Bezpečnostní chyby na mobilní platformě, jejich zneužívání a návrh proaktivního opatření s využitím umělé inteligence

Ing. Milan Oulehla, Ph.D.

Teze disertační práce

Univerzita Tomáše Bati ve Zlíně

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Teze disertační práce

Bezpečnostní chyby na mobilní platformě, jejich zneužívání a návrh proaktivního opatření s využitím umělé inteligence

Security Issues on Mobile Platform, Their Exploiting and Proactive Measure Using Artificial Intelligence

Autor: **Ing. Milan Oulehla, Ph.D.**

Studijní program: Inženýrská informatika P3902
Studijní obor: Inženýrská informatika 3902V023

Školitel: doc. Ing. Zuzana Komínková Oplatková, Ph.D.

Oponenti: pplk. doc. Ing. Petr Hrůza, Ph.D.
prof. Ing. Petr Dostál, CSc.
prof. Ing. Jiří Dvořák, DrSc.

Zlín, únor 2020

Vydala **Univerzita Tomáše Bati ve Zlíně** v edici **Disertační práce**.
Publikace byla vydána v roce 2020

Klíčová slova: *mobilní malware, bezpečnostní chyby mobilní platformy, problematika zabezpečení mobilní platformy, mechanismus detekce mobilního malware, neuronové sítě, strojové učení*

Key words: *mobile malware, mobile security issues, security of mobile platform, mobile malware detection mechanism, neural networks, machine learning*

Práce je dostupná v Knihovně UTB ve Zlíně.

ISBN 978-80-7454-911-3

ABSTRAKT

Dizertační práce se zabývá třemi hlavními oblastmi výzkumu: bezpečností současných mobilních aplikací, mobilním malwarem a detekcí mobilního malwaru pomocí umělé inteligence, především neuronových sítí. Práce popisuje mechanizmy, jejichž prostřednictvím útočníci a tvůrci mobilního malwaru získávají APK balíčky legitimních aplikací, provádějí jejich analýzu a zneužívají nalezené bezpečnostní chyby. Práce je unikátní nejen svým rozsahem a systematickým zpracováním, ale především hloubkou předkládaných poznatků. Publikované informace nemají pouze teoretický charakter, ale obsahují i jedinečné ukázky zdrojových kódů (ve vyšších i nižších jazycích), schémata a snímky obrazovek mobilních zařízení zachycující klíčové situace.

První část dizertační práce pokrývá všechny hlavní oblasti problematiky bezpečnosti mobilních aplikací od rozdílů, jakými jsou zneužívány zranitelnosti nalezené v mobilních aplikacích útočníky a tvůrci mobilního malwaru, přes problematiku APK balíčků a jejich analýzy, až po nalezené zranitelnosti ve vyšetřovaných mobilních aplikacích. Zkoumání zranitelností ve vyšetřovaných mobilních aplikacích vedlo k odhalení celé řady závažných bezpečnostních hrozeb, které byly systemizovány do čtyř kategorií: útoky založené na analýze dat z APK balíčků, APK repackage, útoky na lokální zabezpečení mobilních aplikací a útoky na síťové zabezpečení mobilních aplikací.

V oblasti mobilního malwaru je dizertační práce zaměřena na analýzu mobilního malwaru a charakteristiky mobilního malwaru. Analytická část popisuje získávání vzorků mobilního malwaru a jejich vyšetřovací metody, ve kterých práce přináší nové, dosud nezveřejněné postupy. Unikátní poznatky jsou rovněž publikovány v části zabývající se charakteristikami mobilního malwaru.

Práce se neomezuje pouze na výzkum útočných technik, ale snaží se přispět ke zlepšení bezpečnostní situace proaktivním opatřením, kterým je návrh a experimentální ověření nového způsobu detekce mobilního malwaru pomocí umělé inteligence, především pomocí neuronových sítí. Zde se jako klíčová ukázala datová analýza a tvorba vstupních vektorů pro neuronové sítě, zejména navržený způsob identifikace a redukce problematických složek vektorů. Na kvalitu výzkumu měla pozitivní vliv spolupráce se společností AVG Technologies CZ, nad jejíž datovou sadou probíhaly detekční experimenty. Dosažená přesnost detekce 99,5 % při trénování a 98,23 % při testování při rozsáhlosti a kvalitě datové sady lze označit za vysoce úspěšné a relevantní. Dosažené detekční výsledky ukazují sílu strojového učení a zároveň naznačují jeden z perspektivních směrů, kterými by se měla ubírat problematika detekce mobilního malwaru.

ABSTRACT

This dissertation deals with three main areas of research: security of current mobile applications, mobile malware and detection of mobile malware using artificial intelligence, especially neural networks. This dissertation describes mechanisms by which attackers and mobile malware creators obtain APK packages of legitimate applications, analyse them and exploit found vulnerabilities. The work is unique not only in its scope and systematic processing but mainly in the depth of presented findings. The published information is not only of a theoretical character, but it also contains unique source code samples (in both high-level and low-level programming languages), diagrams as well as screenshots capturing crucial situations.

The first part of the dissertation covers all major areas of mobile application security issues, from different ways how vulnerabilities found in mobile applications are exploited by attackers and mobile malware creators, through the issue of APK packages and their analysis, to vulnerabilities found in investigated mobile applications. Examination of vulnerabilities in investigated mobile applications has revealed a number of serious security threats which have been systematized into four categories: attacks based on analysis of data from APK packages, APK repackage, attacks on local security of mobile applications and attacks on network security of mobile applications.

In the field of mobile malware, the dissertation is focused on mobile malware analysis and mobile malware characteristics. The analytical part describes the acquisition of mobile malware samples and their investigation methods in which the work brings new, unpublished procedures. Unique findings are also published in the part dealing with characteristics of mobile malware.

The dissertation is not only limited to the research of attack techniques but it also tries to contribute to the improvement of the security situation by a proactive measure which is the design and experimental verification of a new way of mobile malware detection using artificial intelligence, especially neural networks. Data analysis and creation of input vectors for neural networks proved to be the key here, especially the suggested method of identification and reduction of problematic vector components. Cooperation with AVG Technologies CZ, whose data set was used for detection experiments, had a positive effect on the quality of the research. The achieved detection accuracy of 99.5% during training and 98.23% during testing can be regarded as highly successful and relevant considering the size and quality of the dataset. The achieved detection results show the power of machine learning and at the same time indicate one of the promising directions which should be taken in the mobile malware detection.

OBSAH

1. ÚVOD

1.1 Úvod do problematiky

Mobilní zařízení, jako jsou chytré mobilní telefony, tablety a v poslední době i nositelný hardware (chytré hodinky a sportovní náramky), se staly běžnou součástí moderní společnosti. Tato skutečnost je dokumentována zprávou Ericsson Mobility Report 2016 [1], podle které v roce 2015 dosáhl počet chytrých mobilních telefonů připojených ke globální telekomunikační síti (smartphone subscriptions) 3,2 miliardy. Přičemž světová populace ve stejném roce byla na základě statistických dat odhadována na 7,3 miliardy [2]. To znamená, že téměř každý druhý obyvatel planety měl v roce 2015 chytrý mobilní telefon, a to včetně kojenců. Do statistiky nebyly započítány ostatní varianty mobilního hardware, jako jsou tablety a nositelný hardware. V následujícím roce došlo, podle Ericsson Mobility Report 2017 [3] k navýšení počtu připojených chytrých mobilních telefonů (smartphone subscriptions) na 4,41 miliardy. Predikce, jež je uvedena v téže zprávě říká, že v roce 2023 počet chytrých mobilních telefonů připojených ke globální telekomunikační síti dosáhne 7,27 miliardy. Dalším významným ukazatelem provázanosti občanské společnosti a mobilních technologií je podíl přístupů na webové stránky. Zatímco v roce 2009, byl podíl mobilních přístupů na webové servery pouhých 0,7 %, v roce 2017 již činil 50,3 % všech dotazů a poprvé tak překonal počty přístupů z osobních počítačů [4].

Mobilní hardware a software tvoří odvětví IT průmyslu, který má roční obrat v řádech miliard dolarů. Mobilní zařízení se stala běžnými komunikačními prostředky jak v osobním, tak v pracovním životě. Z tohoto důvodu často obsahují citlivá osobní data (PII - Personally Identifiable Information, SPI - Sensitive Personal Information), kontakty, e-mailovou korespondenci a korporátní know-how. Navzdory výše uvedeným skutečnostem je bezpečnost mobilní platformy na velmi nízké úrovni. Uvedená situace je způsobena celou řadou faktorů, které jsou podrobně popsány v disertační práci v kapitole 2, jež systematicky mapuje současný stav problematiky bezpečnosti mobilní platformy. Kapitola 2 se také zabývá omezeními, která mají mobilní zařízení oproti osobním počítačům. Například chytré telefony a tablety jsou napájeny bateriemi s omezenou kapacitou. Baterie navíc musí napájet poměrně velké displeje. Kombinace těchto dvou faktorů velmi znesnadňuje běh rezidentních antivirových štítů na pozadí. Skutečně výkonný štít by neúměrně spotřebovával elektrickou energii baterie, což by vedlo k tomu, že by bylo nutné mobilní zařízení nabíjet i několikrát za den. Druhá kapitola se rovněž věnuje analýze současných bezpečnostních standardů a legislativy ve světě a v České republice. Zkoumá, zda jsou zohledňována i bezpečnostní specifika mobilní platformy. Výsledky analýzy odhalily, že, pokud bezpečnostní pravidla a standardy neexistují, nebo dostatečně nezohledňují specifika mobilní platformy, či jsou ignorována v jakékoliv fázi vývojového cyklu, vznikají mobilní aplikace, které obsahují závažné bezpečnostní chyby. Disertační práce se zabývá problematikou bezpečnosti na mobilní platformě, a to

jak z pohledu útoku, tak i z pohledu obrany. Struktura práce je rozvržena následujícím způsobem. Cíle dizertační práce jsou podrobně popsány v kapitole 3. Problematice bezpečnosti mobilních aplikací se věnuje kapitola 4. Bezpečnost mobilních aplikací a mobilní malware (škodlivý software, název vznikl z anglických slov malicious a software) jsou témata, která spolu úzce souvisí. Aplikace, které obsahují bezpečnostní chyby, jsou velmi často zneužívány mobilním malwarem. Pro komplexní vysvětlení problematiky je potřeba nejen popsat chyby v mobilních aplikacích, ale také vysvětlit mechanismy, jakými jsou dané zranitelnosti zneužívány mobilním malwarem (krádeže uživatelských dat, útoky na nezabezpečené poskytovatele obsahu apod.). Výzkum mobilního malwaru je popsán v kapitole 5. Dizertační práce se neomezuje pouze na zkoumání bezpečnostních chyb v mobilních aplikacích a popisu jejich zneužívání prostřednictvím malwaru, ale navrhuje i proaktivní opatření. Kapitola 6 popisuje techniky detekce mobilního malwaru pomocí metod strojového učení (machine learning), umělých neuronových sítí a snaží se tak přispět ke zlepšení bezpečnostní situace na mobilní platformě.

1.2 Použité názvosloví

Bezpečnostní problematika mobilních platforem je mladý obor. Z toho vyplývá naprostá absence výzkumných prací publikovaných v českém jazyce. Veškerá relevantní literatura je téměř výhradně v anglickém jazyce. To znamená, že názvosloví používané mezinárodní i českou odbornou veřejností je anglické. Práce si neklade za cíl vytváření nového názvosloví, ale používá zažité anglické názvosloví, které má ustálenou sémantiku.

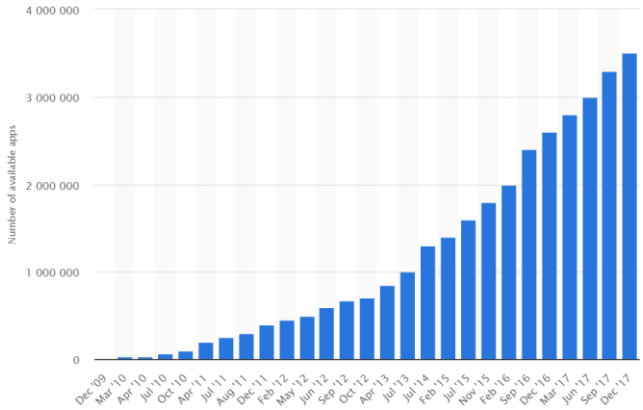
2. SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Rychlost a specifické rysy výzkumu zaměřeného na bezpečnost mobilní platformy se v určitých jeho oblastech projevují nedostatkem kvalitní, systematicky zpracované odborné literatury. Nicméně na kvalitu i kvantitu odborné literatury, a tedy i na stav poznání působí další faktory, které jsou pro problematiku bezpečnosti mobilních aplikací a mobilního malwaru rozdílné. Z tohoto důvodu jsou v disertační práci rozebrány samostatně a velmi podrobně.

2.1 Aktuální stav v oblasti bezpečnosti mobilních aplikací

Vývoj mobilních aplikací je velmi dynamický. Prakticky každá větší společnost či organizace má svoji mobilní aplikaci, jejichž počet od roku 2009 neustále stoupá. Na obrázku (Obr. 2.1) je vidět, že v prosinci 2017 dosáhl počet oficiálních mobilních aplikací pro operační systém (OS) Android 3,5 milionu. Ačkoli mobilní aplikace pronikly do každodenního života většiny obyvatel vyspělého světa, neexistuje jednotný, všeobecně přijímaný standard určený pro bezpečnostní testování mobilních aplikací. V oblasti mobilní platformy zatím nebyl zpracován

komplexní risk management, což znamená, že rizika nejsou systematicky identifikována a hodnocena. Neexistence jednotného standardu velmi znesnadňuje tvorbu odpovídajících penetračních testů. Penetrační test představuje simulovaný útok na testovanou technologii, jehož cílem je včas odhalit možné bezpečnostní chyby. Výsledkem penetračního testu bývá zpráva (penetration report či zkráceně pentest report), která je předávána zadavateli a obsahuje všechny odhalené bezpečnostní problémy. V současné době vznikají penetrační testy na jednotlivých testovacích/forenzních pracovištích spontánně a jsou vzájemně nekompatibilní. Z výše uvedených skutečností vyplývá, že proprietární penetrační testy systematicky nepokrývají všechny možné hrozby. Jinými slovy uživatelé, instituce, ozbrojené složky, soukromé společnosti a jejich data jsou vystaveni prostřednictvím nedostatečně otestovaných aplikací potenciálním závažným bezpečnostním rizikům.



Obr. 2.1: Počet aplikací dostupných na Google Play od prosince 2009 do prosince 2017 [5]

V oblasti bezpečnosti mobilních aplikací na vznik odborné literatury negativně působí dva faktory, které spolu úzce souvisí. Prvním z nich je rychlost a dynamika vývoje operačních systémů, a především pak API (Application Programming Interface). Jedná se o funkcionalitu zapouzdřenou do metod, tříd a protokolů, která je poskytována operačním systémem nebo knihovnami (často systémovými). Uvedenou funkcionalitu používají programátoři při tvorbě mobilních aplikací. Druhým faktorem negativně ovlivňující vznik odborné literatury je časová omezenost zjištěných bezpečnostních poznatků. Například, operační systém Android vyšel od roku 2008 do roku 2017 v patnácti hlavních verzích. Za stejné období vzniklo 27 verzí API [6]. Společnosti zabývající se vývojem mobilních operačních systémů se snaží získat konkurenční výhodu tím, že uvolňují nové verze svých systémů v rychlém sledu za sebou. Což znamená, že upřednostňují nově přidanou funkcionalitu a přívětivost (vývojářskou i uživatelskou) před bezpečností. Často se pak stává, že nové funkce obsahují závažné bezpečnostní chyby, z nichž některé jsou odstraněny prostřednictvím aktualizací dané verze operačního systému a jiné jsou opraveny až v následující verzi operačního systému. Tento výzkum probíhá převážně v komerční sféře,

především prostřednictvím soukromých penetračních laboratoří. Výsledky jejich bezpečnostních výzkumů jsou zahrnovány do penetračních testů, které představují know-how dané laboratoře a nejsou proto veřejně publikovány. Z tohoto důvodu se často pro zkoumání dopadů závažných chyb v mobilních operačních systémech na bezpečnost mobilních aplikací používají jiné informační zdroje než standardní (tištěná) odborná literatura. Jedním z těchto zdrojů jsou takzvané Vulnerability Databases (databáze zranitelností). Vulnerability Databases jsou on-line databáze přístupné přes webové rozhraní, některé poskytují informace o bezpečnostních chybách zdarma, jiné jsou placené.

Dalším důležitým zdrojem informací jsou služby zaměřené pouze na mobilní platformu. Těmito zdroji jsou Android Security Bulletins (https://source.android.com/security/bulletin), vydávány společností Google LLC, a About the security content of iOS, publikovány společností Apple Inc. a vztahující se vždy k určité verzi iOS.

Kromě zranitelností v operačním systému a v systémových komponentách ovlivňují bezpečnost mobilních aplikací negativně i chyby, které vznikají během procesu jejich vývoje. Těchto chyb se dopouštějí analytici při návrhu a programátoři během vývoje mobilních aplikací. Co se týká odborné literatury, jsou bezpečnostní chyby, které vznikají při samotném vývoji mobilních aplikací, mnohem lépe dokumentované, než je tomu u chyb v operačních systémech, které následně negativně působí na bezpečnost mobilních aplikací. Jedná se především o knižní publikace [7] - [9].

Dalším cenným informačním zdrojem – kromě výše uvedených knih a Vulnerability Databases – jsou články, které vznikají v rámci akademického výzkumu. Od roku 2010 do roku 2018 dochází k růstu počtu článků, které se zabývají jak bezpečností mobilních aplikací běžících na operačním systému Android, tak na operačním systému iOS. V roce 2019 vyšlo o 1606,25 % více článků souvisejících s Android než v roce 2010. U článků související s iOS činil nárůst 217,24 %.

Rozbor nejdůležitějších knih a odborných článků, které na toto téma vyšly, je uveden v dizertační práci. Výsledky těchto publikací naznačují potřebu bezpečnostního výzkumu v oblasti mobilních aplikací. Z publikovaných dat rovněž vyplývá, že by se měl výzkum zejména zaměřit na bezpečnost mobilních aplikací běžících pod operačním systémem Android. Uvedená problematika je v dizertační práci popsána v kapitole 4 - Bezpečnost mobilních aplikací.

2.2 Aktuální stav v oblasti malwaru

Problematika malwaru vždy představovala velmi specifickou oblast kybernetické bezpečnosti. Jedním z jejich typických rysů je nedostatek podrobných, systematicky zpracovaných informací. Malware za svoji téměř

padesátiletou historii[1] prošel celou řadou vývojových fází – od ranných neškodných fází přes vojenský malware až po vydírající s kriminálním podtextem.

V současné době má téměř každá moderní armáda jednotku, která se zabývá elektronickým válčením (electronic warfare nebo také electronic cyber-warfare), jehož součástí je i vývoj vojenského malware. Vytvořený malware se používá jako zbraň, která dokáže být v mnoha případech výhodnější než konvenční vojenské řešení. Použitím vojenského malwaru se zabrání ztrátám na lidských životech. Je levnější než tradiční vojenské tažení, přesto může způsobit značné škody. Pro jednotlivé bezpečnostní analytiky, organizace, ale i antivirové společnosti je velmi těžké, ne-li nemožné některé hrozby včas detekovat nebo jim dokonce předcházet. Přestože zatím není oficiálně potvrzen žádný výskyt vojenského malwaru na mobilní platformě, bylo by chybou se domnívat, že takový škodlivý software neexistuje, nebo že není v současné době připravován.

Další významnou oblastí je malware, který je spojen s organizovaným zločinem. Tento malware souvisí s trendem posledních let, kterým je přesun kriminality z reálného světa do kybernetického prostoru. První zdokumentovaný případ ransomware (malware požadující výkupné, název vznikl spojením slov ransom - výkupné a software) pochází z roku 1989. Jmenoval se AIDS a jednalo se o trojského koně[2], jehož škodlivá činnost měla podobu ransomware. AIDS fungoval tak, že poté, co byl počítač infikován, umožnil uživateli ještě devadesát běžných spuštění. Následně malware skryl všechny adresáře a šifroval nebo zamknul jména všech souborů. Pak ransomware AIDS požadoval, aby uživatel napadeného počítače zaplatil společnosti PC Cyborg Corporation sídlící v Panamě sto osmdesát devět dolarů [32].

Nejnovější fázi vývoje malwaru představuje mobilní malware, který souvisí s masovým rozšířením chytrých telefonů a tabletů v posledních deseti letech. Pro tvůrce mobilního malwaru je důležité psát škodlivý software, který bude mít co nejvíce potenciálních obětí. Největší množství mobilního malwaru existuje pro operační systém Android, neboť má od prvního čtvrtletí roku 2011 dominantní pozici na trhu mobilních operačních systémů (Obr. 2.2).

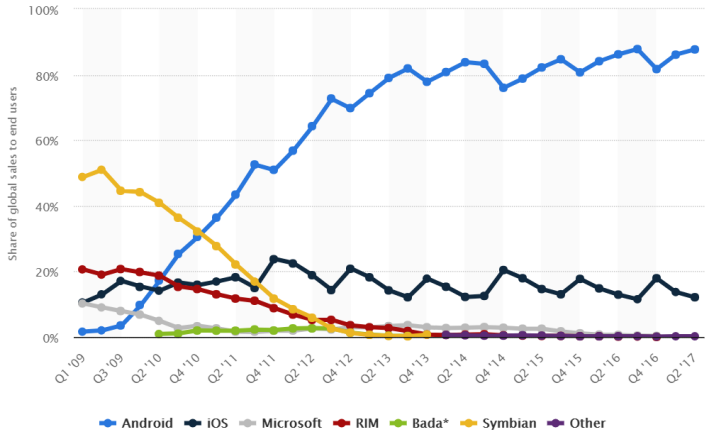
Vývoj malwaru je na mobilní platformě rychlejší než na platformě osobních počítačů. Je to dáno tím, že jednotlivé typy malwaru a jejich principy není potřeba pro mobilní platformu znovu vymýšlet. Stačí je pouze upravit tak, aby respektovala specifika mobilních zařízení [10]. Dnes všechny typy malwaru známé z osobních počítačů již existují i na mobilní platformě. Navíc přibyly zcela nové techniky, které jsou specifické pouze pro mobilní platformu. Například

---

[1] *V otázce prvního malwaru není odborná literatura jednotná. Nejčastěji se uvádí, že předzvěstí malwaru byl Creeper (někdy taky nazývaný Creeper Worm), který vznikl pravděpodobně již v roce 1970. Creeper byl experimentální software, který se uměl sám replikovat (94).*

[2] *Trojský kůň je typ škodlivého softwaru, který předstírá, že dělá něco užitečného a často i něco prospěšného či zábavného skutečně dělá. Může se například jednat o hru či o komprimovací program, který ale obsahuje i skrytou část, jež provádí škodlivou činnost.*

řízení botnetu prostřednictvím SMS zpráv [11]. V oblasti mobilního malwaru představují největší nebezpečí právě zmíněné botnety[3] [12], [13], [14].



Obr. 2.2: Podíl mobilních operačních systémů v letech 2009 až 2017 [10]

Z toho důvodu se dizertační práce ve zvýšené míře zabývá výzkumem mobilních botnetů. Jejich zkoumání odhalilo celou řadu unikátních, dosud nepublikovaných útočných mechanismů a z nich plynoucích zranitelností, které byly použity pro návrh mechanismu detekce mobilního malwaru pomocí neuronových sítí.

Ze všech výše uvedených skutečností vyplývá, že rozsáhlý systematicky vedený výzkum, který se zabývá mobilním malwarem probíhá na komerční (antivirové společnosti), kriminální (zločinecké organizace zabývající se vývojem malware), nebo vojenské bázi. Ani jedna skupina nemá zájem zjištěné poznatky zpřístupnit široké odborné veřejnosti.

Přestože články publikované na téma mobilní malware přináší celou řadu zajímavých a užitečných poznatků, mají i jistá omezení. Autoři pro výzkumné účely často vytváří nástroje, které ale na rozdíl od publikovaných výsledků nejsou veřejně dostupné. Tato skutečnost výrazně komplikuje snahy o replikování výsledků a získání hlubšího vhledu do řešeného problému. Dalším omezením je úzké zaměření článků. Velmi zřídka se vyskytují série na sebe navazujících článků, které by svědčily o systematickém zkoumání autorů v dané oblasti.

2.3 Bezpečnostní specifika současné mobilní platformy

Nově vznikající bezpečností mechanismy nemohou být jednoduše převzaty z PC platformy, neboť musí být vytvořeny s ohledem na vlastnosti, ve kterých se mobilních zařízení liší od osobních počítačů:

[3] *Botnet je síť tvořena infikovanými mobilními zařízeními (tzv. bóti nebo také zombie), která lze vzdáleně ovládat, například pomocí C&C (Command-and-Control) serveru. Prostřednictvím botnetu je možné provádět nejrůznější útoky, například na webové servery. Člověk, který ovládá botnet se nazývá botmaster.*

- chytré telefony a tablety jsou neustále zapnuté (typicky 24/7). To znamená, že jsou vystaveny vlivům malwaru mnohem delší časový rámec, než je běžné u osobních počítačů.
- dochází k častému přepínání mezi celulárními sítěmi (celulární síť je rádiová telekomunikační síť, která zprostředkovává jak hovory a SMS, tak mobilní data) a Wi-Fi sítěmi. Jedno zařízení tak vystupuje jako člen několika naprosto rozdílných sítí. Uvedené chování na jednu stranu způsobuje z pohledu malwaru komplikovanější identifikaci infikovaných zařízení (identifikace je nutná například pro zasílání příkazů a vzdálené ovládání napadených zařízení). Na druhou stranu může jedno napadené mobilní zařízení infikovat celou řadu dalších zařízení v různých sítích,
- mobilní zařízení jsou napájena bateriemi. Uvedená vlastnost velmi znesnadňuje běh rezidentních štítů na pozadí, neboť by neúměrně spotřebovávaly elektrickou energii baterie,
- omezená velikost a odlišný, přísnější, mechanismus správy operační paměti. Paměť je spravována tak, aby byl telefonní subsystém k dispozici za všech okolností (restrikce se vztahují i na antivirové programy),
- Application Sandbox – všechny mobilní aplikace, tedy i antiviry, mají vykonávání svého kódu i svá data oddělena od ostatních aplikací a prostředků operačního systému. Sandboxing na jednu stranu zvyšuje celkovou bezpečnost systému, neboť útočné či infikované aplikace nemohou na neupravovaných zařízeních snadno krást data čistých – neinfikovaných aplikací nebo zneužívat prostředky operačního systému (například fotoaparát, internetové připojení apod.). Na druhou stranu uvedený systém velmi znesnadňuje antivirovým programům komplexní kontrolu mobilních aplikací v reálném čase přímo v mobilních zařízeních (neboť i ony jsou zapouzdřeny pomocí Sandboxingu).

Společnosti, které se zabývají vývojem mobilního hardwaru, preferují zisky před bezpečností. Hardware, ale i software, je vyvíjen v mnohem kratších vývojových cyklech, než je běžné u osobních počítačů. Například výrobci mobilních telefonů a tabletů svoje klíčové modely vydávají každý rok. Operační systém Android vychází průměrně každých osm měsíců. Rychlé tempo vývoje neumožňuje vytvoření robustního designu a jeho bezpečného spojení s operačním systémem a aplikacemi. Uvedený přístup je obvyklý u většiny výrobců, neboť bezpečnost nepřináší přímé zisky a naopak prodražuje vývoj.

Bezpečnostní situaci navíc zhoršují i samotní uživatelé, kteří provádějí zásahy do mobilních zařízení umožňující přístup k jinak zakázaným funkcím nebo zvyšují uživatelský komfort. Například:

- úprava mobilního zařízení, při které uživatel získá práva super uživatele (Root/Rootování Android OS, Jailbreak iOS),
- vypnutí/nepoužívání šifrování perzistentní paměti mobilního zařízení,
- instalace mobilních aplikací:

    - z neoficiálních (neznámých zdrojů),
    - instalace upravených aplikací (placené aplikace, ze kterých byla odstraněna softwarová ochrana),
    - instalace aplikací z oficiálních zdrojů, které požadují více oprávnění, než potřebují vzhledem ke své funkcionalitě,
- chybějící zámek obrazovky,
- používání výchozího pinu SIM karty (například: 1234),
- a podobně.

Výše uvedené informace naznačují nutnost výzkumu detekce mobilního malwaru, který bude respektovat zvláštnosti mobilní platformy. Výzkum provedený v rámci dizertační práce naznačuje, že velmi slibnou oblastí jsou detekční mechanismy založené na technikách umělé inteligence, jmenovitě na umělých neuronových sítích.

2.4 Analýza současných bezpečnostních standardů a legislativy

Obecně lze říci, že bezpečnostní standardy spontánně vznikají pro oblasti, které buď generují velké zisky nebo mají zásadní důležitost. V těchto oblastech jako první vyvstává potřeba vytvářet software, který bude splňovat vysoké nároky na bezpečnost. Z toho důvodu vytvářejí bezpečnostní odborníci přednostně soubory pravidel pro vývoj mobilních aplikací, které vytvářejí velké zisky nebo jsou významné. V České republice (ČR) bezpečnostní standardy zabývající se mobilní platformou zcela chybí. Dizertační práce se proto místo analýzou současných českých standardů zabývá v kapitole 2.4.2 podrobnou analýzou legislativního rámce v České republice, ze kterého budou získány relevantní informace pro problematiku bezpečného vývoje mobilních aplikací. Čeští vývojáři se snaží při vytváření svých aplikací držet zahraničních standardů. Ty však vychází z jiného legislativního rámce a zohledňují specifika států, na jejichž území dané standardy vznikaly. Z výše uvedeného plyne, že zahraniční standardy nejsou mnohdy pro české programátory příliš vhodné. Disertační práce se věnuje analýze standardů, jako je OWASP Mobile Security Project [16], Health Insurance Portability and Accountability Act (HIPAA) Secure [15], PCI Mobile Payment Acceptance Security Guidelines [17], Google Security for Android Developers [18] - [19], deset netechnických problémů, které nejvíce negativně ovlivňují bezpečnost mobilních aplikací [21], vydaných firmou Forrester [20]. Standardy jsou vždy úzce zaměřené (platební operace, zdravotnictví, netechnické problémy…) a jsou vzájemně nekompatibilní, kdy jeden standard nepředpokládá existenci druhého. Některé standardy jsou omezeně použitelné pro české mobilní vývojáře, protože vznikly v jiném legislativním prostředí, než má Český republika. Navíc většina z nich má formu doporučení, kterých by se programátoři měli držet, a nejsou povinné.

3. CÍLE DISERTAČNÍ PRÁCE

Dizertační práce na základě vlastního výzkumu mapuje nejen typické bezpečnostní chyby současných mobilních aplikací, ale popisuje i útočné mechanismy moderního mobilního malwaru, který tyto chyby využívá ve svůj prospěch. Škodlivý software se nezaměřuje jen na chyby v uživatelských aplikacích, ale také zneužívá chyby v operačních systémech a lidské chyby, proto dizertační práce pojednává i o těchto aspektech mobilního malwaru. V neposlední řadě se práce snaží přispět ke zlepšení bezpečnostní situace na mobilní platformě tím, že přináší proaktivní opatření, kterým je detekce mobilního malwaru pomocí metod umělé inteligence. Dizertační práce si klade následující tři cíle.

3.1 Popis závažných bezpečnostních chyb současných mobilních aplikací

Jedním z hlavních cílů dizertační práce je bezpečnostní analýza současných mobilních aplikací. Práce popisuje mechanizmy, pomocí kterých útočníci a tvůrci mobilního malware získávají APK balíčky legitimních aplikací, provádějí jejich analýzu a zneužívají nalezené bezpečnostní chyby. Tento cíl je zpracován v následující struktuře.

- Rozdíl mezi útočníky a tvůrci mobilního malwaru,
- APK balíčky:
    - Získávání APK balíčků,
    - Vytváření a kompilační procesy APK balíčků,
    - Struktura APK balíčků,
    - Dekompilace APK balíčků,
        - Typ 1: dex2jar, JD-GUI,
        - Typ 2: APKTool,
    - Využití informací získaných z dekompilovaných balíčků,
    - Ruční dynamická analýza,
    - Ruční statická analýza,
    - Automatizované metody vyšetřování,
- Zranitelnosti ve vyšetřovaných mobilních aplikacích:
    - Útoky založené na analýze dat z APK balíčků,
    - APK repackage,
    - Útoky na lokální zabezpečení mobilních aplikací,
    - Útoky na síťové zabezpečení mobilních aplikací.

3.2 Popis útočných mechanismů mobilního malwaru

Tato část dizertační práce je zaměřena na dvě hlavní oblasti: analýza mobilního malwaru a charakteristiky mobilního malwaru. První z nich popisuje proces získávání vzorků mobilního malwaru a metody jejich analýzy. Rovněž jsou představeny techniky restaurování kódů malwaru. V druhé části jsou

prezentovány charakteristiky mobilního malwaru, které byly zjištěny na základě výzkumu, například:

- Malware obsahující legitimizující část aplikace (trojští koně),
- Princip infekce legitimních aplikací (APK repackage)
- Malware typu Hidden APK,
- Malware typu Hidden APK s uživatelskou interakcí,
- Výzkum a tvorba experimentálních mobilních botnetů,
- Anti-Analysis techniky,
- Zneužívání komponenty WebView,
- Malware jako spouštěč chráněných částí komerčních aplikací.

3.3 Navržení mechanismu detekce mobilního malwaru pomocí umělé inteligence, především neuronových sítí

Charakteristiky mobilního malwaru jsou dále rozšířeny a využity pro navržení mechanismu detekce mobilního malwaru pomocí umělé inteligence a strojového učení, především umělých neuronových sítí, které do této oblasti spadají. Součástí práce je i vytvoření vhodných trénovacích a testovacích sad pro učení umělých neuronových sítí, navržení vhodného paradigmatu neuronové sítě, její naučení na připravených datech a rozsáhlé testování finálního klasifikačního modelu. Finální model je dále srovnán s dalšími technikami, jako jsou např. metoda podpůrných vektorů (Support Vector Machines), metoda k-nejbližších sousedů (k-nn) či naivní bayesovský klasifikátor.

- Princip detekce mobilního malwaru pomocí neuronových sítí,
- Algebra oprávnění,
- Automatizovaná analýza podezřelých vzorků mobilních aplikací:
  - Moduly statické analýzy,
  - Moduly dynamické analýzy.
- Výzkum v oblasti detekce mobilního malwaru pomocí neuronových sítí:
  - Datová analýza a tvorba vstupních vektorů pro neuronové sítě,
  - Konstrukce neuronové sítě,
  - Testování finálního klasifikačního modelu.
  - Srovnání s dalšími metodami umělé inteligence.

4. BEZPEČNOST MOBILNÍCH APLIKACÍ

V kapitole 4 jsou popsány chyby, které se objevily ve skutečných mobilních aplikacích. Aby nedocházelo k porušování autorských práv, byl použit pouze princip dané zranitelnosti, který byl zakomponován do příslušné demonstrační aplikace. Tyto aplikace byly vytvořeny pod hlavičkou PTLabu (Penetration Testing Laboratory, https://ptlab.fai.utb.cz), ve které autor práce působí. Bezpečnost mobilních aplikací ohrožují útočníci a tvůrci mobilního malwaru. Obě

skupiny pachatelů se snaží najít bezpečnostní chyby v legitimních aplikacích, ale liší se ve způsobu, jakými nalezené chyby zneužívají.

4.1 Rozdíl mezi útočníky a tvůrci mobilního malwaru

Útočníci se snaží v mobilních aplikacích hledat takové chyby v zabezpečení, které jim umožní jejich přímé zneužití. Může se jednat například o odcizení nezabezpečených komerčních dat, které se nacházejí v privátním datovém prostoru mobilní aplikace (/data/data/NameOfApp), nebo využívání funkcionality napadené aplikace, aniž by za ni zaplatili. Mezi další časté způsoby zneužívání patří hledání informací, které by mohli útočníkům posloužit k napadení vzdálených serverů, jejichž služby využívají analyzované mobilní aplikace.

Tvůrci útočného malwaru provádějí analytickou část útoku velmi podobně, často dokonce používají stejné nástroje a metody. Nicméně způsob, jakým zneužívají nalezené bezpečnostní chyby, je zcela odlišný:

- na základě nalezených chyb vytvoří útočný malware, který zneužívá nezabezpečené funkcionality legitimní aplikace, například distribuce spamu přes veřejně dostupné rozhraní e-mailového klienta,
- odstraní ochranu z placené aplikace a pak provedou její infekci, při které do napadené mobilní aplikace vloží škodlivou část kódu.

4.2 APK balíčky

APK balíček přestavuje instalátor mobilní aplikace. Mobilní aplikace běžící pod operačním systémem Android se píší v jazyce Java a nově i v jazyce Kotlin. Proces, kterým jsou ze zdrojových kódů vytvářeny APK balíčky, se nazývá kompilace. Navzdory skutečnosti, že je jazyk Java multiplatformní, je proces kompilace Java programů pro mobilní zařízení a osobní počítače velmi odlišný. Tato skutečnost způsobuje problémy bezpečnostním odborníkům, kteří pracovali s Javou na platformě osobních počítačů a mlčky předpokládají, že je komplikační proces stejný i u mobilních zařízení. Kompletní kompilační proces je zachycen v dizertační práci, kde jsou nejen vysvětleny rozdíly mezi mobilní platformou a osobními počítači, ale i vývoj kompilačního procesu APK balíčků (na tomto poli probíhá intenzivní vývoj). V současné době nejsou ani v primární dokumentaci k operačnímu systému Android, ani v odborné literatuře, uceleně a přehledně zpracovány jednotlivé vývojové fáze kompilačních procesů. Místo toho jsou uvedené postupy rozdrobeny do řady na sebe nenavazujících dokumentů. Z tohoto důvodu dizertační práce systematicky zpracovává uvedenou oblast, přičemž zvláštní pozornost je věnována schématickým znázorněním, na kterých nejlépe vyniknou vývojové rozdíly.

Disertační práce se podrobně v jednotlivých podkapitolách věnuje tématům, jako jsou vytváření, kompilace a dekompilace APK balíčků, včetně dex kompilace, dále podrobné struktuře APK balíčků, která je nutným předpokladem pochopení funkčnosti a problematiky bezpečnosti mobilních aplikací. V

dizertační práci je popsán typický obsah APK balíčku, jako například adresář res či soubory AndroidManifest.xml, classes.dex, resources.arsc a některé další. Soubor AndroidManifest.xml představuje výchozí místo, ve kterém začíná většina útoků, ale i penetračních testů mobilních aplikací běžících pod operačním systémem Android. Manifest, který se nachází v rozbaleném zip archívu, není lidsky čitelný. Aby bylo možné soubor AndroidManifest.xml přečíst, je potřeba provést dekompilaci APK balíčku.

Jakmile útočník/tvůrce mobilního malwaru rozumí kompilačnímu procesu a struktuře APK balíčků, může přikročit k praktické analýze, jejíž první fáze se zabývá získáváním APK balíčků legitimních aplikací. Pro vyhledávání potenciálně hodnotných APK balíčků útočníci a tvůrci mobilního malwaru využívají softwarovou distribuční platformu Google Play, která je dostupná prostřednictvím webového prohlížeče na adrese
https://play.google.com/store/apps.

V procesu výběru vhodné APK aplikace, se zohledňuje několik parametrů: hodnotný datový obsah, počet stažení a oprávnění, které požaduje legitimní aplikace. Nelegitimní požadavky infikované části programu lze nenápadně navázat na legitimní oprávnění hostitelské aplikace.

V dalším kroku je použit nástroj pro stažení APK balíčku (např. APK Downloader nebo přímo z mobilního zařízení, na kterém útočník převzal práva super uživatele (tzv. Root zařízení)).

Procesu rekonstrukce původních zdrojových kódů a zdrojů aplikace z instalačních balíčků se říká dekompilace. Jak bylo vysvětleno výše, existují dva typy dekompilace. První typ je prováděn pomocí nástrojů dex2jar a JD-GUI, pro dekompilaci druhého typu se používá APKTool.

První typ umožňuje získání původních zdrojových kódů v jazyce Java. Dekompilace prvního typu má dvě fáze. Během první fáze je APK balíček pomocí nástroje Dex2jar [22] transformován do souboru *.jar. V druhé fázi je možné otevřít převedený *.jar soubor pomocí Java Decompileru [23], který zajišťuje čtení zdrojových kódů, zkoumání aplikační logiky a hledání bezpečnostních zranitelností přímo v jazyce Java.

Druhý typ dekompilace se provádí pomocí nástroje APKTool a zprostředkovává přístup klíčovým komponentám vyšetřované aplikace, jako jsou AndroidManifest.xml (lidsky čitelná podoba souboru), resources.arsc či datové XML struktury. APKTool také poskytuje aplikační logiku převedenou do jazyka Smali, nižšího nedokumentovaného jazyka podobného bajtkódu, který umožňuje modifikaci aplikační logiky. Práce v jazyce Smali je mnohem náročnější než úpravy prováděné ve vyšších jazycích (například v jazyce Java), proto tvůrci mobilního malwaru i penetrační testeři využívají speciální techniky, jak s tímto jazykem pracovat. Uvedené postupy jsou vysvětleny na příslušných místech dizertační práce.

Během dekompilačního procesu útočníci, tvůrci mobilního malwaru, ale i penetrační testeři získávají informace a data, která mohou být rozdělena do dvou

skupin. Do první skupiny patří data a informace, které lze přímo zneužít. Jsou to například hesla a nezabezpečené multimediální soubory, které byly nalezeny během analýzy APK balíčků. Do druhé skupiny patří informace, které využívají neetičtí vývojáři. Soubory s aplikační logikou odcizené pomocí Java Decompileru slouží k rychlému získání požadované funkcionality bez nutnosti ji programovat. To znamená zlevněný vývoj[4] aplikace na úkor práce jiných programátorů, jejichž kód byl odcizen z dekompilovaných APK balíčků.

4.3 Analýza mobilních aplikací

Pro analýzu mobilních aplikací se používají tři základní metody: ruční dynamická analýza, ruční statická analýza a automatizované metody vyšetřování.

Ruční dynamická analýza je jednou ze základních metod výzkumu používaných, jak pro odhalení nefunkčních bezpečnostních mechanizmů (ohrožena je množina aplikací, které používají danou technologii), tak pro hledání zranitelností v určitých mobilních aplikacích (týká se pouze jedné konkrétní aplikace, např. chybná implementace jinak bezpečné technologie). Dynamická analýza je mnohem rychlejší než metody ruční statické analýzy a klade menší nároky na schopnosti a zkušenosti bezpečnostního analytika. Z tohoto důvodu je dynamická analýza hojně používána, nejen v penetračních a forenzních laboratořích, ale i v akademických laboratořích zaměřených na bezpečnostní výzkum. Ruční dynamickou analýzu provádějí nejen bezpečnostní analytici a penetrační testeři, ale také útočníci a tvůrci mobilního malware. Je nutné si uvědomit, že pro úspěšný útok málokdy stačí pouze vyšetření pomocí metod ruční dynamické analýzy. Dynamické testování je vhodné provádět v kombinaci se statickou analýzou. Bezpečnostní testy provedené pomocí dynamické analýzy mohou poskytnout důležitá vodítka a ukázat možný směr automatizovaného vyšetřování.

Dynamická analýza se zabývá zkoumáním chování aplikace a reakcemi systému na její podměty. Z toho plyne, že metody dynamické analýzy nezkoumají zdrojový kód vyšetřované aplikace. Aby bylo možné realizovat výzkum efektivně a získávat během testování maximum možných informací, je potřeba testy provádět buď ve speciálním softwarovém emulátoru (Android Emulator [24], Nox App Player [25], Genymotion emulator [26], emulátory vytvořené na zakázku) nebo v upraveném fyzickém mobilním zařízení. Neupravená/standardní mobilní zařízení umožňují získat pouze značně omezenou množinu informací.

Navzdory velkým pokrokům na poli dynamické analýzy je ruční statická analýza stále nejdůležitějším nástrojem pro výzkum nových zranitelností a jejich ověřování. Tento typ analýz je časově velmi náročný, klade zvýšené nároky na schopnosti a zkušenosti analytika, který musí být nejen skvělým penetrační testerem, bezpečnostním expertem, etickým hackerem, ale především mobilním

[4] *Nejedná se pouze o snížení nákladů ale také o odhalení technologického know-how konkurence.*

vývojářem schopným porozumět nativním jazykům mobilní platformy. Časová i analytická náročnost se ještě zvyšuje, je-li kód a zdroje aplikace chráněný obfuskací[5]. Nicméně pokud penetrační nebo forenzní laboratoř nechce být závislá na výsledcích výzkumů jiných subjektů, pak se tomuto typu výzkumu nevyhne.

Proces ruční statické analýzy zpravidla zahrnuje kroky:

- Dekompilace typ 1 umožňující získání původních zdrojových kódů.
- Dekompilace typ 2 umožňující získání zdrojů a transformačních kódů aplikace.
- Hledání vstupního místa (tzv. Entry Point) ve zdrojových kódech aplikace.
- Analýza zdrojů a zdrojových kódů aplikace.
- Nalezení třídy nebo metody, která je zodpovědná za zabezpečení mobilní aplikace.
- Nalezení chyb v zabezpečení exponované třídy nebo metody.
- Navržení útočných metod, které zneužijí nalezené zranitelnosti.

Kromě ruční dynamické a statické analýzy se pro výzkum zranitelností aplikací používají i automatizované metody, které jsou často zapouzdřeny do bezpečnostních/útočných frameworků. Prostředky automatizovaného testování kombinují dynamické i statické vyšetřovací metody. Během automatizovaného vyšetřování se využívají všechny informace nashromážděné v předchozích fázích testování. Pro efektivní automatizované vyšetřování je totiž nutné znát alespoň přibližný směr, kterým by se měl výzkum ubírat. Testování bez jakýchkoliv předchozích vědomostí o dané aplikaci je časově náročné a neefektivní.

Patrně nejpoužívanějším nástrojem automatizované analýzy je Drozer. Jedná se o komplexní bezpečnostní/útočný framework pro mobilní platformu obsahující celou řadu účinných nástrojů, pomocí kterých lze provádět efektivní automatizované testy/útoky. Například:

- prostředky pro zjišťování informací z vyšetřovaných APK balíčků, a to včetně manifestu,
- *app.package.attacksurface*, který slouží pro určování perspektivních směrů útoků,
- nástroje pro zjišťování bezpečnostních chyb poskytovatelů obsahu,
- apod.

Významnou skupinu nástrojů automatizovaného vyšetřování tvoří zakázkové syntaktické analyzátory. Jsou vyvíjeny přímo forenzními a penetračními laboratořemi, nebo si je nechávají laboratoře vytvářet na základě svých poznatků a specifikací u třetích stran. Jde o velmi sofistikované nástroje, které tvoří know-how dané laboratoře/společnosti, proto jsou téměř vždy neveřejné. Může se

[5] *Obfuskace je metoda, která převádí zdrojový kód mobilní aplikace do upraveného zdrojového kódu stejného jazyka. Z pohledu funkcionality jde o stejný kód, který se ale velmi špatně čte a staticky analyzuje.*

například jednat o syntaktické analyzátory, které se používají pro automatizované vyšetřování zdrojových kódů, ve kterých se hledají podezřelá místa. Nalezené části potenciálně nebezpečného kódu jsou sestaveny do automaticky generovaného interního reportu, který následně ručně hodnotí analytik. Pracovník laboratoře nemusí procházet tisíce řádků zdrojových kódů, ale zaměřuje se pouze na místa kódu, která mohou být potenciálně nebezpečná. Výhodou automatizovaných testů je jednak jejich rychlost, ale i množství provedených úkonů. Pro komplexní vyšetření mobilních aplikací a pro výzkum nových penetračních technik lze nejlepších výsledků dosáhnout kombinováním ruční statické analýzy, ruční dynamické analýzy a nástrojů automatizovaného vyšetřování.

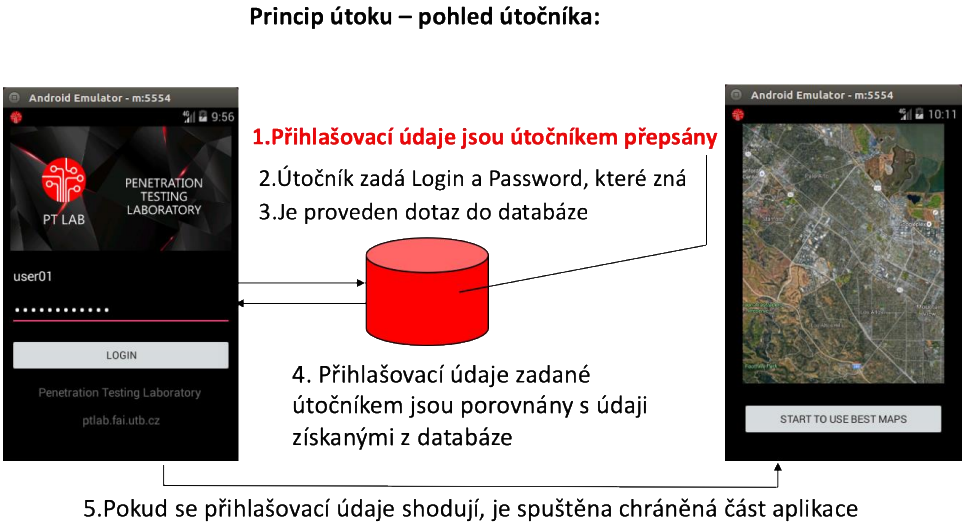
4.4 Zranitelnosti ve vyšetřovaných mobilních aplikacích nalezené v rámci výzkumu

Výzkum provedený v rámci této části dizertační práce trval dva a půl roku s průměrnou tříhodinovou denní dotací. Zkoumání zranitelností ve vyšetřovaných mobilních aplikacích vedlo k odhalení celé řady závažných bezpečnostních hrozeb. Nalezené zranitelnosti jsou systemizované do čtyř hlavních oblastí: útoky založené na analýze dat z APK balíčků, APK repackage, útoky na lokální zabezpečení mobilních aplikací a útoky na síťové zabezpečení mobilních aplikací. V disertační práci jsou nejzávažnější problémy každé z výše uvedených skupin detailně popsány, v tezích je popis z důvodu omezeného místa redukován na jednu zranitelnost, která dokumentuje způsob, jakým je problematika zpracována v dizertační práci.

Jedná se o problém podržení přihlašovacích údajů uložených v databázi, který spadá do kategorie útoků na lokální zabezpečení mobilních aplikací. Častým cílem těchto útoků jsou programy poskytující základní funkcionalitu zdarma, zatímco pokročilé funkce jsou zpoplatněny. Útočníci se snaží modifikovat zabezpečení takovým způsobem, který umožní získat cenná data, zdarma používat zpoplatněnou funkcionalitu, získat přístup k citlivým osobním údajům, odstranit uživatelské nepohodlí plynoucí z používání bezplatné verze, jako je například prodloužený start aplikace (tj. uměle vytvořená prodleva mezi startem a možností program začít používat), odstranění reklamních bannerů apod. Jeden z možných scénářů lokálního přihlašování je nastíněn v aplikaci Best Maps 5. Program byl vytvořen autorem dizertační práce na základě chyb objevených v rámci výzkumu. Očekáváné chování bezpečnostního mechanismu aplikace Best Maps 5 je naznačeno na obrázku (Obr. 4.1). Uživatel zadá do textových polí své uživatelské jméno a heslo. Následně je proveden dotaz do databáze, kde jsou vyhledány validní přihlašovací údaje. Uživatelské jméno a heslo je porovnáno s údaji získanými z lokální databáze. Shodují-li se, je spuštěna chráněná část aplikace nabízející placenou funkcionalitu.

Obr. 4.1: Očekáváné chování aplikace Best Maps 5 [zdroj vlastní]

Uvedený mechanismus má bezpečnostní slabinu, která je vidět na obrázku (Obr. 4.2). Útočník napadne lokální databázi, ve které podvrhne přihlašovací údaje. Uživatelské jméno a heslo bude nahrazeno textovými řetězci, jejichž podobu útočník zná. Po spuštění aplikace útočník zadá do textových polí strážní aktivity podvržené údaje, program provede dotaz do své databáze, ve které je podvržené uživatelské jméno a heslo. Aplikace vyhodnotí shodu údajů a spustí chráněnou část mobilní aplikace.



Obr. 4.2: Princip útoku na databázi aplikace Best Maps 5 [zdroj vlastní]

Části útoku:

- instalace testované aplikace,
- dekompilace APK balíčku,
- získání parametru package z elementu manifest ze souboru AndroidManifest.xml,

- spuštění příkazového interpretu, který je zabudován do mobilního zařízení/emulátoru,
- lokalizace databáze v privátním datovém prostoru,
- otevření databáze prostřednictvím terminálového programu sqlite3 (součást programového vybavení mobilního zařízení/emulátoru),

Dále pak je útok veden následnovně:
Příkazem .tables jsou vypsány tabulky, které obsahuje testovaná databáze:

```
sqlite> .tables
android_metadata  credentials
```

K realizaci útoku je nutné znát strukturu tabulky credentials:

```
sqlite> .schema credentials
CREATE TABLE credentials(_id INTEGER PRIMARY KEY, login TEXT, password TEXT);
```

Nyní je možné v tabulce credentials provést podvržení přihlašovacích údajů:

```
sqlite> UPDATE credentials SET login='m', password='m' WHERE _id=1;
```

Z výpisu obsahu tabulky je vidět, že podvržení přihlašovacích údajů bylo úspěšné:

```
sqlite> SELECT * FROM credentials;
_id         login      password
----------  ----------  ----------
1           m          m
```

Pokud útočník spustí aplikaci Best Maps 5, jejíž databáze byla upravena výše uvedený způsobem, stačí k prolomení bezpečnostního mechanismu zadat místo uživatelského jména a hesla písmeno „m“. Programátoři by si měli být vědomi skutečnosti, že útočníci mohou údaje v databázi nejen číst, ale také modifikovat. Z toho vyplývá, že zabezpečení programů by nemělo spoléhat na údaje v databázi, která se nachází v interní nebo externí perzistentní paměti mobilního zařízení. Rovněž používání logických proměnných (isLicenced, full_version apod.) chránící přístup k placené funkcionalitě není efektivní, neboť tyto položky mohou být snadno útočníkem modifikovány.

5. MOBILNÍ MALWARE

Problematiku bezpečnosti mobilní platformy lze rozdělit na dvě hlavní oblasti. První se týká bezpečnosti mobilních aplikací, která byla nastíněna v kapitole 4. V

kapitole 5 je popsána druhá oblast, kterou tvoří mobilní malware zneužívající bezpečnostní chyby v uživatelských aplikacích a operačních systémech. Obě uvedené oblasti spolu úzce souvisí. Neboť mobilní malware se často zaměřuje na aplikace obsahující závažné bezpečnostní chyby. K úspěchu malwaru přispívá i skupina takzvaných známých bezpečnostních chyb, které se opakují, neboť se jich dopouštějí programátoři s nízkým bezpečnostním povědomím. Mobilní aplikace, které jsou bezpečně navržené, používají moderní bezpečnostní technologie, které jsou korektně naimplementovány, výrazně minimalizují riziko jejich zneužití prostřednictvím malwaru. Na druhou stranu eliminace výskytu samotného malwaru snižuje riziko napadení mobilních aplikací tímto malwarem.

5.1 Analýza mobilního malware

Analýza mobilního malwaru umožnila pochopení principů jeho činnosti. Ať už se jednalo o anatomii útoků na legitimní aplikace, zneužívání hardwarových modulů, jako je GPS či fotoaparát nebo zneužívání mobilních zařízení jako celku, například pro DDoS pomocí vláken běžících na pozadí. Výzkum chování moderního mobilního malwaru byl nezbytný, neboť je nedostatek komplexní odborné literatury, ze které by mohla být problematika studována. Výsledky získané v analytické části výzkumu byly klíčové pro návrh účinného detekčního mechanismu mobilního malwaru pomocí neuronových sítí.

Pro zahájení úvodní fáze výzkumu bylo nejprve nutné získat funkční vzorky současného mobilního malware. Vzorky mobilního malware, které byly zkoumány v rámci dizertační práce, byly získány vyhledáváním podezřelých APK balíčků na file share serverech a pomocí aktivace metody webové infekce. Další skupina analyzovaných vzorků byla získána od společnosti AVG Technologies CZ.

5.2 Charakteristiky mobilního malware

5.2.1 Malware obsahující legitimizující část aplikace a škodlivou část aplikace

Malware tohoto typu je mobilní aplikace, která má dvě části. První z nich je zpravidla viditelná a poskytuje užitečnou či zábavnou funkcionalitu. Druhá část nemá uživatelské rozhraní a obvykle je realizována jako asynchronní vlákno běžící na pozadí nebo jako služba. Skrytá část vykonává bez vědomí uživatele nějakou škodlivou činnost. Může se například jednat o aplikaci na předpověď počasí, která obsahuje bóta umožňujícího vzdálené ovládání mobilního telefonu či tabletu prostřednictvím C&C (Command and Control) serveru. Uvedený typ malwaru lze distribuovat i prostřednictvím Google Play, neboť v současné době existuje celá řada mobilního malwaru schopného překonat bezpečnostní mechanismy Google Play [27].

Z výsledků publikovaných v [8] vyplývá, že uživatelé dávají jednoznačně přednost funkcionalitě před bezpečností, což umožňuje malwaru s legitimizující

částí napáchat velké škody, a to zejména v kombinaci s oprávněními, která jsou aplikaci přidělena.

Jak již bylo uvedeno výše, malware může spouštět svůj škodlivý kód jako asynchronní vlákno běžící na pozadí. Může se například jednat o DDoS útok běžící na pozadí. Dále je v dizertační práci zachycen mechanismus zneužití JavaScriptu v komponentě WebView.

5.2.2 Výzkum infekce legitimních aplikací (APK repackage)

Ačkoliv byl v rámci dizertační práce proveden komplexní výzkum, jehož výsledkem byly experimentální infekce APK balíčků významných výrobců, nejsou z bezpečnostních důvodů tyto výstupy detailně publikovány. Uvedená část dizertační práce je proto koncipována spíše teoreticky. Text se snaží nejen o celkový popis procesu infekce, ale i o upozornění na skutečnosti, které mohou být pro výzkumnou komunitu zabývající se otázkami kybernetické bezpečnosti přínosné. Kompletní výsledky výzkumu včetně zdrojových kódu poskytuje autor v součinnosti s Oddělením informační kriminality Policie ČR. Materiály mohou být poskytnuty na vyžádání členům Policie ČR a Armády ČR.

5.2.3 Analýza malwaru typu Hidden APK

Malware typu Hidden APK je škodlivý software, který neposkytuje uživatelům žádnou užitečnou funkcionalitu. Z tohoto důvodu musí používat techniky, které mu umožní maskovat svou přítomnost v mobilních zařízeních. Malware typu Hidden APK často pro své škodlivé záměry používá komponentu, která se jmenuje broadcast receiver. Broadcast receivery nemají žádné grafické rozhraní, a mohou tedy nepozorovaně reagovat na systémové události, které jsou oznamovány prostřednictvím tzv. broadcastů. Pokud například mobilní zařízení obdrží SMS zprávu, operační systém vyšle broadcast „příchozí SMS“. Všechny aplikace, které mají oprávnění sledovat broadcasty související s SMS zprávami, mohou přijmout broadcast „příchozí SMS“ prostřednictvím svého broadcast receiveru. Informace o systémových událostech jsou nejprve zaslány broadcast receiverům a teprve potom jsou o nich informováni uživatelé. Všechny tyto vlastnosti způsobily, že si broadcast receivery oblíbili tvůrci mobilního malwaru.

První verze malwaru typu Hidden APK byly škodlivé aplikace, které měly veškerou aplikační logiku implementovánu pouze v broadcast receiveru.

V disertační práci je detailně popsáno (a to včetně zdrojových kódů, konfiguračních souborů a snímků obrazovek), jak vytvořit malware založený na broadcast receiverech, který se dokáže v mobilních zařízeních maskovat navzdory zavedenému bezpečnostnímu mechanismu. Postup, jak malware typu Hidden APK lze vytvořit, byl publikovaný článku nazvaném Hidden APK. Tento článek byl oponenty označen za jeden ze tří nejlepších a dostal se na obálku prestižního časopisu Hakin9 - IT SECURITY MAGAZINE. Článek Hiden APK se rovněž dostal mezi 20 nejčtenějších a nejlépe hodnocených článků z let 2016 a 2017 a

proto vyšel ve speciálním čísle Best 20 Hacking Tutorials. Na základě výzkumu Hidden APK, který byl proveden v rámci dizertační práce, byl stanoven detekční postup, který je velmi jednoduchý a pomáhá spolehlivě detekovat malware typu Hidden APK. Bezpečnostní experti mohou tímto způsobem ušetřit čas, neboť lze vynechat časově náročnou dynamickou analýzu.

Kromě základního Hidden APK byla popsána další verze malwaru typu Hidden APK, která vyžaduje neúmyslnou spolupráci uživatelů (tzv. Hidden APK s uživatelskou interakcí). Malware se snaží oklamat uživatele, a přimět ho, aby Hidden APK spustil sám (například pomocí activity-alias).

5.2.4 Mobilní botnety – experimenty

Bót[6] je speciální typ škodlivého softwaru, jehož instalací se mobilní zařízení stává klientem botnetovské sítě. Bot obvykle přijímá příkazy z C&C (Command and Control) serveru. Uvedený typ malwaru se na základě získaných příkazů pokouší provádět škodlivé akce, například poškození firmwaru, posílání spamu, krádeže citlivých informací, zachytávání a odesílání SMS (např. autorizační SMS zprávy internetového bankovnictví), pořizování zvukových záznamů bez vědomí uživatele, podvodné klikání, stahování dalšího škodlivého softwaru [13].

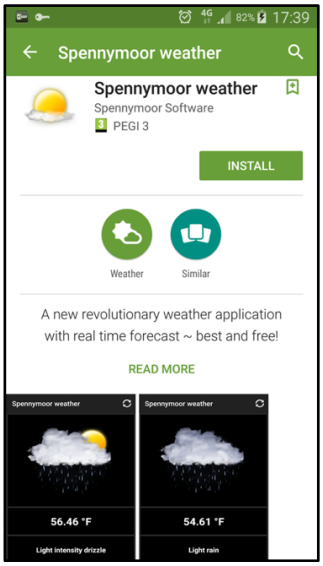
V rámci experimentu byla vytvořena aplikace Spennymoor weather (tj. "testovací aplikace"), která byla rezistentní vůči bezpečnostnímu mechanismu Google Play, neboť byla schválena a zařazena do nabídky programů sloužících k předpovědi počasí, viz Obr. 5.1. Program Spennymoor weather vykazoval celou řadu škodlivých nebo podezřelých vlastností:

- uvnitř Spennymoor weather je zašifrovaný instalační soubor "malwarové aplikace",
- Spennymoor weather obsahuje mechanismus zjišťování, zda je povolena instalace z neznámých zdrojů,
- instalace byla provedena za předpokladu, že je možné instalovat z neznámých zdrojů (pro uvedené chování není legitimní důvod, je tedy velmi podezřelé),
- instalace z lokálního souboru, který nemá příponu * .apk,
- Spennymoor weather provádí podvodnou instalaci "malwarové aplikace", která se snaží vypadat jako aktualizace systému sloužícího k získávání a zpracování předpovědí počasí, ale která ve skutečnosti nemá nic společného s předpovědí počasí.

Všechny výše uvedené nebezpečné rysy Spennymoor weather nebylo možné zjistit dynamickou analýzou nebo analýzou souboru AndroidManifest.xml, protože:

[6] *Někdy je slovem bot označováno celé zařízení, ve kterém běží malware umožňující vzdálené ovládání pomocí C&C serveru.*

- botmaster nevydal příkaz k instalaci během schvalovacího procesu na Google Play, rovněž nebylo prostřednictvím C&C serveru zasláno validní heslo umožňující dešifrování "malwarové aplikace",
- Spennymoor weather nevykazovala v souboru AndroidManifest.xml rozpor mezi funkčními a skutečně požadovanými oprávněními (tj. všechny škodlivé činnosti byly vykonávány pomocí funkčních oprávnění, které sloužily i legitimizující části aplikace),
- Aplikace Spennymoor weather se k C&C serveru nepřipojovala sama od sebe (typický bót tak činí okamžitě po startu). Místo toho čekala na legitimní stahování aktuální předpovědi počasí obsahující i skryté řídící příkazy.

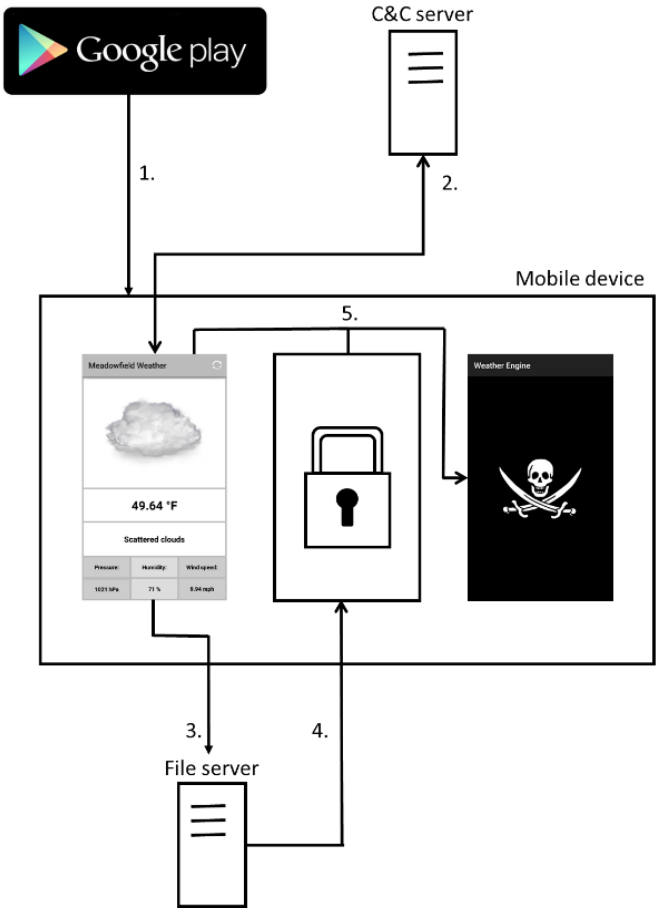


Obr. 5.1: Google Play oficiálně nabízel aplikaci Spennymoor weather obsahující bóta [zdroj vlastní]

Nicméně všechny škodlivé vlastnosti mohly být odhaleny statickou analýzou kódu. Aby byl kód pro bezpečnostní mechanismy Google Play co nejlépe staticky analyzovatelný, nebyla provedena jeho obfuskace. Bez ohledu na skutečnost, že aplikace Spennymoor weather představovala značné bezpečnostní riziko, byla úspěšně publikována prostřednictvím Google Play. Kdokoliv si ji mohl nainstalovat do svého chytrého telefonu nebo tabletu.

Na základě úspěšného publikování aplikace Spennymoor weather na Google Play, byl vytvořen další botnet, jehož cílem bylo potvrzení teorie o nedostatečné statické analýze bezpečnostních mechanismů Google Play. Klient botnetovské sítě, aplikace Meadowfield weather, byla opět kombinací bóta a trojského koně. Mobilní software Meadowfield weather byl naprogramován tak, aby byl jeho škodlivý záměr ještě více zřejmý, než tomu bylo v případě aplikace Spennymoor weather. Celkové schéma činnosti druhého experimentálního botnetu je vidět na Obr. 5.2. Aplikace Meadowfield weather je prostřednictvím Google Play nainstalována do mobilního zařízení oběti (viz bod 1, na Obr. 5.2). Jakmile

botmaster pomocí C&C serveru vydá příkaz k instalaci malwaru (viz bod 2, na Obr. 5.2), bót uvnitř programu Meadowfield weather stáhne zašifrovaný instalační balíček z externího serveru (viz body 3 a 4, na Obr. 5.2). Tento rys botnetu je velmi nebezpečný, protože botmaster může tímto způsobem měnit škodlivý software na externím serveru. To znamená, že do zařízení, která ovládá, může postupně nainstalovat celou řadu malwarů, které mohou vykonávat různorodé škodlivé činnosti. Navíc Google Play nemá přímou kontrolu nad obsahem externího serveru. Botmaster tak může v době schvalovacího procesu aplikace obsahujícího bóta umístit na externí server nějaká neškodná nebo dokonce užitečná data. Jakmile Google Play dokončí schvalovací proces, neškodná data jsou nahrazena nebezpečným instalačním balíčkem, který je navíc zašifrovaný. Po dokončení stahování je provedeno dešifrování[7] a podvodná instalace (viz bod 5, na Obr. 5.2).



Obr. 5.2: Celkové schéma činnosti druhého experimentálního botnetu [zdroj vlastní]

Navzdory velmi nebezpečné funkcionalitě, kterou disponovala aplikace Meadowfield weather, byl schvalovací proces úspěšně dokončen. Meadowfield weather si mohla do svých mobilních zařízení nainstalovat více jak miliarda

[7] *Stejně jako v případě aplikace Spennymoor weather bylo heslo zasláno v JSONu spolu s příkazem k instalaci a předpovědí počasí.*

aktivních uživatelů Google Play. Velmi nebezpečná je zejména kombinace stahování dat z externích zdrojů a následná lokální instalace využívající tato stažená data. Zdrojový kód byl vytvořen tak, aby bylo možné uvedenou činnost co nejsnadněji zjistit pomocí statické analýzy zdrojových kódů[8]. Získané experimentální výsledky potvrzují hypotézu, že těžiště bezpečnostních mechanismů Google Play spočívá v dynamické analýze a v analýze souboru AndroidManifest.xml, zatímco statická analýza zdrojových kódů je podceňována.

Důležité upozornění:

Během prováděných experimentů nebyly shromážděny žádné údaje od skutečných uživatelů Google Play. Botmasterské rozhraní bylo navrženo k provádění škodlivých akcí na základě IP adres, které byly explicitně označeny jako aktivní cíl. Uvedený krok zajistil, že všechny škodlivé činnosti byly prováděny pomocí technologie VPN (Virtual Private Network) pouze na mobilních zařízeních, která byla ve vlastnictví Fakulty aplikované informatiky, Univerzity Tomáše Bati ve Zlíně. Po ukončení experimentů byly botnety okamžitě deaktivovány. Testovací aplikace Spennymoor weather a Meadowfield weather se v současné době na Google Play již nenacházejí.

5.2.5 Ostatní charakteristiky mobilního malwaru

V rámci dizertační práce byly zkoumány některé další charakteristiky mobilního malwaru. Jedná se například o Chameleon malware, který je založen na myšlence, že je snadnější podvodně přimět uživatele, aby uživatelské jméno a heslo vyzradil (napsal) sám uživatel než bojovat se zabezpečovacími mechanismy legitimní aplikace.

Anti-Analysis techniky pod operačním systémem Android mohou používat IMEI[9], protože na rozdíl od jiných identifikátorů ho nebylo možné v Android emulátoru od společnosti Google LLC[10] snadno změnit. Anti-Analysis techniky jsou poměrně rozsáhlým tématem, které bylo podrobně zpracováno v dizertační práci.

Další charakteristiku představuje malware jako spouštěč zabezpečené aplikace. Tvůrce mobilního malwaru může napsat speciální spouštěcí program, který bude chráněnou část aplikace spouštět přímo. To znamená, že startovní aktivita vyžadující login a heslo bude obejita. Uživatel si do svého mobilního zařízení nainstaluje jak aplikaci, která poskytuje svou funkcionalitu pouze platícím zákazníkům, tak spouštěč (obsahující malware). Uživatel nebude placenou aplikaci spouštět přímo ale prostřednictvím spouštěče, který spustí nejen

---

[8] *Například připojovací řetězce na externí server byly přímou součástí zdrojového kódu (Hardcoded).*

[9] *Je-li vrácené IMEI rovno řetězci "000000000000000", je jasné, že je malware spouštěn v emulátoru, protože "000000000000000" je výchozí hodnota, kterou používá emulátor Androidu. Uvedená hodnota je rovněž nepřípustná pro fyzická zařízení.*

[10] *Jedná se o nejpoužívanější emulátor, je součástí Android SDK.*

chráněnou část placené aplikace, ale i svůj škodlivý kód. Jinými slovy činnost malware je spouštěna samotným uživatelem.

6. DETEKCE MOBILNÍHO MALWARE POMOCÍ UMĚLÉ INTELIGENCE, PŘEDEVŠÍM UMĚLÝCH NEURONOVÝCH SÍTÍ

V předchozích kapitolách byly popsány bezpečnostní chyby, kterých využívají tvůrci mobilního malware. Od práce s APK, Hidden APK balíčky přes statickou a dynamickou analýzu až po zjištění, že Google Play nemá dostatečnou kontrolu, především ve statické analýze, jelikož je tam možné malware dostat legitimním způsobem. Miliony uživatelů si pak stáhnou malware, aniž by tušili, že jsou v ohrožení. Proto kromě výše uvedených metod je vhodné vyvíjet i techniky, které malware odhalí, např. součást antivirových programů. V tomto úhlu je pozornost směřována k oblasti strojového učení, především k neuronovým sítím, které jsou známé svými výbornými klasifikačními schopnostmi za předpokladu, že na vstupu dostanou „rozumná“ data. Mezi další techniky strojového učení je možné zařadit např. Naivní Bayesovský klasifikátor [34], Logistickou regresi [35], Metodu podpůrných vektorů (Support Vector Machines) [36], Náhodný les (Random forest) [37], či metodu k-nejbližších sousedů (k-nereast neighbours) [38] a další.

6.1 Princip detekce mobilního malware pomocí umělé inteligence a strojového učení

Moderní operační systémy využívají k ochraně mobilních aplikací takzvaný Application Sandboxing. Jedná se o bezpečnostní mechanismus, který odděluje aplikace a její zdroje od ostatních aplikací, zdrojů, operačního systému a důležitých hardwarových modulů mobilního zařízení (například fotoaparát, mikrofon, …).

Princip oddělení aplikací pomocí technologie Application Sandboxing na jednu stranu výrazně zlepšuje zabezpečení mobilních aplikací, neboť aplikace jsou od sebe odděleny a nemohou tak na sebe přímo útočit či snažit se vzájemně si odcizit data. Na druhou stranu Application Sandboxing velmi znesnadňuje práci mobilním antivirovým programům. Také jsou to aplikace, které běží ve svém Sandboxu a vztahují se na ně veškerá omezení, stejně jako na ostatní mobilní aplikace. Pro antivirové programy je v těchto podmínkách těžké dohlížet na uživatelské mobilní aplikace a provádět bezpečnostní kontroly. Z tohoto důvodu antivirové společnosti hledají alternativní řešení. Jako jedna z velmi efektivních metod se jeví detekce mobilního malwaru pomocí umělé inteligence a strojového učení, např. neuronových sítí.

6.2 Výzkum v oblasti detekce mobilního malware pomocí neuronových sítí

6.2.1 Spolupráce s AVG Technologies CZ

Data, nad kterými byl prováděn výzkum schopností neuronových síti detekovat mobilní malware, byla dodána společností AVG Technologies CZ v rámci výzkumné spolupráce.

Získaná data měla formu numerické abstrakce, nebylo z nich rovněž možné určit významy jednotlivých hodnot (features). Popsaná podoba získaných dat naznačuje sílu neuronových sítí, protože pro jejich úspěšnou detekci nebylo potřeba žádné know-how o nástroji automatizoané analýzy (NAA) systému společnosti AVG Technologies CZ.

6.2.2 Učení nejen umělými neuronovými sítěmi

Umělé neuronové sítě (Artificial Neural Nets, ANN) patří do oblasti umělé inteligence a storojového učení stejně jako např. Naivní Bayesovský klasifikátor [34], Logistická regrese [35], Metoda podpůrných vektorů (Support Vector Machines) [36], Náhodný les (Random forest) [37], či metoda k-nejbližších sousedů (k-nereast neighbours) [38].

Umělé neuronové sítě i další techniky strojového učení jsou tzv. data-driven metody. Každý klasifikátor je jen tak dobrý, jak dobrá a vhodná data dostane na vstupu. Pro získání kvalitních rozhodovacích modelů je třeba připravit vhodným předzpracováním vstupní data ve formě numerických vektorů (podrobněji níže).

ANN jsou inspirovány v biologických neuronových sítích a používají se pro složité a náročné úkoly [28], [29], [30], [31]. Nejčastějším používáním je klasifikace objektů, stejně jako v tomto případě. ANN jsou schopné generalizace a robustního chování. Zvládají nejrůznější typy úloh, jako je klasifikace či rozpoznávání vzorů, řízení systémů, filtrování signálů, regresní úlohy, clustering, rekonstrukce vstupní informace a další.

Umělé neuronové sítě existují ve více variantách. Základní dělení je na supervised (s učitelem) a unsupervised (bez učitele) sítě. Experimenty zaměřené na detekci mobilního malwaru, které byly provedeny v rámci dizertační práce, byly založeny na sítích typu s učitelem, tedy dopředné síti (vícevrstvý perceptron) s Levenberg-Marquardtovým učícím algoritmem. Experimenty prověřily sadu kombinací parametrů a výsledky byly srovnány i s dalšími, výše uvedenými, klasifikačními technikami strojového učení.

6.2.3 Datová analýza a tvorba vstupních vektorů pro neuronové sítě

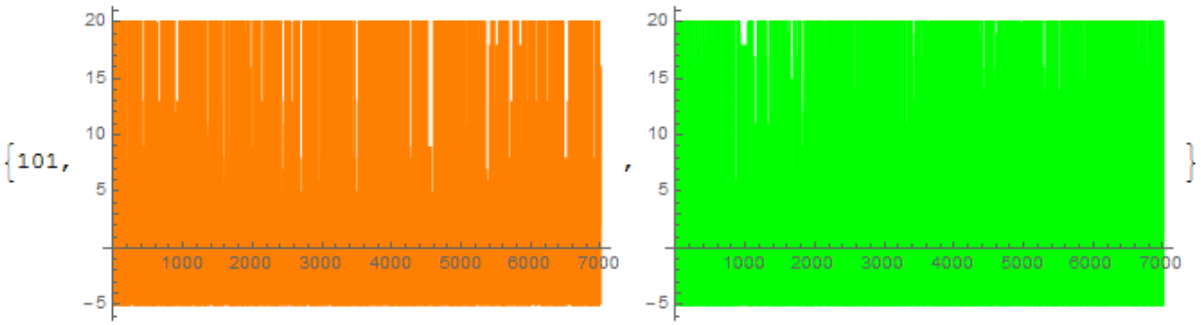
Data, která byla získána ze systému společnosti AVG Technologies CZ, měla svůj specifický firemní formát. Bylo nutné nejprve provést nad daty základní rozbor a seznámit se s jejich vnitřní strukturou. Pro tento účel naprogramoval autor práce ZuMi Parser.

Další část zpracování dat probíhalo v prostředí Wolfram Mathematica, ve kterém byl naprogramován celý analytický modul. Cílem analytického modulu byla úprava problematických features. Modul poskytoval analytické datové a grafické výstupy a vytvářel trénovací i testovací množiny vstupních vektorů pro neuronové sítě.

Vstupní reprezentace jednoho vzorku vyšetřované APK aplikace byla realizována jako vektor. Jeden prvek vektoru vždy odpovídá jedné feature. Klíčovou součástí analýzy byly vizualizace zkoumaných dat, které sloužily:

- pro zobrazení aktuálního stavu dat, například zobrazení neupravených dat po prvním načtení souborů ze ZuMi parseru, nebo po provedení transformačních procesů,
- pro eliminaci nežádoucích features,
- pro výběr nejvýhodnějšího postupu potřebného pro další fázi výzkumu.

Pro eliminaci nežádoucích features byly hledány takové, jejichž hodnoty byly stejné nebo velmi podobné jak pro malware, tak i pro legitimní aplikace. To znamená features, které zhoršovaly detekční schopnosti neuronových sítí. Pozornost byla zaměřena na skupiny features, které měly stejný typ grafu (vizualizaci), viz Obr. 6.1. Uvedené features byly označeny jako kandidáti k vyřazení. Aby nebyl vyřazen typ atributů, které by obsahovaly cenná data, bylo provedeno experimentální ověření. Features daného typu byly dočasně vyřazeny, pak byly vytvořeny množiny trénovacích a testovacích vektorů. Neuronová síť byla naučena pomocí množiny trénovacích vektorů. Poté bylo provedeno experimentální zjištění detekčních schopností neuronové sítě na množině testovacích vektorů. Pokud byly detekční schopnosti lepší, byl daný typ features vyřazen. Pokud klesla detekční schopnost neuronové sítě, byly všechny features daného typu ponechány.



Obr. 6.1: Feature s velmi podobnými hodnotami pro malware i legitimní aplikace [zdroj vlastní]

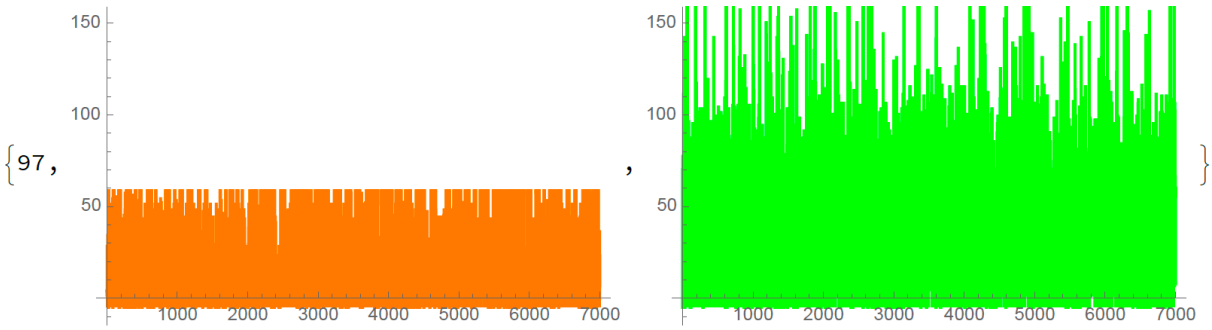
Obecně lze říci, že čím více rozdílů se mezi daty malware a čistých, neinfikovaných aplikací podaří najít, tím větší detekční schopnosti budou neuronové sítě mít. Proto byly odstraněny takové features, které zhoršovali detekci prováděnou pomocí ANN:

- features u niž převládala umělá hodnota u. Odstraňování daných features je citlivý proces, jehož špatné provedení může negativně ovlivnit detekční schopnosti, proto byla stanovena dolní a horní mez a provedeny množinové operace.,
- features, které byly z pohledu umělých neuronových sítí nerozlišitelné (jednalo se o stejné nebo velmi podobné hodnoty vyskytující u malwaru i u legitimních aplikací).

Odstraňování problematických features, je poměrně komplikovaný proces, který je plně objasněn v dizertační práci. Cílem bylo zachovat pouze takové features, které umožňovaly dosahovat dobrých detekčních výsledků. Jednalo například o features, které měly u jedné skupiny (např. u malwaru) převládající pravidelné hodnoty, viz Obr. 6.2 vlevo, zatímco u druhé skupiny (např. u legitimních aplikací) převládaly nepravidelné hodnoty, viz Obr. 6.2 vpravo.

Zajímavé se ukázaly i features s kombinovanými vlastnostmi, ve kterých zároveň: převládalo střídání pevných hodnot, rozsahy pevných hodnot byly pro obě skupiny různé, v obou skupinách se objevovaly i nepravidelné hodnoty, apod.

Přínosné byly rovněž features, které měly řídký hřebenový graf pro malware, hustý hřebenový graf pro legitimní aplikace a naopak.



Obr. 6.2: Feature 97 převládající pravidelné hodnoty (vlevo) versus nepravidelné hodnoty (vpravo) [zdroj vlastní]

6.2.4 Nastavení neuronových sítí

Byla provedena celá řada testů (trénování a testování) s různým počtem neuronů (1-15) ve skryté vrstvě a různou kombinací přenosových funkcí (logistická sigmoida, hyperbolický tangens, saturovaná lineární) s 200 učícími epochami a trénovacím algoritmem Levenberg-Marquardtovým.

6.2.5 Dosažené výsledky s využitím neuronových sítí

V Tabulka 6.1 jsou zobrazeny nejlepší dosažené hodnoty pro jednotlivé kombinace. Dle získaných hodnot lze konstatovat, že nalezené modely mají vysokou přesnost s téměř žádnou hladinou přeučení.

Tabulka 6.1: Nejlepší výsledky pro neuronové sítě v jednotlivých kombinacích

	Sigmoid - Tanh 6	Tanh - Tanh 15	Sigmoid - Sigmoid 6	Tanh - Sigmoid 2	SaturatedLin - SaturatedLin 13
Accu-train	99.5	98.8143	96.5	97.5571	99.0714
F1-train	99.5004	98.8198	96.6024	97.6081	99.0729
Accu-test	98.2286	98.0286	96.1429	96.7	98.2143
F1-test	98.2311	98.0392	96.2635	96.7859	98.2135

Jako nejlepší kombinaci lze z výše uvedené tabulky (Tabulka 6.1) prohlásit kombinaci přenosové funkce logistické sigmoidy ve skryté vrstvě, přenosové funkce hyperbolického tangentu ve výstupní vrstvě a 6 neuronů ve skryté vrstvě. Mezi sledovanými parametry byla přesnost 99,5 % při trénování a 98,23 % při testování úspěšnosti detekce vzorků malwaru. Ne vždy je přesnost u klasifikátorou směrodatnou mírou. Používají se i další parametry, mezi které lze zařadit například specificitu, sensitivitu, přesnost či F1 score. Ve výše zmíněné tabulce (Tabulka 6.1) je kromě přesnosti (accuracy) použito F1 score, které bylo v nejlepším případě rovno 99,5 % při trénování a 98,23 % při testování.

Pro tuto kombinaci pak detailní výsledky (konkrétní absolutní počty korektně i nekorektně klasifikovaných dat a jednotlivé klasifikační míry) zobrazují konfúzní matice (matice záměn) v následujících tabulkách (Tabulka 6.2 pro trénování a Tabulka 6.3 pro testování).

Tabulka 6.2: Konfúzní matice (záměn) trénovací pro nejlepší kombinaci

Konfúzní matice trénovací - 6SigmoidTanh.txt				
		P		N
predikce True	TP	3485	FP	15
predikce False	FN	20	TN	3480
	TPR	99.57	TNR	99.43
	FPR	0.43	FNR	0.57
Přesnost (Accu)	99.5			
Přesnost (Precis)	99.57			
F1 score	99.5			
TPR-Sensitivita	TNR-Specificita	FPR-False positive rate	FNR-False negative rate	

Tabulka 6.3: Konfúzní matice (záměn) testovací pro nejlepší kombinaci

Konfúzní matice testovací - 6SigmoidTanh.txt				
		P		N
predikce True	TP	3443	FP	57
predikce False	FN	67	TN	3433
	TPR	98.37	TNR	98.09
	FPR	1.63	FNR	1.91
Přesnost (Accu)	98.23			
Přesnost (Precis)	98.37			
F1 score	98.23			
TPR-Sensitivita	TNR-Specificita	FPR-False positive rate	FNR-False negative rate	

6.2.6 Srovnání výsledků s dalšími metodami

Pro srovnání s výše uvedenými neuronovými sítěmi byly využity další techniky z oblasti umělé inteligence a strojového učení - naivní bayesovský klasifikátor, logistická regrese, metoda podpůrných vektorů (Support Vector Machines),

náhodný les (Random forest), metoda k-nejbližších sousedů (k-nereast neighbours), které jsou využity v prostředí Mathematica (www.wolfram.com) ve vnitřní funkci Classify. Výsledky srovnání jsou uvedeny v následující celkové tabulce (Tabulka 6.4).

Tabulka 6.4: Srovnání metod umělé inteligence v úspěšnosti detekce malwaru

	Neuronové sítě	Náhodný les (rozhodovací stromy) = (Random Forest)	Naivní bayesovský klasifikátor	Metoda podpůrných vektorů (Support vector machines)	Metoda k-nejbližších sousedů	Logistická regrese
Accu-train	**99.5**	98.34	95.6	99	96.76	98.4
F1-train	**99.5**	98.36	96	99	96.8	98.4
Accu-test	**98.23**	97.93	95.93	98.36	96.69	98.03
F1-test	**98.23**	97.96	96.01	98.36	96.73	98.03

Z výsledků ve výše uvedené v tabulce (Tabulka 6.4) lze konstatovat, že nalezený model neuronových sítí byl nejlepší ze srovnávaných metod se sekundací metody podpůrných vektorů.

Celkově lze uzavřít výsledky prohlášením, že v současné době, kdy je v digitálním světě kolem nás malwaru opravdu velké množství, je třeba hledat metody s vysokou přesností detekce. Je třeba hledat a přeučovat techniky umělé inteligence s cílem dosažení nejlépe 100% přesnosti, jinak hrozí časté falešné poplachy i při 95% přesnosti z důvodu vysokého množství potenciálních aplikací – nosičů malwaru. Falešné poplachy zákazníci vyvinutých antimalwarových aplikací neocení. Ač jsou útočníci většinou o krok před penetračními testery či vývojáři antimalwarových programů, výhody technik umělé inteligence nabízí výbornou detekci malwaru i u neznámých (nových) vzorků. Neznamená to ale, že není potřeba aktualizace učení se zařazenými nově objevenými vzorky malwaru.

6.2.7 Popis možné implementace naučené neuronové sítě do komerčních detekčních procesů

Po dokončení výzkumné části práce je možné přikročit k vytvoření systému použitelného v technické praxi. Transformační software, datový analyzátor (který upravuje data a vytváří vstupní vektory) a naučená neuronová síť se ve vyšším jazyce implementují do jediného systému. Modul neuronové sítě bude mít navíc rozhraní pro rychlé přeučování na nové typy malwaru.

U všech vyšetřovaných APK balíčků, které byly detekovány jako malware, se vypočítá signatura (například pomocí funkce hash [33]). A následně se zařadí do virové databáze, která tvoří základní know-how antivirových společností. Mobilní antivirová aplikace bude obsahovat virovou databázi a mechanismus jejich bezpečných aktualizací. Z důvodu Application Sandboxing, který byl vysvětlen v oddílu Princip detekce mobilního malware pomocí neuronových sítí, mobilní aplikace (tzn. mobilní antivirový program) sama nebude provádět identifikaci malwaru na základě jeho škodlivého chování přímo v mobilním zařízení. Místo toho bude pouze počítat signatury nainstalovaných APK aplikací a porovnávat je s virovou databází. Pokud mobilní antivirová aplikace nalezne signaturu infikované APK aplikace, nabídne uživateli její odinstalování a další ochranné postupy v závislosti na typu malwaru.

7. PŘÍNOS PRO VĚDU A PRAXI

Bezpečnostní výzkum popsaný v dizertační práci byl zaměřen na nalezení závažných bezpečnostních chyb současných mobilních aplikací, analýzu mobilního malwaru a návrh a experimentální ověření nového způsobu detekce mobilního malwaru pomocí umělé inteligence, především pak pomocí neuronových sítí. V každé z těchto oblastí byly vytvořeny výstupy, které mohou pomoci nejen v navazujícím vědeckém bádání, ale mohou být přínosné pro technickou praxi.

Jedním z typických rysů, příznačných pro oblast bezpečnosti mobilních aplikací a mobilního malwaru, je nedostatek odborné, komplexně zpracované literatury. Na vznik odborné literatury týkající se bezpečného vývoje mobilních aplikací negativně působí rychlost a dynamika vývoje mobilních operačních systémů. Například operační systém Android vyšel za posledních jedenáct let ve dvaceti devíti verzích. Relevantní systematicky vedený výzkum zabývající se mobilním malwarem je veden především v soukromých společnostech zabývajících se kybernetickou bezpečností, ve zločineckých organizacích a v technologicky vyspělých armádách. Ani jedna z těchto skupin nemá z pochopitelných důvodů zájem sdílet výsledky svých výzkumů s odbornou veřejností. Analýzy, vyšetřování, testování, psaní experimentálních malwarů ale i praktické ověřování publikovaných závěrů dalo vzniknout práci, která tuto mezeru v odborné literatuře zaplňuje. Obsah práce není pouze teoretický, ale přináší celou řadu konkrétních postupů či vylepšení. Například u ruční dynamické analýzy práce navrhuje metodiky testovaní včetně vyvolání neošetřených výjimek aplikace, vyvolání pádu celé aplikace se sledováním systémového logu, dosahování nekonzistentního stavu testovaných aplikací, vytváření škvíry odlišnosti apod. Pro tento typ analýzy byl rovněž zaveden diskrétní čas vyšetřování, včetně důležitých bodů každého dynamického testu. Pro ruční statickou analýzu byla mimo jiné navržena technika hledání vstupního místa (tzv. Entry Point) ve zdrojových kódech aplikace. Dále byly popsány analýzy zdrojů

(např. xml konfigurační soubory, nepředkompilovaná data) a zdrojových kódů aplikace, a to včetně praktických ukázek. V rámci statické analýzy byla rovněž objasněna problematika nalezení třídy nebo metody, která je zodpovědná za zabezpečení mobilní aplikace. Pro technickou praxi je možné použít postupy, jako jsou nalezení chyb v zabezpečení exponované třídy nebo metody. V softwarových společnostech je zase možné využít objasnění útočných metod při návrhu robustního zabezpečení.

V oblasti automatizovaných vyšetřovacích metod práce mimo jiné představuje pozitiva i negativa automatizovaných vyšetřovacích frameworků a doporučený postup, jak je spolu s ostatními analytickými metodami používat. Výzkum provedený v oblasti bezpečnosti mobilních aplikací trval dva a půl roku a v dizertační práci je popsán na sto třicet jedna stranách. Tato část práce rovněž unikátní nejen svojí rozsáhlostí a komplexností ale především tím, že byla provedena z pohledu útočníka. Uvedený způsob je zajímavý nejen pro mobilní vývojáře, kterým poskytne nový pohled „druhé strany“, ale i pro bezpečnostní analytiky a penetrační testery. V softwarových společnostech může být tato část dizertační práce použita při vytváření, úpravě či obohacení interních pravidel a směrnic vztahujících se k bezpečnostním aspektům mobilního vývoje. Ve forenzních a penetračních laboratořích může zase sloužit jako podklad pro vytváření testovacích/penetračních metodik.

Části dizertační práce, které se věnují problematice mobilního malwaru a jeho detekci, jsou unikátní nejen svým rozsahem, ale i obsahem a celkovou koncepcí. Výzkum trval dva roky a je popsán na sto třiceti pěti stranách. Pro výzkumníky z řad akademické obce, ale i pro bezpečnostní experty, mohou být zajímavé části, které se týkají získávání infikovaných vzorků mobilních aplikací, stejně jako detailní vyšetřovací metody, které se například zabývají problematicky analyzovatelným zdrojovým kódem, který je obfuskovaný, restaurováním poškozených částí škodlivých kódů, které byly poškozeny dekompilačními procesy, rekonstrukcí dat, které byly před obfuskací uložené v R.class, zneužíváním komponenty WebView pomocí JavaScriptu, pokročilým analytickým používáním nástroje GNU grep apod. Další část, která je pro technickou praxi dobře využitelná je oblast popisující útočné mechanismy a charakteristiky mobilního malwaru. Například malware, který má dvě části. První z nich je zpravidla viditelná a poskytuje užitečnou či zábavnou funkcionalitu. Druhá část nemá uživatelské rozhraní a obvykle je realizována jako asynchronní vlákno běžící na pozadí nebo jako služba. Přínosné rovněž mohou být i další publikované mechanismy, jako například infekce legitimních aplikací, či navržený postup detekce malwaru typu Hidden APK. Článek nazvaný Hidden APK, který vytvořil autor dizertační práce, byl oponenty označen za jeden ze tří nejlepších a dostal se na obálku prestižního časopisu Hakin9 - IT SECURITY MAGAZINE. Článek Hiden APK se rovněž dostal mezi 20 nejčtenějších a nejlépe hodnocených článků z let 2016 a 2017 a proto vyšel ve speciálním čísle Best 20 Hacking Tutorials. Technologicky přínosné byly i experimenty s mobilními

botnety a softwarovou distribuční platformou Google Play, které naznačují nutnost zlepšení statické analýzy schvalovaných mobilních aplikací.

Práce se neomezuje pouze na bezpečnostní analýzu mobilních aplikací a současného mobilního malware, ale přichází i s proaktivním opatřením, kterým je detekce mobilního malware pomocí umělé inteligence. Zde jsou z hlediska navazujících výzkumů významné tři oblasti: algebra oprávnění, unikátní postup datové analýzy včetně tvorby vstupních vektorů, výsledky detekce mobilního malwaru pomocí metod umělé inteligence, především pomocí neuronových sítí.

Algebra oprávnění je formalizovaný aparát umožňující vytvářet systémy, které jsou schopny detekovat mobilní malware používající různé útočné mechanismy i různé ochrany znesnadňující jeho analýzu a tím i jeho kompromitaci. Algebra oprávnění je postavena na skutečnosti, že existuje místo, kde musí každá aplikace, tedy i mobilní malware poodhalit své skutečné záměry. Tím místem je soubor AndroidManifest.xml a prostředkem jsou takzvané Android Permissions (oprávnění).

Jedním z nejvýznamnějších přínosů pro vědu a praxi, kterých bylo v dizertační práci dosaženo je nejen návrh a experimentální ověření mechanismu detekce mobilního malwaru pomocí metod umělé inteligence, především pomocí neuronových sítí, ale především dosažené výsledky. Detekce vzorků malwaru a legitimních aplikací měly přesnost 99,5 % při trénování (vzorky, které naučená neuronová síť znala) a přesnost 98,23 % při testování (vzorky, které naučená neuronová síť neznala). V tomto kontextu je významný ještě jeden fakt a to, že výše uvedených výsledků bylo dosaženo na profesionální datové sadě společnosti AVG Technologies CZ, která byla dostatečně rozsáhlá i kvalitní.

8. ZÁVĚR

Výzkum provedený v rámci dizertační práce byl zaměřen na tři hlavni oblasti: bezpečnost mobilních aplikací, mobilní malware a návrh a experimentální ověření nového způsobu detekce mobilního malwaru pomocí umělé inteligence, především neuronových sítí.

V první části se dizertační práce zabývá analýzou a hledáním závažných bezpečnostních chyb v současných mobilních aplikacích. Zpracování této fáze je z hlediska kontextu ostatní odborné literatury unikátní hned v několika ohledech. Dizertační práce systematicky pokrývá všechny hlavní oblasti této problematiky od rozdílů, jakými jsou zneužívány zranitelnosti nalezené v mobilních aplikacích útočníky a tvůrci mobilního malwaru, přes problematiku APK balíčků a jejich analýzy, až po nalezené zranitelnosti ve vyšetřovaných mobilních aplikacích. Dalším unikátním rysem je obsáhlost řešeného tématu. Jen tato část dizertační práce má sto třicet jedna stran, které kromě teoretických popisů obsahují i unikátní ukázky zdrojových kódů, schémata a snímky obrazovek mobilních zařízení zachycující klíčové situace. Uvedená část práce je rovněž výjimečná tím, že je zpracována z pohledu útočníka a tvůrce mobilního malwaru. Většina publikací je psána pro programátory a má formu „best practices“, zatímco analyticky cenný pohled „druhé strany“ je zcela opomíjen.

Dizertační práce se v ucelené podobě zabývá komplexní analýzou APK balíčků, která zahrnuje kompilační proces, strukturu APK balíčků, dekompilaci a získávání původních zdrojových kódů, na které je možné zaútočit. V práci jsou vysvětleny odlišnosti kompilace Java programů pro mobilní zařízení a osobní počítače. Pozornost je věnována zejména mobilní dex kompilaci a navazujícím operacím. Rychlost a dynamika vývoje mobilních operačních systémů má za následek velké změny kompilačních procesů. Správné pochopení kompilace APK balíčků je nezbytným vstupním předpokladem pro bádání v oblasti bezpečnosti mobilních aplikací a mobilního malwaru. Navzdory uvedené skutečnosti je současná primární dokumentace vztahující se ke kompilačnímu procesu roztříštěna do většího počtu na sebe nenavazujících dokumentů. Proto je kompilace v dizertační práci systematicky zpracována, přičemž zvláštní pozornost je věnována i schématům, na kterých nejlépe vyniknou jednotlivé vývojové rozdíly.

Dizertační práce se podrobně také věnuje ruční dynamické analýze, ruční statické analýze a automatickým vyšetřovacím metodám. Patrně jednu z nejcennějších částí této práce představuje oddíl 4.6 Zranitelnosti ve vyšetřovaných mobilních aplikacích, nalezené autorem a jejichž výzkum trval dva a půl roku s průměrnou tříhodinovou denní dotací. Zkoumání zranitelností ve vyšetřovaných mobilních aplikacích vedlo k odhalení celé řady závažných bezpečnostních hrozeb, které takto uceleně dosud nebyly v odborné literatuře publikovány. Nalezené zranitelnosti jsou systemizované do čtyř hlavních oblastí:

útoky založené na analýze dat z APK balíčků, APK repackage, útoky na lokální zabezpečení mobilních aplikací a útoky na síťové zabezpečení mobilních aplikací.

Odhalené bezpečnostní chyby z provedených útoků založených na analýze dat z APK balíčků jsou v disertační práci rozděleny do tří kategorií. Do první skupiny patří dosud nepublikované zranitelnosti a jejich zneužití, jako jsou například Redirecting, String Digging, Single-Jump Attack, Multi-Jump Attack, Negation Attack, Removal Attack a další. Práce se nespokojuje pouze s teoretickým popisem zranitelností, ale ukazuje i celý proces zneužití, a to včetně zdrojových kódů v jazyce Java (hledání zranitelností, příprava na útok) a Smali (samotné provedení útoku), realizaci opětovné kompilace včetně chyb, které se mohou vyskytnout, i jejich odstranění. Do druhé skupiny zranitelností spadají takové chyby, které jsou sice známé, ale jejich závažnost či možné důsledky nebyly v odborné literatuře dostatečně zdůrazněny. Například ukládání citlivých informací a dat do privátního datového prostoru, což je způsobeno nevhodnou vývojářskou dokumentací od společnosti Google LLC. Nepočítá totiž se zařízeními, kde útočník převzal práva super uživatele. Do poslední skupiny patří bezpečnostní chyby, které jsou dobře známé, ale současná literatura se omezuje pouze na jejich popis. V takových případech jsou v dizertační práci navržena doporučení, která mobilním vývojářům umožní zlepšit bezpečnost jejich aplikací; například jak čelit nedostatku binární ochrany. Tato část práce v neposlední řadě přináší úplný, dosud nepublikovaný popis reverzování aplikačního protokolu, Server Authentication Redirecting, a celou řadu dalších zranitelností a útoků.

Další část dizertační práce se věnuje problematice mobilního malwaru. Uvedená část je unikátní nejen svým rozsahem, ale především hloubkou. Práce systematicky prochází všechny hlavní oblasti dané problematiky od získávání vzorků (z file share serverů pomocí mechanismu webové infekce), přes vyšetřovací metody až po charakteristiky mobilního malwaru. V oblasti analýzy mobilního malwaru jsou významné zejména části, které se zabývají rekonstrukcí klíčového XML layoutu podezřelé aplikace, mechanismem falešných jmen mobilního malwaru, analýzou obfuskovaných kódů pomocí Java Decompileru, získáním a analýzou oprávnění, která malware požaduje prostřednictvím souboru AndroidManifest.xml. Analyticky unikátní jsou i postupy navržené autorem, jako například rekurzivní vyhledávání klíčových výrazů uvnitř struktury tříd jazyka Java, které mohou lokalizovat škodlivou část kódu, restaurování poškozených částí kódů malwaru či kontrolované testování zrestaurované funkcionality malwaru. V oddílu 5.2 Charakteristiky mobilního malwaru byla představena celá řada dosud nepublikovaných mechanismů, z nichž nejdůležitější je analýza malwaru typu Hidden APK a experimenty s mobilními botnety. Experimenty s mobilními botnety potvrdily hypotézu o nedostatečné statické analýze na Google Play. Bóti byly totiž vytvořeni tak, aby je bylo možné snadno identifikovat staticky, zatímco dynamická analýza, která by vedla k jejich kompromitaci, byla téměř nemožná. Naznačený způsob se ukázal jako účinný, protože navzdory velmi nebezpečné funkcionalitě, kterou disponovaly aplikace Spennymoor

weather a Meadowfield weather (bóti), byly schvalovací procesy na Google Play úspěšně dokončeny. Tyto škodlivé aplikace si mohla pak do svých mobilních zařízení nainstalovat více než miliarda aktivních uživatelů Google Play.

Poslední výzkumná část dizertační práce se zabývá problematikou detekce mobilního malwaru pomocí neuronových sítí. Tato část obsahuje unikátní formalizovaný aparát pro detekci mobilního malwaru, kterým je algebra oprávnění. Velmi významná byla spolupráce s AVG Technologies CZ, neboť experimenty s detekcí mobilního malwaru pomocí neuronových sítí probíhaly na profesionální datové sadě od této společnosti. Uvedená sada obsahuje dostatečný počet prokazatelně infikovaných aplikací i čistých legitimních aplikací. Rozsáhlost a kvalita datové sady umožnila označit získané výsledky za vysoce relevantní. Jako klíčová pro přesnost detekce pomocí neuronových sítí se ukázala datová analýza a pečlivá tvorba vstupních vektorů, která se skládala zejména z identifikace a redukce problematických složek vektorů. Další významnou část představují pokusy s neuronovými sítěmi, které zahrnovaly experimentování s různými počty neuronů ve skryté vrstvě a kombinací různých přenosových funkcí ve skryté a výstupní vrstvě (logistická sigmoida, hyperbolický tangens a lineární saturovaná funkce). Pro výzkumníky zabývající se problematikou detekce mobilního malwaru bude určitě zajímavé i srovnání výsledků na této rozsáhlé datové sadě, dosažené i pomocí dalších klasifikačních metod umělé inteligence, jako je metoda podpůrných vektorů (Support vector machines), metoda náhodného lesa (Random Forest), naivní Bayesovský klasifikátor, metoda nejbližších sousedů či logistická regrese. Neuronové sítě dosáhly 99,5 % přesnosti při trénování a 98,23 % při testování se sekundací metody podpůrných vektorů s 99 % přesnosti při trénování a 98,36 % při testování. Ostatní uvedené metody byly zhruba o jedno až čtyři procenta horší. Tyto výsledky ukazují sílu strojového učení a zároveň naznačují jeden z perspektivních směrů, kterými by se měla ubírat problematika detekce mobilního malwaru.

Vývoj komponent mobilních aplikací i mobilních operačních systémů je velmi dynamický. Některé staré chyby jsou v nových operačních systémech částečně nebo zcela opraveny, ale často nové verze operačních systémů přináší chyby nové, na které je potřeba reagovat. Komplexnost a systematické zpracování dělá z této dizertační práce unikátní zdroj informací, který mohou využít nejen výzkumníci z řad akademické obce, ale i penetrační testeři a analytici mobilního malwaru jako odrazový můstek sloužící pro zorientování se v této oblasti.

SEZNAM POUŽITÉ LITERATURY

[1] Ericsson Mobility Report: On the Pulse of the Networked Society. *Ericsson* [online]. Stockholm: Ericsson, 2016, 2016 [cit. 2018-05-03]. Dostupné z: https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf

[2] 2015 World Population Data Sheet with a special focus on women's empowerment. *Population Reference Bereau* [online]. Washington, DC: Population Reference Bereau, 2015, 2015 [cit. 2018-05-03]. Dostupné z: http://www.prb.org/pdf15/2015-world-population-data-sheet_eng.pdf

[3] Ericsson Mobility Report: On the Pulse of the Networked Society. *Ericsson* [online]. Stockholm: Ericsson, 2017, 2017 [cit. 2018-05-03]. Dostupné z: https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-november-2017.pdf

[4] Percentage of all global web pages served to mobile phones from 2009 to 2018. *Statista: The Statistics Portal* [online]. New York: Statista, 2018, 2018 [cit. 2018-05-03]. Dostupné z: https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/

[5] Number of available applications in the Google Play Store from December 2009 to December 2017. *Statista: The Statistics Portal* [online]. New York: Statista, 2018, 2018 [cit. 2018-05-03]. Dostupné z: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

[6] Android History. *Android* [online]. Android - Google, 2014, 2014 [cit. 2018-05-03]. Dostupné z: https://www.android.com/history/

[7] SIX, Jeff. *Application security for the Android platform*. Sebastopol, CA: O'Reilly, 2012. ISBN 978-1449315078.

[8] VELU, Vijay Kumar. *Mobile Application Penetration Testing*. Birmingham: Packt Publishing, 2016. ISBN 978-1785883378.

[9] HOU, Rui, Zhigang JIN a Baoliang WANG. Investigation of taint analysis for Smartphone-implicit taint detection and privacy leakage detection. *EURASIP Journal on Wireless Communications and Networking*. 2016, 2016(1). DOI: 10.1186/s13638-016-0711-4. ISSN 1687-1499. Dostupné také z: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-016-0711-4.

[10] Rizwan AHMED and Rajiv V. DHARASKAR. Study of Mobile Botnets: An Analysis from the Perspective of Efficient Generalized Forensics Framework for Mobile Devices. *IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2012)*, pp. 5-8, 2012.

[11] GENG, Guining, Guoai XU, Miao ZHANG, Yanhui GUO, Guang YANG a Cui WEI. The Design of SMS Based Heterogeneous Mobile Botnet: Proof of concept. *Journal of Computers*. IEEE, 2012, 2014, **7**(1), -. DOI: 10.4304/jcp.7.1.235-243. ISBN 978-1-4799-5692-0. ISSN 1796-203X. Dostupné také z: http://ojs.academypublisher.com/index.php/jcp/article/view/5338

[12] ABDULLAH, Zubaile, Madihah Mohd SAUDI a Nor Badrul ANUAR. Mobile botnet detection: Proof of concept. *2014 IEEE 5th Control and System Graduate Research Colloquium*. IEEE, 2014, 2014, 257-262. DOI: 10.1109/ICSGRC.2014.6908733. ISBN 978-1-4799-5692-0. Dostupné také z: http://ieeexplore.ieee.org/document/6908733/

[13] LA POLLA, Mariantonietta, Fabio MARTINELLI, Daniele SGANDURRA, Yanhui GUO, Guang YANG a Cui WEI. A Survey on Security for Mobile Devices: Proof of concept. *Journal of Computers*. IEEE, 2013, 2014, **15**(1), 446-471. DOI: 10.1109/SURV.2012.013012.00028. ISBN 978-1-4799-5692-0. ISSN 1553-877X. Dostupné také z: http://ieeexplore.ieee.org/document/6170530/

[14] PIETERSE, Heloise, Martin S OLIVIER, Daniele SGANDURRA, Yanhui GUO, Guang YANG a Cui WEI. Android botnets on the rise: Trends and characteristics. *2012 Information Security for South Africa*. IEEE, 2012, 2012, **15**(1), 1-5. DOI: 10.1109/ISSA.2012.6320432. ISBN 978-1-4673-2159-4. ISSN 1553-877X. Dostupné také z: http://ieeexplore.ieee.org/document/6320432/

[15] Summary of the HIPAA Security Rule. *HHS* [online]. Washington, D.C.: U.S. Department of Health & Human Services, 2013, 2013 [cit. 2018-05-03]. Dostupné z: https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html

[16] OWASP Mobile Security Project. *Owasp* [online]. London: owasp, 2017, 2017 [cit. 2018-05-03]. Dostupné z: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#Top_Ten_Mobile_Risks

[17] EMERGING TECHNOLOGIES, ed. PCI Mobile Payment Acceptance Security Guidelines for Developers. *PCI Security Standards* [online]. Wakefield: Security Standards Council, 2017, 2017 [cit. 2018-10-03]. Dostupné z: https://www.pcisecuritystandards.org/documents/PCI_Mobile_Payment_Acceptance_Security_Guidelines_for_Developers_v2_0.pdf

[18] Security for Android Developers. *Android Developers* [online]. [cit. 2019-02-20]. Dostupné z: https://developer.android.com/topic/security/index.html

[19] App Security Improvement Program. *Android Developers* [online]. [cit. 2019-02-20]. Dostupné z: https://developer.android.com/google/play/asi.html
[20] Forrester Research. *Forrester* [online]. Cambridge: Forrester, 2018, 2018 [cit. 2018-05-03]. Dostupné z: https://go.forrester.com/research
[21] Address The Top 10 Nontechnical Security Risks In Mobile App Development: How To Change Culture And Strengthen App Security. *Forrester* [online]. Cambridge: Forrester, 2016, 2016 [cit. 2018-05-03]. Dostupné z: https://www.forrester.com/report/Address+The+Top+10+Nontechnical+Security+Risks+In+Mobile+App+Development/-/E-RES112801
[22] dex2jar. *Sourceforge* [online]. 2016, [cit. 2018-10-10]. Dostupné z: https://sourceforge.net/projects/dex2jar/?source=navbar
[23] Java Decompiler. *Java Decompiler* [online]. [cit. 2018-09-29]. Dostupné z: http://jd.benow.ca
[24] Run apps on the Android Emulator. *Android Studio* [online]. [cit. 2018-08-10]. Dostupné z: https://developer.android.com/studio/run/emulator.html
[25] The perfect Android emulator to play mobile games on PC. *NoxPlayer* [online], 2019, [2019-03-01]. Dostupné z: https://www.bignox.com
[26] Android as a Service, Run Android virtual devices. *Genymotion* [online], 2019, [cit. 2019-01-09]. Dostupné z: https://www.genymotion.com
[27] BALAJI, N. 2 Android Apps From Google Play Store Launching Banking Malware With Sophisticated Evasion Techniques. *Gbhackers* [online], 2019, [cit. 2019-02-01]. Dostupné z: https://gbhackers.com/android-apps-banking-malware/
[28] GURNEY, Kevin. *An introduction to neural networks*. London: UCL Press, 1997. ISBN 18-572-8673-1.
[29] HERTZ, John, Anders KROGH a Richard G. PALMER. *Introduction to the theory of neural computation*. Redwood City, Calif.: Addison-Wesley Pub. Co., c1991. ISBN 978-0201515602.
[30] WASSERMAN, Philip D. *Neural computing: theory and practice*. New York: Van Nostrand Reinhold, c1989. ISBN 978-0442207434.
[31] FAUSETT, Laurene V. *Fundamentals of neural networks: architectures, algorithms, and applications*. Englewood Cliffs, NJ: Prentice-Hall, c1994. ISBN 978-0133341867.
[32] KIM, Daniel. On tje Current State of Ransomware. *Tufts* [online]. 2015, [cit. 2017-06-07]. Dostupné z: http://www.cs.tufts.edu/comp/116/archive/fall2015/dkim.pdf
[33] Hash Function. WolframMathWorld [online]. [cit. 2018-05-03]. Dostupné z: http://mathworld.wolfram.com/HashFunction.html
[34] RUSSELL, Stuart a Peter NORVIG, 2002. Artificial Intelligence: A Modern Approach. 2 edition. Upper Saddle River, N.J: Prentice Hall. ISBN 978-0-13-790395-5.

[35] HOSMER Jr., David W., Stanley LEMESHOW a Rodney X. STURDIVANT, 2013. Applied Logistic Regression. 3 edition. Hoboken, New Jersey: Wiley. ISBN 978-0-470-58247-3.
[36] CORTES, Corinna a Vladimir VAPNIK, 1995. Support-vector networks. *Machine Learning* [online]. **20**(3), 273–297. ISSN 1573-0565. Dostupné z: doi:10.1007/BF00994018
[37] HO, Tin Kam, 1995. Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Dostupné z: https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf
[38] COVER, T.M. a Peter E. HART, 1967. "Nearest neighbor pattern classification" (PDF). IEEE Transactions on Information Theory. 13 (1): 21–27, doi:10.1109/TIT.1967.1053964.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ANN	Artificial Neural Nets
API	Application Programming
APK	Android aPplication pacKage
C&C	Command and Control
CVE	Common Vulnerabilities and Exposures
DoS	Denial of Service
DDoS	Distributed Denial of Service
FTP	File Transfer Protocol
FTPS	FTP s SSL/TLS
GNU	GNU's Not Unix
GUI	Graphical User Interface
iOS	iPhone Operating Systém
IP	Internet Protocol
HIPAA	Health Insurance Portability and Accountability Act
JSON	JavaScript Object Notation
NAA	Nástroj Automatizované Analýzy
PII	Personally identifiable information
PNG	Portable Network Graphics
OS	Operating System
OWASP	Open Web Application Security Project
SDK	Software Development Kit
SPI	Sensitive Personal Information
URI	Uniform Resource Identifier
VPN	Virtual Private Network
XML	Extensible Markup Language

PUBLIKAČNÍ AKTIVITY AUTORA

Mezinárodní patentová přihláška podaná k patentovému řízení v roce 2019

Jašek R. [25], Oulehla M. [35], Žáček P. [6], Krňávek J. [14], Lázecký V. [5], Makowski J. [5], Malík T. [5], Malík J. [5] Identity and License Verification System for Working with Highly Sensitive Data

Časopisy

[A.1] Oulehla M. [90], Malanik D. [10] Comparison of cryptographic methods Triple DES, AES and a method based on the arithmetic of elliptic curves (ECC) on the Android mobile platform. *International Journal of Computers and Communications.* 2015, (9), s. 62-67. ISSN 2074-1294. Dostupné také z: http://www.naun.org/main/UPress/cc/2015/a182012-145.pdf

[A.2] Oulehla M. [90], Malanik D. [10] Techniques that Allow Hidden Activity Based Malware on Android Mobile Devices. *International Journal of Scientific Engineering and Applied Science (IJSEAS).* 2016, 2(3), s. 409-419. ISSN 2395-3470.

Konference

[A.3] Oulehla, M. [90], Malanik, D. [10] Přenos šifrovaných souborů mezi mobilními zařízeními. In *Mezinárodní konference Bezpečnostní management a společnost.* Brno: Univerzita obrany. 2013. str. 351 - 357. ISBN 978-80-7231-928-2.

[A.4] Oulehla, M. [90], Malanik, D. [10] Využití mobilních aplikací a šifrování při logistických procesech. In *First International Conference On Application of Modern Information Technologies in Logistics (ITL 2013)*. Přerov: College of Logistics in Přerov. 2013. str. 15-23. ISBN 978-80-87179-32-1.

[A.5] Oulehla, M. [90], Malanik, D. [10] Comparison of cryptographic methods based on the arithmetic of elliptic curves (ECC) with symmetric cryptography methods. In *2014 International Conference Mathematics and Computers in Sciences and Industry (MCSI 2014)*. Bulgaria, 2014, 6 p.

[A.6] Oulehla, M. [90], Malanik, D. [10] Techniques Allowing Broadcast Receiver Malware on Android Platform. In *2015 Recent Advances in Systems - Proceedings of the 19th International Conference on Systems (part of CSCC '15),* Zakynthos Island, Greece, 2015, str. 235 - 239. ISBN: 978-1-61804-321-4.

[A.7] Oulehla, M. [90], Malanik, D. [10] Techniques that Allow Hidden Activity Based Malware on Android Mobile Devices. In *2015 International Conference on Cyber Warfare and Security (ICCWS)*, Kruger National Park, South Africa, 2015, Poster.

[A.8] Oulehla, M. [100] Investigation Into Google Play Security Mechanisms Via Experimental Botnet, In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Abu Dhabi, United Arab Emirates: IEEE, 2015, ISBN 978-1-5090-0480-5.

[A.9] Oulehla, M. [90], Malaník, D. [10] Insight into Contemporary Dissemination Techniques of Mobile Botnet Clients (Bots). In *The Tenth International Conference on Emerging Security Information, Systems and Technologies: SECURWARE 2016.* France: IARIA, 2016, s. 117-123. ISBN 978-1-61208-493-0. ISSN 2162-2116.

[A.10] Oulehla, M. [80], Kominkova Oplatkova, Z. [10], Malanik, D. [10] Detection of mobile botnets using neural networks In *FTC 2016 - Future Technologies Conference 2016*, San Francisco, United States, ISBN 978-1-5090-4171-8.

[A.11] Kominkova Oplatkova, Z. [50], Oulehla, M. [50] Mobile Botnet Detection via Artificial Neural Networks In *2017 International Conference on Logistics, Informatics and Service Sciences (LISS)*. Kyoto, Japan: IEEE, 2017, p. 5. ISBN 978-1-5386-1047-3.

ODBORNÝ ŽIVOTOPIS AUTORA

europass Životopis

OSOBNÍ ÚDAJE

Ing. Oulehla Milan

Moutnická 1381/4, 627 00 Brno (Česká republika)

+420 732 118 306

oulehla.milan@gmail.com

PRACOVNÍ ZKUŠENOSTI

2013–do současnosti — Mobile Security Section
Penetrační laboratoř - Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně
http://ptlab.fai.utb.cz/oulehla/

2007–do současnosti — Správce sítě
ABCERT

2005–do současnosti — Projektant a analytik informačních systémů
Mendelova univerzita v Brně

2002–2005 — Správce databázových systémů
Friends of the Earth Czech Republic

1994–2002 — Servisní technik
firma Milan Křížek - Prodej paliva a kontejnerová autodoprava

VZDĚLÁNÍ, ODBORNÁ PŘÍPRAVA A KURZY

2013–do současnosti — Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně, obor Inženýrská informatika, doktorský typ studia, kombinovaná forma

2011–2013 — Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně, obor Informační technologie, magisterské navazující studium, kombinovaná forma, zakončeno státní závěrečnou zkouškou

2007–2011 — Přírodovědecká fakulta, Univerzita Palackého, obor Aplikovaná informatika, bakalářské studium, kombinovaná forma, zakončeno státní závěrečnou zkouškou

2003 — Microsoft Certified Professional, zakončeno mezinárodní zkouškou

1990–1994 — Gymnázium Uničov, přírodovědná větev, zakončeno maturitní zkouškou

OSOBNÍ DOVEDNOSTI

Mateřský jazyk: čeština

Další jazyky

	POROZUMĚNÍ		MLUVENÍ		PÍSEMNÝ PROJEV
	Poslech	Čtení	Ústní interakce	Samostatný ústní projev	Psaní
angličtina	C1	C1	B2	B2	C1
2015 - doktorská zkouška - Academic writing					

Úrovně: A1 a A2: Začátečník - B1 a B2: Nezávislý uživatel - C1 a C2: Způsobilý uživatel
Společný evropský referenční rámec pro jazyky

Odborné dovednosti

Výzkum v oblasti bezpečnosti mobilních a informačních technologií, programování v jazycích Java (se zaměřením na mobilní platformu Android), C#, C++, HTML, JavaScript, PHP, SQL, správa počítačových sítí (včetně správy serverů)

V jazyce Java a Bash autor naprogramoval systém zabezpečeného sdílení dat mezi mobilními zařízeními v reálném čase. Dále autor vytvořil dva experimentální botnety, pro účely zkoumání zabezpečení softwarového tržiště Google Play. O svém bezpečnostním výzkumu autor přednáší na významných komerčních (mDevTalk, HACKERFest, SecPublica, OWASP Czech Chapter Meeting, atd.) i akademických konferencích (viz publikační činnost). Části výzkumu, které nemohou být z bezpečnostních důvodů prezentovány veřejně, jsou autorem pravidelně prezentovány v rámci Policie ČR a Armády ČR.

Milan Oulehla

Bezpečnostní chyby na mobilní platformě, jejich zneužívání a návrh proaktivního opatření s využitím umělé inteligence

Security Issues on Mobile Platform, Their Exploiting and Proactive Measure Using Artificial Intelligence

Teze disertační práce

Vydala Univerzita Tomáše Bati ve Zlíně,
nám. T. G. Masaryka 5555, 760 01 Zlín.

Náklad: vyšlo elektronicky

Pořadí vydání: první

Sazba: Milan Oulehla

Publikace neprošla jazykovou ani redakční úpravou.

Rok vydání 2020

ISBN 978-80-7454-911-3