `https://source.android.com/security/enhancements/index.html`, 2016

[6] Android Developers: Requesting Permissions at Run Time, Dostupné na internete: `http://developer.android.com/training/permissions/requesting.html`, 2016

[7] GRACE, M.C. et. al.: Unsafe exposure analysis of mobile in-app advertisements, *WISEC '12 Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, 2012, s. 101 - 112

[8] ZHOU, Y. - JIANG, X.: Dissecting Android Malware: Characterization and Evolution, *Proceedings of the 33rd IEEE Symposium on Security and Privacy (Oakland 2012)*, 2012, 15 s.

[9] ENCK, W. et. al.: TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, *OSDI'10*, 2010, 15 s.

[10] SHABTAI, A. - FLEDEL, Y. - ELOVICI, Y.: Automated Static Code Analysis for Classifying Android Applications Using Machine Learning, *Computational Intelligence and Security (CIS), 2010 International Conference on*, 2010, s. 329 - 333

[11] TAM, K. et. al.: CopperDroid: Automatic Reconstruction of Android Malware Behaviors, *22nd Annual Network and Distributed System Security Symposium, NDSS 2015 San Diego, California, USA, February 8-11, 2015*, 2015, 15 s.

[12] VARGA, J. - ZAJAC, P.: Mobile Security Experience of IT Students, *ELOSYS. Elektrotechnika, informatika a telekomunikácie 2012*, 2012, s. 161 - 164

[13] FELT, A.P. et. al.: Android Permissions Demystified, *CCS'11*, 2011, 11 s.

[14] FELT, A.P. et. al.: Android Permissions: User Attention, Comprehension, and Behavior, *Symposium on Usable Privacy and Security (SOUPS) 2012*, 2012, 14 s.

[15] WEI, X. et. al.: Permission Evolution in the Android Ecosystem, *ACSAC '12*, Dec. 3-7, 2012, Orlando, Florida USA, 2012, 10 s.

[16] VARGA, J. et. al.: Mitigating Possible Threats from Overprivileged Android Applications, *IN-TECH 2015: Proceedings of the International conference on innovative technologies, Dubrovnik, Croatia, 09.-11.09.2015*, 2015, s. 42 - 45

[17] VARGA, J. - MUSKA, P.: Presenting risks introduced by Android application permissions in a user-friendly way, *Tatra Mt. Math. Publ. 60 (2014)*, 2014 s. 85 - 100

# 5 Zoznam prác dizertanta

**Vedecké práce v zahraničných časopisoch registrovaných v databázach Web of Science alebo SCOPUS**

HROMADA, Viliam - VARGA, Juraj. Phase-shift fault analysis of Trivium. In Studia Scientiarum Mathematicarum Hungarica. Vol. 52, No. 2 (2015), s. 205-220. ISSN 0081-6906. V databáze: WOS: 000357757000005.

**Vedecké práce v domácich časopisoch registrovaných v databázach Web of Science alebo SCOPUS**

VARGA, Juraj - MUŠKA, Peter. Presenting risks introduced by android application permissions in a user-friendly way. In Tatra Mountains Mathematical Publications. Vol. 60, (2014), Issue: 1, s. 85-100. ISSN 1210-3195. V databáze: SCOPUS.

VARGA, Juraj - ŠVANDA, Dominik - VARCHOLA, Marek - ZAJAC, Pavol. Authentication based on gestures with smartphone in hand. In Journal of Electrical Engineering. Vol. 68, No. 4 (2017), s. 256-266. ISSN 1335-3632. V databáze: WOS: 000410953500002.

**Vedecké práce v ostatných zahraničných časopisoch**

VARGA, Juraj - KOSTRECOVÁ, Eva. OS Android - Architecture and Security. In Sdělovací technika. Roč. 61, č. 10 (2013), s.54-55. ISSN 0036-9942.

VARGA, Juraj - KOSTRECOVÁ, Eva. Android Applications - Design and Security. In Sdělovací technika. Roč. 61, č. 9 (2013), s.54-55. ISSN 0036-9942.

VARGA, Juraj - KOSTRECOVÁ, Eva. Android - Mobile Telecommunication Platform. In Sdělovací technika. Roč. 61, č. 7 (2013), s.46-47. ISSN 0036-9942.

VARGA, Juraj - KOSTRECOVÁ, Eva. Google Play another Way of Securing OS Android. In Sdělovací technika. Roč. 61, č. 11 (2013), s.50-51. ISSN 0036-9942.

**Publikované príspevky na zahraničných vedeckých konferenciách**

ANTAL, Eugen - VARGA, Juraj. Zodiac. In Mikulášská kryptobesídka 2010 : Sborník příspěvků. Praha, 2.-3.12.2010. Praha : Trusted Network Solutions, 2010, s.89-90. ISBN 978-80-904257-1-2.

VARGA, Juraj - GULÁŠOVÁ, Michala - OREM, Martin - DOBROČKA, Pavol - NOVOTNÝ, Daniel - BOLEDOVIČ, Andrej. Mitigating possible threats from overprivileged android applications. In IN-TECH 2015 : Proceedings of the International conference on innovative technologies. Dubrovnik, Croatia. 09.-11.09.2015. Rijeka : Engineering University of Rijeka, 2015, S. 42-45. ISSN 1849-0662.

VARGA, Juraj. Overview of android security mechanisms. In Norwegian-slovakian workshop in crypto : Bergen, Norway. February 8 - 10, 2016. 1. ed. Bratislava : Slovak University of Technology, 2016, S. 59-64. ISBN 978-80-227-4541-3.

**Ostatné publikácie**

BALOGH, Štefan - VARGA, Juraj. Threats in Mobile Security. In ELITECH'12 [elektronický zdroj] : 14th Conference of Doctoral Students. Bratislava, Slovak Republic, 22 May 2012. Bratislava : Nakladateľstvo STU, 2012, s.CD-ROM, [6] s. ISBN 978-80-227-3705-0.

VARGA, Juraj - ZAJAC, Pavol. Mobile Security Experience of IT Students. In ELOSYS. Elektrotechnika, informatika a telekomunikácie 2012 [elektronický zdroj] : Trenčín, 9.-12.10.2012. Bratislava : FEI STU, 2012, s.CD-ROM, s. 161-164. ISSN 1335-2547.

VARGA, Juraj - HROMADA, Viliam. Extracting Randomness from Mobile Devices. In EE časopis pre elektrotechniku, elektroenergetiku, informačné a komunikačné technológie : konferencia ELOSYS, Trenčín, 15.-18.10.2013. Roč. 19, mimoriadne č (2013), s.20-22. ISSN 1335-2547.

ANTAL, Eugen - SÝS, Marek - VARGA, Juraj. Evaluation Functions in the Cryptoanalysis of Homophonic Substitution. In ISCAMI 2012 : Book of abstracts. Malenovice, Czech Republic, 10.-13.5.2012. Ostrava : University of Ostrava, 2012, s.12.

HROMADA, Viliam - VARGA, Juraj. Accelometers as Sources of Randomness in Mobile Device. In ISCAMI 2013 : Book of abstracts. Malenovice, Czech Republic, May 2-5, 2013. Ostrava : University of Ostrava, 2013, s.34.

HROMADA, Viliam - VARGA, Juraj. Phase-shift Fault Analysis of Trivium. In Central European Conference on Cryptology 2014 : conference pre-proceedings. Budapest, Hungary, May 21-23, 2014. Budapest : Alfréd Rényi Institute of Mathematics, 2014, s. 54-55.

HROMADA, Viliam - VARGA, Juraj. Entropy assessment of Android OS. In ISCAMI 2014 : book of abstracts. Malenovice, ČR, 27. - 30. 3. 2014. 1. vyd. Ostrava : Ostravská univerzita, 2014, s. 29.

**Rôzne publikácie v spoluatorstve so študentmi**

GAZDÍK, Martin - VARGA, Juraj. Bezpečnosť mobilnej platformy Android. In ŠVOČ 2013 [elektronický zdroj] : Zborník vybraných prác, Bratislava, 23. apríl 2013. 1. vyd. Bratislava : FEI STU, 2013, s.CD ROM, s. 10-12. ISBN 978-80-227-3909-2.

BOLEDOVIČ, Andrej - VARGA, Juraj. Practical implementation of McEliece cryptosystem on android. In CECC 2016 : The 16th central european conference on cryptology. Piestany, Slovakia. June 22 - 24, 2016. Bratislava : STU, 2016, S. 15-18.

VARGA, Juraj - GAZDÍK, Martin. Exploiting Missing Permission in Android. In 43. konference EurOpen.CZ : Vranov, Czech Republik; 29. 9.-2.10.2013. Plzeň : EurOpen.CZ, 2013, s.9-20. ISBN 978-80-86583-26-6.

## 5.1 Ohlasy a citácie (bez autocitácií)

Abdella, Ozuysal a Tomur v článku *CA-ARBAC: privacy preserving using context-aware role-based access control on Android permission system* citujú článok *Presenting risks introduced by android application permissions in a user-friendly way.*

Bhatt, Gupta a Mittal v článku *iABC: Towards a hybrid framework for anayzing and classifying behaviour of iOS applications using static and dynamic analysis*

citujú článok *Presenting risks introduced by android application permissions in a user-friendly way.*

Kumar, Shanker a Verma v článku *Context Aware Dynamic Permission Model: A Retrospect of Privacy and Security in Android System* citujú článok *Presenting risks introduced by android application permissions in a user-friendly way.*

Shah a Reznik v článku *Permission-based Security Evaluation System for Android Applications* citujú článok *Presenting risks introduced by android application permissions in a user-friendly way.*

Pavol Zajac v článku *Using Local Reduction fot the Experimental Evaluation of the Cipher Security* cituje článok *Phase-shift fault analysis of Trivium.*

Hromada a Petho v článku *Phase-shift Analysis of Grain v1* citujú článok *Phase-shift fault analysis of Trivium.*

## 5.2 Aktívna prezentácia výsledkov a prednášky

Miesto: ISCAMI 2012 (International Student Conference on Applied Mathematics and Informatics), Malenovice, CZ
Rok: 2012
Príspevok: Evaluation functions in the cryptoanalysis of homophonic substitution

Miesto: Elitech 2012, FEI STU, Bratislava
Rok: 2012
Príspevok: Threats in Mobile Security

Miesto: ELOSYS 2012, Trenčín
Rok: 2012
Príspevok: Mobile Security Experience of IT Students

Miesto: ELOSYS 2013, Trenčín
Rok: 2013
Príspevok: Extracting Randomness from Mobile Devices

Miesto: ISCAMI 2013 (International Student Conference on Applied Mathematics

and Informatics), Malenovice, CZ
Rok: 2013
Príspevok: Accelometers as Sources of Randomness in Mobile Device

Miesto: EurOpen, Vranov nad Dyjí, CZ
Rok: 2013
Príspevok: Exploiting Missing Permission in Android

Miesto: ISCAMI 2014 (International Student Conference on Applied Mathematics and Informatics), Malenovice, CZ
Rok: 2014
Príspevok: Entropy assessment of Android OS

Miesto: 16th FRUCT (Finnish-Russian University Cooperation in Telecommunications) Conference Oulu, Fínsko
Rok: 2014
Príspevok: Presenting Risks Introduced by Android Application Permissions in a User-friendly Way

Miesto: Prednáška pre študentov z USA (v rámci projektu NATO SPS 984520), FEI STU, Bratislava
Rok: 2014
Príspevok: Introduction to Android Security

Miesto: Pracovné stretnutie na Ruhr Universität, Bochum, Nemecko
Rok: 2015
Príspevok: User-friendly system for detection of over privileged Android apps

Miesto: IN-TECH 2015 International conference on innovative technologies. Dubrovnik, Chorvátsko
Rok: 2015
Príspevok: Mitigating possible threats from overprivileged android applications.

Miesto: Prednáška pre študentov z Nórska (v rámci projektu EEA Grant SK06-IV-01-001), FEI STU, Bratislava
Rok: 2015
Príspevok: Android Security - Basic Overview

Miesto: Norwegian-Slovakian Workshop in Crypto, Bergen, Nórsko
Rok: 2016
Príspevok: Overview of android security mechanisms.

# Summary

At the beginning, this dissertation summarizes the state of the art of Android security and relevant security mechanisms implemented in this OS. This introductory part is divided in two chapters. In the first one we inspect the Android as a whole. We describe all the security mechanisms implemented since it's introduction. In the second chapter we investigate the most significant directions in Android security, namely problems of physical access to the device and, with greater focus, mobile malware targeted on this platform.

During our research we found out, that Android permission model [2] is the core component, which manages access to system resources (e.g. Internet connection, telephony) and user data (e.g. contact list). The original permission model did not allow users to manage permissions as they wanted [3], [4]. It wasn't until version 6.0, that introduced so called runtime permissions [5]. This new model allows users to manage resources that applications want to access. Still, this model is flawed, which allows mobile malware to operate on vulnerable devices. Currently, there are many ways how to detect mobile malware, for example marking data flows [9], static analysis [10], dynamic analysis (behavioral) [11] or their combinations.

Therefore, in chapter 3, we introduce our solution for mobile malware detection. We also decided to allow users to participate in evaluating of applications intalled on their devices. They use client application to request analysis. This request is sent via secure web service to server. Server part handles the analysis. Analysis is based on modular solution, currently consisting of four different modules. Each module produces an outcome, which has a certain weight in final verdict. These partial results are further processed in aggregator, which calculates the final result. This result is sent back to the client application and is shown to the user. Following that, the users are allowed to input their own personal evaluation. We can further use these inputs to adjust weights of certain modules to produce a more precise results. We call our solution Distributed malware detection with social aspects.

To test our solution (Chapter 4) we used a malware database collected by Zhou a Jiang [8] and almost 900 benign applications. We primary tested precision of our solution and user experience, secondary, we followed existing works, e.g. [13], [14] or [15] and investigated permission distribution in available samples. The conclusions are:

- **Analysis time:** currently, the users are limited only by the speed of their Internet connection, but only in the case they need to send the *.apk* file from their device to the server. Otherwise, we achieved the performance between 5 to 17 seconds to analyze a specific application, depending on it's size and

complexity.

- **Analysis accuracy:** we thoroughly tested our solution on all available samples. Despite the simplicity of our solution we achieved good results on tested samples. More than 60% of malware samples were correctly marked as malware. If we take into account samples from range $< 0.4; 0.6)$ this percentage rises to 75.66%. We could change weights of analytic modules to achieve even better results, but that would make scores of benign applications worse. Now we achieve almost 90% success rate in correctly marking these application as benign.

- **Distribution of permissions:** we decided to follow up on aforementioned works and conduct similar tests. We managed to test all available samples and the results closely correspond with the ones achieved in these papers. Therefore they support correctness of our approach and testing methodology. Moreover, this approach is easy to understand even for technically not skilled users.

- **User evaluation:** even with the best possible module weights settings, there are still benign applications labeled as malware and vice versa. Because of this situation, we decided to give the users an opportunity to input their own evaluation of a specific application. This way they contribute to corrections in module weight settings and improve our results. Unfortunately, not all of the users cooperate and input false evaluations. But these fake values could be filtered out by using sufficient number of user database.

In this work we managed to improve results published in the past [16] or partial [17] versions of this project. The final design is modular, therefore it can be arbitrarily expanded by other analysis modules (e.g. dynamic analysis). Client application is user-friendly and easy to understand. Analytic modules can be added and combined, thus gradually build a knowledge base system for malware detection. However, full potential of this design could be only achieved in context of vast number of samples and users, for example by integrating it within Google Play.