Kauno technologijos universitetas
Informatikos fakultetas

Išmaniųjų įrenginių saugumo nustatymų stebėjimo metodas

Baigiamasis magistro studijų projektas

Nerijus Šatkauskas
Projekto autorius

Prof. Jevgenijus Toldinas
Vadovas

Kaunas, 2019 m.

Kauno technologijos universitetas

Informatikos fakultetas

Išmaniųjų įrenginių saugumo nustatymų stebėjimo metodas

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Nerijus Šatkauskas

Projekto autorius

Prof. Jevgenijus Toldinas

Vadovas

Doc. Stasys Maciulevičius

Recenzentas

Kaunas, 2019 m.

Kauno technologijos universitetas

Informatikos fakultetas

Nerijus Šatkauskas

Išmaniųjų įrenginių saugumo nustatymų stebėjimo metodas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Nerijaus Šatkausko, baigiamasis projektas tema „Išmaniųjų įrenginių saugumo nustatymų stebėjimo metodas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)	(parašas)

Kauno technologijos universitetas

Informatikos fakultetas

Baigiamojo magistro projekto užduotis

Projekto tema _____

Reikalavimai ir sąlygos
(tikslinti pavadinimą pagal poreikį)

Vadovas / Vadovė _____

(vadovo pareigos, vardas, pavardė, parašas) (data)

Šatkauskas, Nerijus. Išmaniųjų įrenginių saugumo nustatymų stebėjimo metodas. Magistro studijų baigiamasis projektas / vadovas prof. Jevgenijus Toldinas; Kauno technologijos universitetas, informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informacijos ir informacinių technologijų sauga (kodas 621E10003)

Reikšminiai žodžiai: Išmanioji aplinka, išmanusis įrenginys, Android, pavojingi leidimai, jautri informacija, informacijos nutekėjimas, stebėjimo metodas.

Kaunas, 2019 m. 65 p. (be priedų)

Santrauka

Išmaniųjų įrenginių arba išmaniosios aplinkos samprata yra gan plati. Bet kaip nurodoma šaltiniuose, tai tokia aplinka, kuri užpildyta įrenginiais ir jutikliais, gebančiais sąveikauti su žmogumi [2]. Ją galim suskirstyti į 3 pagrindinius lygmenis [3]: jutiminis, perdavimo ir programinis. Šiame darbe pagrindinis dėmesys skiriamas programiniam lygmeniui.

Įvairiuose šaltiniuose galima rasti teiginių, kad „Android“ įrenginių skaičius jau viršijo 2 000 000 000. Viename jų [1] apie tai rašyta jau 2017 m., remiantis „Google“ generalinio direktoriaus Sundar Pichai žodžiais. Tuo tarpu „Apple“, kaip nurodoma tame pačiame šaltinyje, teigusi, kad 2016 m. pasiekė 1 000 000 000 „iOS“ įrenginių ribą. „Android“ operacinė sistema pasirinkta dėl didesnio populiarumo ir prieinamumo.

„Android“ operacinė sistema yra atvirojo kodo, sukurta „Linux“ pagrindu. Kiekvienos norintis gali įkelti savo sukurtas programėles į „Play Store“ programėlių parduotuvę. Nors „Google“ stengiasi kiek išgalėdama, visų kenkėjiškų programėlių aptikti ir pašalinti, prieš jas paskelbiant „Play Store“, nepavyksta.

Kitas nerimą keliantis saugumo aspektas – net ir nekenkėjiškomis laikomos programėlės linkusios prašyti per daug leidimų. Kadangi „Android“ OS laikomos informacijos konfidencialumas priklauso nuo suteiktų leidimų, trečiųjų šalių programėlės tokiu būdu gauna nevaržomą prieigą. Kaip teigiama įvairiuose pastebėjimuose, naudotojas, net ir žinodamas, kad tam tikram leidimui priskiriama informacija yra naudojama, nežino kaip konkrečiai ji naudojama.

Sekimo programėlės, nors skirtos vaikų ir darbuotojų, besinaudojančių įmonės įrenginiais, stebėjimui, gali būti pritaikytos kitais tikslais. Jomis galima sekti nieko neįtariantį artimos aplinkos asmenį ir sužinoti beveik viską: skambučius, žinutes, buvimo vietą ir t. t.

Šatkauskas, Nerijus. Security Permission Monitoring Method for Smart Devices. Master's Final Degree Project / supervisor prof., Jevgenijus Toldinas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Security of Information and Information Technology (Reference 621E10003).

Keywords: Smart environment, smart device, Android, dangerous permissions, sensitive information, information leak, monitoring method.

Kaunas, 2019. 65 pages (excl. the annex).

Summary

The concept of smart devices or smart environment is quite abstract. However, as different sources state it is such an environment which is full of devices and sensors that are able to interact with humans [2]. It can be divided into 3 main levels [3]: perception, transportation and application. This paper mainly focuses on the application level.

It is possible to find some statements in different sources that Android has already exceeded the limit of 2 000 000 000. One of them [1] has already claimed about it on 2017, as it was a part of Sundar Pichai, Google CEO, report. Meanwhile Apple as the same source announced has reached a limit of 1 000 000 000 iOS device limit in 2016. Android has been chosen due to higher popularity and availability.

Android OS is an open-code platformed which is developed on the basis of Linux. Anyone at his / her own choice can upload any developed applications to Android Play Store. Google tries as hard as it can but is not possible to discover all the malware and remove it before it is uploaded to Google Play Store.

Another security aspect which raises concerns – even benign applications tend to request too many permissions. As the confidentiality of any data stored on Android OS relies on granted permissions, third party applications get unrestricted access in that manner. As different sources claim, a user does not know how in particular that data is used even he / she knows that certain data is within the scope of the granted permission.

Spying applications are mainly developed for monitoring children or employees who use business smart devices, however, they can be used for other purposes. They can be used to spy on a victim from a close environment who has no clue about it in order to learn almost everything: call logs, SMS messages, location etc.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

1. ĮVADAS

Darbo problematika ir aktualumas

Šiuo metu „Android“ operacinė sistema, kuri skirta mobiliesiems įrenginiams, yra viena populiariausių pasaulyje. Naujausiais duomenimis teigiama, kad „Android“ operacinę sistemą naudojančių įrenginių skaičius jau perkopė 2 000 000 000 [1]. Deja, didžiojoje dalyje įrenginių naudojama pasenusi operacinė sistema, kuri nėra saugi. Dar vienas svarbus klausimas – kiek ta operacinė sistema yra patikima, net jei ji ir yra pačios naujausios versijos? O ar duomenys, saugomi „Android“ išmaniuosiuose įrenginiuose, yra saugūs? Kokių programėlės turi galimybių pasiekti asmeninius / slaptus duomenis? Ar yra metodų, pagal kuriuos galima įvertinti, kiek programėlės turi galimybių nuskaityti slaptą asmeninę informaciją?

Darbo tikslas ir uždaviniai

Tikslas: įvertinti išmaniosios aplinkos informacijos laikymo saugą „Android“ operacinėje sistemoje. Nustatyti, ar pašaliniai asmenys per tam tikras programėles gali perimti informaciją. Kokiais metodais būtų galima saugą sustiprinti.

Uždaviniai

- Nustatyti ir klasifikuoti išmaniojo įrenginio vietas, kur yra saugoma jautri asmeninė informacija – el. paštas, adresai, užrašai, telefonų numeriai, nuotraukos, prisijungimo informacija ir t. t.
- Sudaryti informacijos jautrumo požymius.
- Nustatyti programėlių galimybes pasiekti ir nuskaityti jautrią asmeninę informaciją.
- Pasiūlyti programėlių galimybių pasiekti ir nuskaityti jautrią asmeninę informaciją aptikimo metodus.
- Realizuoti skanerio programėlės prototipą, kuris pagal pasiūlytos informacijos jautrumo požymius informuos naudotoją ir pateiks išmaniojo įrenginio programėlių sąsają su jautria informacija (t. y. programėlių sąrašas ir programėlių pasiekiamos jautrios informacijos vietų sąrašas).

Darbo rezultatai ir jų svarba

Išanalizuota išmanioji aplinka ir jos sudedamoji dalis „Android“ OS. Aptartos saugumo grėsmės, kylančiose šiuose įrenginiuose saugomiems duomenims. Saugomos informacijos jautrumui įvertinti pasiūlytas V-S ašių metodas. Pasiūlytas prototipas, vertinantis programėlių galimybes nutekinti jautrią informaciją. Iš viso ištirta 100 saugių programėlių, 41 kenkėjiška programėlė ir 28 komercinės sekimo programėlės.

Prototipas padeda paprastai ir aiškiai įvertinti programėlių galimybes nutekinti jautrią informaciją, nes bendra taškų suma apskaičiuojama pagal prašomus leidimus (Y ašis) ir informacijos vertę (X ašis). Įvardijami dažniausiai prašomi leidimai, iš kurių galima spręsti, kokio tipo informacijai dažniausiai kyla nutekinimo pavojus.

Darbo struktūra

Šį magistrinį darbą sudaro šios pagrindinės dalys:

- Pirmojoje dalyje analizuojama išmanioji aplinka ir jos sandara. Pagrindinis dėmesys skiriamas programiniam lygmeniui. „Android“ operacinė sistema pasirenkama dėl populiarumo. Išanalizuojamos šios OS grėsmės. Bendrai įvardijami bendri grėsmių aptikimo metodai ir galimybės jų išvengti. Pristatomas V-S informacijos klasifikavimo metodas.
- Antroje dalyje aptariami saugumo nustatymų stebėjimo metodai, leidžiantys programėlių leidimus nuskaityti skirtingais būdais, ir prašomų leidimų išsaugojimas. Aptariamas „Android“ leidimų skirstymas ir pristatomas prototipas. Analizuojamos ir aprašomos prototipo „Programėlių monitoringo sistema“ savybės.
- Trečioje dalyje aprašomi 3 atlikti tyrimai: (1) dažniausiai naudojamų „Android“ programėlių tyrimas, (2) „Android“ kenkėjiškų programėlių tyrimas ir (3) „Android“ šnipinėjimo programėlių tyrimas. Išnagrinėjami šių programėlių tipams būdingi leidimai, programėlių panašumai ir skirtumai leidimų požiūriu, prototipu gauti rezultatai ir programėlių galimybės nutekinti jautrią informaciją. Surinkti duomenys atvaizduojami grafiškai.
- Pateikiamos bendros magistrinio darbo išvados.
- Priede pateikiami parašyti moksliniai straipsniai.

2. IŠMANIEJI ĮRENGINIAI IR JŲ SAUGUMO ANALIZĖ

2.1. Išmanioji aplinka ir išmanieji įrenginiai

Išmanioji aplinka [2] yra įprasta aplinka su įtaisyta vaizdo ir garso aptikimo sistema, sklaidos įrenginiais, jutikliais ir tinklais, kurie gali suprasti žmogų, jaučia vykdomą žmogaus veiklą ir reaguoja į tą veiklą. Jų naudojimas yra akivaizdus dėl fakto, kad įvairios išmaniosios aplinkos yra integruojamos visose gyvenimo situacijose. Šiems išmaniosios erdvės elementams reikia mikroprogramininės įrangos, standartų ir sąsajos technologijos, kad būtų galima valdyti jų sudėtingą sistemą.

Netolimoje ateityje, kaip manoma, išmaniąją aplinką sudarys milijardai išmaniųjų įrenginių, kurie galės apdoroti, jausti ir įvertinti situaciją, būdami prijungti prie interneto arba vidinio tinklo. Integravus socialinio tinklo idėją į išmaniąją aplinką (daiktų internetą), atsiranda socialinės išmaniosios aplinkos (daiktų interneto) sąvoka, leidžianti žmonėms ir prijungtiems įrenginiams sąveikauti ir palengvinanti informacijos dalinimąsi. Visgi vidinio suderinamumo, saugumo ir privatumo problemos kelia daug klausimų dėl išmaniosios aplinkos ir išmaniųjų įrenginių, todėl susidaro papildomų kliūčių, bet jos taip pat veikia kaip paskatinimas sukurti patikimą ir iš vidaus suderintą sistemą. Jei šios problemos ir sunkumai nebus išspręsti, išmanioji aplinka su išmaniaisiais įrenginiais, tikriausiai, nesulauks didelio populiarumo, todėl visas potencialas gali būti prarastas.

2.2. Išmaniosios aplinkos lygmenys ir galimos grėsmės

Bendrai išmaniosios aplinkos sistemą ir jos įrenginius galima visiškai perteikti ir aprašyti trimis pagrindiniais lygmenimis [3]: jutiminis, perdavimo ir programinis.

1 lentelė. Tipiškos išmaniosios aplinkos grėsmės

Lygmuo	**Pagrindinės grėsmės**
Jutiminis lygmuo	Aparatinės įrangos klastojimas
	Kenkėjiško mazgo įrengimas
	Kenkėjiško kodo įterpimas
	Miego režimo išjungimo išpuolis
	WSN (belaidžio jutiklių tinklo) mazgų interferencija triukšmu
	RFID įrenginių interferencija radijo bangomis
Perdavimo lygmuo	Perdavimo srauto analizės išpuoliai
	RFID šnipinėjimas
	RFID neautorizuota prieiga
	Visų signalų nukreipimas į vieną tašką
	Ataka „Žmogus viduryje“
	Maršruto parinkimo informacijos ataka
Programinis lygmuo	„Phishing“ išpuoliai
	Virusai, kirminai, Trojos arkliai, šnipinėjimo programos
	Aptarnavimo perkrovos ataka (DoS)
	Programinės įrangos pažeidžiamumas

Kiekvienas šis sisteminis lygmuo pasižymi savitomis technologijomis, kurios turi savo trūkumų ir gali susilpninti saugumą. Toliau kiekvieno lygmens ypatybes ir saugumą paanalizuosime atskirai.

Jutiminis lygmuo

Šis lygmuo yra susijęs su fiziniais išmaniosios aplinkos jutikliais, kurie renka duomenis ir juos apdoroja skirtingomis technologijomis, pavyzdžiui, RFID (identifikacija radijo dažniu), WSN (belaidžių jutiklių tinklas), RSN (RFID jutiklių tinklas) ir GPS. Šiam lygmeniui priklauso įvairūs jutikliai ir aktyvatoriai, atliekantys įvairiausius matavimus (pvz., temperatūros, greitėjimo, drėgmės ir t. t.) ir vykdantys tokias funkcijas kaip vietos koordinačių nustatymas. Dėl ribotų mazgo galimybių ir organizuotosios struktūros pagrindiniai šio lygmens saugumo pavojai yra tokie:

- **Aparatinės įrangos klastojimas** [4]. Išpuoliai prieš aparatinę įrangą yra dažniausiai pasitaikantys jutiminiame lygmenyje. Jutiminį lygmenį iš esmės sudaro WSN, RFID, „Zigbee“ ir kitokio tipo jutikliai. Išpuolio sumanytojas turi būti arti tinklo arba tinklo mazgų. Tada gali bandyti pakeisti aparatinės įrangos dalis arba visiškai pakeisti mazgą. Tokiu būdu išpuolio sumanytojas gali sužinoti visą informaciją apie tą tinklą, įskaitant maršruto parinkimą, komunikacinį raktą, kriptografijos raktą, radijo bangų perdavimo raktą, ir sukelti grėsmę visam tinklui.
- **Kenkėjiško mazgo įrengimas**. Išpuolio sumanytojas gali įrengti suklastotą mazgą tarp tinklo mazgų ir taip gauti prieigą prie tinklo, kontroliuoti tinkle perduodamą duomenų srautą. Netgi galima priversti mazgą nebeperduoti tikrųjų duomenų. Taip sugadinamas ištisas tinklas.
- **Kenkėjiško kodo įterpimas**. Išmaniosios aplinkos tinklo mazgui taip pat galima pakenkti įterpus kenkėjišką kodą. Mazguose dažnai pasitaiko DoS tipo atakų ir virusų.
- **Miego režimo išjungimo išpuolis.** Išmaniosios aplinkos tinklo mazgai dažniausiai maitinami vienkartinėmis baterijomis. Mazgai yra užprogramuoti veikti budėjimo režimu, kai jais nesinaudojama, kad baterija ilgiau neišsektų. Visgi išpuolio sumanytojas neleidžia mazgui įjungti budėjimo režimo, todėl greitai išeikvojamos jo baterijos ir jis nustoja veikti.
- **WSN (belaidžio jutiklių tinklo) mazgų interferencija triukšmu.** Belaidžių jutiklių tinklui naudojami radijo bangų signalai. Galima sutrikdyti jų darbą, tinklu siunčiant triukšmo signalus arba slopinant WSN signalus. Tai neleidžia mazgams tarpusavyje komunikuoti.
- **RFID įrenginių interferencija radijo bangomis**. RFID veikimas pagrįstas radijo signalais. Šiuo atveju kenkėjiškam asmeniui nereikia slopinti signalų. Užtenka vien tik tinkle siųsti triukšmo signalus, kad mazgai nustotų veikti.

Perdavimo lygmuo

Šio tipo atakos nukreipiamos į išmaniosios aplinkos sistemos tinklą. Atliekant tokius išpuolius, nereikia būti fiziškai arti tinklo.

- **Perdavimo srauto analizės išpuoliai**. Belaidžio perdavimo tinkle gali būti šnipinėjama siekiant sužinoti konfidencialios informacijos. Tokiais atvejais programišiai pirmiausiai sužino su tinklu susijusią informaciją, šnipinėdami duomenų paketus arba nuskaitydami prievadus. Tada suruošia išpuolį dėl konkrečiai dominančios informacijos.
- **RFID šnipinėjimas**. RFID šnipinėtojas nusitaiko į RFID signalus, kad galėtų gauti prieigą prie informacijos, saugomos RFID žymeklyje. Kai signalas šnipinėjant sužinomas,

programišius gali tuo pasinaudoti ir perduoti savo signalą su pirminiu ID. Taip gaunama visiška prieiga prie sistemos.

- **RFID neautorizuota prieiga.** Kadangi nėra saugaus tapatybės nustatymo RFID sistemose, bet kas gali gauti prieigą prie žymenų. Dėl to galima lengvai žymenis manipuliuoti.
- **Visų signalų nukreipimas į vieną tašką.** Per ataką visi signalai iš belaidžių jutiklių tinklo mazgo nukreipiami į tą patį tašką. Dėl to sumažėja saugumas. Paketai gali būti nepristatyti gavėjui.
- **Ataka „Žmogus viduryje“.** Išpuolio sumanytojas tinklu sutrikdo du jutiklių mazgus, kad gautų prieigą prie privačios informacijos. Taip pakenkiama mazgų privatumui. Tokia ataka gali būti atliekama per išmaniosios aplinkos komunikacijos protokolą, todėl programišiui nereikia būti toje vietoje, kur įrengtas tinklas.
- **Maršruto parinkimo informacijos ataka.** Tai momentinės atakos, per kurias priešiškas asmuo, sufalsifikuodamas arba pakeisdamas maršruto parinkimo duomenis, gali sutrikdyti sistemos darbą ir suaktyvinti maršruto parinkimo ciklus, suteikti arba atmesti leidimus, siųsti klaidinančius pranešimus apie klaidas ir t. t.

Programinis lygmuo

Kompiuteriuose arba išmaniuosiuose įrenginiuose su „Android“ operacine sistema daug žalos gali padaryti išpuoliai, orientuoti į programinę įrangą. Kenkėjui pasinaudojus Trojos arkliais, kirminais, virusais, šnipinėjimo programomis ir kenkėjiškais kodais, sistema gali pradėti blogai veikti, gali būti suklastoti duomenys arba neteikiamos paslaugos.

- **„Phishing“ išpuoliai**. Kai vykdomi sukčiavimo apsimetant (phishing) išpuoliai, išpuolio organizatorius per tam tikrą svetainę arba el. paštu gali sužinoti naudotojo tapatybę ir kitus konfidencialius duomenis.
- **Virusai, kirminai, Trojos arkliai, šnipinėjimo programos.** Jei išmanusis įrenginys su operacine sistema užkrečiamas šiomis žalingomis programomis, duomenys gali būti klastojami, prarandami arba sistema gali neatlikti savo funkcijų.
- **Aptarnavimo perkrovos ataka (DoS)**. Per šias atakas stengiamasi nukreipti per didelį srautą į sistemas arba tinklus, kad nukentėjusiojo asmens techninė įranga nepajėgtų jo apdoroti, todėl teisėtam asmeniui būtų neįmanoma ja naudotis.
- **Programinės įrangos pažeidžiamumas.** Pažeidžiamumų atsiranda dėl nestandartinio kodo, kurį gali parašyti programuotojai. Ypač dėl buferio perpildymo. Juo pasinaudoję programišiai gali sustabdyti sistemos darbą.

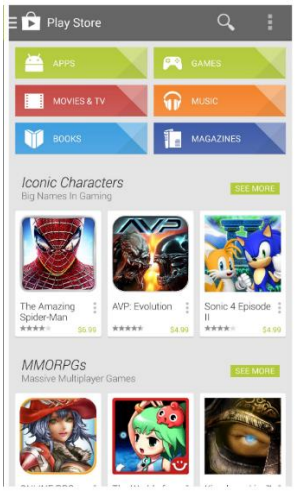
2.3. „Android“ operacinė sistema ir ją naudojantys įrenginiai

„Android“ operacinė sistema

„Android“ šiuo metu yra viena populiariausių išmaniųjų įrenginių operacinių sistemų [5]. „Android“ yra tokia mobiliųjų įrenginių operacinė sistema, kurią sukūrė „Google“ įmonė „Linux“ branduolio pagrindu. Ji iš esmės pritaikyta lietimu valdomiems mobiliesiems įrenginiams, pavyzdžiui, išmaniesiems telefonams ir planšetiniams kompiuteriams. „Android“ naudotojo sąsaja valdoma liečiamaisiais gestais: braukiant, bakstelint ir suspaudžiant ekrane. Taip pat ekrane atsiranda virtualioji klaviatūra, kuria įvedamas tekstas. „Android“ programėlės kuriamos „Java“ („.java“ failai) programavimo kalba, o kompiliuojamos JVM bitkodu („.class“ failai), kuris po to perleidžiamas

„Android Dalvik Virtual Machine“ (DVM) („.dex“ arba „.odex“). „Dalvik“ failus sudaro JVM baitkodas ir tam tikros JAR bibliotekos. Galiausiai, DEX failas ir bet kokie susiejęs resursai kaip paveikslėliai yra suarchyvuojami į APK failą (vadinamą „Android“ paketu). Šiuos APK galima atsisiųsti ir įdiegti „Android“ įrenginyje. „Android“ programėlių parduotuvėse šiuo metu yra nesuskaičiuojama galybė programėlių, kurias „Android“ naudotojai gali patogiai atsiųsti ir įdiegti.

Visgi nemažai programėlių gali būti pažeidžiamos dėl informacijos nutekėjimo, nes jos gali pasidalinti tokiais naudotojo duomenimis kaip vietos koordinatės, kontaktai ir t. t. Daug asmeninių duomenų būna socialinių tinklų ir bankininkystės programėlėse. Kad būtų saugiau, turi būti taikomi tam tikri stebėjimo ir kontroliavimo metodai, taip pat duomenys turi būti saugiau laikomi ir perduodami.

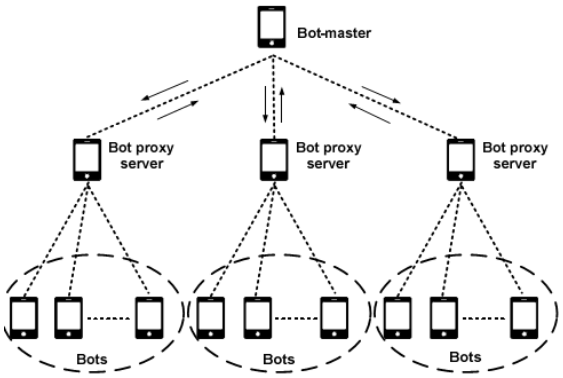


2.1 pav. „Android“ programėlių parduotuvė

Išmaniojo įrenginio su operacine sistema ir tinklų pažeidžiamumas

Išmaniosios aplinkos vizija yra tokia, kad kiekvienas fizinis objektas turėtų virtualųjį komponentą, galintį atlikti tam tikras funkcijas. Prie išmaniųjų aplinkų taip pat prijungiami tokie išmanieji įrenginiai kaip išmanieji telefonai, planšetiniai kompiuteriai [6]. Visgi į bendrą tinklą sujungto namo pažeidžiamumas pasireiškia tada, kai toks tinklas patenka į blogas rankas. Naudojant išmanųjį elektros tiekimo tinklą, galima per išpuolį atjungti elektros tiekimą. Visai įmanoma, kad galybė į bendrą tinklą sujungtų išmaniųjų įrenginių gali tapti kibernetinių atakų židiniu. O jei prie šio tinklo būtų prijungti robotai, tai išmaniosios erdvės naudotojams ir net visai nacionalinei infrastruktūrai gali užsibaigti fatalinėmis pasekmėmis.

Pagal tam tikrą statistiką buvo nustatyta, kad vidutiniškai žmogus turi 3 prie interneto prijungiamus išmaniuosius įrenginius. Tai gali būti išmanieji telefonai arba planšetiniai kompiuteriai. Dėl tokios išmaniųjų įrenginių gausos ir jų evoliucijos bei galingų procesorių, jie gali tapti tinkamais objektais kenkėjiškam „botnet“ tinklui. Tokį mobiliųjų įrenginių tinklą sudaro išmanieji prietaisai, į kuriuos buvo įsilaužta, ir jie gali būti valdomi nuotoliniu būdu.



2.2 pav. Mobiliųjų įrenginių „botnet“ tinklas

Užkrėstų kompiuterių tinklai yra dažniausiai pasitaikantys objektai, organizuojant saugumo atakas, o mobiliųjų įrenginių tinklai yra rečiau pasitaikantys ir laikomi ne tokiais pavojingais. Bent jau kol kas. Taip yra dėl to, kad mobilieji įrenginiai pasižymi ribota akumuliatorių talpa, resursais ir prisijungimo prie interneto apribojimais. Todėl ir tyrimų dėl šių įrenginių atlikta mažiau. Visgi toks požiūris gali pasikeisti, nes mobiliųjų įrenginių naudojimas itin auga. Milijonai žmonių naudojasi išmaniaisiais įrenginiais, nes jie pasižymi puikiomis kompiuterį pavaduojančiomis savybėmis, praktiškumu ir turi prieigą prie interneto. Bet blogiausia tai, kad išmaniuosiuose įrenginiuose dažniausiai yra gausu asmeninių duomenų arba duomenų apie darbą ar įmonę, taip pat tokių duomenų, kurie naudojami internetiniai bankininkystei. Didelio populiarumo sulaukę atvirojo šaltinio operacinės sistemos išmanieji įrenginiai kaip „Android“ ir jiems trečiųjų šalių kuriamos programėlės suteikia puikiausių galimybių kenkėjiškų programų kūrėjams. Nekyla abejonių, kad laikui bėgant išmanieji įrenginiai taps dar labiau viliojančiu kibernetinių nusikaltėlių taikiniu.

Silpniausia IT saugumo sistemos grandis – žmogus. Išmaniųjų įrenginių naudotojai laikosi nuomonės, kad viskas veiks taip, kaip priklauso. Jie pasitiki įrenginio gamykliniais nustatymais ir nesivargina analizuoti sudėtingų naudojimo instrukcijų. Visgi programinės ir aparatinės įrangos teikėjai turėtų atsižvelgti į tokį požiūrį ir prisiimti atsakomybę ir pareigą apsaugoti tinklą. Bet vos tik naudotojas prijungia „Android“ įrenginį prie savo kompiuterio USB kabeliu, SD kortelės turinį galima nuskaityti, įrašyti jį arba pašalinti. Jei kompiuteryje naudotojui nežinant buvo įdiegta kenkėjiška programinė įranga, taip ji gali persikelti į „Android“ įrenginį ir atvirkščiai. Bus gauta prieiga prie asmeninių duomenų [7]. Tai tik vienas variantas iš galybės galimų.

Nors seniau mobilieji įrenginiai laikyti saugiais, viskas pasikeitė, juose pradėjus naudoti operacinę sistemą. Šiais laikais mobilieji įrenginiai gali būti naudojami sveikatos priežiūrai. Toks sveikatos priežiūros įrenginys gali būti prijungtas prie namų tinklo, kuriuo belaidžiu būdu galima perduoti informaciją ligoninėms arba kitoms suinteresuotoms institucijoms. Dauguma gamintojų nededa jokių papildomų pastangų, kad užtikrintų, jog šie įrenginiai veiktų saugiai. Bet jei į juos kas nors įsilaužtų, nutekėtų ne tik įrenginio naudotojo asmeninė informacija, bet išpuolio sumanytojas galėtų pakeisti net ir įrenginio parametrus, o tai gali turėti milžiniškos neigiamos įtakos sveikatai. Jau buvo pademonstruota, kad įmanoma įsilaužti į širdies stimuliatorių ir nuskaityti visus įrenginyje saugomus duomenis.

Kenkėjiškoms programėlėms būdinga ir tai, kad jos gali sumažinti procesoriaus darbinį greitį. Taip pat sumažėja įrenginio komunikacijos greitis internetu. Dėl kenkėjiškų programėlių gali itin greitai išsieikvoti akumuliatorius, ypač kai labai įkyriai rodomos reklamos. Priklausomai nuo pobūdžio, kenkėjiškos programėlės gali šnipinėti atliekamus veiksmus su įrenginiu, perduoti

asmeninius duomenis nuotoliniam serveriui, įdiegti Trojos arklių, leidžiančių programišiams gauti nuotolinę prieigą. Galima paminėti ir tai, kad užkrėstas įrenginys be savininko žinios gali skambinti itin brangiai apmokestinamais numeriais, siųsti žinutes. Esant tokiam žalingam poveikiui, įrenginys gali strigti, išsijunginėti arba nuolat persikrauti.

„Android“ operacinės sistemos pažeidžiamumas, remiantis statistika

„Android“ operacinė sistema nukenčia nuo įvairiausių išpuolių organizatorių [8]. Lyginant su kitomis platformomis, „Android“ įrenginiai gerokai lenkia kenkėjiškų programų skaičiumi. „SophosLabs“ nuo 2010 m. užregistravo daugiau nei 1,5 milijonų „Android“ kenkėjiškų programų pavyzdžių. Remiantis šaltinio [9] duomenimis, vidutiniškai 1,4 milijardo įrenginių naudoja „Android“ operacinę sistemą. 90 % naudoja pasenusią operacinės sistemos versiją. Išpuolio organizatoriai pasinaudoja operacinės sistemos saugumo spragomis, siekdami gauti prieigą prie įrenginio. Jei šias saugumo spragas aptinka operacinės sistemos kūrėjai, jie jas ištaiso ir išleidžia atnaujinimus. Bet jei operacinė sistema nėra atnaujinama, įrenginiu gali pasinaudoti programišiai. Rastas nemažas skaičius klaidų „Stagefright“ bibliotekose, kurios naudojamos programėlėms. Šie pažeidžiamumai yra itin grėsmingi, nes leidžia nuotoliniu būdu paleisti kodą nieko neįtarančio asmens telefone, vien tik nusiuntus specialiai paruoštą MMS žinutę. Tokie nuotolinės prieigos įrankiai (RAT), kokie naudojami su „Stagefright“, lengvai įsigyjami internete. Dėl tokio gausaus „Android“ įrenginių paplitimo net ir santykinai žemas užkrečiamumo lygis prilygsta dideliam užkrėstų įrenginių skaičiui.

Dar viena priežastis, dėl kurios „Android“ įrenginiai yra viliojantis objektas – tas faktas, kad naudotojai gali atsisiųsti programėlių iš trečiųjų šalių arba jas tiesiogiai įkelti į įrenginį. Nėra būtinybės naudotis vien tik „Play Store“. Kaip teigia šaltinis [10], iš 150 milijonų programėlių, kurios buvo skenuotos „Play Store“ per pastaruosius 3 mėnesius, 37 milijonuose aptikta kenkėjiškumo požymių. 2015 m. ketvirtajame ketvirtyje dar aptikta 2,4 milijono naujų kenkėjiškų programų. Visgi „Google“ pasižadėjo kas mėnesį išleisti atnaujinimus. „iOS“ naudotojams yra daug mažesnė tikimybė, kad atsitiktinai įdiegs kenkėjišką programėlę į savo įrenginį, nes „Apple“ reikalauja, kad jos naudotojai siųstųsi programėles tik iš oficialios programėlių parduotuvės „Apple Store“ [11].

Kaip išvengti kenkėjiškų programėlių

Galimybė išvengti kenkėjiškų programėlių ne visada yra lengva užduotis. Visgi yra keletas metodų arba strategijų, kurie sumažina užsikrėtimo tikimybę. Vienas iš lengviausių metodų – atsisiųsti tik patikimas programėlės iš oficialios programėlių parduotuvės „Play Store“. Kadangi „Play Store“ parduotuvėje taikomos tikrinimo procedūros, su kuriomis programėlės susiduria, kad patektų į „Play Store“, tai itin apsunkina išpuolio sumanytojus.

„Android“ įrenginiuose numatyta galimybė įkelti programėles per USB jungtį. Bet tai nėra geras sumanymas. Jei prijungiamas užkrėstas USB kaupiklis prie „Android“ įrenginio, jis gali persikelti į įrenginį. Reikia atsižvelgti į tai, kad tokia programėlė gali nebūti atsisiųsta iš „Play Store“, kurioje atliekama griežta patikra. Dėl to „Mcafee“ rekomenduoja išjungti telefono gebėjimą automatiškai aktyvinti MMS žinutes arba leidimą įdiegti programėles ne iš oficialiosios parduotuvės „Play Store“. Dar labai svarbu yra nespausti nuorodų iš nežinomų šaltinių. Jos dažniausiai sutinkamos su elektroninio pašto virusais. Jei paspausite, programišiai gali tuo pasinaudoti ir įdiegti virusą į jūsų įrenginį.

Nors šios strategijos yra naudingos apsisaugant nuo piktavališkų programėlių, jos nėra absoliutus sprendimas nuo visų pavojų. Reikia išvystyti metodus, kurie aptiktų jas, kol jos dar nėra įdiegtos.

2.4. Kenkėjiškų kodų paieška „Android“ operacinėje sistemoje

Statinis ir dinaminis aptikimo metodas

Tyrėjų bendruomenė įvardija du [12] metodus, skirtus aptikti kenkėjiškas programėles: statinė analizė ir dinaminė analizė. Statinis metodas veikia programėlės dvejetainio kodo išskaidymo principu. Tokiu atveju, nepaleidus programėlių, yra tikrinamas kodas ir susiję metaduomenys. Visgi statinės analizės metodai, kuriems naudojamos tradicinės statinės signatūros, gali neaptikti naujausios kenkėjiškos programinės įrangos, kuri dar mažai žinoma, kurių signatūros dar neturima. Šiai problemai išspręsti egzistuoja kitas kenkėjiškų programėlių aptikimo metodas. Jis vadinamas dinamine analize. Kai programėlė veikia, šiuo metodu galima nustatyti, ar pastebimi kenkėjiškos programėlės požymiai. Galima veiksmingai naudotis dinaminės analizės metodais norint aptikti programėles, kurių signatūra dar nežinoma. Siekiant sumažinti žalą, kurią gali padaryti kenkėjiškos programėlės, reikėtų naudotis dinamine analize.

Tiriamajame darbe [12] yra siūlomas „Linux“ įrankis „Strace“. Juo galima naudotis, kai norima stebėti tam tikros programos signalus, kuri jau yra paleista ir naudojasi proceso ID. Kiekvienam procesui priskiriamas PID identifikacijos numeris, kai jis „Linux“ operacinėje sistemoje sukuriamas. PID visada sudaro teigiamas sveikasis skaičius ir kiekvienam procesui priskiriamas unikalus PID. Kad būtų gautas PID, būtina paleisti programėlę. „Android“ programėlę galima paleisti per ADB sistemą. Kai sužinomas programėlės PID, galima paleisti programėlę naudojant įrankį „Monkey. Tokiu atveju sukuriamas imitacinis naudotojo veiksmų srautas – paspaudimai, palietimai arba gestai.

Rekomenduojama kiekvieną programėlę paleisti atskirai, naudojant „Android VM“ ir „Monkey“ įrankį. Per vieną minutę turėtų būti generuojama apie 500 gestų. Delsa tarp kiekvieno veiksmo siekia 0,5 s. Kad būtų galima atlikti statistinę analizę, verta surinkti „Strace“ gautus rezultatus ir sudėti į vieną duomenų lapą.

„Android“ OS saugumo įvertinimas mašininio mokymosi algoritmu

„Android“ sistemos naudoja leidimais pagrįstą mechanizmą [13], suteikiantį daugiau saugumo ir apribojantį programėles. Įdiegiant ir paleidžiant „Android“ programėles, reikia įvairiausių leidimų, tokių kaip nuskaityti kontaktus, interneto prieiga, prieiga prie nuotraukų, prieiga prie žinučių ir t. t. Diegimo priemonės paketas parodo leidimų sąrašą, per kurį naudotojas gali suteikti arba nesuteikti šiuos leidimus. Jei naudotojas suteikia tokį leidimą, tada programėlė gali nevaržomai naudotis informacijos šaltiniu arba prieiga, be to, vėliau galimybė atšaukti šiuos leidimus yra prieinama per nustatymų skyrių.

Šulcas ir kiti [14] buvo pirmieji, kurie pasiūlė naudoti mašininio mokymosi algoritmą „Bayes“, kad būtų galima aptikti kenkėjiškas programėles. Jie net pranešė, kad rezultato sėkmingumas lygus 97,11 %. Firdausi ir kiti [15], ieškodami kenkėjiškų programėlių, pasinaudojo J48 sprendimų medžio mašininio mokymosi algoritmu. Jų praneštas sėkmingumo rezultatas siekė 97 %. Kaip pastebėta, sprendimų medžiai pasižymi aukštu veiksmingumu, klasifikuojant tinklo saugos problemas.

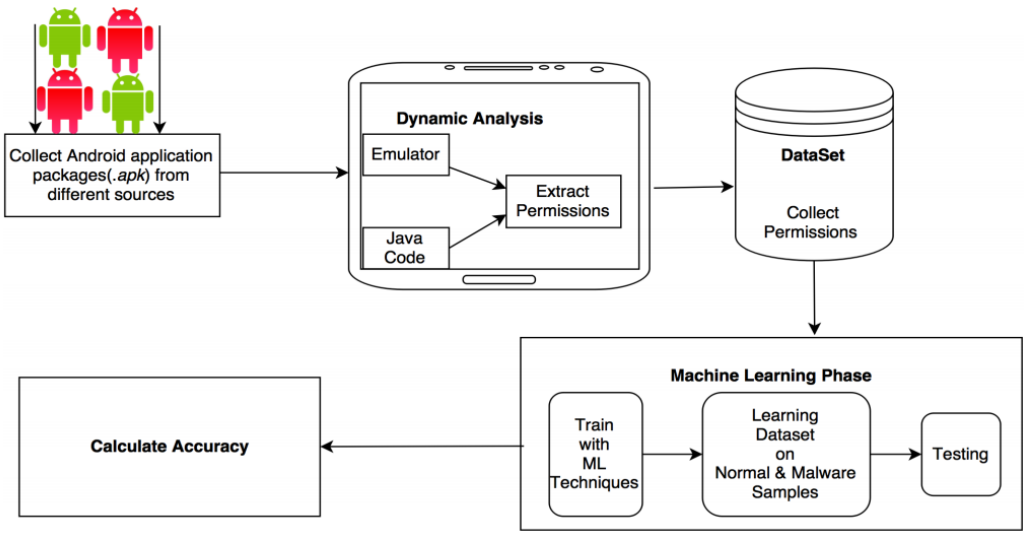
„Android“ programėlę sudaro į APK failą sudėti programiniai kodai [16]. Kiekvienoje programėlėje, kai ji išarchyvuojama, yra failas, vadinamas „Androidmanifest.XML“. Šiame faile yra leidimų, kurių programėlė norės gauti, sąrašas. Toliau paveikslėlyje pateikiamas „Androidmanifest.XML“ failo pavyzdys.

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<uses-permission android:name="com.android.vending.CHECK_LICENSE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.MODIFY_PHONE_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.READ_LOGS" />
```

2.3 pav. „AndroidManifest.xml“ failo pavyzdys

Kad būtų galima naudotis mašininio mokymosi algoritmais klasifikuojant „Android“ kenkėjiškas programėles, pirmiausia reikia išskleisti visų programėlių leidimų duomenis. Duomenų bazė suformuojama įrašant 1, jei „Androidmanifest.XML“ faile yra leidimas. Jei ne, tada įrašoma 0 [17]. Toliau nurodytais veiksmais išskiriami leidimų duomenys:

- Reikia atsisiųsti originalias „Android“ programėles iš „Google Play Store“.
- Surinkti kenkėjiškos programėlės duomenis iš standartinių duomenų lapų.
- Išgauti „Androidmanifest.XML“ failą iš programėlių.
- Sužinoti leidimus kiekvienai programėlei ir pasiruošti „Excel“ duomenų lapą.
- Konvertuoti duomenis į formatą WEKA.ARFF.



2.4 pav. Mašininio mokymosi algoritmo taikymas

Toliau aprašoma keletą mašininio mokymosi algoritmų.

Naive Bayes. Šiam metodui naudojama sąlyginio nepriklausomumo sąvoka. Per mokymo etapą reikia atskirai apsvarstyti kiekvieną savybę kiekvienoje klasėje. Per testavimo etapą apskaičiuojama sąlyginė tikimybė.

Decision Tree. Naudojamas diskretinių verčių tikslinės funkcijos metodas, kuriuo išmokta funkcija atvaizduojama kaip „jei taip, tada...“ taisyklių rinkiniu. „Decision Tree“ (sprendimų medis) susideda iš mazgų, kurie sudaro medį. Mazgas vadinamas šaknimi.

Random Forest. Susideda iš medžio struktūros klasifikatorių $\{h(X, \Theta_k), k = 1, 2, 3, \ldots\}$, kur $\{\Theta_k\}$ yra nepriklausomas identiškai pasiskirstęs atsitiktinis vektorius. Kiekvienas medis pateikia savo balsą už populiariausią klasę įvestyje X.

k-star. „k-star“ yra pasyvusis mokymosi metodas. Mokomųjų duomenų apibendrinimas atidedamas tol, kol nėra sistemai pateiktos užklausos. Šis būdas yra priešingybė aktyviajam mokymuisi, kur sistema mėgina apibendrinti mokymosi duomenis prieš gaunant užklausas.

2.5. Pavojai „Android“ įrenginiuose saugomiems duomenims

„Google“ komandos paruoštoje 2017 m. ataskaitoje [18] siekiama suvienodinti terminus apie potencialiais žalingas programėles. Joje šios žalingos programėlės aprašomos ir įvardijami, kokie kyla pavojai duomenų saugumui.

Šnipinėjimo programėlės

Tai tokios programėlės, kurios perduoda bet kokią jautrią (slaptą) informaciją iš įrenginio be naudotojo sutikimo, be to, naudotojas nėra įspėjamas, kad tai vyksta. Dažniausiai tokie duomenys perduodami trečiajai šaliai:

- Adresų knygelės įrašai
- Nuotraukos arba failai
- El. pašto laiškų informacija
- Skambučių žurnalas
- SMS
- Interneto naršymo istorija arba naršyklės žymelės

Tokia informacija gali būti naudojama nepageidaujamų laiškų siuntimui ir t. t.

„Phishing“ (sukčiavimas apsimetant)

Pagrindinis „phishing“ atakos tikslas yra išvilioti tokius slaptus duomenis kaip prisijungimo duomenys prie bankinės sistemos, kredito kortelės numarį, socialinių tinklų prisijungimo duomenis. Tokia programėlė siekia įtikinti naudotoją, kad ji yra patikima, bet iš tikrųjų slaptus duomenis perduoda trečiosioms šalims, kur jie gali būti neteisėtai panaudojami.

Trojos arkliai

Tokios programėlės, kurios gali atrodyti nepavojingos, bet jos naudotojui nežinant atlieka jam kenksmingus veiksmus. Kitaip sakant, trojanai yra užmaskuojami. Pavyzdžiui, žaidžiant atsisiųstą

žaidimą, programėlė be naudotojo žinios gali siųsti SMS žinutes žmonėms iš adresų knygelės arba brangiai apmokamais numeriais.

Išpirkos reikalaujančios programėlės

Tokios programėlės blokuoja prieigą prie visų išmaniojo įrenginio duomenų arba prie dalies iš jų ir reikalauja išpirkos, kad prieiga būtų sugrąžinta. Jos tuo tikslu net gali užšifruoti duomenis ir gali veikti administratoriaus teisėmis, todėl naudotojui tokios programėlės nepavyks pašalinti.

2.6. Jautrios informacijos apsaugojimas nuo grėsmių

Paprasčiausias apsaugos būdas

Patariama prieigą prie išmaniųjų įrenginių apsaugoti slaptažodžiu. Ypač jei jie išsinešami iš namų. Galima net įdiegti sekimo programėlę, kuri leidžia duomenis ištrinti nuotoliniu būdu, jei įrenginys patenka į blogas rankas.

Reikėtų susikurti atsargines tų duomenų kopijas, kurių labai svarbu neprarasti tokiais išskirtiniais atvejais. Atsargines kopijas galima kurti tiek debesyje, tiek ir kompiuteryje ar išimamoje atminties kortelėje.

Programėlės ir leidimai

Rekomenduojama kuo mažiau įsidiegti programėlių, nes taip bus suteikiama mažiau nereikalingų leidimų. Taip pat diegiant programėlę reikėtų gerai pagalvoti, ar jos prašomi leidimai jai yra būtinas. Be reikalo leidus naudoti geografines koordinates arba adresų knygelę, galima turėti neigiamų padarinių.

Atsargiai elgtis su „Wi-Fi“

Mobilusis duomenų perdavimo ryšys gali brangiai kainuoti, ypač lankantis užsienyje. Viešieji „Wi-Fi“ prisijungimo prie interneto taškai yra puiki alternatyva, bet tik tada, kai jie patikimi. Vienas iš sprendimų tokiais atvejais – naudotis VPN (virtualiu asmeniniu tinklu), kuriuo duomenys šifruojami, todėl „Wi-Fi“ šnipinėtojai netenka galimybės nuskaityti duomenų. „Android“ įrenginiuose yra integruotas VPN klientas. Jei nėra galimybės naudoti VPN kliento, tada reikėtų jungtis tik prie svetainių su saugiu HTTP protokolu (HTTPS).

Naudotis įvairesniais paslaugų teikėjais

Patogiausia „Android“ sistemoje naudotis „Google“ paskyra. Bet reikėtų nepamiršti, kad kuo daugiau paslaugų yra valdoma viena paskyra, tuo lengviau piktavaliams. Įsilaužus į vieną paskyrą, galima gauti prieigą prie įvairiausių duomenų. Geriau naudotis, pavyzdžiui, „Microsoft OneDrive“ debesiu, „Firefox“ naršykle ir t. t.

Nesakyti teisybės

Kadangi internetinis verslo modelis yra pagrįstas stebėjimu ir norint naudotis tam tikromis paslaugomis nėra kitos išeities, kaip tik su tuo sutikti, specialistai pataria ne visada nurodyti tikslius duomenis. Todėl kai prašoma nurodyti augintinio vardą, geriau kaip slaptąjį atsakymą įrašyti, kad jo vardas yra H2O. Bus ir lengva prisiminti, ir sunku atspėti.

2.7. „Android“ informacijos saugojimo priemonės

„Android“ informacijos saugojimo atminties tipai

Dažniausiai kylantis saugumo klausimas dėl „Android“ programėlių ir duomenų – ar įrenginyje išsaugoti duomenys yra prieinami kitoms programėlėms? Yra trys pagrindinės vietos, kur duomenys išsaugomi:

- Vidinė atmintis
- Išorinė atmintis
- Programėlės kūrėjo serveryje / debesyje

Pagal numatymą failai, kurie yra sukuriami vidinėje atmintyje, prieinami tik atitinkamai programėlei. „Android“ taiko apsaugos priemones ir dažniausiai to pakanka.

Jei failai sukuriami išorinėje atmintyje, tokioje kaip SD kortelės, jos yra globaliai nuskaitomos ir įrašomos. Taip pat išorinę atminties priemonę galima išimti ir modifikuoti bet kuria kita programėle, todėl išorinėje atmintyje geriau nesaugoti slaptų duomenų.

Jei duomenys saugomi programėlės kūrėjo serveryje, ten taikomas struktūrizuotas saugojimo mechanizmas, kur duomenų prieinamumas galimas tik skirtajai programėlei. Jei duomenys eksportuojami, galima suteikti prieigą kitoms programėlėms.

„Android“ skirstiniai ir katalogai

„Android“ operacinėje sistemoje duomenims saugoti naudojami skirstiniai ir katalogai [19]. Toliau pateikiamas skirstinių pavyzdinis sąrašas.

Partition	Explanation
/boot	kernel & Co.
/cache	app cache
/data	user data partition[1]
/data/data	app data[1]
/dev	devices[2]
/mnt/asec	encrypted apps (App2SD)
/mnt/emmc	internal sdcard[3]
/mnt/sdcard	external sdcard[3]
/proc	process information[2]
/recovery	used in recovery mode
/system	system ROM (read-only)

2.5 pav. Skirstiniai ir katalogai

Pagal numatymą visos programėlės saugo savo duomenis adresu */data/data* [20]. Programėlės taip pat tam gali naudoti SD kortelę, jei jos tokio leidimo paprašo, kai programėlė yra įdiegiama. Jei mus domintų kontaktinių asmenų nuotraukų saugojimo vieta, jos adresas būtų */data/data/com.android.providers.contacts/files/photos*

Kontaktų ir skambučių saugojimo vieta

Kontaktai ir skambučiai saugomi toje pačioje duomenų bazėje. Kontaktus nebūtinai įtraukia naudotojas. Jie gali būti automatiškai įtraukiami, kai per „Gmail“ išsiunčiamas el. laiškas arba kai tam tikras žmogus įtraukiamas į „Google+“. Įmanomi ir kiti būdai.

Katalogo pavadinimas: *com.android.providers.contacts*

Dominantys failai:

- /files/
  - photos/
  - profile/
- /databases/
  - contacts2.db

2.6 pav. Kontaktų katalogas

Kataloge „files“ patalpinamos visos nuotraukos, skirtos naudotojų kontaktams. Tai šiuo atveju „files“ -> „photos“ talpina kitų naudotojų nuotraukas, o naudotojo profilio nuotrauka – kataloge „profile“. Duomenų bazėje „contacts2.db“ saugoma visa informacija apie priimtus ir atliktus skambučius, taip pat įtraukiami visi kontaktai, esantys naudotojo „Google“ paskyroje. Duomenų bazėje saugomos tokios lentelės su atitinkamais duomenimis: *accounts, calls, contacts, data, deleted_contacts, groups, raw_groups*.

„Gmail“ duomenų saugojimo vieta

„Gmail“ yra el. pašto paslauga, kurią teikia „Google“. Kai įrenginys pirmą kartą nustatomas, vos tik jį įsigijus, dažnai prašoma „Gmail“ paskyros, nors ji ir nėra būtina.

Katalogo pavadinimas: *com.google.android.gm*

Dominantys failai:

- /cache
- /databases/
  - mailstore.<username>@gmail.com.db
  - databases/suggestions.db
- /shared_prefs/
  - MailAppProvider.xml
  - Gmail.xml
  - UnifiedEmail.xml

2.7 pav. El. pašto katalogas

Kataloge */cache*, kuris yra programėlių aplanke, talpinami naujausi failai, kurie buvo pridėti prie el. laiškų – tiek siųstų, tiek gautų. Šie priedai yra būtent čia išsaugomi, net jei juos naudotojas ir nėra išskirtinai atsisiuntęs.

Duomenų bazės faile *mailstore.<username>@gmail.com.db* įtraukiama įvairiausia naudinga informacija. Šios duomenų bazės lentelės: *attachements, conversations*.

„Google Chrome“ duomenų saugojimo vieta

„Google Chrome“ yra interneto naršyklė ir ji yra gamykliškai diegiama įrenginiuose su „Android“ operacine sistema. „Chrome“ duomenys įrenginyje yra unikalūs dėl to, kad jie apima ne tik mobiliojo įrenginio naudojimo duomenis, bet ir kitus įrenginius, kuriais naudotojas buvo prisijungęs prie „Chrome“. Todėl dažnai pasitaiko, kad mobiliojo „Android“ įrenginio duomenų bazėje būna duomenų, kurie gauti naršant kompiuteriu. Taip susikaupia dideli kiekiai duomenų, kuriuos gali tekti išanalizuoti, jei kažko ieškoma.

Katalogo pavadinimas: *com.android.chrome*

Dominantys failai:

- /app_chrome/Default/
  - Sync Data/SyncData.sqlite3
  - Bookmarks
  - Cookies
  - Google Profile Picture.png
  - History
  - Login Data
  - Preferences
  - Top Sites
  - Web Data
- /app_ChromeDocumentActivity/

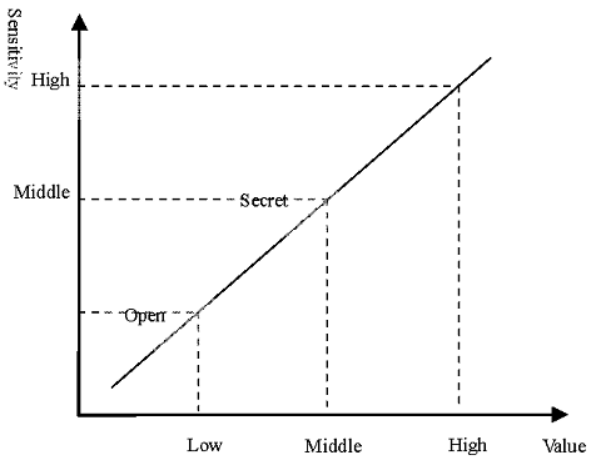
2.8 pav. „Chrome“ katalogas

Visi failai, kurie yra įtraukti į katalogą */app_chrome/Default/,* išskyrus tik tą, kuris baigiasi plėtiniu „.png“, *Bookmarks* ir *Preferences,* yra „SQLite“ duomenų bazės. Duomenų bazė *SyncData.sqlite3* pasižymi tuo, kad joje yra tų duomenų sąrašas, kurie buvo sinchronizuoti iš naudotojo paskyros įrenginyje su „Google“ serveriais.

2.8. Informacijos klasifikavimas V-S ašių metodu

Išanalizavus tam tikrus informacijos klasifikavimo metodus, nuspręsta daugiau dėmesio skirti V-S informacijos klasifikavimo metodui, nes jis labiausiai atitinka šio darbo poreikius.

Šiuo metodu informacija klasifikuojama remiantis informacijos (X) verte ir informacijos (Y) jautrumu (slaptumu) [21]. Šie kriterijai leidžia priskirti informaciją skirtingoms kategorijoms. Tokiu atveju galima taikyti skirtingas saugumo kontrolės strategijas, atsižvelgiant į skirtingas informacijos gavimo, laikymo, apdorojimo, pritaikymo ir dalinimosi kategorijas.



2.9 pav. Duomenų klasifikavimas V-S ašių metodu

Pirmiausiai informaciją reikia suskirstyti į 3 lygius vertės (X) ašyje: žema, vidutinė, aukšta. Tada reikės įvertinti informacijos jautrumą (slaptumą) Y ašyje, suskirstant jos jautrumo reikšmes į tris lygius: žema, vidutinė, aukšta. Kai bus nustatyta informacijos vertė ir jautrumas, tada konkrečiai informacijai bus galima nustatyti V-S (vertės-jautrumo) reikšmes. Tai leis informaciją suklasifikuoti.

Atsižvelgiant į visas galimas vertės ir jautrumo reikšmes, iš viso gali būti gaunamos 9 V-S reikšmės. Pagrindinės informacijos klasifikavimo reikšmės iš šių 9 būtų tokios 3: nereikšminga ir vieša = ŽŽ (žema-žema), vidutiniškai reikšminga ir vidutiniškai slapta = VV (vidutinė-vidutinė), itin reikšmingas ir itin slapta = AA (aukšta-aukšta).

2.9. Informacijos klasifikavimas pagal vertės (X) ašį

Informacijos vertę nulemia du veiksniai: informacijos naudingumas ir informacinės išlaidos. Informacijos naudingumas pasitarnauja planuojant, organizuojant, kontroliuojant ir koordinuojant įvairius veiksmus. Toliau pateikiami pagrindiniai veiksniai, į kuriuos reikėtų atsižvelgti, kai analizuojamas informacijos naudingumas:

- Informacijos patikimumas
- Informacijos kiekis
- Informacijos savalaikiškumas
- Informacijos prieinamumas
- Informacijos praktiškumas naudotojams

Jei laikytumėmės nuomonės, kad informacijos naudingumas yra linijinė funkcija, kurią sudaro įvairūs veiksniai, nulemiantys informacijos naudingumą, būtų galima pritaikyti tokią formulę:

$$[21]\ (1)\ U = f\ (u_1, u_2, u_3 \dots)$$

Jei laikytume, kad informacijos išlaidas sudaro bendrosios išlaidos, susidedančios iš informacijos gavybos, apdorojimo, perdavimo ir panaudojimo, tada jas sudarytų tokie veiksniai:

- Informacijos gavybos išlaidos
- Informacijos tarnybinės išlaidos
- Informacijos pritaikymo išlaidos
- Informacijos vienkartinio arba daugkartinio panaudojimo išlaidos

Jei laikytumėmės nuomonės, kad informacinės išlaidos yra linijinė funkcija, kurią sudaro įvairūs veiksniai, nulemiantys informacines išlaidas, būtų galima pritaikyti tokią formulę:

$$[21]\ (2)\ C = g(c_1, c_2, c_3 \dots)$$

Išanalizavus informacijos naudingumo ir informacinių išlaidų veiksnius, galima naudoti tolesnę apibendrinimo formulę, kaip pasiūlyta surasto mokslinio tyrimo ataskaitoje [21].

$$[21]\ (3)\ V = \frac{U}{C} = \frac{f(u_1,u_2,u_3\dots)}{g(c_1,c_2,c_3\dots)}$$

Autoriai savo darbe teigia, kad reikėtų normalizuoti reikšmes U ir C, ir kad jų intervalas turėtų atitikti [0, 1]. Bet šių skaičiavimų atvejų V gaunama reikšmė gali būti nuo – ∞ iki + ∞. Kad būtų patogiau skaičiuoti, reikėtų įtraukti konstantą 1 tiek prie skaitiklio, tiek prie daliklio. Tada gaunamos reikšmės intervalas siektų [0,5, 2]. Aukščiau pateiktą formulę būtų galima perrašyti į:

$$[21]\ (4)\ V' = \frac{1+f\prime(u_1,u_2,u_3\dots)}{1+g\prime(c_1,c_2,c_3\dots)}$$

2.10. Informacijos klasifikavimas pagal jautrumo (Y) ašį

Informacijos jautrumas yra toks mato vienetas, kuris įvertina praradimus ir neigiamą poveikį, kurį patiria informacijos savininkai, kai ta informacija prarandama arba paviešinama. Mažas informacijos jautrumas reiškia, kad nuostoliai ir neigiamas poveikis, kurį patiria informacijos savininkas, kai ta informacija sugadinama, prarandama arba atskleidžiama, yra nedidelis. Toliau pateikiami pagrindiniai veiksniai, nulemiantys informacijos jautrumą:

- Priklausomumo laipsnis
- Atskleidimo toleravimas
- Platinimo toleravimas
- Atkūrimo išlaidos
- Atkūrimo trukmė

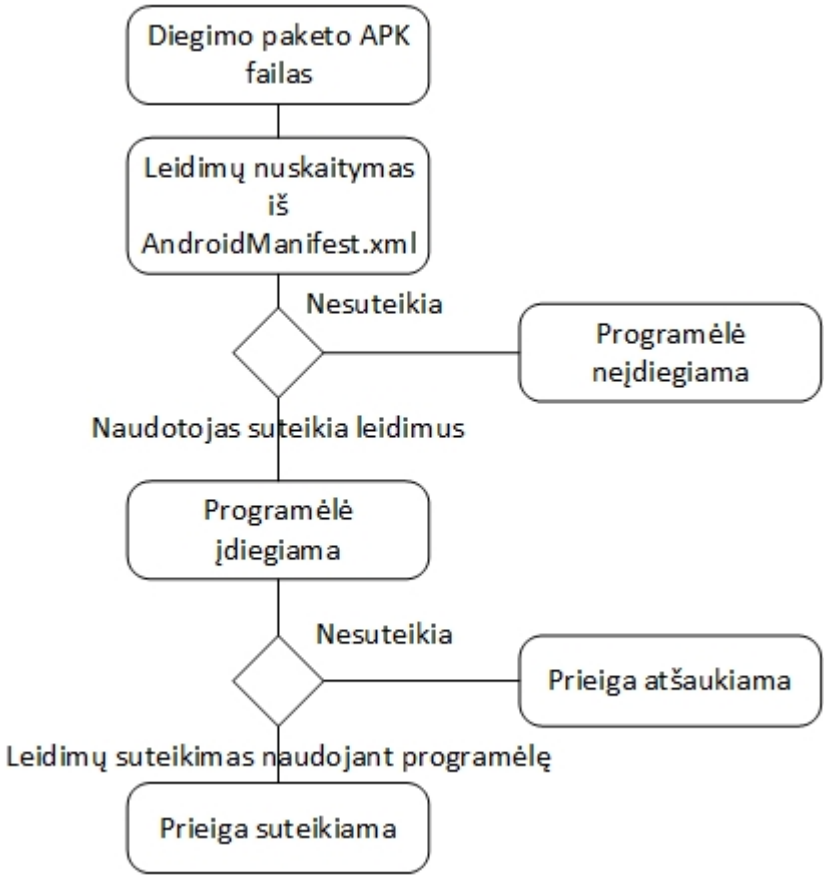
2.11. Išmaniosios aplinkos analizės išvados

1. Išanalizavus „Android“ operacinės sistemos veikimo principus, galima teigti, kad programėlės turi įvairių galimybių pasiekti konfidencialią asmeninę informaciją. Skirtumas tik tas, ar naudotojas nori, kad asmeninė informacija būtų prieinama tam tikroms programėlėms, ar nenori. Kadangi „Android“ operacinės sistemos įrenginių programėlėms naudojamas teisių suteikimo modelis, naudotojas prieš įdiegdamas tam tikrą programėlę pats gali leisti tai programėlei naudoti savo atitinkamus asmeninius duomenis: kontaktus, žinutes, naršymo istoriją, paveikslėlius ir t. t. Visgi didesnių problemų kyla tada, kai naudotojas nėra suteikęs tokių teisių, bet programėlė slaptai arba kenkėjiškai ar apgaulingai naudoja tokius duomenis. Norint sumažinti tokių atvejų tikimybę, reikėtų bent jau įdiegti operacinės sistemos ir programėlių naujinius, taip pat programėles atsisiųsti iš patikimo šaltinio.
2. Darbe, remiantis patikimais šaltiniais, išanalizuotos įrenginio vietos, kur saugoma asmeninė informacija. Šie skirstiniai ir katalogai pasiekiami, tik jei turima prieiga prie šakninio („root“) katalogo. Saugumo sumetimais operacinės sistemos kūrėjas gamykliškai nesuteikia prieigos prie šakninio katalogo.
3. Kad būtų galima užtikrinti informacijos saugumą „Android“ sistemoje, reikalingas informacijos klasifikavimas, atsižvelgiant į informacijos jautrumą (Y) ir vertę (X). Tai sudarys galimybę taikyti informacijos apsaugos metodus, priklausančius nuo informacijos jautrumo lygio.

3. IŠMANIOJO ĮRENGINIO PROGRAMĖLIŲ SAUGUMO STEBĖJIMO METODAS

3.1. „Android“ OS numatytasis programėlių leidimo kontrolės mechanizmas

Leidimų kontrolė yra esminis „Android“ programėlių saugos mechanizmas. Visos „Android“ saugumo problemos sprendžiamos per leidimus. Programėlės gali naudotis įrenginio funkcijomis tik tada, jei turi tam leidimą. Iš viso „Android“ sisteminių leidimų yra per 100 [22] ir jie skirstomi į 4 grupes: įprasti (normal), pavojingi (dangerous), signatūriniai (signature) ir specialieji (special).

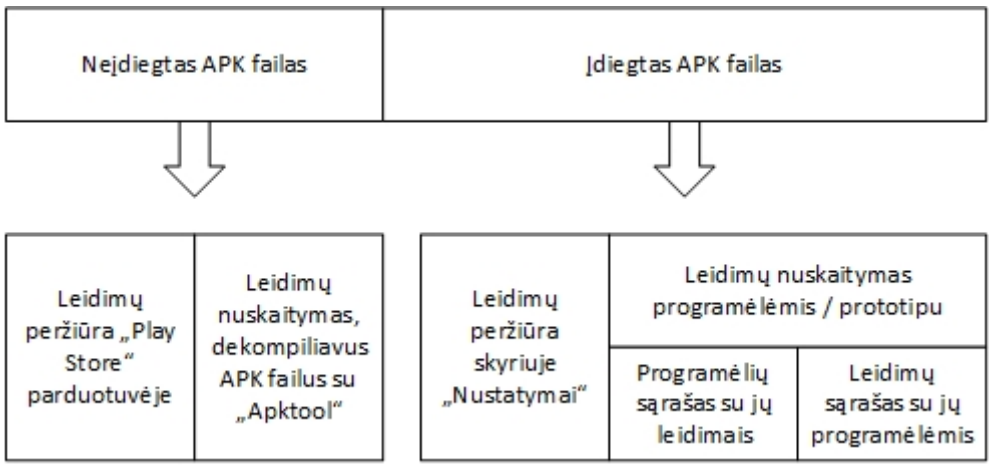


3.1 pav. Leidimų kontrolė, diegiant programėlę

Kai programėlę reikia įdiegti, „Android“ paketų valdymo sistema praneša naudotojui, kokių leidimų reikia programėlei. Jei naudotojas suteikia reikiamus leidimus, diegimas prasideda. Kitaip įdiegimas atšaukiamas. Kai programėlė paleidžiama ir naudojama arba atnaujinama, jai gali prireikti papildomų leidimų. Jei tie leidimai suteikiami, ji sėkmingai veiks. Kitu atveju gali nebeveikti.

3.2. Leidimų aptikimo ir stebėjimo metodai

Leidimų aptikimo sistemų yra tokių, kurias galima naudoti kompiuteryje, kaip „APKtool“, ir mobiliajame įrenginyje, atsisiuntus trečiųjų šalių tos paskirties programėlę arba naudojantis „Android“ OS parametrų skyriaus parinktimis. Taip pat leidimus galima sužinoti „Play Store“ parduotuvėje prieš programėlę įdiegiant. Naudojant vieną iš šių metodų, galima aptikti įdiegtos arba neįdiegtos programėlės leidimus. Bendrai vertinant, metodus aptikti ir stebėti leidimus, kurie nurodyti tyrime [23], dar papildžius, galima apipavidalinti, kaip toliau nurodyta.



3.2 pav. Leidimų aptikimo ir stebėjimo metodai

Leidimų nuskaitymą trečiųjų šalių programėlėmis / prototipu „Android“ įrenginyje dar galima skirstyti į dvi dalis pagal tai, kaip pateikiami leidimai: (1) įdiegtų programėlių sąrašas, nurodant kiekvienos programėlės iš sąrašo leidimus, ir (2) naudojamų leidimų sąrašas, kiekvienam leidimui nurodant, kokios programėlės jį naudoja.

3.3. Leidimų modulis, dekompiliavus programėlę

Norint nuskaityti programėlės leidimus, saugomus „AndroidManifest.xml“ faile, reikia APK failą dekompiliuoti. Tam tikslui galima naudoti dekompiliavimo įrankį „Apktool“.

Dekompiliavimui naudojama komanda: apktool d C://failas.apk

```
android:name="android.permission.INTERNET"/>
android:name="android.permission.ACCESS_WIFI_STATE"/>
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
android:name="android.permission.ACCESS_NETWORK_STATE"/>
android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
android:name="com.android.vending.BILLING"/>
android:name="android.permission.CAMERA"/>
android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
android:name="com.google.android.c2dm.permission.RECEIVE"/>
```

```
Administrator: C:\Windows\system32\cmd.exe

C:\Users\Nerijus msi>apktool d 0x.apk
Input file (0x.apk) was not found or was not readable.

C:\Users\Nerijus msi>apktool d C://0x.apk
I: Using Apktool 2.3.3 on 0x.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (C:\Users\Nerijus msi\AppData\Local\
work), using C:\Users\NERIJU~1\AppData\Local\Temp\ instead...
S: Please be aware this is a volatile directory and frameworks coul
 please utilize --frame-path if the default storage directory is un
I: Loading resource table from file: C:\Users\NERIJU~1\AppData\Loca
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

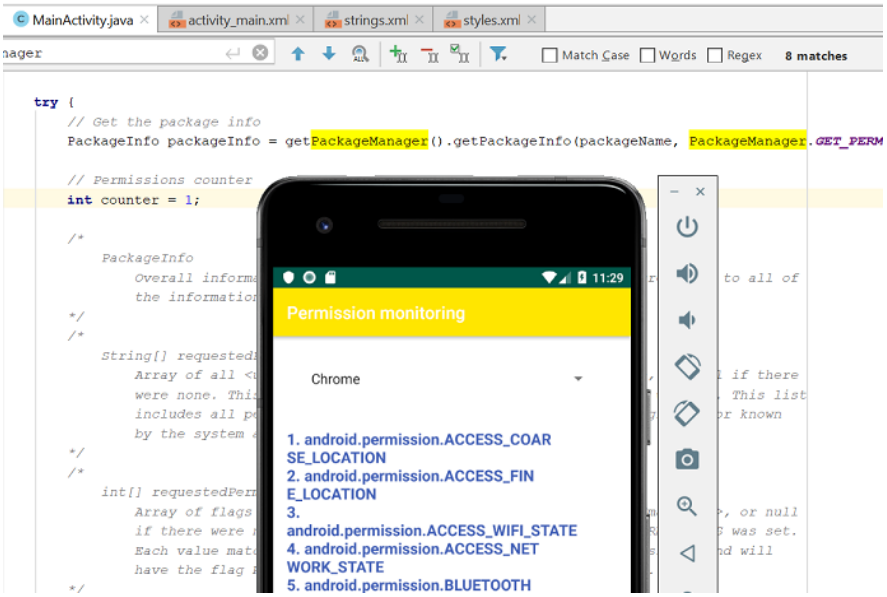
3.3 pav. Failo „AndroidManifest“ fragmentas, dekompiliavus APK failą

Iliustracijoje matyti, kad programuotojai yra numatę, jog „Oxford“ žodyno programėlei reikės tokių leidimų kaip internetas, „Wi-Fi“, duomenų įrašymas į išorinę atmintį, kamera ir kt.

3.4. Leidimų nuskaitymas su „PackageManager“ klase

„Android“ sistema pasižymi dideliu skaičiumi API (programėlių programavimo sąsajų), skirtų programėlių valdymui. Norint nuskaityti leidimus „PackageManager“ klase, reikia aktyvinti „PackageManager“ API. Tada galima naudoti tokias funkcijas, pavyzdžiui:

- Context#getPackageManager – klasė, skirta įdiegtų APK informacijos nuskaitymui
- getInstalledPackages – gauti įdiegtų APK failų (programėlių) sąrašą
- getPackageInfo(PackageName,PackageManager.GET_PERMISSIONS); – nurodomi programėlės prašyti leidimai
- getInstallerPackageName – nurodomas programinio paketo pavadinimas



3.4 pav. Leidimų nuskaitymas su „PackageManager“

Šis modulis gali būti naudojamas nuskaitant leidimus pagal programėlę arba nuskaitant programėles pagal tam tikrą leidimą arba leidimus.

3.5. „Android“ leidimų klasifikavimas

Leidimams numatyti keturi klasifikaciniai lygiai, kurie taikomi visom programėlėms: įprasti, signatūriniai, pavojingi ir specialieji. Visą jų sąrašą galima rasti „Android“ programuotojams skirtoje svetainėje [24].

3.5 pav. „Android“ leidimų klasifikavimas

Įprastieji leidimai

Įprastieji leidimai yra tokie, kai programėlei reikia prieigos prie duomenų arba resursų, nepriklausančių programėlės vidinėms funkcijoms, bet tai kelia labai mažą pavojų naudotojo privatumui arba programėlių funkcionalumui. Pavyzdžiui, leidimas nustatyti laiko juostą laikomas įprastuoju leidimu.

Jei programėlė „manifest“ faile nurodo, kad jai reikia įprastojo leidimo, sistema automatiškai suteikia jai tokį, kai ji yra įdiegiama. Sistema neragina naudotojo suteikti įprastųjų leidimų, be to, naudotojai negali atšaukti šių leidimų.

Operacinėje sistemoje „Android 8.1“ tolesni leidimai yra klasifikuojami kaip įprastieji PROTECTION_NORMAL:

1. ACCESS_LOCATION_EXTRA_COMMANDS
2. ACCESS_NETWORK_STATE
3. ACCESS_NOTIFICATION_POLICY
4. ACCESS_WIFI_STATE
5. BLUETOOTH
6. BLUETOOTH_ADMIN
7. BROADCAST_STICKY
8. CHANGE_NETWORK_STATE
9. CHANGE_WIFI_MULTICAST_STATE
10. CHANGE_WIFI_STATE
11. DISABLE_KEYGUARD
12. EXPAND_STATUS_BAR
13. GET_PACKAGE_SIZE
14. INSTALL_SHORTCUT
15. INTERNET
16. KILL_BACKGROUND_PROCESSES
17. MANAGE_OWN_CALLS
18. MODIFY_AUDIO_SETTINGS
19. NFC
20. READ_SYNC_SETTINGS

21. READ_SYNC_STATS
22. RECEIVE_BOOT_COMPLETED
23. REORDER_TASKS
24. REQUEST_COMPANION_RUN_IN_BACKGROUND
25. REQUEST_COMPANION_USE_DATA_IN_BACKGROUND
26. REQUEST_DELETE_PACKAGES
27. REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
28. REQUEST_INSTALL_PACKAGES
29. SET_ALARM
30. SET_WALLPAPER
31. SET_WALLPAPER_HINTS
32. TRANSMIT_IR
33. USE_FINGERPRINT
34. VIBRATE
35. WAKE_LOCK
36. WRITE_SYNC_SETTINGS

Signatūriniai leidimai

Sistema įdiegiant suteikia šiuos programėlių leidimus, bet tik tada, kai programėlė, kuri siekia naudoti leidimą, yra pasirašyta tuo pačiu sertifikatu kaip ir programėlė, kuri apibrėžia leidimą. Kai kurių signatūrinių leidimų trečiųjų šalių programėlės negali naudoti.

Operacinėje sistemoje „Android 8.1“ tolesni leidimai yra klasifikuojami kaip signatūriniai PROTECTION_SIGNATURE:

1. BIND_ACCESSIBILITY_SERVICE
2. BIND_AUTOFILL_SERVICE
3. BIND_CARRIER_SERVICES
4. BIND_CHOOSER_TARGET_SERVICE
5. BIND_CONDITION_PROVIDER_SERVICE
6. BIND_DEVICE_ADMIN
7. BIND_DREAM_SERVICE

8. BIND_INCALL_SERVICE
9. BIND_INPUT_METHOD
10. BIND_MIDI_DEVICE_SERVICE
11. BIND_NFC_SERVICE
12. BIND_NOTIFICATION_LISTENER_SERVICE
13. BIND_PRINT_SERVICE
14. BIND_SCREENING_SERVICE
15. BIND_TELECOM_CONNECTION_SERVICE
16. BIND_TEXT_SERVICE
17. BIND_TV_INPUT
18. BIND_VISUAL_VOICEMAIL_SERVICE
19. BIND_VOICE_INTERACTION
20. BIND_VPN_SERVICE
21. BIND_VR_LISTENER_SERVICE
22. BIND_WALLPAPER
23. CLEAR_APP_CACHE
24. MANAGE_DOCUMENTS
25. READ_VOICEMAIL
26. REQUEST_INSTALL_PACKAGES
27. SYSTEM_ALERT_WINDOW
28. WRITE_SETTINGS
29. WRITE_VOICEMAIL

Pavojingi leidimai

Pavojingi leidimai yra susiję su tokiomis sritimis, kai programėlei reikia duomenų arba išteklių, sudarančių naudotojo privačią informaciją, arba kai tai gali turėti įtakos naudotojo turimiems duomenims ar kitų programėlių funkcionavimui. Pavyzdžiui, leidimas nuskaityti naudotojo kontaktinius duomenis yra laikomas pavojingu. Jei programėlė nurodo, kad jai reikia pavojingo leidimo, naudotojas privalo išskirtinai suteikti leidimą tai programėlei. Kol naudotojas leidimo nepatvirtina, programėlė negali vykdyti tos funkcijos, kuri priklauso nuo to leidimo.

Kad programėlė galėtų naudoti pavojingą leidimą, ji naudotoją paragina per paleistį suteikti minėtą leidimą. Operacinėje sistemoje „Android 8.1“ tolesni leidimai yra klasifikuojami kaip pavojingi:

2 lentelė. Pavojingi leidimai

Leidimų grupė	**Leidimai**
CALENDAR (kalendorius)	READ_CALENDAR Leidžia programėlei nuskaityti naudotojo kalendoriaus duomenis. WRITE_CALENDAR Leidžia programėlei rašyti duomenis į naudotojo kalendorių.
CALL_LOG (skambučių žurnalas)	READ_CALL_LOG Leidžia programėlei nuskaityti naudotojo skambučių žurnalą. WRITE_CALL_LOG Leidžia programėlei rašyti duomenis (bet ne skaityti) į naudotojo skambučių žurnalą. PROCESS_OUTGOING_CALLS Leidžia programėlei matyti numerį, kuriuo naudotojas skambina, turint galimybę peradresuoti skambutį kitu numeriu arba nutraukti skambutį.
CAMERA (kamera)	CAMERA Šio leidimo reikia, kad būtų galima naudotis įrenginio kamera. Tai automatiškai suteiks galimybę manifesto failo elementui naudoti *visas* kameros funkcijas. Jei nereikia visų kameros funkcijų arba programėlė gali puikiai veikti, net jei kamera nėra prieinama, tada reikia atitinkamai pakeisti manifesto failą, kad būtų galim įdiegti programėlę tuose įrenginiuose, kuriuose neveikia visos kameros funkcijos.
CONTACTS (kontaktai)	READ_CONTACTS Leidžia programėlei nuskaityti naudotojo adresų knygelės duomenis. WRITE_CONTACTS Leidžia programėlei rašyti duomenis naudotojo adresų knygelėje. GET_ACCOUNTS Suteikia prieigą prie paskyrų sąrašo.
LOCATION (vieta)	ACCESS_FINE_LOCATION Leidžia programėlei sužinoti tikslią vietą. ACCESS_COARSE_LOCATION Leidžia programėlei sužinoti apytikslią vietą.
MICROPHONE (mikrofonas)	RECORD_AUDIO Leidžia programėlei įrašyti garsą.
PHONE	READ_PHONE_STATE

(telefonas)	Leidžia tik nuskaityti telefono būseną, įskaitant įrenginio telefono numerį, esamą mobiliojo tinklo informaciją, atliktų skambučių būseną ir bet kokių įrenginyje užregistruotų telefono paskyrų sąrašą. READ_PHONE_NUMBERS Leidžia nuskaityti prieigą prie įrenginio telefono numerių. Tai antrinis galimybių rinkinys, kurį suteikia READ_PHONE_STATE. CALL_PHONE Leidžia programėlei inicijuoti telefono skambutį, nenaudojant telefono rinkiklio naudotojo sąsajos, leidžiančios naudotojui patvirtinti skambutį. ANSWER_PHONE_CALLS Leidžia programėlei atsiliepti į priimamą telefono skambutį. ADD_VOICEMAIL Leidžia programėlei pridėti sistemoje balso pašto pranešimų. USE_SIP Leidžia programėlei naudoti SIP paslaugą.
SENSORS (jutikliai)	BODY_SENSORS Suteikia programėlei prieigą prie duomenų, gautų iš jutiklių, kuriuos naudoja naudotojas, kad galėtų įvertinti, kokie procesai vyksta naudotojo organizme, pavyzdžiui, sužinoti pulsą.
SMS (SMS žinutės)	SEND_SMS Leidžia programėlei siųsti SMS pranešimus. RECEIVE_SMS Leidžia programėlei gauti SMS pranešimus. READ_SMS Leidžia programėlei skaityti SMS pranešimus. RECEIVE_WAP_PUSH Leidžia programėlei gauti „WAP push“ pranešimus. RECEIVE_MMS Leidžia programėlei sekti gaunamus MMS pranešimus.
STORAGE	READ_EXTERNAL_STORAGE Leidžia programėlei nuskaityti iš išorinės atminties. Bet kuriai programėlei, kuri deklaruoja WRITE_EXTERNAL_STORAGE leidimą, šis leidimas suteikiamas besąlygiškai. WRITE_EXTERNAL_STORAGE Leidžia programėlei įrašyti į išorinę atmintį.

Specialieji leidimai

Yra dar 2 leidimai, kurie nėra priskiriami nei įprastiems, nei pavojingiems. SYSTEM_ALERT_WINDOW ir WRITE_SETTINGS leidimai yra itin jautrūs, todėl daugumą programėlių neturėtų jų naudoti. Jei programėlei reikia vieno iš šių leidimo, tas leidimas turi būti

užregistruojamas „manifest“ faile, taip pat naudotojui turi būti pateikiamas prašymas dėl suteikimo. Sistema apie šį ketinimą įspėja parodydama naudotojui išsamų valdymo rodinį.

1. SYSTEM_ALERT_WINDOW
2. WRITE_SETTINGS

3.6. Saugumo monitoringo programos struktūra

Saugumo monitoringo programą sudaro 2 pagrindiniai moduliai: įdiegtų programėlių leidimų nuskaitymo modulis pagal programėles ir įdiegtų programėlių leidimų nuskaitymo modulis pagal leidimus.

Nuskaitymas pagal programėles. Pateikiamas visų įdiegtų programėlių sąrašas, o pasirinkus konkrečią programėlę ir ją spragtelėjus, pateikiama išsami informacija apie paprašytus leidimus. Taip pat pateikiama kita naudotojui svarbi ir su sauga susijusi (arba santykinai susijusi) informacija kaip įdiegimo laikas, atnaujinimo laikas, įdiegimo vieta (įrenginyje arba SD kortelėje), programėlės įdiegimo paketo dydis, prašomi programėlės leidimai.

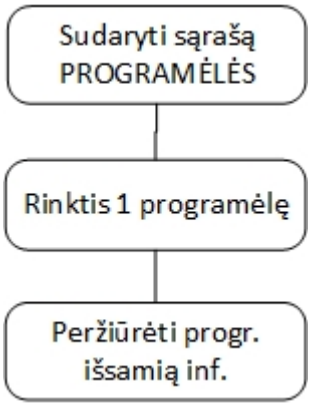
Tam tikslui būtų galima naudoti klasės AppInfoProvider funkciją getAllApps(), kuri pateikia visų sistemoje esančių programėlių sąrašą. Naudodami funkciją requestPermissions gauname informacijos apie leidimus, packageName – paketo pavadinimui, loadIcon – įkeliama programėlės piktograma, loadlabel – nuskaitomas programėlės pavadinimas.

3.6 pav. Nuskaitymas pagal leidimus

Nuskaitymas pagal leidimus. Šio modulio tikslas – naudojant pavojingus (jautrius) leidimus (jie nustatomi pagal oficialų leidimų klasifikavimą ir taip pat pagal naudotojo asmeninius pageidavimus, jei tam tikras leidimas yra svarbus) pateikti sąrašą, kiek programėlių vienus ar kitus leidimus naudoja.

Galima naudoti klases SimpleCursorAdapter ir ListView siekiant pateikti užklausą duomenų bazės failui ir atvaizduoti jo informaciją, pateikiant sąraše pavojingus (jautrius) leidimus. Kad sužinotume kuris leidimas paspaustas ekrane, naudojama funkcija onItemClick(). Tada rodomos programėlės pagal tą konkretų leidimą. Kad būtų gautas programėlių sąrašas, galima naudoti

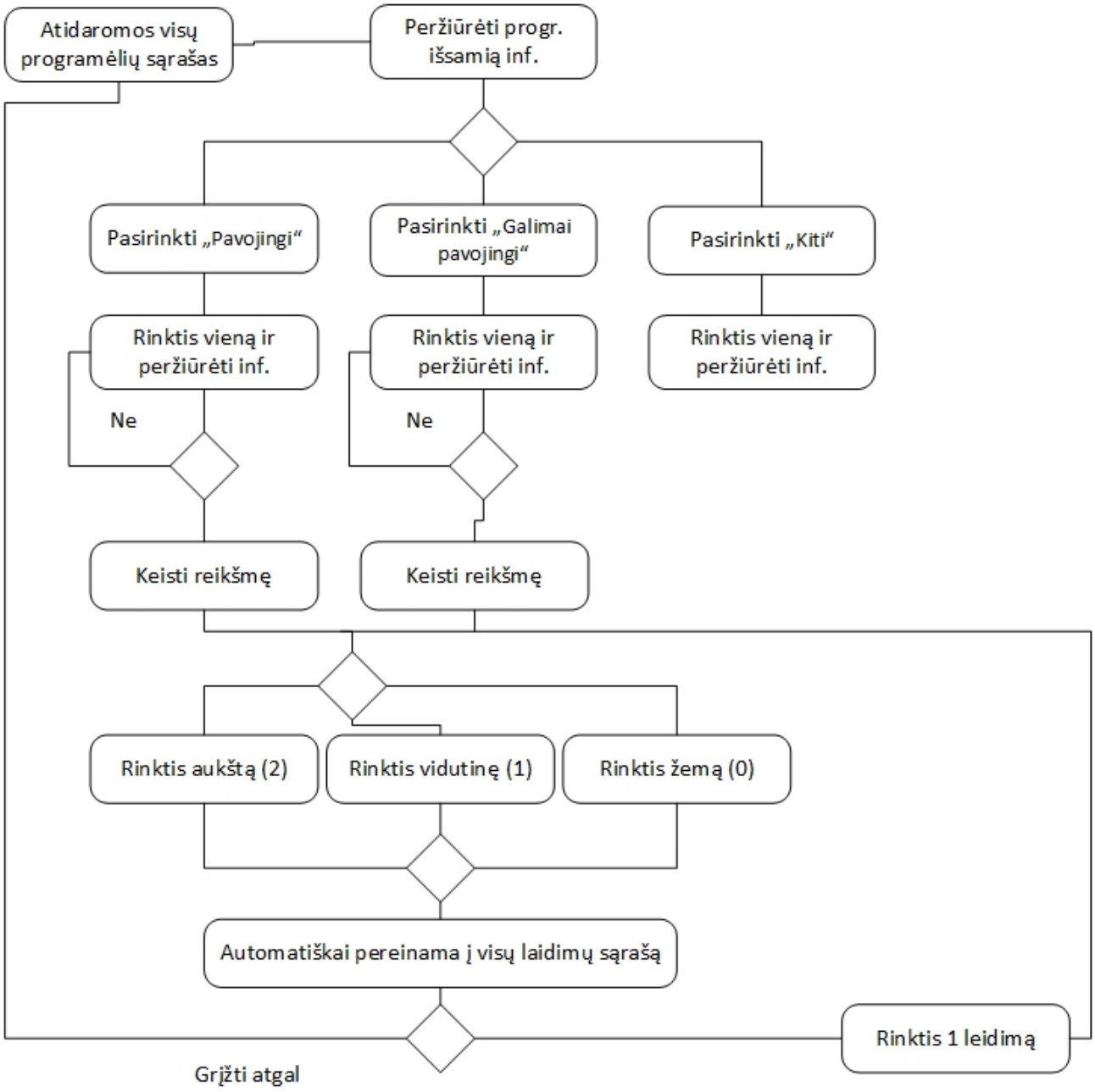
„PackageManager“ klasę ir nuskaityti visų programėlių leidimus, ieškant pasirinktų jautrių leidimų. Jei randama programėlė su tuo leidimu, ji įtraukiama į sąrašą.



3.7 pav. Nuskaitymas pagal programėles

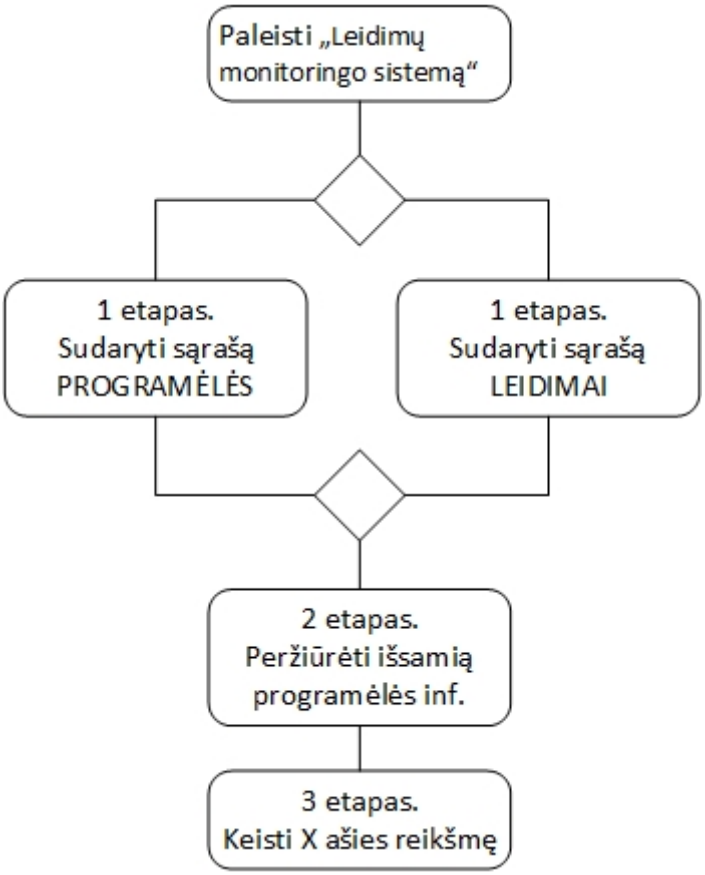
Informacijos vertės (X ašyje) keitimas. Kadangi leidimai Y ašiai yra priskiriami ir jų vertė nekeičiama, galima keisti tik informacijos (X ašies) vertes, kurios taip pat priskirtos pagal numatymą. Į šią programėlės dalį patenkama, kai peržiūrima išsami informacija apie programėlę. Pasirinkus vieną iš leidimų meniu parinktį – (1) pavojingi, (2) galimai pavojingi ir (3) kiti – galima keisti (1) pavojingiems ir (2) galimai pavojingiems leidimams priskirtos informacijos reikšmę vertės (X) ašyje. Paspaudus meniu punktą „Keisti reikšmę“, galima rinktis tos informacijos reikšmę: aukšta (2 taškai), vidutinė (1 taškas) ir žema (0 taškų).

Kai reikšmė pakeičiama arba paliekama ta pati, pereinama į visų leidimų sąrašą. Galima pasirinkti iš to sąrašo bet kurį kitą leidimą ir pakeisti (X ašies) reikšmę, kurie pagal numatymą susieta su tuo leidimu Y ašyje.



3.8 pav. Informacijos vertės (X ašyje) keitimas

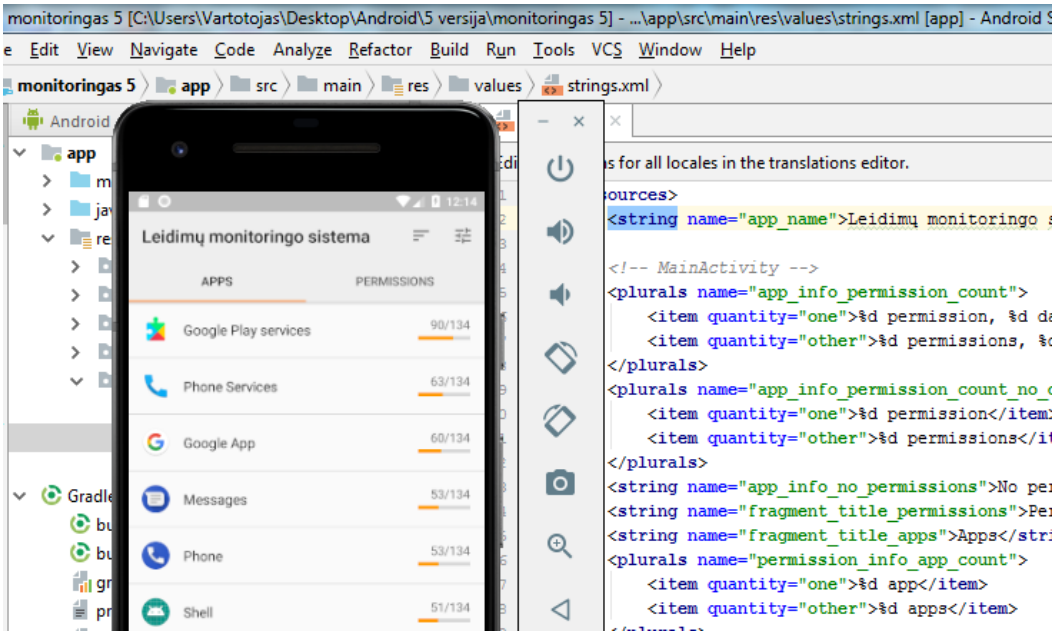
Išmaniųjų įrenginių saugumo nustatymų stebėjimo metodas. Pagal tai, kaip veikia prototipas, jį sudaro 3 pagrindiniai etapai: (1) programėlių arba leidimų sąrašo sudarymas, (2) išsamios informacijos peržiūrėjimas apie programėlę ir (3) vertės, kuri pagal numatymą priskiriama, remiantis Y ašies leidimais, X ašyje keitimas.



3.9 pav. Išmaniųjų įrenginių saugumo nustatymų stebėjimo metodas

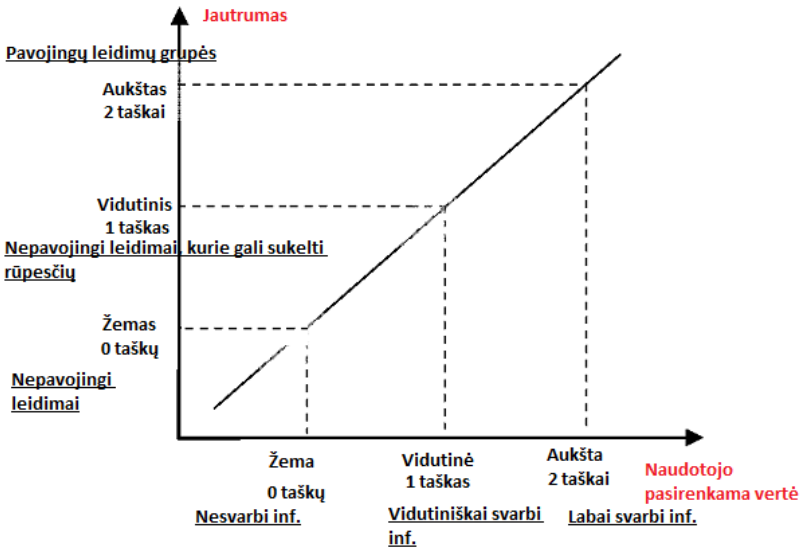
3.7. Programėlių saugumo monitoringo prototipas

Saugumo monitoringo prototipui kurti naudota „Android Studio 3“ programavimo aplinka. Šioje aplinkoje programuojama „Java“ kalba. Ši programų kūrimo aplinka susideda iš derinimo programos, bibliotekų, programavimo pavyzdžių, emuliatoriaus, dokumentacijos ir pan.



3.10 pav. Leidimų monitoringo sistemos vaizdas, paleidus emuliatorių

Kai „Android“ įrenginyje paspaudžiama „Leidimų monitoringo sistemos“ piktograma, paleidžiama programėlė. Programėlės pagrindiniame lange pateikiami 2 pagrindiniai sąrašai: PROGRAMĖLĖS ir LEIDIMAI. Šie sąrašai leidžia atitinkamai išdėstyti programėles. Sąraše PROGRAMĖLĖS yra išdėstomos programėlės pagal pavojingumo taškų skaičių. Daugiausiai taškų turinčios programėlės yra daugiausiai pavojingų leidimų ir naudotojo nuomone jautrių leidimų naudojančios programėlės. Pavojingumo taškai apskaičiuojami taip:



3.11 pav. Pavojingumo taškų skaičiavimas pagal 2 ašis

Programėlių jautrumo ašyje vertinami leidimai, kuriuos naudoja programėlė. Įprastieji (normal permissions) laikomi nepavojingais, todėl apie juos nereikia naudotojo įspėti ir dėl to skiriama 0 taškų. Įprastųjų leidimų sąrašą galime rasti „Android“ svetainėje.

Visgi ne visi įprastieji leidimai visiškai nekelia grėsmės jautrios informacijos nutekinimui. Gali būti, kad jei tokie leidimai ir nesukeltų grėsmės, bet dėl jų naudotojas turėtų nepatogumų. Pavyzdžiui, leidimas CHANGE_NETWORK_STATE, leidžia programėlei keisti prisijungimo prie tinklo būseną. Programėlė tinklu gali perduoti jautrią informaciją arba prisijungti prie nesaugių tinklų. Už tokio vieno leidimo naudojimą skiriamas 1 taškas.

2 taškai skiriami už pavojingų leidimų grupes. Jų sąrašą galima rasti „Android“ svetainėje. Programėlės, naudojančios šiuos leidimus, gali nutekinti jautrią informaciją.

Naudotojo pasirenkamos duomenų vertės ašyje taip pat yra trys reikšmės. Už nesvarbią informaciją skiriama 0 taškų, nes jei naudotojas ją prarastų arba kas nors ją nutekintų, jokios reikšmės tai neturėtų. Už vidutinio svarbumo informaciją skiriamas 1 taškas. 2 taškai skiriami už labai svarbią informaciją. Naudotojo pasirenkamos vertės ašiai naudojamos pirminiam apskaičiavimui numatytosios vertės, bet naudotojas vėliau jas gali pakeisti, todėl programėlės pavojingumo rezultatai taip pat pasikeistų. Numatytosios vertės yra tokios: žema informacijos vertė (0), kai ir pavojingumas jautrumo ašyje (Y) yra mažas, vidutiniška vertė (1) – pavojingumas jautrumo ašyje (Y) vidutinis, o naudotojo informacijos vertės numatytoji pasirenkama reikšmė yra aukšta (2), kai ir pavojingumas jautrumo ašyje (Y) yra aukštas.

Tokiu atveju skaičiuojama pagal toliau pateiktą lentelę. Vertės yra sudauginamos.

3 lentelė. Pavojingumo taškų skaičiavimas

Pavojingi leidimai	2	0	2	4
Galimai pavojingi	1	0	1	2
Nepavojingi leidimai	0	0	0	0
		0	1	2
		Nesvarbi inf.	Vidutiniškai svarbi	Svarbi inf.

Didžiausia taškų suma, kurią gali programėlė surinkti, apskaičiuojama įtraukiant visus galimus leidimus.

4 lentelė. Didžiausia taškų suma už pavojingus leidimus

El. Nr.	Leidimų grupė	Kiekis	Atskiri leidimai	Y ašis	X ašis (maks.)	Sandauga
1	CALENDAR	1	READ_CALENDAR	2	2	4
		2	WRITE_CALENDAR	2	2	4
2	CALL_LOG	3	READ_CALL_LOG	2	2	4
		4	WRITE_CALL_LOG	2	2	4
		5	PROCESS_OUTGOING_CALLS	2	2	4
3	CAMERA	6	CAMERA	2	2	4
4	CONTACTS	7	READ_CONTACTS	2	2	4
		8	WRITE_CONTACTS	2	2	4
		9	GET_ACCOUNTS	2	2	4
5	LOCATION	10	ACCESS_FINE_LOCATION	2	2	4
		11	ACCESS_COARSE_LOCATION	2	2	4
6	MICROPHONE	12	RECORD_AUDIO	2	2	4
7	PHONE	13	READ_PHONE_STATE	2	2	4
		14	READ_PHONE_NUMBERS	2	2	4
		15	CALL_PHONE	2	2	4
		16	ANSWER_PHONE_CALLS	2	2	4
		17	ADD_VOICEMAIL	2	2	4
		18	USE_SIP	2	2	4
8	SENSORS	19	BODY_SENSORS	2	2	4
9	SMS	20	SEND_SMS	2	2	4
		21	RECEIVE_SMS	2	2	4
		22	READ_SMS	2	2	4
		23	RECEIVE_WAP_PUSH	2	2	4
		24	RECEIVE_MMS	2	2	4

10	STORAGE	25	READ_EXTERNAL_STORAGE	2	2	4
		26	WRITE_EXTERNAL_STORAGE	2	2	4
	Didžiausia taškų suma už pavojingus leidimus					104

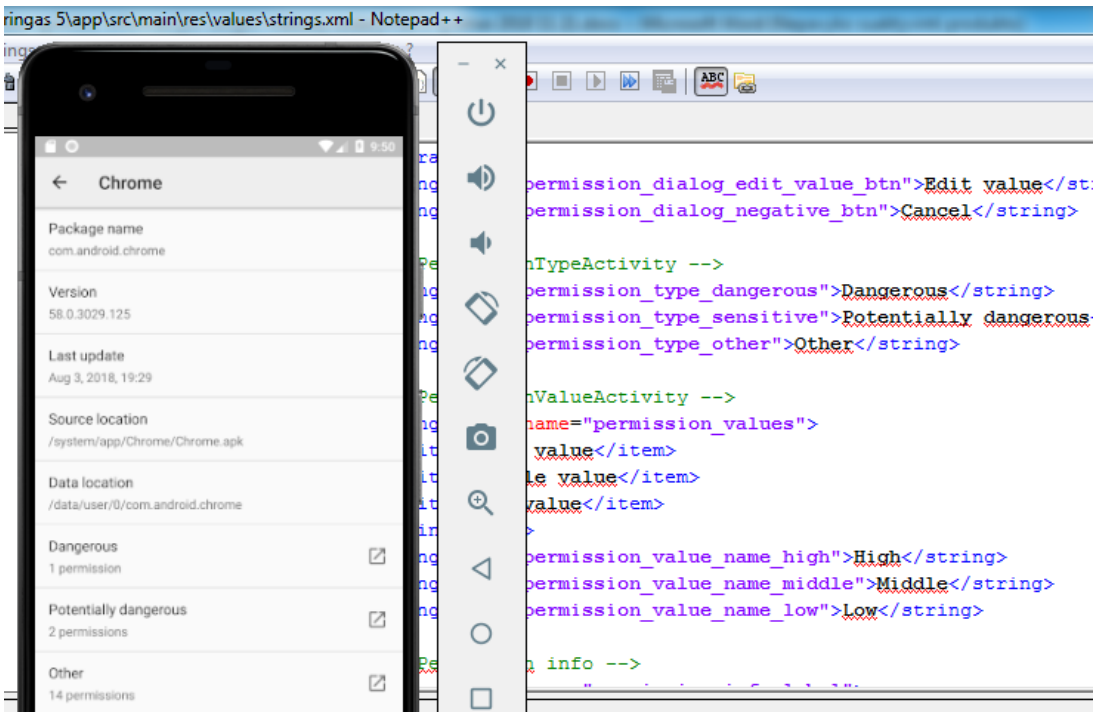
5 lentelė. Didžiausia taškų suma už galimai pavojingus leidimus

Eil. Nr.	Leidimai	Y ašis	X ašis	Sandauga
1	CHANGE_NETWORK_STATE	1	2	2
2	CHANGE_WIFI_STATE	1	2	2
3	MODIFY_AUDIO_SETTINGS	1	2	2
4	REQUEST_DELETE_PACKAGES	1	2	2
5	NFC	1	2	2
6	REORDER_TASKS	1	2	2
7	REQUEST_INSTALL_PACKAGES	1	2	2
8	FLASHLIGHT	1	2	2
9	GET_TASKS	1	2	2
10	BILLING	1	2	2
11	SET_ALARM	1	2	2
12	DISABLE_KEYGUARD	1	2	2
13	SET_WALLPAPER	1	2	2
14	SYSTEM_ALERT_WINDOW	1	2	2
15	WRITE_SETTINGS	1	2	2
	Didžiausia taškų suma už vidutinio pavojingumo leidimus			30

6 lentelė. Didžiausia taškų suma už pavojingus leidimus ir galimai pavojingus leidimus

Didžiausia taškų suma už pavojingus leidimus	104
Didžiausia taškų suma už vidutinio pavojingumo leidimus	30
Didžiausia taškų suma už pavojingus ir vidutinio pavojingumo leidimus	134

Toliau pateikiamas pavyzdys, kaip apskaičiuoti vienos iš programėlės pavojingumo taškai. „Google Chrome“ surenka 10 pavojingumo taškų iš 134 galimų.



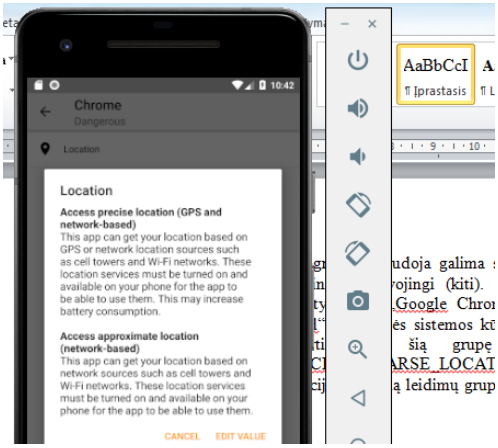
3.12 pav. „Google Chrome“ leidimai

Paspaudus programėlių sąraše programėlę „Chrome“, atsiranda meniu, kuriame pateikiama tam tikra informacija apie programėlę. Tą informaciją sudaro programinio paketo pavadinimas, versija, paskutinio atnaujinimo data, APK paketo vieta, įdiegimo vieta ir naudojamų leidimų sąrašas pagal pavojingumą. „Google Chrome“ naudoja 1 pavojingą leidimą, 1 galimai pavojingą ir 14 nepavojingų.

7 lentelė. „Google Chrome“ pavojingumo taškai

Leidimų grupės jautrumo ašyje (Y)	**Naudotojo pasirenkamos vertės ašyje (X)**
1 * 2 (pavojingų leidimų grupė su 2 pavojingais leidimais) * 2 (numatytoji vertė jautrumo ašyje) = 4	4 * 2 (numatytoji naudotojo pasirenkama vertė) = 8
Galimai pavojingų 2 * 1 (numatytoji vertė jautrumo ašyje) = 2	2 * 1 (numatytoji naudotojo pasirenkama vertė) = 2
Nepavojingų 14 * 0 (numatytoji vertė jautrumo ašyje) = 0	0 * 0 (numatytoji naudotojo pasirenkama vertė) = 0
	10 pavojingumo taškų

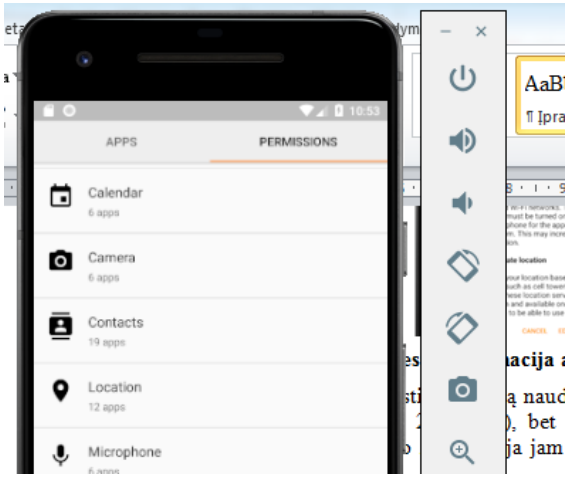
Kiek ir kokių leidimų programėlė naudoja galima sužinoti paspaudus dominančius meniu punktus: pavojingi, galimai pavojingi, nepavojingi (kiti). Jei šiuo atveju paspaudus meniu punktą „pavojingi“, galima pamatyti, kad „Google Chrome“ naudoja pavojingą leidimų grupę „Vieta“ (angl. Location). „Android“ operacinės sistemos kūrėjo puslapyje iš duotų pavojingų leidimų grupių sąrašo matyti, kad šią grupę sudaro 2 pavojingi leidimai: ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION. Paspaudus leidimų grupę „Vieta“, galima sužinoti išsamesnės informacijos apie šią leidimų grupę.



3.13 pav. Išsamesnė informacija apie leidimų grupę „Vieta“

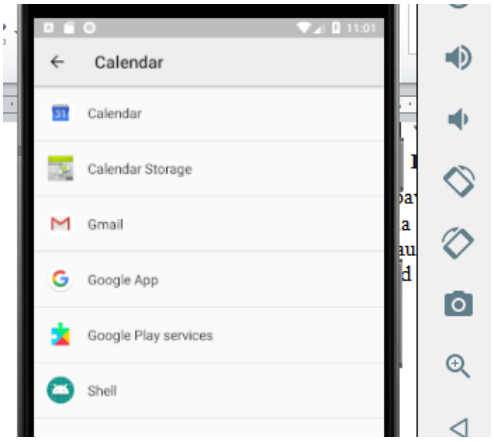
Šioje vietoje galima pakeisti numatytą naudotojo pasirenkamą vertę. Numatytoji naudotojo vertė su pavojingais leidimais yra 2 (aukšta), bet naudotojas gali ją pakeisti į 0 (žema) arba 1 (vidutinė), jei mano, kad tokio tipo informacija jam nėra tiek reikšminga ir jis tiek nesibaimina dėl tokios informacijos nutekėjimo.

Sąraše LEIDIMAI yra pateikiamos pavojingų leidimų grupės ir nurodoma, kiek programėlių naudoja atitinkamus leidimus.



3.14 pav. Sąrašas LEIDIMAI

Kaip matyti šiuo atveju, pavojingų leidimų grupę „Kalendorius“ naudoja 6 programėlės, o „Kontaktai“ – 19 ir t. t. Galima sužinoti kokios konkrečiai programėlės naudoja dominančią pavojingų leidimų grupę, paspaudus atitinkamą punktą. Jei mus domina leidimų grupė „Kalendorius“, galime pamatyti, kad ją naudoja tolesnės programėlės.



3.15 pav. Leidimų grupę „Kalendorius“ naudojančios programėlės

Paspaudus sąraše dominančią programėlę, galima sužinoti apie ją papildomos informacijos. Ta informacija sudaro jau anksčiau minėtą: paketo pavadinimas, atnaujinimas, versija ir t. t.

3.8. Išvados

1. Pasiūlytas programėlių saugumo įvertinimo prototipas. Kadangi „Android“ programėlių konfidencialumas arba galimybė nutekinti informaciją priklauso nuo suteiktų leidimų, prototipo veikimas pagrįstas suteiktų leidimų identifikavimu.
2. Prototipas pateikia 2 pagrindinius sąrašus pagrindiniame prototipo lange. Pirmajame programėlės išrikiuojamos pagal pavojingumo taškus. Antrąjį sąrašą sudaro leidimai. Jie surikiuojami pagal panaudojimo dažnumą.
3. Prototipo išskirtinė savybė – įvedamos 2 ašys. Ašis Y skirta leidimų pavojingumui, o ašis X – naudotojo pasirenkamai vertei. Bendra taškų suma apskaičiuojama sudauginus abiejų ašių vertes. Naudotojui pasirinkus skirtingą nei numatytąją vertę, pavojingumo taškai perskaičiuojami.

4. TYRIMAS

4.1. Tyrimui naudota įranga

Aparatinė įranga

Atliekant tyrimą ir išbandant prototipą bei kitas tiriamąsias programėles naudota tolesnė aparatinė įranga. Kompiuteris naudotas kuriant prototipą ir jį paleidžiant emuliatoriuje. Mobilusis telefonas ir planšetinis kompiuteris naudotas įdiegiant prototipo „.apk“ failą ir tiriant juo atsiųstas programėles.

8 lentelė. Aparatinė įranga

Įrenginys	**Techninės savybės**	
Kompiuteris „Lenovo Yoga 530“	Operacinė sistema	Windows Pro 10
	Sistemos tipas	64 bitų operacinė sistema, x64 pagrindo procesorius
	Procesorius	Intel® Core™ i3-8130U CPU @ 2,20 Ghz
	Įdiegta atmintis (RAM)	16,0 GB
Mobilusis telefonas „Samsung Galaxy S8“	Modelis	SM-G950F
	„Android“ versija	8.0.0
	„Samsung Experience“ versija	9.0
	„Knox“ versija	Knox 3.0 Knox API level 24 TIMA 3.3.0
Planšetinis kompiuteris „Samsung Tab A“	Modelis	SM-T585
	„Android“ versija	8.1.0
	„Samsung Experience“ versija	9.5
	„Knox“ versija	Knox 3.2 Knox API level 26 TIMA 3.3.0

Programinė įranga

Atliekant tyrimą ir išbandant prototipą bei kitas tiriamąsias programėles naudota tolesnė programinė įranga / programėlės.

9 lentelė. Programinė įranga

Programinė įranga / programėlė	**Techninės savybės**	
Android Studio 3.2.1	Versija	#AI-181.5540.7.32.5056338, 2018 m. spalio 9 d.
	JRE	1.8.0_152-release-1136-b06 amd64
	JVM	OpenJDK 64-bit Server VM by JetBrains s.r.o

	„Android“ virtualusis įrenginys	Pixel 2 API 26 1080 x 1920; 420dpi Android 8.0 CPU x86
„Android“ operacinės sistemos modulis „Programos“	Modulį „Programos“ galima įjungti	Nustatymai > Programos
Prototipas „Leidimų monitoringo sistema“	-	-

4.2. Dažniausiai naudojamų programėlių saugos tyrimas

Šio eksperimentinio tyrimo apie galimybes nutekinti „Android“ įrenginyje saugomą informaciją iškelti tikslai yra tokie:

1. Kurios programėlių kategorijos iš tikrintų kelia didžiausią informacijos nutekinimo pavojų?
2. Kurios programėlės iš visų tikrintų kelia didžiausią informacijos nutekinimo pavojų?
3. Kurių leidimų programėlės prašo dažniausiai?
4. Ar didesnis informacijos nutekinimas, paprašius daugiau leidimų, leidžia programėlių kūrėjams pelnyti daugiau patikimo žvaigždučių „Play Store“ parduotuvėje?

Siekiant atlikti tyrimą, kategorijos „Play Store“ parduotuvėje išrinktos atsitiktine tvarka. Tiriamosios programėlės atitinkamoje kategorijoje pasirinktos pagal populiarumą, nes jos aktualiausios daugumai naudotojų, o tai reiškia, kad turėtų daugiausiai galimybių nutekinti informaciją.

Toliau nurodomos kategorijos, iš kurių po 20 atsisiųsta tuo metu buvusių populiariausių programėlių.

1. Apsipirkimas
2. Finansai
3. Ryšiai
4. Švietimas
5. Verslas

Tirtos „Android“ programėlės

Toliau nurodoma 20 populiariausių programėlių iš kiekvienos kategorijos, kurios naudotos tyrimui. Gretimame stulpelyje nurodomi pavojingumo taškai, kurie apskaičiuojami pagal prototipą „Leidimų monitoringo sistema“. Taip pateikiamas programėlių įvertinimas, kuris „Play Store“ parduotuvėje apskaičiuojamas iš visų naudotojų skirtų „patikimo žvaigždučių“.

10 lentelė. „Android“ programėlės iš kategorijos „Apsipirkimas“

1.	**Apsipirkimas**		
Nr.	Programėlės pavadinimas	Pavojingumo taškai	Naudotojų vertinimas
1.	Joom	8	4,5
2.	Wish	25	0
3.	AliExpress	25	4,2
4.	Pigu.lt	8	4,6
5.	Vinted.lt	24	4,5
6.	Alibaba.com	37	4,4
7.	Lidl	10	4
8.	Barbora.lt	4	0
9.	eBay	21	3,8
10.	Senukai	12	4
11.	Zaful	39	4,8
12.	Mano akcijos	4	3,9
13.	Skelbiu.lt	20	4,7
14.	Mūsų mažyliai	24	4,7
15.	ASOS	13	4,5
16.	Vova	30	4,5
17.	Banggood	13	0
18.	NEWMOOD.lt	8	4,6
19.	Cashback App	28	4,5
20.	SHEIN	53	4,8

11 lentelė. „Android“ programėlės iš kategorijos „Finansai“

2.	**Finansai**		
Nr.	Programėlės pavadinimas	Pavojingumo taškai	Naudotojų vertinimas
1.	Swedbank Lietuva	12	3,8
2.	SEB Lietuva	20	2,6
3.	Luminor DNB Lietuva	20	3,4
4.	Swedbank 2019 Lietuva	5	4,8
5.	Revolut	41	3,1
6.	MoQ	34	3,9
7.	Western Union LT	32	4,4
8.	PayPal Mobile Cash	43	4,3
9.	Paysera mobilioji piniginė	28	3,9
10.	Viena sąskaita	5	4
11.	Curve: One card for all your accounts	12	3,8

12.	Libertex	20	4,6
13.	Monese	29	3,2
14.	Šiaulių bankas	8	4,1
15.	Luminor Nordea Lietuva	8	4,2
16.	SEB	20	4,7
17.	1Money	9	4,6
18.	TransferWise Money Transfer	17	4,3
19.	TransferGo	21	4,6
20.	Fast Budget	9	4,8

12 lentelė. „Android“ programėlės iš kategorijos „Ryšiai“

3.	**Ryšiai**		
Nr.	Programėlės pavadinimas	Pavojingumo taškai	Naudotojų vertinimas
1.	Facebook Messenger	79	4,4
2.	Viber Messenger	56	4,4
3.	WhatsApp Messenger	53	4,4
4.	Messenger Lite	35	4,5
5.	Discord	21	4,4
6.	Telegram	47	4,4
7.	Naršyklė Opera	29	4,2
8.	Messenger: All-In-One Messaging	11	4,4
9.	Naršyklė Firefox	33	4,4
10.	Free Adblocker Browser	39	3,8
11.	Mobile Number Locator	36	4,3
12.	Messenger for SMS	86	4,3
13.	Kontaktai	26	4,3
14.	Call Recorder	34	4,5
15.	Naršyklė Opera Mini	28	4,1
16.	Mobile Location Tracker	25	4,1
17.	Opera Touch	13	3,9
18.	Mano Labas	8	4,6
19.	Signal Private Messenger	73	4,6
20.	Mail.ru	30	4,6

13 lentelė. „Android“ programėlės iš kategorijos „Švietimas“

4.	**Švietimas**		
Nr.	Programėlės pavadinimas	Pavojingumo taškai	Naudotojų vertinimas
1.	WordBit	14	4,7
2.	PhotoMath	4	4,5
3.	Pinkfong Baby Shark	9	4,7

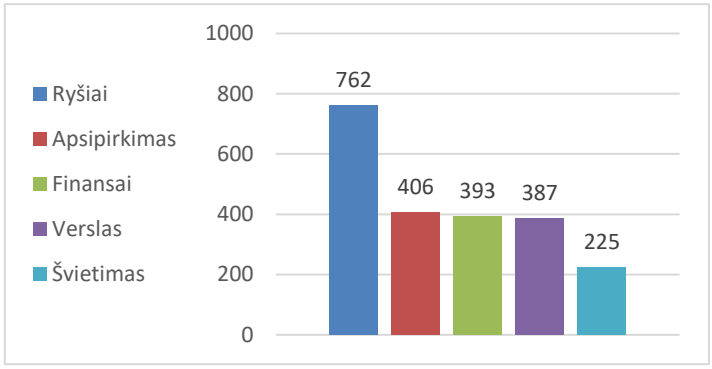
4.	Duolingo: Learn Language Free	22	4,6
5.	Anglų kalba pradedantiesiems	10	4,1
6.	TO-FU Oh!SUSHI	9	1,9
7.	TAMO išmaniesiems	8	3,8
8.	Miga Town: My Pets	13	4
9.	TO-FU OH!Fire	9	4,2
10.	Main Street Pets Village	1	4,2
11.	Kahoot!	9	4,7
12.	KETBILIETAI testai B kat 2019	9	3,1
13.	Mano dienynas	30	4,1
14.	Pizza Chef	0	4,7
15.	Calculator+	22	4,5
16.	Pingfong Singing Phone	13	4,3
17.	Kids Learn Professions	13	4,5
18.	Sago Mini Holiday Trucks and Diggers	9	4,3
19.	Mokomes skaityti	16	4,7
20.	Learn Languages with Memrise	5	4,7

14 lentelė. „Android“ programėlės iš kategorijos „Verslas“

5.	**Verslas**		
Nr.	Programėlės pavadinimas	Pavojingumo taškai	Naudotojų vertinimas
1.	WordBit	14	4,7
2.	PhotoMath	4	4,5
3.	Pinkfong Baby Shark	9	4,7
4.	Duolingo: Learn Language Free	22	4,6
5.	Anglų kalba pradedantiesiems	10	4,1
6.	TO-FU Oh!SUSHI	9	1,9
7.	TAMO išmaniesiems	8	3,8
8.	Miga Town: My Pets	13	4
9.	TO-FU OH!Fire	9	4,2
10.	Main Street Pets Village	1	4,2
11.	Kahoot!	9	4,7
12.	KETBILIETAI testai B kat 2019	9	3,1
13.	Mano dienynas	30	4,1
14.	Pizza Chef	0	4,7
15.	Calculator+	22	4,5
16.	Pingfong Singing Phone	13	4,3
17.	Kids Learn Professions	13	4,5
18.	Sago Mini Holiday Trucks and Diggers	9	4,3
19.	Mokomes skaityti	16	4,7

20.	Learn Languages with Memrise	5	4,7

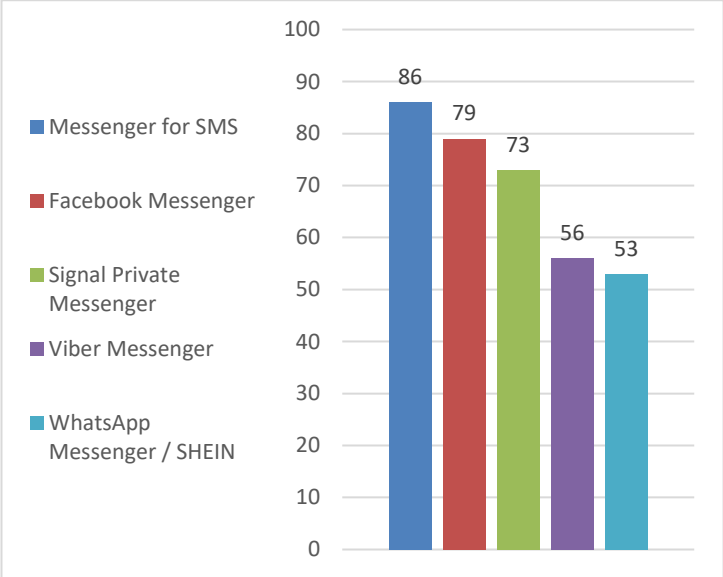
1. Tikrintų programėlių kategorijos pagal informacijos nutekinimo pavojų išsidėsto tolesne tvarka. Kategorijos pavojingumas apskaičiuojamas susumuojant visų 20 populiariausių programėlių pavojingumo taškus.



4.1 pav. Kategorijos pagal pavojingumo taškus

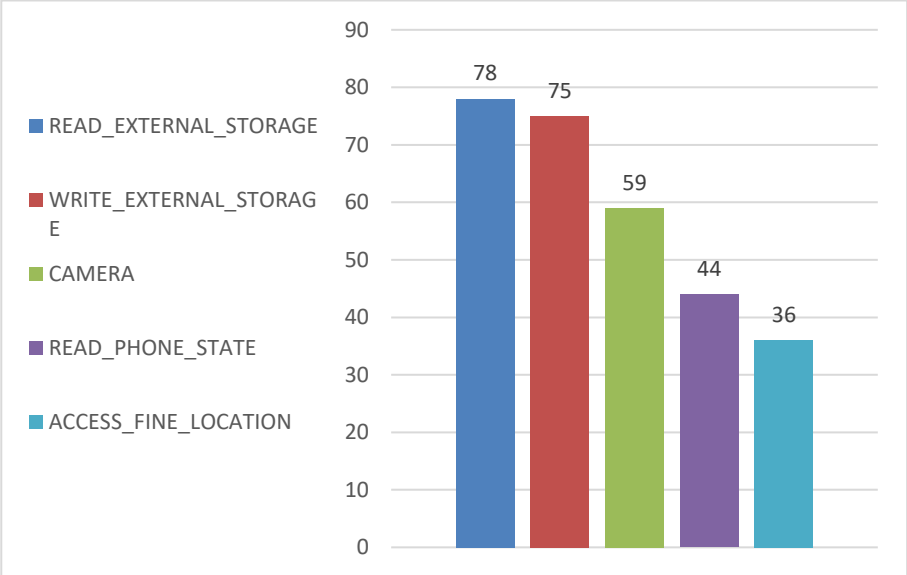
Pavojingiausia kategorija, kaip matyti, iš tirtų kategorijų yra „Ryšiai“. Jai reikia daugiausiai leidimų, todėl turi daugiausiai galimybių nutekinti jautrią informaciją.

2. Šios programėlės kelia didžiausią informacijos nutekinimo pavojų. Jų iš viso tikrinta 100. Kaip galima pastabėti iš šios diagramos, daugiausiai leidimų prašo bendravimui skirtos programėlės, kurios priklauso kategorijai „Ryšiai“.



4.2 pav. Pavojingiausios programėlės

3. Toliau nurodomi pavojingi leidimai, kurių programėlės prašo dažniausiai. Kaip matyti, dažniausiai prašoma galimybės nuskaityti duomenis iš išorinės laikmenos. Beveik tiek pat dažnai prašoma leidimo įrašyti duomenis į išorinę laikmeną. Šie 2 leidimai lyg ir nėra išskirtinai pavojingi, bet įvairiuose šaltiniuose pabrėžiama, kad išorinėje laikmenoje saugomi duomenys yra mažiau saugūs, nes laikmeną galima lengvai išimti ir nuskaityti bet kuriuo kitu įrenginiu, jei ji nėra šifruota.



4.3 pav. Dažniausiai prašomi leidimai

4. Programėlių kūrėjams yra svarbus geras jų programėlės įvertinimas „Android Play Store“ parduotuvėje. Tai užtikrina, kad naudotojai labiau pasitikės programėle, jie labiau ją norės įdiegti, o kūrėjai žinos, kad visi darbai tinkamai atlikti. Tikėtina, kad gero įvertinimo yra siekiama įvairiomis priemonėmis. Savo naudotojo žinojimas gali suteikti privalumų. Taigi, ar didesnis informacijos nutekinimas, paprašius daugiau leidimų, leidžia programėlių kūrėjams pelnyti daugiau patikimo žvaigždučių „Play Store“ parduotuvėje?

15 lentelė. Ryšys tarp informacijos nutekinimo ir programėlės įvertinimo

Vieta	Įvertinimas „Play Store“	Pavojingumas	Programėlė
1	4,8	4	Mano akcijos
2	4,8	41	Revolut
3	4,7	13	ASOS
4	4,7	8	Joom
5	4,7	22	Duolingo: Learn Language Free

Visgi kaip matyti iš šios lentelės, kuri sudaryta pagal aukščiausią programėlių įvertinimą, labiausiai patinkančios naudotojams programėlės nutekina gan mažai duomenų. Jų pavojingumo taškų kiekis žemas. Iš to galima spręsti, kad naudotojo palankumui užsitarnauti neužtenka vien kuo daugiau apie juos žinoti.

4.3. Kenkėjiškų programėlių informacijos nutekinimo galimybių tyrimas

Laikomasi nuomonės, kad saugiausia atsisiųsti programėles iš „Play Store“, bet įvairūs šaltiniai teigia, kad vargiai įmanoma užtikrinti absoliutų saugumą. Šaltinyje [25] teigiama, kad produktų vadovas Andrew Ahn nurodo, jog 2017 m. jie iš „Play Store“ pašalino 700 000 kenkėjiškų programėlių. Tai sudaro maždaug 2 000 programėlių per dieną. Taip pat Andrew Ahn nurodo, kad jie aptiko 99 % kenkėjiškų programėlių ir pašalino jas prieš suteikdami galimybę įkelti jas į „Play

Store“. Bet 1 % procentas programėlių – o tai maždaug 20 per dieną – vis tiek buvo neaptikta ir jas galėjo atsisiųsti nieko neįtariantys naudotojai.

Šio eksperimentinio tyrimo apie kenkėjiškų programėlių galimybes nutekinti „Android“ įrenginyje saugomą informaciją iškelti tikslai yra tokie:

1. Ištirti kenkėjiškas programėles prototipu „Leidimų monitoringo sistema“ ir apskaičiuoti pavojingumo taškus.
2. Nustatyti dažniausiai tiriamų kenkėjiškų programėlių prašomus leidimus.
3. Palyginti kitų atliktų mokslinių tyrimų rezultatus dėl galimybės nutekinti jautrią informaciją su rezultatais, gautais naudojant prototipą.

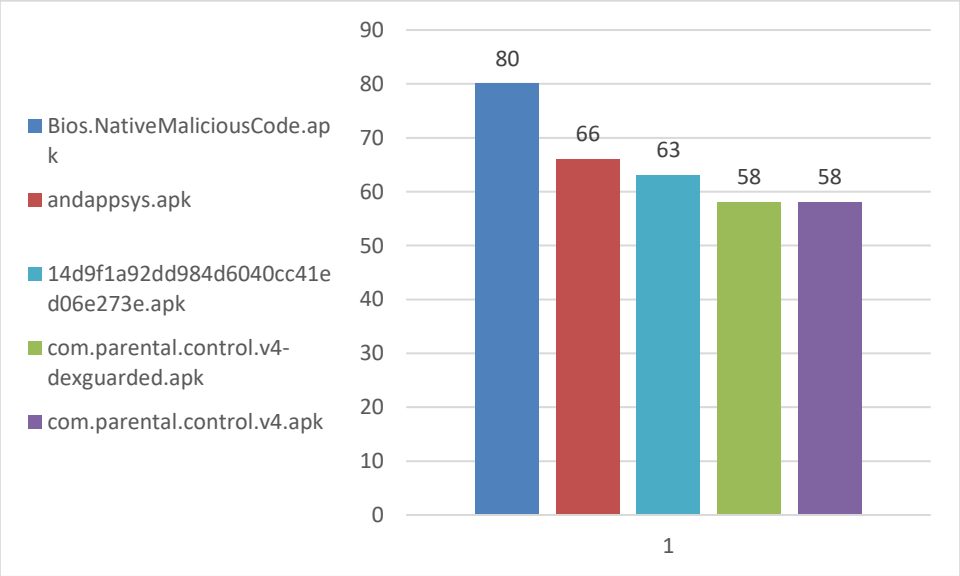
Kad būtų galima atlikti tyrimą, reikia atsisiųsti ir įdiegti veikiančias kenkėjiškas programėles. Jos atsisiųstos iš „GitHub“ [26]. Tyrimui naudota 41 kenekėjiška programėlė.

16 lentelė. Tirtos kenkėjiškos „Android“ programėlės

Nr.	**Programėlės pavadinimas**	**Pavojingumo taškai**
1.	com.fdhgkjhrtjkjbx.model.apk	14
2.	com.c101421042723.apk	35
3.	BadNews.A.apk	13
4.	Bios.NativeMaliciousCode.apk	80
5.	Masnu.apk	47
6.	Omigo.apk	31
7.	SmsThief.apk	16
8.	SmsWorker.apk	31
9.	Claco.A.apk	37
10.	DropDialer.apk	9
11.	FakeBank.B.apk	44
12.	FakeCMCC.A.apk	24
13.	Agent.apk	20
14.	Blatantsms.apk	46
15.	FakeDoc.apk	31
16.	FakeValidation.apk	49
17.	Fobus.apk	40
18.	Opfake.apk	16
19.	JiFake.A.apk	20
20.	NotCompatible.A.apk	0
21.	Obad.A.apk	39
22.	Oldboot.A.apk	15
23.	Samsapo.A.apk	40
24.	AndroidFileSystem.apk	24
25.	andappsys.apk	66

26.	XTaoAD.A.apk	40
27.	com.parental.control.v4-dexguarded.apk	58
28.	com.parental.control.v4.apk	58
29.	Google-play.apk	33
30.	org.benews.apk	12
31.	brother.apk	4
32.	mcpef.apk	4
33.	14d9f1a92dd984d6040cc41ed06e273e.apk	63
34.	021d55c415ff951c8e7b1ce3f94399bb.apk	55
35.	skype.apk	4
36.	com.cattss.apk	29
37.	com.chistescortos.apk	29
38.	com.chistespicanticos.apk	29
39.	com.funnyys.apk	29
40.	krep.itmtd.ywtjexf-1.apk	43
41.	towelroot.apk	12

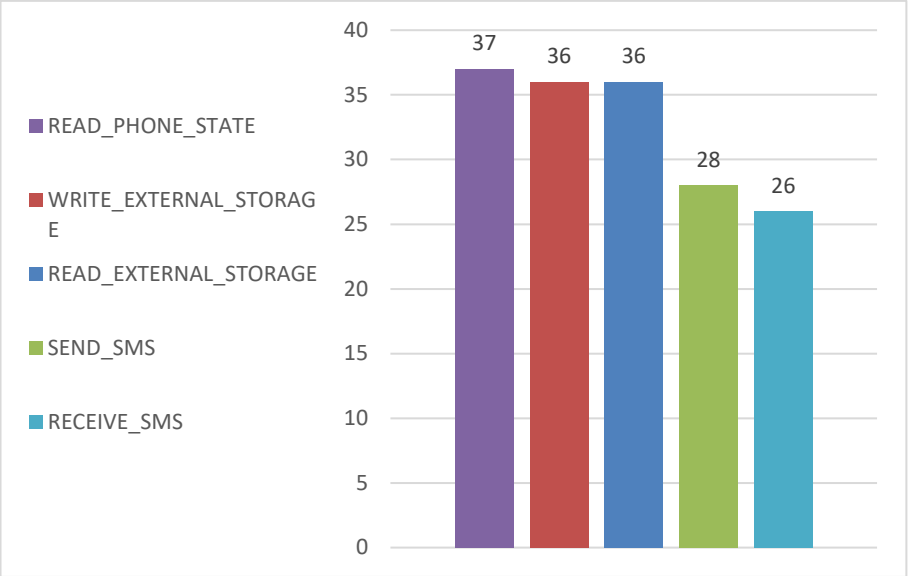
1. Prototipu įvertintos kenkėjiškų programėlių galimybės nutekinti jautrią informaciją. Visgi reikėtų atsižvelgti į tai, kad pavojingumo taškai arba programėlės naudojami leidimai **gali ir neatspindėti tikro programėlės pavojingumo**. Nors naudotojas ir suteikia leidimus kenkėjiškai programėlei kaip įprastai programėlei, nes neįtaria, kad naudojasi kenkėjiška programėle, bet nei vienu, nei kitu atveju naudotojas nežino, kaip tie leidimai yra naudojami. Kenkėjiška programėlė „SmsThief.apk“ su 16 pavojingumo taškų neatrodo labai pavojinga, bet jei ji be naudotojo žinios siunčia SMS žinutes gavėjams iš užsienio šalių, į kurias tarifai yra labai aukšti, galima patirti daug finansinių nuostolių.



4.4 pav. Daugiausiai informacijos nutekinančios „Android“ kenkėjiškos programėlės

Pagal pavojingumo taškus pateikiamos 5 kenkėjiškos programėlės iš 41 tirtos, kurios nutekina daugiausiai duomenų.

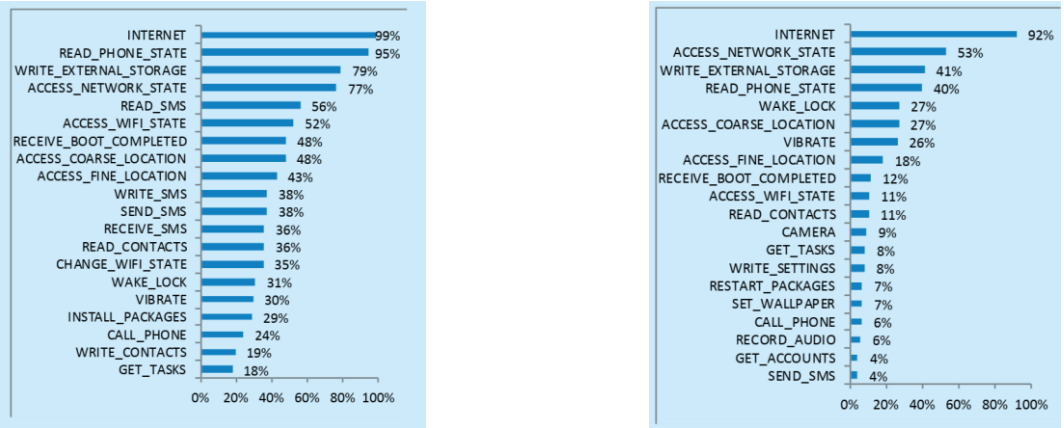
2. Atlikus kenkėjiškų programėlių tyrimą, nustatyti dažniausiai jų prašomi leidimai. Pastebėta, kad itin dažnai prašomas leidimas yra INTERNET. Kadangi prieiga prie interneto yra itin svarbi tiek įprastoms programėlėms, tiek kenkėjiškoms, ir kadangi leidimas INTERNET nėra priskiriamas prie pavojingų leidimų, nuspręsta į ji neatsižvelgti, nes jis nesuteikia galimybės daryti konkrečių išvadų.



4.5 pav. Dažniausiai prašomi „Android“ kenkėjiškų programėlių leidimai

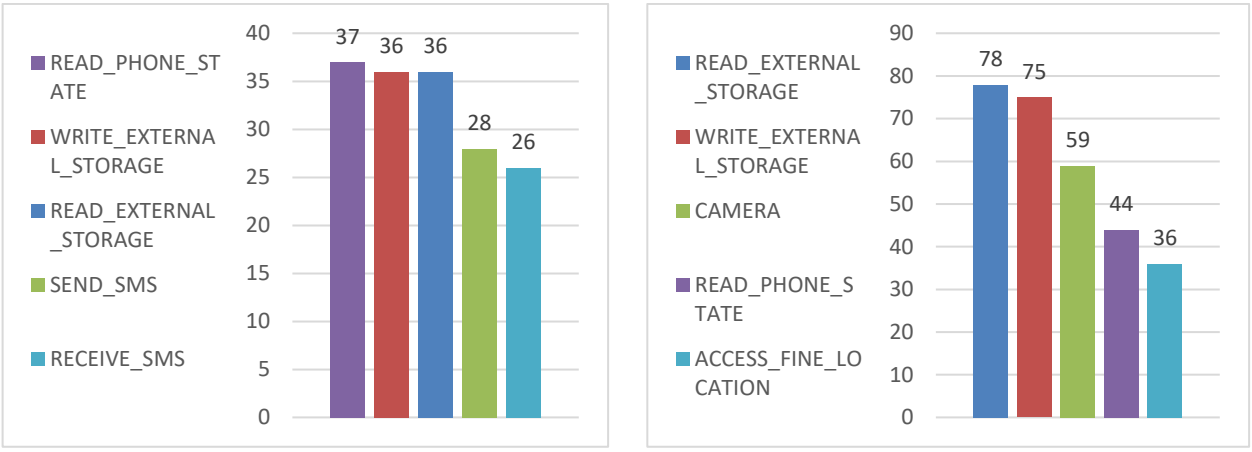
Kaip matyti iš tyrimo rezultatų, daugiausiai tirtos kenkėjiškos programėlės prašė leidimo READ_PHONE_STATE. Taip pat labai populiarūs leidimai, skirti įrašyti ir nuskaityti duomenis iš išorinės laikmenos. Leidimų siųsti ir gauti SMS prašė daugiau nei pusė tirtų programėlių. Prielaidai dėl SMS leidimų populiarumo būtų galima pasitelkti statistiką [27], kurioje teigiama, kad „SophosLab“ įrašė beveik 1,5 milijono unikalių „Android“ kenkėjiškų programėlių pavyzdžių. 55,6 % iš jų buvo skirtos naudotojui nežinant siųsti ir gauti SMS. Kaip teigiama, taip yra greičiausiai dėl to, kad kibernetiniams nusikaltėliams yra vienas iš lengviausių būdų pasipelnyti.

3. Kaip teigiama atliktame tyrime [28], kenkėjiškų programėlių prašomi leidimai akivaizdžiai skiriasi nuo įprastų programėlių dažniausiai prašomų leidimų. Palyginus nurodyto tyrimo gautą statistiką, matyti, kad didžiausias skirtumas yra tarp leidimo READ_PHONE_STATE, nes jo prašė 95 % kenkėjiškų programėlių, o įprastų – tik 40 %. Dažnai prašyti leidimai READ_SMS ir WRITE_SMS, nors įprastų programėlių statistikoje jų iš viso nėra.



4.6 pav. Kairėje – kenkėjiškų programėlių leidimai, dešinėje – įprastų (kitų tyrėjų duomenys)

Atlikus tyrimą prototipu „Programėlių monitoringo sistema“ ir palyginus 5 populiariausius leidimus, pastebėta panašumų. Tik šiuo atveju didesnis dėmesys skirtas pavojingiems leidimams, nes jie turi daugiau galimybių nutekinti jautrią informaciją.



4.7 pav. Kairėje – kenkėjiškų programėlių leidimai, dešinėje – įprastų (atlikus tyrimą)

Kaip matyti, labiausiai kenkėjiškas programėles domina leidimas READ_PHONE_STATE, išskyrus leidimą INTERNET. Taip pat dažnai naudojami SMS, išorinės laikmenos leidimai. O leidimas CAMERA visai nedomina arba beveiks visai nedomina, nors jis gan įprastas tarp nekenksmingų programėlių.

Vien iš leidimų nėra galimybės tiksliai nuspręsti, ar programėlė kenkėjiška. Kenkėjiškos programėlės turi sau būdingus tipiškus leidimus, lyginant statistiškai, bet kiekviena programėlė yra unikali, todėl jos leidimų derinys taip pat gali būti vienoks ar kitoks.

Visgi verta paminėti, kad žinant programėlės paskirtį ir žinant jos leidimus, galima spręsti, ar ji tikrai sukurta nurodomai paskirčiai. Pavyzdžiui, jei išmaniojo telefono fonų paveikslėliams sukurta programėlė prašo leidimo siųsti arba gauti SMS, galima įtarti, jog ši programėlė galbūt nėra tokia, kokios tikimasi.

Taip pat prototipe „Leidimų monitoringo sistema“ galima nustatyti daugiau pavojingumo taškų atskirai tiems leidimams, kurie yra populiariausi tarp kenkėjiškų programėlių. Tai naudotoją informuotų ne tik apie nutekinamos informacijos tipą, bet ir gal būt paskatintų pasinaudoti antivirusine programine įranga, kuri tiksliau įvertintų įtartiną programėlę.

4.4. Šnipinėjimo programėlių tyrimas

„Android Play Store“ programėlių parduotuvėje galima rasti daug ir įvairių šnipinėjimo programėlių. Oficialiai jos pristatomos kaip suteikiančios galimybę stebėti savo vaikus arba darbuotojus, naudojančius įmonės išmaniuosius įrenginius. Visgi realybė gali būti ir kitokia. Tyrime [29] teigiama, kad smurtaujantys partneriai yra linkę įdiegti tokias programėles smurtą patiriančio asmens įrenginyje. Tai suteikia jiems galimybę savo auką sekti, perimti žinutes, skambučių žurnalus ir kitą informaciją. Vėliau toks jautrios informacijos nutekinimas priveda prie dar didesnio smurto.

Ši situacija yra būdinga komercinėms sekimo programėlėms, bet nereikėtų pamiršti ir kenkėjiškos programinės įrangos, sukurtos sekimo tikslais. Ją nieko neįtariantis naudotojas gali atsisiųsti ir įdiegti savo įrenginyje, kuri taip pat nekontroliuojamai nutekintų jautrią asmeninę informaciją. Tik šiuo atveju nebūtų apsiropojama artima aplinka, kai ją įdiegia vienas iš artimų žmonių.

Šioje tyrimo dalyje dėmesys bus skiriamas komercinėms sekimo programėlėms. Išsikelti tikslai yra tokie:

1. Ištirti komercines šnipinėjimo programėles prototipu „Leidimų monitoringo sistema“ ir apskaičiuoti pavojingumo taškus.
2. Nustatyti dažniausiai tiriamų komercinių šnipinėjimo programėlių prašomus leidimus.

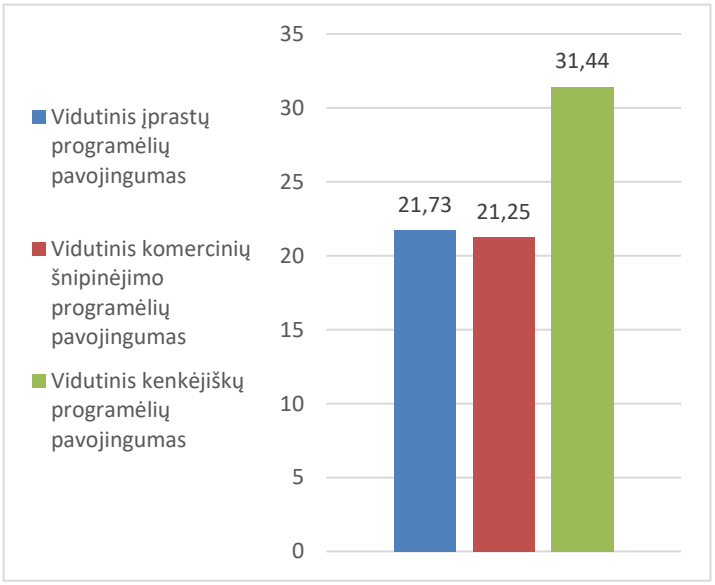
Kad būtų galima atlikti tyrimą, iš „Play Store“ programėlių parduotuvės arba kitų parduotuvių ar programėles sukūrusių organizacijų svetainių atsisiųstos tolesnės programėlės. Jos pasirinktos atsitiktine tvarka, įvedus į paieškos laukelį atitinkamus raktinius žodžius.

17 lentelė. Tirtos komercinės sekimo programėlės

Nr.	Programėlės pavadinimas	Pavojingumo taškai
1	Find Device	8
2	AllTracker Family	45
3	GPS Locator	41
4	Life360	41
5	Mobile Location Tracker	37
6	Mobile Phone Number	36
7	GeoZilla	33
8	Remore Monitoring	29
9	Safe365	29
10	Fameelee	28
11	FindMyPhone	28

12	TrackView	28
13	Mobile Tracker for Android	20
14	Cell Phone Tracker	17
15	Employee Work Spy	17
16	Family Link	17
17	FamilyTime Dashboard	17
18	SMS Tracker	17
19	Spy Tracker	17
20	mLite	17
21	SafeMinor	16
22	Phone Tracker: Family Locator	8
23	Chat Message Tracker	4
24	Message and Call Tracker	4
25	WW Parent App	4
26	Texting, Chat, Phone Spy	17
27	Phone Tracker	11
28	Parent App	9
	Iš viso	595

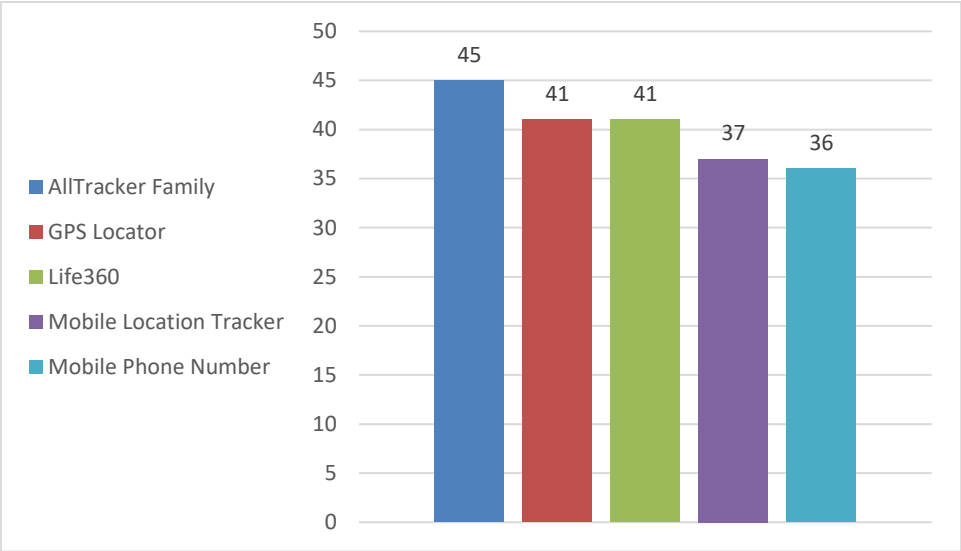
1. Ištyrus komercines sekimo programėles, matyti, kad jų vidutinis pavojingumas pagal prašomus leidimus nė kiek neišsiskiria iš kitų programėlių. Šiuo atveju lenkia bendravimui skirtos programėlės iš „Play Store“ kategorijos „Ryšiai“. Jos linkusios prašyti daug leidimų, todėl ir tikimybė, kad nutekins informacija, yra didesnė.



4.8 pav. Vidutinis atskirų programėlių grupių pavojingumas

Vidutiniškai trečdaliu daugiau leidimų prašo tik kenkėjiškos programėlės. Bet vien leidimais įvertinti programėlės pavojingumą būtų sunku, nes viskas priklauso nuo to, kur ir kaip tie jautrūs

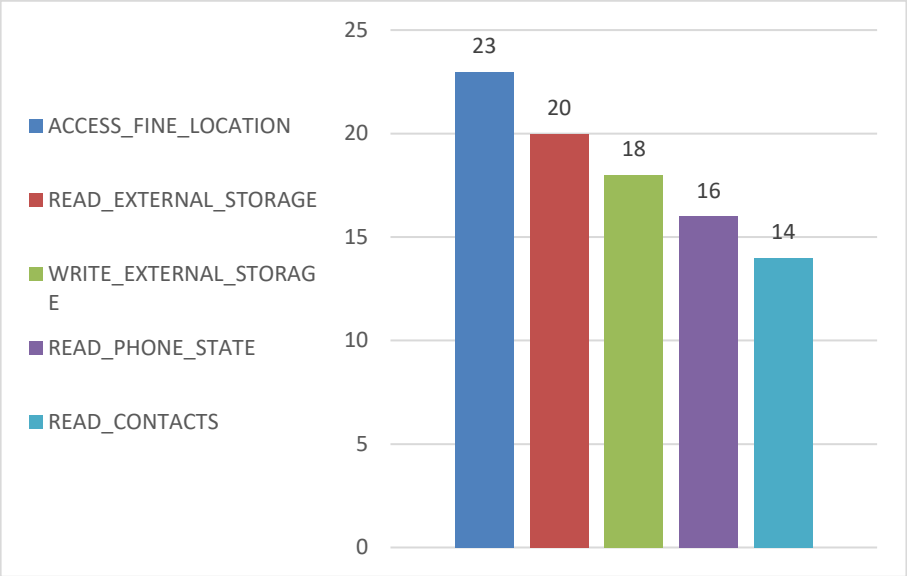
nutekinti duomenys yra naudojami. Jei, kaip buvo nurodyta tyrime [29], sekimo programėlė naudojama smurtaujant artimoje aplinkoje, pasekmės gali būti nenuspėjamos.



4.9 pav. Komercinės sekimo programėlės pagal pavojingumo taškus

Pagal pavojingumo taškus sekimo programėlės neatrodo išskirtinės, bet tyrimą ribojo faktas, kad daugiausiai sekimo funkcijų turinčios programėlės kaip „mSpy“ ir panašios yra brangiai kainuojančios. Jų nėra galimybės atsisiųsti iš „Play Store“. Dėl to neįmanoma sužinoti, kokių leidimų jos prašo. Taip pat, vien naudojantis leidimais, apie sekimą būtų galima ir nieko nesužinoti, jei naudojamos tokios gamyklinės funkcijos kaip „Samsumg“ gamintojo „Ieškoti mano mobiliojo“. Jos įjungiamos ir išjungiamos per nustatymus. Jų paskirtis – apsaugoti savo pamestą išmanųjį įrenginį. Visgi savo paskyroje galima užregistruoti artimo asmens įrenginį ir per platformą https://findmymobile.samsung.com stebėti jį nuotoliniu būdu kaip savo pamestą įrenginį, taip pat atlikti tam tikras operacijas.

2. Sekimo programėlės, kaip ir būtų galima tikėtis, prašo savo paskirčiai būdingų leidimų. Joms svarbu sužinoti tikslią įrenginio naudotojo vietą, todėl pirmauja leidimas ACCESS_FINE_LOCATION. Taip pat dažnai prašo prieigos prie vidinės ir išorinės atminties, bet tai būdinga ir visoms kitoms programėlėms. Tipiškas leidimas šioms programėlėms dar būtų READ_CONTACTS, nes kitų grupių programėlės jo vidutiniškai prašo rečiau.



4.10 pav. Dažniausiai komercinių sekimo programėlių prašomi leidimai

4.5. Eksperimentinės dalies išvados

1. Eksperimentinėje dalyje atlikti 3 tyrimai: (1) dažniausiai naudojamų „Android“ programėlių tyrimas, (2) „Android“ kenkėjiškų programėlių tyrimas ir (3) „Android“ šnipinėjimo programėlių tyrimas. Iš viso ištirta 100 saugių programėlių, 41 kenkėjiška programėlė ir 28 komercinės sekimo programėlės. Tiriamųjų programėlių leidimams nuskaityti naudotas prototipas „Programėlių monitoringo sistema“.
2. Iš 5 tirtų saugiųjų programėlių kategorijų matyti, kad labiausiai išsiskiria kategorija „Ryšiai“. Bendravimui skirtos programėlės linkusios daugiausiai prašyti leidimų, todėl jos turi daugiausiai galimybių nutekinti jautrią informaciją. Ir nors didesnis žinojimas apie savo klientą visada būna naudingas, programėlių atveju šio ryšio nepastebėta. Aukšto įvertinimo sulaukusios programėlės „Play Store“ parduotuvėje neišsiskyrė dideliu leidimų kiekiu.
3. Nors remiantis oficialiais pranešimais „Google“ kiek įmanoma stengiasi atpažinti kenkėjiškas programėlės ir jas pašalinti, tai ne visada pavyksta [25]. Vidutiniškai 1 % kenkėjiškų programėlių lieka „Play Store“ ir jas nieko neįtariant galima atsisiųsti. Šios programėlės prašo vidutiniškai 1/3 daugiau leidimų nei kitos. Joms būdingos savitos prašomų leidimų kombinacijos, kurioms, prototipą tobulinant, būtų galima suteikti didesnį pavojingumo taškų skaičių. Nors tai neatstotų saugumo įvertinimo antivirusine programa, bet labiau paskatintų atkreipti dėmesį į konkrečias programėles.
4. Sekimo programėlės oficialiai pristatomos kai priemonės stebėti vaikus arba įmonės darbuotojus, besinaudojančius darbo išmaniuoju įrenginiu, bet niekas netrukdo jas panaudoti prieš nieko neįtariančius artimos aplinkos žmones. Šių programėlių išskirtinis bruožas tai, kad dažniausiai prašoma prieigos prie tikslių įrenginio vietos koordinačių. Jei įrenginyje aktyvintos gamyklinės funkcijos, leidžiančios sekti įrenginį kaip pamestą, to galima nepastebėti net ir skenuojant leidimus trečiųjų šalių programėlėmis.

5. IŠVADOS

1. Šiuo metu pasaulyje yra daugiau nei 2 000 000 000 [1] „Android“ įrenginių, remiantis 2017 m. teiginiais. „Android“ operacinė sistema yra atvirojo kodo, kuri sukurta „Linux“ pagrindu. Programėlės atsisiunčiamos ir įdiegiamos iš „Google Play Store“ arba kitų šaltinių.
2. „Android“ OS saugumas pagrįstas leidimų modeliu. Tai reiškia, kad programėlė gali naudoti atitinkamą informaciją, jei jai suteiktas leidimas. Įvairiuose tyrimuose pabrėžiama, kad naudotojai dažniausiai nežino, kaip ta informacija yra naudojama ir kas kiek laiko. Žinoma tik tai, kad ją naudoti yra suteiktas leidimas.
3. Nekenksmingos programėlės yra dažnai linkusios prašyti daugiau leidimų nei joms reikia, todėl padidėja jautrios informacijos nutekinimo pavojus. Kenkėjiškos programėlės savo ruožtu jautrią informaciją gali nutekinti nekontroliuojamai, naudotojui nieko apie tai neįtariant. Anot „Google“ produktų vadovo Andrew Ahn žodžiais, nepavyksta pašalinti visų kenkėjiškų programėlių prieš jas įkeliant į „Play Store“ [25]. 1 % kenkėjiškų programėlių patenka į „Play Store“, todėl naudotojų jautri informacija gali patekti į neteisėtas rankas. O šnipinėjimo programėlės, nors oficialiai pristatomos kaip priemonė stebėti vaikus arba darbuotojus, gali būti naudojamos sekti artimos aplinkos žmonės, sužinant apie juos beveik viską: skambučius, žinutes, buvimo vietos koordinates ir t. t.
4. Informacijos jautrumo požymiams sudaryti ir jos jautrumui įvertinti pasiūlytas V-S ašių metodas [21]. Informacijos vertei naudojama X ašis su 3 reikšmėmis: nesvarbi (0), vidutiniškai svarbi (1) ir labai svarbi (2). Informacijos jautrumui pagal leidimus naudojama Y ašis su taip pat 3 reikšmėmis: įprasti leidimai (0), kai kurie įprasti leidimai, galintys sukelti rūpesčių (1) ir pavojingi leidimai (2).
5. Pasiūlytas prototipas, kuris veikia ašių V-S pagrindu. Pavojingumo taškai arba galimybės nutekinti jautrią informaciją apskaičiuojamos pagal leidimų pavojingumo (Y) ir informacijos vertės (X) reikšmių sandaugas. Gauta taškų suma lyginama su didžiausia pavojingumo taškų suma, lygia 134. Įdiegtos programėlės po skenavimo pateikiamos sąraše, o šalia jų dešinėje pusėje pateikiamas grafinis pavojingumo atvaizdavimas.
6. Eksperimentinę dalį sudaro 3 tyrimai, naudojant prototipą: (1) dažniausiai naudojamų „Android“ programėlių tyrimas, (2) „Android“ kenkėjiškų programėlių tyrimas ir (3) „Android“ šnipinėjimo programėlių tyrimas. Iš viso ištirta 100 saugių programėlių, 41 kenkėjiška programėlė ir 28 komercinės sekimo programėlės.
7. Pastebėta, kad iš 5 tirtų programėlių kategorijų, pasirinktų „Play Store“ atsitiktine tvarka, labiausiai linkusios nutekinti informaciją yra iš kategorijos „Ryšiai“. Jos skirtos bendravimui kaip „Facebook Messenger“, „Viber Messenger“. Visgi didesnis informacijos nutekinimas neprisideda prie to, kad naudotojai suteiktų daugiau patikimo žvaigždučių.
8. Kenkėjiškos programėlės vidutiniškai prašo 1/3 daugiau leidimų, bet vien leidimai neatspindi jų jautrios informacijos nutekinimo galimybių, nes viskas priklauso nuo to, kaip nutekinta informacija naudojama. Pavyzdžiui, SMS kenkėjiška programėlė gali prašyti mažai leidimų ir atrodyti nepavojinga, bet ji gali paslapčia siųsti SMS žinutes bangiais numeriais.
9. Sekimo programėlės dažniausiai prašo ACCESS_FINE_LOCATION leidimo, leidžiančio sužinoti tikslią vietą. Leidimų skaičius visgi gali priklausyti nuo to, kiek programėlė gali atlikti funkcijų. Kai kurioms gali tekti gauti prieigą prie ROOT (šakninio) katalogo. Nutekinama gali būti pati įvairiausia informacija, bet didžiausias apribojimas, kad tokią programėlę reikia įdiegti į įrenginį. Dėl to jos naudojimo galimybės apsiriboja artimų žmonių ratų.

10. Tiek kenkėjiškoms programėlėms, tiek sekimo programėlėms, tiek ir atskirtų kategorijų programėlės yra būdingos tam tikros prašomų leidimų kombinacijos. Kaip tolesnį prototipo patobulinimo galimybę būtų galima toms tipiškoms leidimų kombinacijoms suteikti daugiau taškų arba išskirti jas kitu būdu. Tai leistų vadovautis prielaidomis, kad tam tikra programėlė gali būti kenkėjiška, todėl ją reikėtų įvertinti antivirusine programa ir pan.
11. Jei naudojamos tokios gamyklinės funkcijos kaip „Samsumg“ gamintojo „Ieškoti mano mobiliojo“ arba panašios, vien leidimų stebėjimu jų nepavyks aptikti. Jos įjungiamos ir išjungiamos per nustatymus. Jų paskirtis – apsaugoti pamestą išmanųjį įrenginį. Savo paskyroje užregistravus artimo asmens įrenginį, galima per platformą https://findmymobile.samsung.com stebėti jį nuotoliniu būdu kaip savo pamestą prietaisą, taip pat atlikti tam tikras veiksmus.

6. LITERATŪRA

[1] Joe Rossignol, „Google Says There Are Now More Than 2 Billion Monthly Active Android Devices“, 2017 m. [Tinkle]: https://www.macrumors.com/2017/05/17/2-billion-active-android-devices/ [Kreiptasi 2019 04 30]

[2] ACM DIGITAL LIBRARY, „State of the art smart spaces application models and software infrastructure“, 2006 m. [Tinkle]: http://ubiquity.acm.org/article.cfm?id=1167869 [Kreiptasi 2017 11 10]

[3] IEEEXplore, „Evaluating critical security issues of the IoT world: Present and Future challenges“, 2017 m. [Tinkle]: http://ieeexplore.ieee.org/document/8086136/ [Kreiptasi 2017 11 10]

[4] IEEEXplore, „IoT Security: A Layered Approach for Attacks & Defenses“, 2017 International Conference on Communication Technologies (ComTech), [Tinkle]: http://ieeexplore.ieee.org/document/8065757/ [Kreiptasi 2017 11 12]

[5] IEEEXplore, „Critical Review of Static Taint Analysis of Android Applications for Detecting Information Leakages“, 2017 8th International Conference on Information Technology (ICIT), [Tinkle]: http://ieeexplore.ieee.org/document/8080041/ [Kreiptasi 2017 11 12]

[6] IEEEXplore, „Mobile Malware and Smart Device Security: Trends, Challenges and Solutions“, 2013 m. [Tinkle]: http://ieeexplore.ieee.org/document/6569314/ [Kreiptasi]: 2017 11 13.

[7] ACM DIGITAL LIBRARY, „Smartphone Security: Review of Challenges and Solution“, 2016 m. [Tinkle]: https://dl.acm.org/citation.cfm?id=2905214&CFID=833008306&CFTOKEN=40518203 [Kreiptasi]: 2017 11 13

[8] ACM DIGITAL LIBRARY, „A Comparison of Features for Android Malware Detection“, 2017 m. [Tinkle]: https://dl.acm.org/citation.cfm?id=3077288&CFID=1005942318&CFTOKEN=47079180 [Kreiptasi 2017 11 14]

[9] „Forbes“, „Report: 97% Of Mobile Malware Is On Android. This Is The Easy Way You Stay Safe“, 2017 m. [Tinkle]: https://www.forbes.com/sites/gordonkelly/2014/03/24/report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe/#13fe25cf2d4f , [Kreiptasi 2017 11 14]

[10] „Mcafee“. „Mobile Threat Report: Whats on the Horizon for 2016“, 2016 m. [Tinkle]: http://www.mcafee.com/us/resources/reports/rp-mobile-threat-report-2016.pdf , [Kreiptasi 2017 11 14]

[11] „Sophos“. „When Malware Goes Mobile“, 2016 m. [Tinkle]: https://www.sophos.com/en-us/security-news-trends/security-trends/malware-goes-mobile.aspx, [Kreiptasi 2017 11 14]

[12] IEEEXplore, „Malware Detection in Android based on Dynamic Analysis“, 2017 m. [Tinkle]: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8074847, [Kreiptasi 2017 11 15]

[13] ACM DIGITAL LIBRARY, „Dynamic Permissions based Android Malware Detection using Machine Learning Techniques“, 2017 m. [Tinkle]: https://dl.acm.org/citation.cfm?id=3021485, [Kreiptasi 2017 11 17]

[14] M., Schultz, E., Eskin, F., Zadok, and S. Stolfo, „Data Mining Methods for Detection of New Malicious Executables,“ 2001 m. [Tinkle]: https://scholar.google.lt/scholar?q=Data+Mining+Methods+for+Detection+of+New+Malicious+Executables&hl=en&as_sdt=0&as_vis=1&oi=scholart&sa=X&ved=0ahUKEwiP9_rcy8HXAhUsIJoKHWPNDjQQgQMILjAA, [Kreiptasi]: 2017 11 15

[15] IEEEXplore, „Analysis of Machine Learning Techniques Used in Behavior Based MalwareDetection“, 2010 m. [Tinkle]: http://ieeexplore.ieee.org/document/5675808/ [Kreiptasi]: 2017 11 15

[16] IEEEXplore, „Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms“, „International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)“, 2017 m. [Tinkle]: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8074847, [Kreiptasi 2017 11 15]

[17] ACM DIGITAL LIBRARY, „A novel rough set 64tribute reduction based on Ant Colony Optimisation“, „International Journal of Intelligent Systems Technologies and Applicaitons“, 2015 m. [Tinkle]: https://dl.acm.org/citation.cfm?id=2886072, [Kreiptasi 2017 11 15]

[18] „The Google Android Security Team’s Classifiations for Potentially Harmful Applications“. 2017 m. https://source.android.com/security/reports/Google_Android_Security_PHA_classifications.pdf

[19] „StackExchange Network“, „Android Folder Hierarchy“. [Tinkle]: https://android.stackexchange.com/questions/46926/android-folder-hierarchy/46934#46934, [Kreiptasi 2018 01 05]

[20] Rohit Tamma, Donnie Tindall; „Learning Android Forensics“, 2015 m. Spausdino: „PACKT publishing“.
[21] IEEEXplore, „Research on Supply Chain Information Classification Based on Information Value and Information Sensitivity“, 2007 m. [Tinkle]: http://ieeexplore.ieee.org/document/4280248/, [Kreiptasi 2018 01 06]
[22] Google.Android Reference:Manifest File-Permissions, 2018. [Tinkle]: http://developer.android.com/guide/topics/manifest/manifest-intro.html. [Kreiptasi]: 2018 05 23
[23] Jinxin Zhang, Xiaohui Yang, Tao Li ir Jiamin Bao, „A detection system of Android application based on permission analysis“, 2014 m. „Communications Security Conference“ (CSC 2014), Pekinas, 2014 m. [Tinkle]: https://ieeexplore.ieee.org/document/6992223. [Kreiptasi]: 2018 05 23.
[24] „Protection levels“, 2018 m. [Tinkle]: https://developer.android.com/guide/topics/permissions/overview#normal-dangerous. [Kreiptasi]: 2018 06 05 https://developer.android.com/reference/android/Manifest.permission
[25] Patrick Howell O'Neill, „Google killed 700,000 malicious apps in the Play Store in 2017“ [Tinkle:] https://www.cyberscoop.com/android-malware-google-play-store-2017/. [Kreiptasi]: 2019 04 08.
[26] „GitHub, Inc.“, „Ashishb android malware“. [Tinkle]: https://github.com/ashishb/android-malware, [Kreiptasi]: 2019 04 08.
[27] Rowland Yu & William Lee, „Will Android Trojans, Worms or Rootkits Survive in SEAndroid and Containerization?“, 2016 m. [Tinkle]: https://www.virusbulletin.com/virusbulletin/2016/02/vb2015-paper-will-android-trojans-worms-or-rootkits-survive-seandroid-and-containerization/, [Kreiptasi]: 2019 04 08.
[28] IEEEXplore, „Android malware detection with contrasting permission patterns“, 2014 China Communications. [Tinkle]: https://ieeexplore.ieee.org/document/6911083 [Kreiptasi 2019 04 09]
[29] R. Chatterjee et al., „The Spyware Used in Intimate Partner Violence“, 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, 2018 m., pp. 441-458. [Tinkle]: https://ieeexplore.ieee.org/document/8418618. [Kreiptasi]: 2019 04 27.

7. PRIEDAI

Visi magistrinio darbo priedai, kaip ir pats darbas elektroniniame formate, yra į rašyti į kompaktinę plokštelę. Priedus sudaro:

1. Mokslinėje konferencijoje IVUS 2019 pristatytas straipsnis „Composition of the Information Security Methods for a Smart Environment and the Research“. Nuoroda: http://ivus.vdu.lt/wp-content/uploads/2019/04/IVUS-2019-conference-proceedings.pdf
2. Mokslinė konferencijoje „Lietuvos magistrantų informatikos ir IT tyrimai“ pristatyta santrauka „Security Permission Monitoring Method for Smart Devices“.
3. Moksliniam leidiniui „International Journal of Computer Trends & Technology“ publikuoti paruoštas straipsnis „Android Information Leak Potential in Benign, Malware, and Commercial Spyware Applications“. Nuoroda: https://www.ijcttjournal.org/archives/ijctt-v67i5p105
4. Kompaktinė plokštelė su visais duomenimis.

7.1. Mokslinė konferencija IVUS 2019.

IVUS
2019

CERTIFICATE OF PARTICIPATION

This is to certify that

NERIJUS ŠATKAUSKAS
Kaunas University of Technology

attended the 24th International
Conference on Information Technology
IVUS 2019
in Kaunas, Lithuania, April 25, 2019
and presented the paper

Nerijus Šatkauskas „Composition of the information security methods for a smart enviroment and the research“

Dean of Vytautas Magnus University
Faculty of Informatics
Prof. Dr. Daiva Vitkutė-Adžgauskienė

Programme Committee Chair
Prof. Dr. Tomas Krilavičius

VYTAUTAS MAGNUS UNIVERSITY
ktu kaunas university of technology 1922
Kaunas Faculty
UNIVERSITÀ degli STUDI di CATANIA
Silesian University of Technology

Composition of the Information Security Methods for a Smart Environment and the Research

Nerijus Šatkauskas
Department of Computer Sciences
Kaunas University of Technology
Kaunas, Lithuania
nerijus.satkauskas@ktu.edu

***Abstract*—Smart devices and the smart environment itself is getting more and more popular. A big part of smart devices uses the Android operating system. Since any information on these devices can become available to the third parties on the basis of granted permissions, it is very important to consider it properly before granting them. A permission monitoring system prototype has been proposed for this purpose.**

***Keywords*—dangerous permission group, dangerous permission, information leakage, android operating system, smart environment, smart device, information value, information sensitivity, Android permissions, permission monitoring**

I. INTRODUCTION

Smart environment is rather an abstract conception and it may refer to a number of more specific areas in question. If we referred to one of many definitions for the smart environment, it would sound like [1] "ordinary environments equipped with visual and audio sensing systems, pervasive devices, sensors and networks that can perceive and react to people…". It is expected that the number of such devices will only increase in the future.

One of the smart devices which makes a big part of the smart environment is a smartphone. A dominating operating system currently is Android [2]. This operating system has been created by Google on the basis of Linux. The operating system due to its nature of being an open source one has to be well controlled and maintained in order to keep it as safe as possible.

The purpose of this research is to analyze security issues the Android operating system faces. It assesses the security of the smart environment information storage in the Android operating system. It attempts to detect whether any unauthorized parties can get an access to this information. The methods which may strengthen the security are considered.

A prototype has been proposed for this purpose. This prototype shall classify the tested applications based on their permissions which suggest any potential information leakage. The results will be compared with some other applications which are currently available on the Play Store for the same purpose.

II. SMART ENVIRONMENT THREATS

Mobile devices once were considered as safe ones but everything has changed as soon as operating systems were introduced. Installing an application is not only an additional comfort. It can be an additional concern as well. Especially if it is a malware which can leak any information.

IoT environment or the smart environment in this particular case since the issues are rather common can be divided into three main levels [3]: application level, transportation level and perception level. All these three levels bear threats which are typical to them.

TABLE I. SMART ENVIRONMENT THREAT LEVELS

Layer	Main Threats
Application level – provides customer requested services like air temperature	Data leakage: stealing data
	DoS attacks: making services unavailable
	Malicious code injection: exploiting known vulnerabilities
Transportation level – transmits and receives any collected data	Routing attacks: intermediate malicious nodes
	DoS attacks: making nodes unavailable
	Data transit attacks: attacks in networks
Perception level – physical sensors to collect any data and to process it	Physical attacks: node tempering, replacing
	Impersonation: fake identity for attacks
	DoS attacks: making nodes unavailable
	Routing attacks: intermediate malicious nodes
	Data transit attacks: sniffing, man-in-the-middle

This research focuses on the application level. The operating system Android is picked due to its leading positions in the market.

III. ANALYSIS OF THE CURRENTLY AVAILABLE ANDROID DATA LEAKAGE MONITORING TOOLS

Data availability to third parties in the Android operating system relies on the permission model [4]. Permissions are such labels which should be assigned by developers to their application. The application must define in the manifest file which sensitive resources it needs to have an access to. The user during the installation has a chance either to grant these permissions or not.

A. Preinstalled permission manager

As Android 6.0 "Marshmallow" has been introduced in 2015, the ability was provided to toggle any granted dangerous permission groups for any specific application [5]. The accessibility of this tool may vary depending on the manufacturer of a device, but it can be accessed in general via Settings > Apps / Application Manager > Permissions.

A screenshot is provided below of the operating system Android 8.1.0. It gives an access to the list of all the installed applications. Dangerous permission groups can be reviewed, granted or revoked at any time.

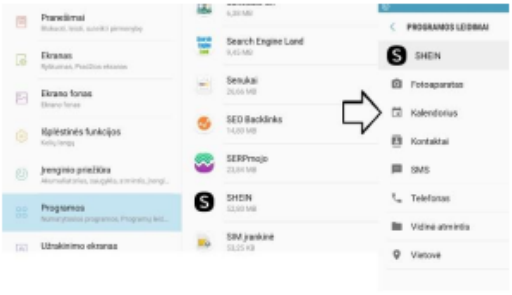


Fig. 1. Preinstalled permission manager

B. *Application Inspector*

A good alternative which is available on Play Store for the permission management is Application Inspector. This is a third-party application which is developed by UBQSoft.

The tool once it is launched provides a list of all the installed applications. One can see more details after picking any particular application within that list concerning libraries, last update time etc. Involved permissions are described as well as their level they belong to is provided: dangerous, normal, signature. The status of granted or not granted is available which can be changed after tapping and being directed to relevant Settings submenus.

Fig. 2. Application Inspector

C. *Apk Analyzer*

It is a very extensive analyzer and it provides an access to different statistical data after a specific application is picked within a general scanned applications list. There is a tab for used permissions. These permissions are listed after tapping the tab, but the information resources are very limited. There are no descriptions about these permissions. Which level they belong to is undefined. There is no information if any of these permissions in the manifest file are granted or not.

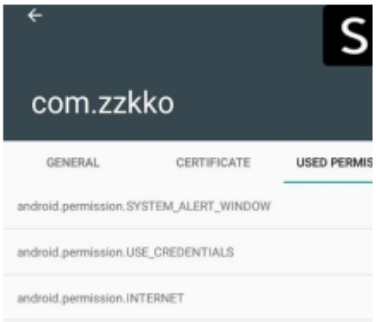


Fig. 3. Apk Analyzer

D. *PackageInfo*

Another application which can be helpful for scanning any installed applications on the device is PackageInfo. It gives a list of applications after scanning which are available for a more detailed review after picking any of them. It gives some package information, including the list of permissions. There are no detailed descriptions of these permissions. The state whether they are granted or not is unidentified.

GIDs	3003
Split Names	Unavailable
Split Revision Codes	Unavailable
Version Code	251
Version Name	6.5.0
Base Revision Code	0
Permissions	android.permission.SYSTEM_ALERT_ android.permission.USE_CREDENTIAI android.permission.INTERNET android.permission.WRITE_EXTERNA android.permission.READ_EXTERNAL android.permission.CAMERA android.permission.GET_ACCOUNTS

Fig. 4. PackageInfo

It becomes obvious after the analysis of some currently available tools for permission scanning and monitoring that the focus given on the permissions may not be enough for a regular user. A regular user may not want to search for any explanatory information about the granted permissions in external sources. It may lead the user to underestimating any potential threat due to personal information leakage.

IV. V-S AXIS INFORMATION SENSITIVITY ASSESSMENT

Different data classification methods were taken into consideration but V-S method [6] was chosen as the most appropriate one in this case. This method classifies any available information based on 2 axis which stand for information value and sensitivity. As the authors suggest who have introduced this method it is possible to assign the data to different information classes while implementing different security measures.

A. *V-S axis method in the prototype*

In order to able to use the proposed V-S axis method for the data on Android device, we first need to define the value of the vertical axis for information **sensitivity (Y)**. Sensitivity axis has tree levels: low (0), middle (1), and high (2).

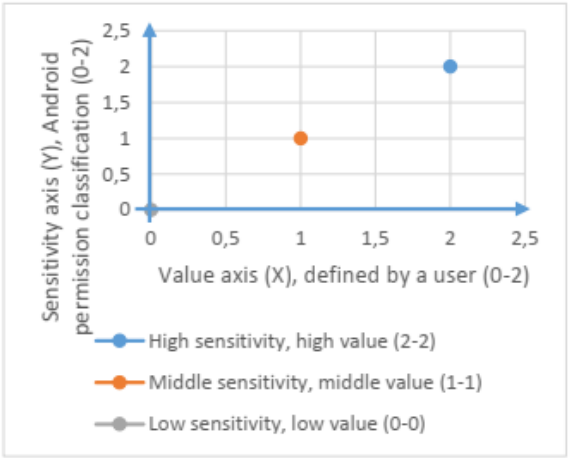


Fig. 5. V-S axis chart

The horizontal axis for information **value (X)** has also three levels. The levels are correspondingly: low (0), middle (1), and high (2).

The official classification of permissions available on Android developers portal was used for that purpose [7]. Permissions are classified there into four groups: normal, dangerous, signature, and special ones. This official classification reflects different information sensitivity levels to any potential information leakage. These permissions were assigned to the **sensitivity (Y) axis** in the following manner:

1) *Low (0):* Normal permissions are assigned to this level due their low potential threat. These permissions are granted to any installed application on a smart device without any intervention on the user side.

2) *Middle (1):* Some normal permissions are assigned to this level. Applications with these permissions may cause some inconvenience to users like CHANGE_NETWORK_STATE which allow to change the connectivity to wireless networks.

3) *High (2):* Dangerous permissions groups were assigned to this level. It is officially confirmed and classified as having negative impact once the information which belongs to the above class is unintentionally exposed to any third part parties.

Signature permissions and special permissions were not further considered in this research. Therefore, they were not assigned to any axis level.

The **horizontal (X) axis** for information value is used for a personal assessment of the information stored on the smart device. The values for this axis are selected by default in the prototype but a user can change them any time.

1) *Low (0):* This information is not valuable to the user or the user will not have any significant issues upon losing it. Permissions of low sensitivity (Y) axis level are matched to this value (X) axis level by default which results in 0 as a score.

2) *Middle (1):* This information might have some value to the user or the user might have some issues upon losing it. Permissions of middle sensitivity (Y) axis level are matched to this value (X) axis level by default which results in 1 as a score.

3) *High (2):* This information is valuable to the user. Losing it might cause considerable issues or financial losses. Permissions of high sensitivity (Y) axis level are matched to this value (X) axis level by default which results in 4 as a score.

B. Proposed prototype based on V-S classification

The proposed prototype Permission Monitoring System gives a quick review of the installed applications. It consists of 2 main lists. The first one is made of applications which are sorted in the order of the highest danger point score to the lowest one. A total danger point score for a specific application is compared to the maximum possible danger point score (maximum point score is 134). It gives that way a quick review of the data leakage potential.

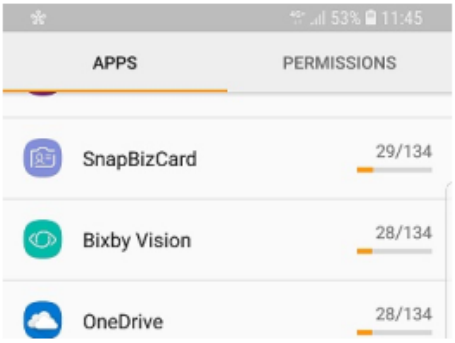


Fig. 6. Permission Monitoring System

V-S axis classification method is used both for the maximum danger point score calculation and for the current danger point score calculation of any specific application. As mentioned above, permission classification and default information values which a user can adjust to his / her own priority any time are taken into consideration.

If a user taps any application in the application list provided by the prototype, further options are available. The user can see the package name, version number, last update time etc. It also provides the number of dangerous, potentially dangerous and normal permissions. These permissions can be further explored after tapping their titles in this submenu.

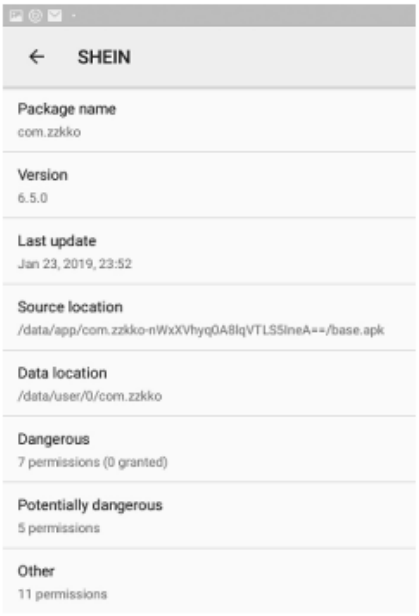


Fig. 7. Specific application data

As soon as a submenu option for permissions is tapped, one can see the dangerous permissions which of them are namely granted. If these dangerous permission groups are not granted but they are still in the list, it means that the manifest file contains that permission group, and as the application is used, sooner or later this permission group will be requested by the corresponding application. It is also the place where a user can change the value (horizontal one) axis level to a preferred one if he / she thinks that the default value does not meet his / her expectations. E.g. if a user feels that there is no important information in his / her contacts book and exposing it unintentionally to any third parties is not a big concern, it is

possible to change the value axis level to the middle one or the low one. Danger point score will be recalculated accordingly.

C. V-S classification of permissions

The following default values were used to calculate the score for any permission used within an application.

TABLE II. MAX. OF POINTS FOR A SPECIFIC PERMISSION

Dangerous permissions	2	0	2	4
Potentially dangerous	1	0	1	2
Normal permissions	0	0	0	0
		0	1	2
		Low value	Average value	High value

The maximum amount of points for a **dangerous** permission is 4. Meanwhile, the maximum amount of points for **potentially dangerous** permissions is 2.

The following formula was used to calculate the danger point score:

$$(Y_D * X_D) + (Y_{PD} * X_{PD})$$

All the permission groups which belong to the dangerous protection level are used for this prototype. As it was mentioned above, they belong to level High (2) on the sensitivity (Y) axis. Further details are provided below.

TABLE III. PERMISSION GROUPS AND MAX. POINT SCORE

Permission group	Permissions and max. score on both axis			
	Permissions	Y_D	X_D	*Multiplication*
CALENDAR	READ_CALENDAR	2	2	4
	WRITE_CALENDAR	2	2	4
CALL_LOG	READ_CALL_LOG	2	2	4
	WRITE_CALL_LOG	2	2	4
	PROCESS_OUTGOING_CALLS	2	2	4
CAMERA	CAMERA	2	2	4
CONTACTS	READ_CONTACTS	2	2	4
	WRITE_CONTACTS	2	2	4
	GET_ACCOUNTS	2	2	4
LOCATION	ACCESS_FINE_LOCATION	2	2	4
	ACCESS_COARS_LOCATION	2	2	4
MICROPHONE	RECORD_AUDIO	2	2	4
PHONE	READ_PHONE_STATE	2	2	4
	READ_PHONE_NUMBERS	2	2	4
	CALL_PHONE	2	2	4
	ANSWER_PHONE_CALLS	2	2	4
	ADD_VOICEMAIL	2	2	4
	USE_SIP	2	2	4
SENSORS	BODY_SENSORS	2	2	4
SMS	SEND_SMS	2	2	4
	RECEIVE_SMS	2	2	4
	READ_SMS	2	2	4
	RECEIVE_WAP_PUSH	2	2	4
	RECEIVE_MMS	2	2	4
STORAGE	READ_EXTERNAL_STORAGE	2	2	4
	WRITE_EXTERNAL_STORAGE	2	2	4
Maximum point score for dangerous permissions				104

Some normal protection level permissions are used for the sensitivity (Y) axis with the default value set to Middle. These values officially are considered as not dangerous, but a user may find it uncomfortable if their status becomes uncontrollable. Therefore, the level on the sensitivity axis (Y) is 1, and the level on the value axis (X) which can be changed by a user is also 1 by default. However, this default value is considered as 2 when calculating the maximum danger point score. The following table provides further calculation details.

TABLE IV. MAX. SCORE FOR POTENTIALLY DANGEROUS

Permissions	Y_{PD}	X_{PD}	Multiplication
CHANGE_NETWORK_STATE	1	2	2
CHANGE_WIFI_STATE	1	2	2
MODIFY_AUDIO_SETTINGS	1	2	2
REQUEST_DELETE_PACKAGES	1	2	2
NFC	1	2	2
REORDER_TASKS	1	2	2
REQUEST_INSTALL_PACKAGES	1	2	2
FLASHLIGHT	1	2	2
GET_TASKS	1	2	2
BILLING	1	2	2
SET_ALARM	1	2	2
DISABLE_KEYGUARD	1	2	2
SET_WALLPAPER	1	2	2
SYSTEM_ALERT_WINDOW	1	2	2
WRITE_SETTINGS	1	2	2
Maximum point score for potentially dangerous permissions			30

TABLE V. TOTAL MAXIMUM SCORE

Maximum danger point score	Max. score
Maximum point score for dangerous permissions	104
Maximum point score for potentially dangerous permissions	30

Maximum danger point score	Max. score
Maximum point score for dangerous permissions + potentially dangerous permissions	134

The maximum danger point score therefore is 134. If a user changes the level on the value (X) axis for any dangerous permission group or a potentially dangerous permission to low, it means that this permission will be multiplied by 0 which leads this permission to be unconsidered in the total danger point score for applications.

V. EXPERIMENTAL FINDINGS

The purposes of completing information leakage experiments based on permissions were the following ones:

1) Which categories do pose the highest risk of an information leakage among the tested ones?
2) Which applications do pose the highest risk of an information leakage among the tested ones?
3) Which permissions are requested the most frequently?

The following devices were used in one or other way in order to download the applications for testing them with the prototype.

TABLE VI. USED DEVICES

Device	Basic specifications
Lenovo Yoga 530	Windows Pro 10 Intel® Core™ i3-8130U CPU @ 2,20 Ghz 16,0 GB RAM
Samsung Galaxy S8	Android 8.0.0 Octa-core (2.3GHz Quad + 1.7GHz Quad), 64 bit, 10nm processor 4 GB RAM (LPDDR4)
Samsung Tab A (SM-T585)	Android 8.1.0 Octa-core (4x1.6 GHz Cortex-A53 & 4x1.0 GHz Cortex-A53) 3 GB RAM

Applications were downloaded based on different categories. Applications within the categories were picked while using the most popular application list since these applications are the most relevant ones to the biggest number of users.

The most popular 20 applications from the categories below were downloaded and installed.

- Shopping
- Finance
- Communication
- Education
- Business

Tested categories according to the results of the information leakage risk are distributed on the chart in the following way.

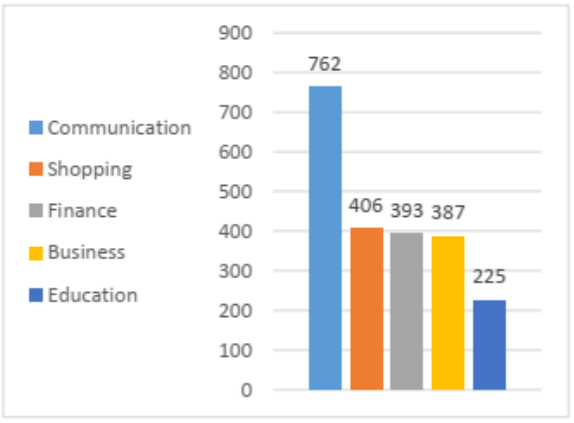


Fig. 8. Distribution of the tested categories

There results were calculated by summing up the danger point score of all the tested applications within that category. It was 20 top applications in it based on their popularity.

The following applications pose the highest risk of an information leakage among the tested ones.

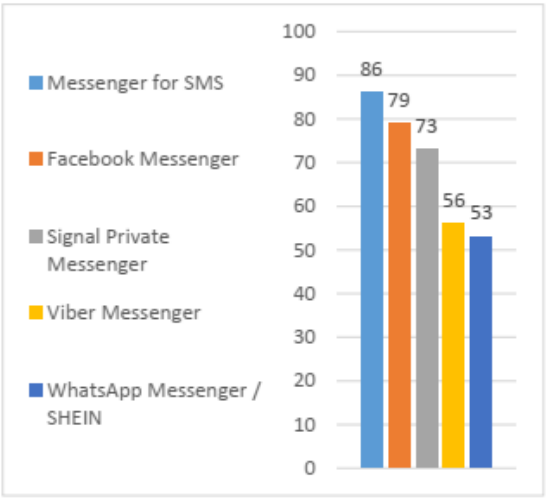


Fig. 9. The most dangerous applications

These applications were picked by looking for the highest danger point score among all the tested applications. The number of the tested applications is 100 at the moment.

The following dangerous permissions are requested the most frequently by the downloaded applications which were used for this research.

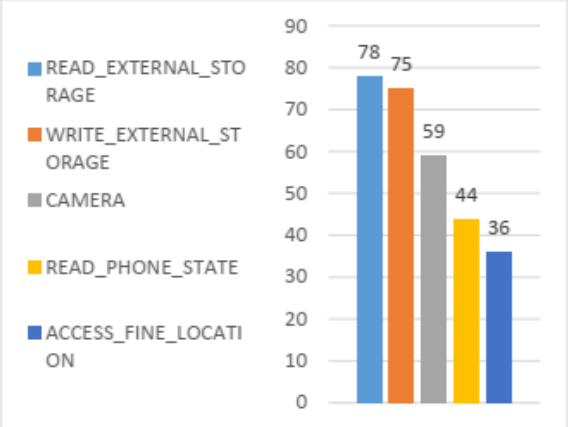


Fig. 10. The most frequent permissions

Numbers of the usage of different dangerous permissions were calculated in this test. As 100 Android applications were currently tested in this research, the chart numbers suggest the amount of instances the corresponding permission was requested or was to be requested. It means in this case that the permission READ_EXTERNAL_STORAGE was requested by 78 applications out of 100 tested applications. Top 5 permissions with the highest usage number were picked.

VI. CONCLUSIONS

Android OS security is based on the permission model. However, granting the permissions can be underestimated by a regular user due to a lack of available information or interest in his/her personal security.

A prototype has been offered which provides a simple risk assessment of any information leakage. A user does not need to have any awareness of permissions to understand the results.

100 applications in total from 5 different categories were tested. The results are provided in charts for a comparative purpose.

REFERENCES

[1] (2006) ACM DIGITAL LIBRARY, "State of the art smart spaces application models and software infrastructure". [Online]. Available: http://ubiquity.acm.org/article.cfm?id=1167869

[2] (2017) IEEEXplore, "Critical Review of Static Taint Analysis of Android Applications for Detecting Information Leakages", 8th International Conference on Information Technology (ICIT). [Online]. Available: http://ieeexplore.ieee.org/document/8080041/

[3] (2017) IEEEXplore, "Evaluating critical security issues of the IoT world: Present and Future challenges". [Online]. Available: http://ieeexplore.ieee.org/document/8086136/

[4] (2017) IEEEXplore, "Android Permissions Unleashed". [Online]. Available: https://ieeexplore.ieee.org/document/7243742

[5] Google Play Help, "Control your app permissions on Android 6.0 and up", [Online]. Available: https://support.google.com/googleplay/answer/6270602?hl=en-GB

[6] (2007) IEEEXplore, "Research on Supply Chain Information Classification Based on Information Value and Information Sensitivity". [Online]. Available: http://ieeexplore.ieee.org/document/4280248/

[7] (2018) "Protection levels". [Online]. Available: https://developer.android.com/guide/topics/permissions/overview#normal-dangerous

7.2. Mokslinė konferencija „Lietuvos magistrantų informatikos ir IT tyrimai“

Pažymėjimas

Nerijus Šatkauskas
Kauno technologijos universitetas

dalyvavo Lietuvos mokslų akademijos
mokslinėje konferencijoje

Lietuvos magistrantų informatikos ir IT tyrimai

ir skaitė pranešimą

Security Permission Monitoring Method for Smart Devices

Konferencijos organizacinio komiteto pirmininkas
akad. GINTAUTAS DZEMYDA

2019 m. gegužės 14 d.
Vilnius

Security Permission Monitoring Method for Smart Devices

Nerijus Šatkauskas

Kaunas University of Technology, Faculty of Informatics, Studentų g. 50, Kaunas

nerijus.satkauskas@ktu.edu

SUMMARY

It is said that there are over 2 billion of smartphones [1] currently in the world. To be more accurate, it can be about 2.7 billion in 2019. A smartphone is known to have an operating system which enables a user to download and install any required application. This comfort does come at a price. Android operating system which is among the leading ones for its popularity has also its vulnerabilities. Some resources may say that Google trades its users' privacy... [2]. Android OS is based on permissions. If a user grants certain permission, relevant information can be used by that application. Sensitive information can be exposed.

Another concern is malware. Andrew Ahn, the product manager, has said that they had removed 700 000 malware applications from their Play Store in 2017 before it was uploaded. It makes 99 %. But 1 % was still not removed on time. Malware is even more dangerous due to its uncontrollable sensitive information leak.

The purpose of this research is to propose a method to asses any information leak in Android devices. A prototype which allows to assess the extent of an information leak shall be offered. It calculates a total danger point score based on requested permissions (on Y axis) and information value (on X axis).

V-S axes method [3] was used for this purpose. Permission sensitivity (Y) axis: low (0 points), middle (1 point), and high (2 points). Official Android permissions were distributed on this axis [4]. Information value (X) axis: low (0 points), middle (1 point), and high (2 points). Default values were used for this axis, but a user can change it any time.

100 applications as benign ones were tested. They were downloaded from Play Store while picking 20 top applications from 5 random categories. 41 malware applications were downloaded [5]. Top 5 applications among the tested ones which have the highest information leak potential have been listed. 5 most frequent permissions for benign and malware applications were detected.

Table 1. Top 5 applications and permissions.

Highest leak potential in danger point score	The most common permissions in benign	The most common permissions in malware
Messenger for SMS	READ_EXTERNAL_STORAGE	READ_PHONE_STATE
Facebook Messenger	WRITE_EXTERNAL_STORAGE	WRITE_EXTERNAL_STORAGE
Signal Private Messenger	CAMERA	READ_EXTERNAL_STORAGE
Viber Messenger	READ_PHONE_STATE	SEND_SMS
WhatsApp Messenger	ACCESS_FINE_LOCATION	RECEIVE_SMS

The prototype allows to assess quickly any sensitive information leak potential by giving a total score in points. The research gives a list of applications one has to pay an increased attention to in order to avoid any leak. The most common permission patterns may suggest using an anti-virus application for a malware scan.

Literature

[1] Deuthche Welle, (2019) "Smartphones: Live longer, be greener". [Online]. Available at: https://www.dw.com/en/smartphones-live-longer-be-greener/a-46423527.

[2] Kaspersky Lab DAILY, "Google Trades Privacy and Security for Hangouts". Available at https://www.kaspersky.com/blog/google-privacy-hangouts/1993/.

[3] X. Shi, D. Li, H. Zhu and W. Zhang, "Research on Supply Chain Information Classification Based on Information Value and Information Sensitivity," 2007 International Conference on Service Systems and Service Management, Chengdu, 2007, pp. 1-7. Available at: http://ieeexplore.ieee.org/document/4280248/

[4] "Protection levels". Available at: https://developer.android.com/guide/topics/permissions/overview#normal-dangerous

[5] GitHub, Inc., "Ashishb Collection of Android Malware Samples". Available at: https://github.com/ashishb/android-malware

7.3. „International Journal of Computer Trends & Technology“ straipsnis

International Journal of Computer Trends and Technology (IJCTT) – Volume 67 Issue 5- May 2019

Android Information Leak Potential in Benign, Malware, and Commercial Spyware Applications

Nerijus Šatkauskas[#1]

Department of Computer Sciences , Kaunas University of Technology
Kaunas, Lithuania

Abstract

There are well over 2 billion smartphones currently in the world. Their number is only increasing.A big part of OS is Android. It is not only an OS with huge resources. Android is notorious for an increased information leak potential. Information availability is based on granted permissions, but a user may underestimate it due to a lack of interest or skills. Application developers are often blamed for asking too many permissions. Meanwhile malware and commercial spyware means that the information leak in question is uncontrollable.
Permission Management System, the prototype, has been offered which gives a simplified review of any potential information leak due to permissions. A comparative study has been completed on benign, malware and commercial spyware applications.

Keywords *— Android, permission,dangerous permission, information leak, smartphones,permission monitoring, permission management, benign, malware, commercial spyware*

I. INTRODUCTION

As soon as smartphones emerged, they have become an inseparable part of our lives. It is said that the number of currently used smartphones in the world is well over 2 billions [1]. The comfort they give though is not for free. It comes at a price. And we pay with our privacy for this.

There were severalOSfor smartphones since their introduction but few of them survived up to the current day. One of these OS is Android.It was 2008 when the first device was launched with it. Ever since, it is getting bigger, more powerful and more popular. Now it is more than a smartphone OS. It is available in wearable electronics, IoT, TV boxes etc.

Plenty of scientific studies have been completed due to Android OS privacy concerns. Google Play Store is a relatively safe place for downloading any application. They have a variety of security procedures to check the uploaded applications. However, there were some reports that a malware is not always removed on time to prevent users from getting infected. And repackaged applications seem to be the easiest method to inject any malicious code into an application which looks like an original one. One study claims that about 86 % of malware samples [2] were repackaged applications.

It is not only malware which is know for information leakage to third parties. Benign Android applications are also notorious for this. One can hear such claims like “Google trades privacy and security for… [3]”. In addition to this, there are plenty spying applications which cantransmit your location, messages, calls and other data. The primary goal of spying applications is to monitor your children or to keep track of businesssmartphones [4] as it is officially suggested, but the truth is they are often used to spy on somebody you may have an access to his/her devices to configure it for, and it can be a victim in this particular case [5].

This research aims to analyze any information leak potential the Android OS faces throughout the usage of different applications. It will focus on the methods of the information leak, and the type of data which can be made available to any unauthorized persons.

The experimental part includes a prototype which was used to test the applications. Benign and spying applications were downloaded from Google Play Store and their official dealers, meanwhile malware applicationswere obtained from ashishb [6] collections of Android malware samples.

II. INFORMATION LEAK THREATS

Benign applications. Android OS security is based on a permission model as an application is downloaded and installed. Developers of any Android application are required to define in the AndroidManifest.xml file the permissions which their application will need to run correctly. These permissions may not be required immediately. A request to grant it will made to the user if he/she uses a particular function which needs specific hardware resources. Once these permissions are granted, the application is enabled to transmit any corresponding data to relevant third parties.

TABLE I
ANDROID PERMISSION MODEL

Permissions	Details
Normal	They are expected to pose very low risk. A system will grant them automatically at the moment the application is being installed. It cannot be cancelled. E.g. SET_ALARM
Signature	These permissions are also given at the time of an installation but it should be there the compliance of the certificates E.g. READ_VOICEMAIL
Special	Special permissions actually belong to the signature permissions but they act slightly in a different way. They are extra sensitive, therefore applications would rather avoid asking for them SYSTEM_ALERT_WINDOW and WRITE_SETTINGS
Dangerous	These permissions are organized in certain groups. If one permission is granted to an application, the other permissions within that group is also granted. E.g. READ_CONTECTS and WRITE_CONTECTS

Officially these permissions are divided into 4 classes [7]. However, there are 3 protection levels. Special permissions do not have a separate protection level. The purpose of these permissions and classification of them is to protect the privacy of an Android user. If one grants the permission for an application to read or write contacts or SMS messages, a potential threat will be a misuse of the above-mentioned information when the relevant application uses it improperly.

The downside of this model is that a regular user may not always be aware of the significance of these permissions. A lack of interest of a personal security may lead to personal information being exposed to some unauthorized parties.

Android malware. After a permission is granted, there are no limitations on how the recourses of a smartphone are used [8]. Information leak in benign applications is an open question but malware can exploit it to a higher degree.

SophosLabs has collected almost 1.5 million of unique Android malware samples [9]. Their chart gives a suggestion what kind of threats Android users may expect.

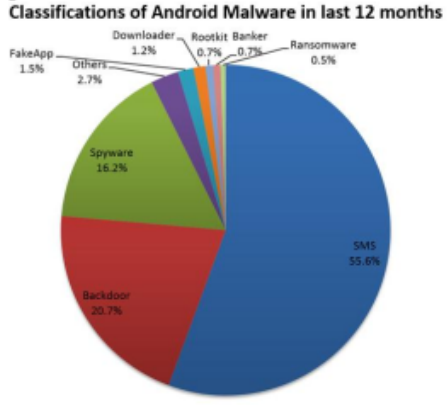


Fig. 1. SophosLabs malware statistics

The popularity of SMS malware accounts for the easiest way to attackers to profit on their malign code distribution. An extra attention should be paid while granting this permission.

Android spying applications. There is a number of officially available spying applications on Play Store or available via Google Search after submitting any relevant keywords. Some of them are heavily promoted and corresponding ads will appear during the search.

If these spying applications were used as they are officially introduced, it would rise no concerns. They are introduced as a children monitoring tool or an office smartphone control application. The reality as the research [10] shows can be different.Violent partners tend to install these applications on their victims' devices in order to track their location and communication which leads to a higher degree violence.

This research will focus on the above-mentioned types of applications. As all the security is based on the permissions in Android, Permission Management System, the prototype, has been introducedfor any potential information leak threats.

III. OVERVIEW OF THE CURRENTLY COMPLETED RESEARCHES

A huge number of researches has been completed on the AndroidOS, and a considerable number of them is dedicated to its permission model. Permissions may hardly be enough to tell if that application is malware or a benign one, but it is good enough to give a general view of what kind of information could be made available to someunauthorized parties.

Tianliang Lu and Su Hou have proposed a two-layered malware detection model [11] in order to improve the accuracy.Malware requested permissions are often similar to the permissions requested by benign applications. Sensitive permission model analysis along with the machine learning would do a better job as it is implied. Random forest is used as a machine learning algorithm for the first layer, meanwhile sensitive permission rules are used by the second layer.

The research completed by Gurol Canbek and others [12] highlights the importance of a regular Android user to understand permissions which are requested by an application instead of making statistics on the most frequently requested permissions by malware and benign applications. Their solution was to group semantically 251 Android permissions into 12 clusters. They have also proposed a visualization approach which is to look more conventional to end users and experts.

Another attempt to use permissions in order to detect any malware which leads to uncontrollable information leak is made by Abdirashid Ahmed Sahal and the others [13]. They have introduced a new weighting method which they call TF-IDFCF. They

claim their detection rate is above 95.3 %. They decompile Android application files in order to read their requestable permissions which are stored in the AndroidManifest.xml file. Unique identified permissions in malware and benign applications are used to build a binary matrix. An enhanced TF-IDF method is used to select features while building their datasets. Finally, in order to detect any potentially negative application, multiple classifiers are trained.

What concerns spyware applications and a detection of them, there are also some papers available. Mustafa Hassan Saad has proposed in his paper a spyware application for a better understanding of the ways the spyware applications work and a solution to fight any spyware which is called DroidSmartFuzzer [14].

It is stated in the abovementioned research that spyware is a concern for privacy as it can overtake SMS messages, incoming and outgoing calls, andit transmits data via internet.DroidSmartFuzzer is based on a Fuzz testing which is an effective technique to find any security vulnerabilities. The software gets an input of a big amount of diverse data, and it is being monitored during this process for any unusual behavior, crashes and fails. A specific goal of DroidSmartFuzzer in that particular case was to spot any internet usage by some unauthorized applications using the following permissions:

- RECEIVE_SMS
- PROCESS_OUTGOING_CALLS
- READ_PHONE_STATE

As the test was completed, it has confirmed that according to the authors their application was successful to report any spying activity.

IV. ANALYSES OF THE CURRENTLY AVAILABLE ANDROID INFORMATION LEAK MONITORING TOOLS

Since Android permission model is so important for users' sensitive information, relevant attention should be paid to any existing tools. Developers define in the manifest file which permissions are needed for that application. A user can either grant it or not but not granting may lead to an improper functioning.

A. Android Play Store permission review section

Android Play Store is the very first place where a user can review these permissions in order to assess any sensitive information leak.However, getting an access to this section might be slightly complicated at first. One has to pick the required applications, click on the title READ MORE, scroll to the bottom of the pop-up window, and click "View Details"underthe "Permissions" title.

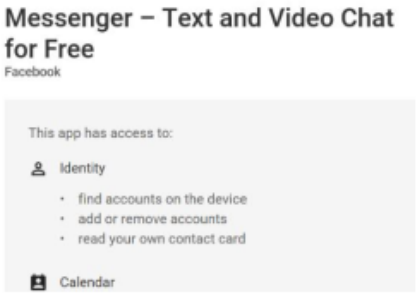


Fig. 2. A part of Messenger permissions

B. Permission monitoring via settings

It was not possible to toggle any granted permissions before Android 6.0 "Marshmallow" has been released in 2015 [15]. Usually that option is available via Settings > Apps / Application Manager > Permissions. It may differ however due to a manufacturer.

A screenshot is provided below. Dangerous permission groups can be granted or revoked by using a toggle switch. Normal permissions are granted automatically.

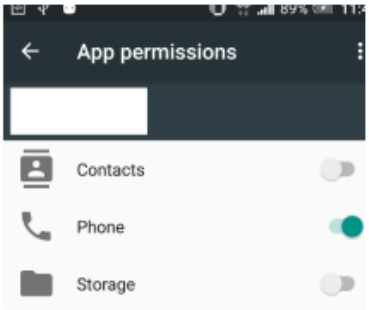


Fig. 3. Permission overview via settings

C. Third party applications

There are somethird-party applications for a permission review which gives a good idea of any potentialleak of sensitive information. To name a few of them:

- Application Inspector
- APK Analyzer
- Package Info

These applications will usually scan the device for any installed applications in order to produce a list of them. As that list is further explored, one can see after picking a particular application some more details about it. It may include the version number, installation path, update time, libraries, granted permissions and permissions to be requested as well as some other details.

APK Analyzer has a good function which allows a downloaded APK package to be scanned by that application before it is installed. It gives a chance for a permission review one more time.

A tool which gives a more focused review on permissions might be useful. A regular user might not

be persistent enough to look online for further explanations on certain permissions and how that type of information will be used. A personal factor on information sensitivity value might also introduce a better understanding of any potential information leak.

V. INFORMATION LEAK MONITORING ON V-S AXES

It was decided that V-S axis method [16] is the most appropriatefor the sensitivity assessment of permissions and their associated information.One axis isfor**information value (X)**and the other one is for **permission sensitivity (Y)**.As different levels are assigned on these 2 axes, different security measures can be applied.

The levels of the axis are the following ones:

Permission**sensitivity (Y)**: low, middle, and high.

Information **value (X)**: low, middle, and high.

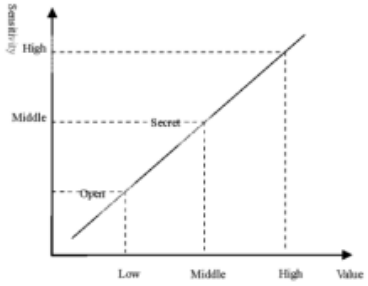


Fig. 4. V-S axes and their levels

The official Android classification of permissions [17] was used for permission sensitivity (Y). Permissions originally are classified into 4 groups: normal, dangerous, signature, and special ones. These permissions were assigned to the sensitivity axis (Y) in the following way:

TABLE II
SENSITIVITY AXIS (Y) BASED ON PERMISSIONS

Axis (Y) level	Points	Assigned permissions
Low	0	Normal–they are not dangerous officially. Granted automatically.
Middle	1	Normal – they are not dangerous officially by may cause issues. E.g . CHANGE_NETWORK_STATE
High	2	Dangerous –dangerous permission groups. They may cause some sensitive information leak

Information value axis (X) is dedicatedto a personal assessment of the stored information. The prototype uses the default levels for this axis, but they are available for adjustingat any time.

TABLE III
VALUE AXIS (X) BASED ON PERSONAL VIEW

Axis (X) level	Points	Information value
Low	0	Low value information. A user is not concerned to lose it. Low sensitivity (Y) is matched with low value (X) by default.
Middle	1	Average value information. A user may regret to lose it. Middle sensitivity (Y) is matched with middle value (X) by default.
High	2	High value information. A user does not want to lose it. High sensitivity (Y) is matched with high value (X) by default.

Permission Management System, the proposed prototype,is based on these two axes. As this prototype is launched, it starts scanning all the installed applications. APPS list is produced by default where applications are ranked according to their danger point score. The second list PERMISSIONS is the one where permissions are ranked by the frequency of their usage. It gives a user a quick review of any potential sensitive information leak.

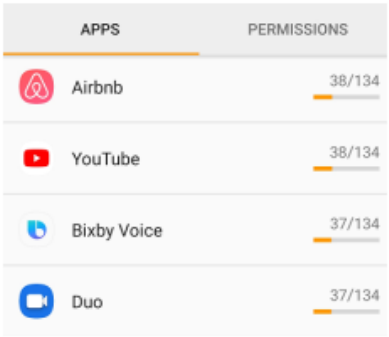


Fig. 5. The prototype

All the permissions from **dangerous** permission group were used for permission sensitivity (Y) axis.Dangerous permissions are assigned by default to the level **High (2)**.

TABLE IV
PERMISSION GROUPS AND MAX. POINT SCORE

Permission group	Permissions and max. score on both axis			
	Permissions	*Y*	*X*	*Y * X*
CALENDAR	READ_CALENDAR	2	2	4
	WRITE_CALENDAR	2	2	4
CALL_LOG	READ_CALL_LOG	2	2	4
	WRITE_CALL_LOG	2	2	4
	PROCESS_OUTGOING_CALLS	2	2	4
CAMERA	CAMERA	2	2	4
...	...	...	...	...
Maximum point score for dangerous permissions				104

Some normal permissions were picked for using themwith the permission sensitivity (Y) axis.The default value isset to**Middle (1)**. It is set to High (2) when maximum point score is calculated which equals 134.Normal permissions are considered to be not dangerous and they are granted automatically as an application is getting installed, but they may cause some inconvenience. Besides some normal permissions like CHANGE_WIFI_STATE are very common among malware applications.

TABLE V
MAX. SCORE FOR POTENTIALLY DANGEROUS

Permissions	Y	X	Y * X
CHANGE_NETWORK_STATE	1	2	2
CHANGE_WIFI_STATE	1	2	2
MODIFY_AUDIO_SETTINGS	1	2	2
REQUEST_DELETE_PACKAGES	1	2	2
NFC	1	2	2
REORDER_TASKS	1	2	2
REQUEST_INSTALL_PACKAGES	1	2	2
FLASHLIGHT	1	2	2
GET_TASKS	1	2	2
BILLING	1	2	2
SET_ALARM	1	2	2
DISABLE_KEYGUARD	1	2	2
SET_WALLPAPER	1	2	2
SYSTEM_ALERT_WINDOW	1	2	2
WRITE_SETTINGS	1	2	2
Maximum point score for dangerous permissions			30

The maximum danger point score is 104 + 30 = 134. Default levels are used for the information value (X) axis but a user can change it. As the tables above suggest, the default information value (X) axis has the level High (2) when it is matched with dangerous permissions the axis Y. The default level on the information value (X) axis is Middle (1) when it is matched with some picked normal permissions on the axis Y. If the level Low (0) is chosen, it will be multiplied by 0 which renders that permission unconsidered.

VI. EXPERIMENTAL FINDINGS

This experiment has been completed with the following purposes:

1) Do commercial spyware and malware have on average a higher score over benign applications?

2) Which permissions are the most common for benign, malware and spyware applications?

TABLE VI
USED DEVICES

Device	Basic specifications
Lenovo Yoga 530	Windows Pro 10 Intel® Core™ i3-8130U CPU @ 2,20 Ghz 16,0 GB RAM
Samsung Galaxy S8	Android 8.0.0 Octa-core (2.3GHz Quad + 1.7GHz Quad), 64 bit, 10nm processor 4 GB RAM (LPDDR4)
Samsung Tab A (SM-T585)	Android 8.1.0 Octa-core (4x1.6 GHz Cortex-A53 & 4x1.0 GHz Cortex-A53) 3 GB RAM

The test includes 100 benign applications, 41 malware and 28 commercial spyware applications.

Benign applications were downloaded from Play Storeusing 5 categories: shopping, finance, communication, education and business. These categories were selected randomly. Top 20 applications were selected from each of these 5 categories.

Malware applications were downloaded from GitHub [6]. Meanwhile commercial spyware applications were randomly downloaded from Play Store or from their original distributors.

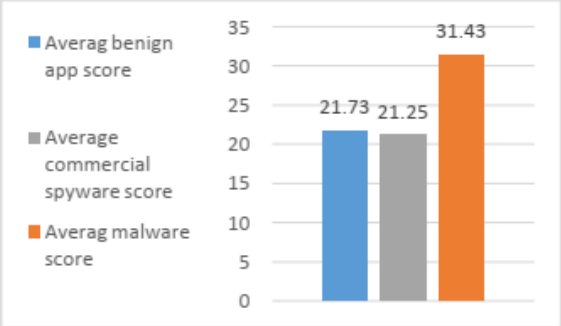


Fig. 6. Average danger point score

As one can see in the chart, malware applications have on average a higher danger point score by 1/3.The paper [18] claims that malware applications usually tend to request more permissions than benign ones. That could be the case.

Permissions solely may not however reflect the whole danger of malware due to its uncontrollable information leak.A malware application may ask just a few permissions to look completely safe but if it includes e.g. SEND_SMS, it can send SMS messages to bring a high financial loss. Permissions may not reflect the danger of spying either. If e.g. a physical attack follows spying, it is more than an information leak.

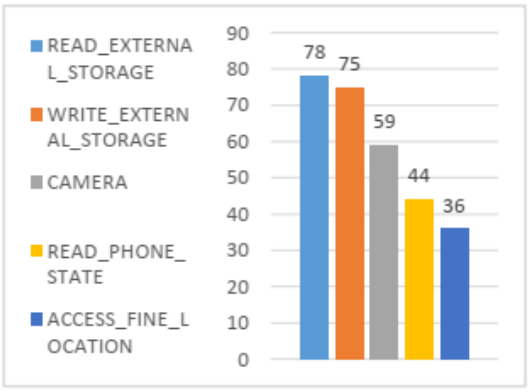


Fig. 7. The most frequent benign application permissions

Benign applications are mostly eager to use the external storage of a device.They would also need access to a camera or location.

Malware would tend to use a different set of applications as it was noticed in the paper [18].More attention was paid to dangerous ones in this case due to more sensitive nature.

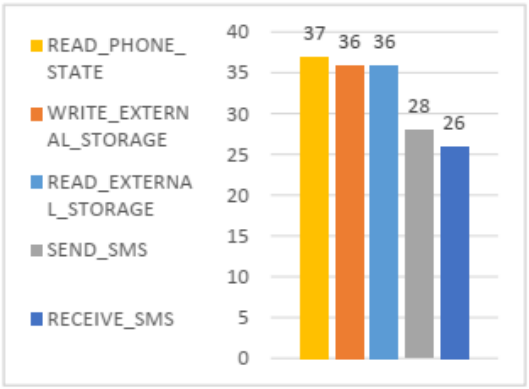


Fig. 8. The most frequent malware permissions

READ_PHONE_STATE dominates but an access to the external storage is also very important. Permissions for SMS messages are common.

Commercial spying applications will most frequently ask for the location of the device. Except reading the storage and the phone state, they will also need the contacts. The set of permissions will mainly depend of the functionality.

No scan of permissions may work if the device in enabled for such default services like Find My Mobile (Samsung). It is meant to find the lost device but it could be used to spy on close people to some extentwhen these devices are registered on the same account. Tracking services as for the lost phoneare available at https://findmymobile.samsung.com.

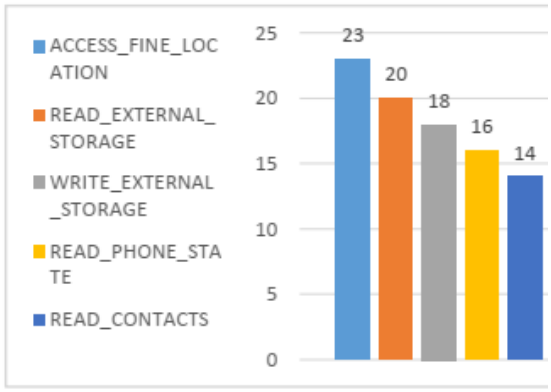


Fig. 9. The most frequent commercial spyware permissions

Further work can be done to assign a higher point score for the most common malware or commercial spyware permission sets which would allow to identify it easier. It would also suggest to double check a corresponding application with an anti-virus tool or just remove it.

VII. CONCLUSIONS

Android OS uses permission protection levels. These permissions are not always explanatory enough to understand their importance. Granting a permission keeps one informed that this type of information is used but there are no methods to reveal how it is used.

The prototype provides a quick and user-friendly assessment of a potential sensitive information leak. The danger of an information leak may not always be reflected with permissions if any further information misuse is involved for a physical attack or violence.

The research includes 100 benign applications, 41 malware and 28 commercial spyware applications.They seem to have their typical set of permissions. A further study of these sets may lead to increased safety capabilities.

REFERENCES

[1] (2019) Deuthche Welle, "Smartphones: Live longer, be greener". [Online]. Available at:https://www.dw.com/en/smartphones-live-longer-be-greener/a-46423527.

[2] Q. Chen, J. Wang and Y. Wang, "An Online Approach for Detecting Repackaged Android Applications Based on Multi-user Collaboration," 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, 2015, pp. 312-315.Available at: https://ieeexplore.ieee.org/abstract/document/7518244

[3] Kaspersky Lab DAILY, "Google Trades Privacy and Security for Hangouts". Available at https://www.kaspersky.com/blog/google-privacy-hangouts/1993/

[4] Spyzie, "All-Inclusive Phone Spy". Available at https://www.spyzie.com/ad/phone-spy-amp.html?gclid=EAIaIQobChMI9u3YsvO-4QIVV-d3Ch08ggReEAAYASAAEgKg3_D_BwE

[5] R. Chatterjee et al., "The Spyware Used in Intimate Partner Violence," 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, 2018, pp. 441-458. Available at: https://ieeexplore.ieee.org/document/8418618

[6] GitHub, Inc., "Ashishb Collection of Android Malware Samples".Available at: https://github.com/ashishb/android-malware

[7] Permissions Overview, 2019. Available at: https://developer.android.com/guide/topics/permissions/overview#normal-dangerous

[8] O. S. J. Nisha and S. M. S. Bhanu, "Detection of repackaged Android applications based on Apps Permissions," 2018 4th International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, 2018, pp. 1-8. Available at: https://ieeexplore.ieee.org/document/8388984

[9] Rowland Yu & William Lee, "VB2015 paper: Will Android Trojans, Worms or Rootkits Survive in SEAndroid and Containerization?", Sophos, Australia. Available at: https://www.virusbulletin.com/virusbulletin/2016/02/vb2015-paper-will-android-trojans-worms-or-rootkits-survive-seandroid-and-containerization/

[10] R. Chatterjee et al., "The Spyware Used in Intimate Partner Violence," 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, 2018, pp. 441-458. Available at: https://ieeexplore.ieee.org/document/8418618

[11] X. Liu and J. Liu, "A Two-Layered Permission-Based Android Malware Detection Scheme," 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Oxford, 2014, pp. 142-148. Available at: https://ieeexplore.ieee.org/document/6834956

[12] G. Canbek, N. Baykal and S. Sagiroglu, "Clustering and visualization of mobile application permissions for end users and malware analysts," 2017 5th International Symposium on Digital Forensic and Security (ISDFS), Tirgu Mures, 2017, pp. 1-10. Available at: https://ieeexplore.ieee.org/document/7916512

[13] A. Sahal, S. Alam and I. Soğukpinar, "Mining and Detection of Android Malware Based on Permissions," 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, 2018, pp. 264-268. Available at: https://ieeexplore.ieee.org/document/8566510

[14] M. H. Saad, A. Serageldin and G. I. Salama, "Android spyware disease and medication," 2015 Second International Conference on Information Security and Cyber Forensics (InfoSec), Cape Town, 2015, pp. 118-125. Available at: https://ieeexplore.ieee.org/document/7435516

[15] Google Play Help, "Control your app permissions on Android 6.0 and up", [Online]. Available: https://support.google.com/googleplay/answer/6270602?hl=en-GB

[16] X. Shi, D. Li, H. Zhu and W. Zhang, "Research on Supply Chain Information Classification Based on Information Value and Information Sensitivity," 2007 International Conference on Service Systems and Service Management, Chengdu, 2007, pp. 1-7. Available at: http://ieeexplore.ieee.org/document/4280248/

[17] "Protection levels". Available at: https://developer.android.com/guide/topics/permissions/overview#normal-dangerous

[18] P. Xiong, X. Wang, W. Niu, T. Zhu and G. Li, "Android malware detection with contrasting permission patterns," in China Communications, vol. 11, no. 8, pp. 1-14, Aug. 2014. Available at: https://ieeexplore.ieee.org/document/6911083