LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

ATĻAUJU ORGANIZĒŠANAS MODEĻI LIETOTNĒS

BAKALAURA DARBS

Autors: **Krišjānis Segliņš**
Studenta apliecības Nr.: ks14035
Darba vadītājs: profesors Dr. dat. Uldis Straujums

RĪGA 2018

ANOTĀCIJA

Atļaujas, ko lietotājs dod lietotnēm, nosaka, kādas darbības lietotne var veikt, kādus datus lietotne var ievākt par lietotāju un ko ar tiem iesākt, tāpēc šo atļauju efektīva organizēšana lietotājam ir ļoti svarīga.

Darbā tiek pētīti iOS un Android atļauju organizēšanas modeļi, to problēmas un problēmu risinājumi, kā arī, ņemot vērā izpētes rezultātus, tiek aprakstīts jauns uzlabots atļauju organizēšanas modelis, kas apvieno abu operētājsistēmu labākās kā arī ietver jaunas problēmu risināšanas idejas.

Rezultātā aprakstītais modelis tiek aprakstīts un salīdzināts ar iOS un Android lietotajiem modeļiem un izdarīti secinājumi par to, kā pēc iespējas efektīvāk veidot šādus modeļus.

Atslēgvārdi: atļaujas, lietotņu atļauju organizēšanās modeļi, iOS, Android, lietotāju datu privātums.

ABSTRACT

Permission organization models in applications

Permissions that a user gives to the applications regulate what kind of activities the app will be able to do, what kind of data it may collect and what it could do with them, therefore an effective organization of these permissions is of significant importance to the user.

In this thesis document the permission organization models of iOS and Android are researched and in accordance with the results a new, better model is described that combines the best practices from both Android and iOS approaches as well as includes currently unused ideas.

As a result, the new model is described and compared with that by Android and iOS. Conclusions are drawn about how such permission organization models should be made more effectively.

Keywords: permissions, application permission organization models, iOS, Android, user data privacy.

SATURS

APZĪMĒJUMU SARAKSTS

Android	-	Google pārvaldīta viedierīču operētājsistēma
IOS	-	Apple pārlvaldīta un veidota viedierīču operētājsistēma
API	-	Lietojumprogrammatūras saskarne
Bluetooth	-	Radiotehnoloģija, kas izmanto mazas jaudas raidītājus un kas nodrošina pārnesājamu datu pārraides ierīču savienošanu savā starpā vai to pievienošanu datoram vai internetam [1]
IP adrese	-	Skaitliska adrese, kas viennozīmīgi identificē katru datoru internetā un kas izveidota kā četru ar punktiem atdalītu skaitļu virkne, piem.: 192.100.81.101. [1]
MAC adrese	-	12 ciparu heksadecimāla adrese, kas ieprogrammēta datora tīkla kartē un identificē datoru šajā tīklā
VPN	-	Virtuāls privāts tīkls
Cocoa	-	Satvars,kas paradzēts iOS un MacOS lietotņu izstrādei
VoIP lietotnes	-	Lietotnes, kas izmanto balss pārraidi ar interneta protokolu palīdzību (*Voice over IP*)
Root	-	Sistēmas galvenais lietotājs
Sticky Broadcast	-	Rīks, kas palīdz Android lietotņu izstrādātājiem komunicēt ar citām lietotnēm
Wi-Fi Multicast	-	Bezvadu grupu komunikācija
Google Play store	-	Android galvenais lietotņu izplatīšanas veikals
Apple App store	-	Apple galvenais lietotņu izplatīšanas veikals
IMEI numurs	-	Internacionālās mobilā aprīkojuma identifikācijas numurs, kas identificē katru mobilo ierīci
Power user	-	Lietotājs, kas ir padziļināti zinošs par datoriem un elektronikas iekārtām

IEVADS

Atļaujas ir viens no pamata elementiem datu drošības nodrošināšanai sistēmās. Laiku pa laikam parādās ziņās, ka ir notikušas datu noplūdes vai ziņa, ka kāda kompānija ir ievākusi lielu daudzumu datus, lietotājus skaidri neinformējot. Datu privātums un līdz ar to arī atļaujas ir aktuāla tēma, kas periodiski kļūst aktuāla arī plašākai sabiedrībai. Lai gan lielu daļu laika cilvēki pavada lieki par to nesatraucoties, ir notikumi, kas sabiedrības uzmanību tomēr vērš uz to, kādus datus viņi par sevi publisko, kā tie tiek ievākti, kam izmantoti un kāpēc.

Nesenākais šāda veida notikums, kas pievērsa ļoti plašas publikas uzmanību, bija ziņa, ka kompānija Facebook nodevusi vairāk nekā 80 miljonu lietotāju personīgo datu informāciju kompānijai Cambridge Analytica, bez lietotāju informēšanas, kas gan vērsa uzmanību uz to, kādi dati tiek izgūti un kā izgūtie dati tiek izmantoti pēc to ievākšanas [30]. Taču šādi gadījumi aktualizē arī atļauju organizēšanas jautājumu, jo atļaujas ir tās, kas nosaka, kādiem resursiem lietotnes var piekļūt, kādu informāciju ievākt un kā to potenciāli izmantot.

Tāpēc šī darba mērķis ir veikt apskatu par iOS un Android atļauju modeļiem, to problēmām un problēmu risinājumiem un no šīs informācijas veidot jaunu uzlabotu atļauju organizēšanas modeli, kas savienotu abu operētājsistēmu labākās prakses un pievienotu jaunas problēmu risināšanas idejas. Darbā tiek apskatītas tikai Android un iOS operētājsistēmas, jo pēc 2018.gada aprīļa datiem tās izmanto 94,89% no visām viedierīcēm pasaulē [32].

Mērķa sasniegšanai ir izvirzīti šādi uzdevumi:

1. Apskatīt un salīdzināt iOS un Android atļauju organizēšanas modeļus;
2. Apskatīt, kādas problēmas ir sastopamas šajos atļauju organizēšanas modeļos un kā tās iespējams risināt;
3. Izvēlēties labākos risinājumus un izveidot uzlabotu atļauju organizēšanas modeli, balstoties uz šiem risinājumiem, un salīdzināt šo modeli ar jau eksistējošajiem iOS un Android atļauju modeļiem.

Darbs sastāv no trim nodaļām un katrā no tām aprakstīts risinājums kādam no augstāk minētajiem uzdevumiem. Darba nobeigumā autors apkopo darba rezultātus un apraksta darba veidošanas gaitā gūtos secinājumus par dažādajiem atļauju organizēšanas modeļiem Android un iOS operētājsistēmās, kā arī secinājumus par autora aprakstīto uzlaboto atļauju organizēšanas modeli.

1. ATĻAUJAS

Atļaujas ir svarīga daļa moderno viedierīču drošības sistēmās. Šajā nodaļā tiek apskatīts, kas ir atļaujas un kas ir atļauju organizēšanas modeļi. Aplūkotas tiek arī problēmas Android un iOS operētājsistēmās, kas ir saistītas ar atļaujām, un atļauju evolūcija Android un iOS sistēmās.

1.1 Atļauju apraksts

Lietotnēm bieži ir nepieciešama pieeja informācijai par lietotāju, sistēmas resursiem vai citām funkcionalitātēm. Atļaujas nodrošina šo pieeju, bet reizē arī ļauj to ierobežot. Galvenais uzdevums atļaujām ir nevis sniegt pieeju, bet tieši ierobežot pieeju un tādējādi aizsargāt lietotāja datus un sistēmas funkcionalitātes no nelabvēlīgas izmantošanas. Atļaujas vienmēr piešķir kādam, tas var būt lietotājs vai pati sistēma, vai arī lietotne.

Ikdienā izmantotas ierīces, kā piemēram tālruņi vai planšetdatori, pamatā strādā ar Android un iOS operētājsistēmām. Lietotājam ar šīm ierīcēm pavadot gandrīz visu laiku, ierīces un to lietotnes ievāc krietnu daudzumu ar datiem par lietotāju, sākot no lietotāja atrašanās vietas līdz zvanu vai īsziņu vēsturei. Lielā datu apjoma dēļ ierīču drošība un lietotņu pieeja šiem datiem ir diezgan aktuāla tēma. Lietotnēm lietotāja dati ir nepieciešami, lai sniegtu lietotājam kādu pakalpojumu vai funkcionalitāti. Pieejas sniegšana ierīcēs, kas strādā ar Android un iOS operētājsistēmām, ir vienkārša - ja lietotnei ir nepieciešama pieeja, tad lietotājam tiek parādīts logs, kur ir jāuzspiež poga “atļaut”, un atļauja lietotnei ir sniegta. Lietotne pēc atļaujas piešķiršanas var ievākt datus par lietotāju, atkarībā no atļaujas tipa. Pēc 2015.gada datiem 90% no lietotņu lietotājiem ir norādījuši, ka zināšana, kādiem lietotāja datiem lietotne vēlēsies pieeju ir ļoti svarīga vai svarīga pirms lietotnes uzstādīšanas [8]. Kā arī 60% no lietotājiem ir izvēlējušies lietotnes neuzstādīt pēc uzzināšanas, cik daudz lietotāja personīgajiem datiem lietotne pieprasa pieeju [8]. Pēc 2017.gada aptaujas datiem 32% vienmēr un 42% ik pa laikam pārbauda, kādas pieejas lietotne pieprasa, un tikai 17% reti pārbaude, bet 9% nepārbauda [7]. Statistikas dati liecina, ka lielākai daļai lietotāju ir svarīgi zināt, kādiem datiem lietotnēm ir pieeja. Šī iemesla pēc ir svarīgi nodrošināt skaidri saprotamas un drošas atļaujas, lai lietotāji varētu uzzināt un kontrolēt, kādus datus lietotne izmanto un kādus datus nē.

1.2 Atļauju organizēšanas modeļi

Liela daļa viedierīču operētājsistēmas izmanto stingru trešās puses lietotņu norobežošanu no pašas sistēmas. Katrai lietotnei ir savs lietotājs, kam tiek piešķirtas atļaujas, un tā atrodas savā mapē, kurā veica arī visas savas darbības un bez specifiskas atļaujas tai nav pieeja ārpus šīs mapes. Kāda daļa no norobežošanas ir atļauju sistēma, kas nodod lietotājam kādiem personīgajiem datiem lietotne vēlas piekļūt un sniedz iespēju piešķirt vai atteikt atļauju, lai šiem datiem piekļūtu [18]. Atļauju sistēmas tiek veidotas pēc atļauju organizēšanas modeļiem, kas apraksta, kādas atļaujas sistēmā ir pieejamas, kā tās tiek kārtotas, kā katrai atļaujai var piekļūt, vai vajag lietotāja piekrišanu vai arī atļauju var piešķirt automātiski, kad lietotājam tiek pieprasītas atļaujas un kā tiks pieprasītas atļaujas. Atļauju sistēma ir realizēts atļauju organizēšanas modelis.

Lietotājiem ir svarīgi zināt, kādiem datiem lietotnei ir piekļuve, un svarīgi ir iespēja arī piekļuves mainīt. Atļauju organizēšanas modeļi apskata gan piekļuves, gan kā tās pārvaldīt, tāpēc šajā darbā tiek tuvāk aplūkoti jau esoši atļauju organizēšanas modeļi un tiek mēģināts uzlabot šos modeļus, lai tie labāk aizsargātu lietotāja personīgos datus.

1.3 Ar atļaujām saistītās problēmas

Atļauju pamata mērķis ir aizsargāt lietotāja datus un pārvaldīt pieeju tiem, tāpēc problēmas ar atļauju modeļiem lielākoties ir saistītas ar to, kā tiek aizsargāti lietotāja dati un pārvaldītas pieejas tiem. Android un iOS operētājsistēmu atļauju organizēšanas modeļos ir 3 pamata problēmas:

1. lietotne var piekļūt pārāk daudz datiem, jo atļauja pietiekoši neierobežo pieeju;
2. lietotājam var tikt liegta lietotnes izmantošana, ja viņš atsakās piešķirt kādu atļauju;
3. lietotājiem ir pārāk maz iespēju pārvaldīt piešķirtās atļaujas vai tas ir sarežģīti.

Pirmā problēma, ietver sevī, ka lietotne var ievākt datus par lietotāju un lietotāju pašu par to neinformēt. Atļauja, kas nodrošina pieeju pie kontaktu grāmatiņas atļauj šos datus ievākt jebkurā lietotnes darbības laikā, pat strādājot fona režīmā. Fona režīmu lietotnes var izmantot arī lietotāja pārraudzīšanai. Lietotne var maldināt par atļaujas nepieciešamību un kad atļauja tiks izmantota. Piemēram, Android kameras piekļuves atļauja darbina arī zibspuldzi, turklāt nav atsevišķas atļaujas, kas kontrolētu piekļuvi zibspuldzei, tāpēc pat lietotnes, kas izmanto zibspuldzi kā gaismas

lukturīti pieprasa lietotājam pilnu piekļuvi kamerai. Tāpēc teorētiski lietotne varētu, lietotājam nezinot, izmantot arī pārējās kameras funkcijas.

Otrā problēma ir, ka lietotne var nestrādāt bez kādas atļaujas piešķiršanas, pat ja atļauja nav saistīta ar lietotnes pamata funkcionalitāti. Piemēram, ir bijuši gadījumi ar tām pašām gaismas lukturīšu lietotnēm, kad šādas lietotnes pieprasīja atrašanās vietas atļauju, kas nekādi saistīta ar lietotnes veicamo funkcionalitāti, bet bez kuras piešķiršanas lietotni nebija iespējams darbināt. Turklāt vēl nemaz ne tik sen - līdz Android 6.0 versijas iznākšanai – visas Android lietotnes bija iespējams lejupielādēt tikai, ja tika apstiprinātas visas pieprasītās atļaujas bez izņēmumiem. Līdz ar Android 6.0 versijas iznākšanu, šī problēma tiek risināta, taču lietotnes, kas veidotas pirms šīs versijas un vēl darbojas uz mūsdienu viedierīcēm, joprojām atļaujas pieprasa pēc šīs vecās "visu vai neko" pieejas [15].

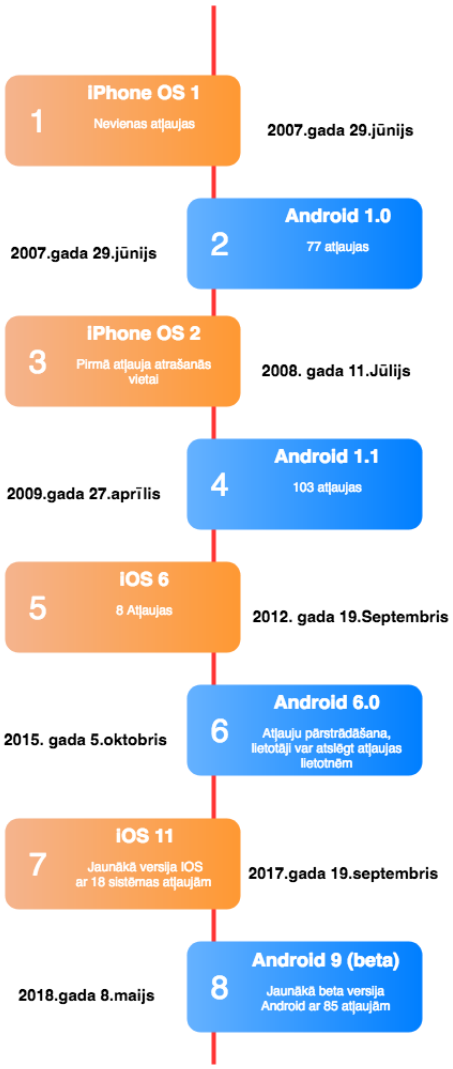
Trešā problēma ir, ka lietotājam ir sarežģīti pārvaldīt lietotņu atļaujas un piešķirt atļaujas uz noteiktu laiku. Gadījumos, kad lietotājs vēlās sarakstes lietotnēs izmantot mikrofonu, lietotne pieprasīs piekļuvi mikrofonam, ko lietotājs visticamāk apstiprinās, taču pēc lietotāja sarunas, kad atļauja mikrofona pieejai vairs nav nepieciešama, lietotājam ir jāveic garāka virkne ar darbībām, lai šo atļauju atspējotu. Rezultātā lietotājs vai nu šo atļauju atstās un lietotnei būs nepamatota pieeja mikrofonam, vai nu lietotājs būs neapmierināts, jo būs patērējis ilgāku laiku, atļauju atspējojot.

Ar laiku šīs problēmas lielākā vai mazākā mērā tiek risinātas gan Android, gan iOS platformās. Daži no iespējamajiem un plānotajiem risinājumiem tiks apskatīti nākamajās šī darba nodaļās.

1.4 Atļauju evolūcija

Jau kopš API 1 versijas atļauju modelis Android vienmēr bijis operētājsistēmas sastāvdaļa. Android pirmā versija tika izdota 2008.gada 23.septembrī. Pirmajā versijā bija iekļautas 77 atļaujas. API 3 jeb *Cupcake* versijā, kas bija pirmā plaši izmantotā versija, tika iekļautas jau 103 atļaujas [19, 21]. Pēdējā Android P versijā ir paredzētas 85 atļaujas, tomēr skaitot darbspējīgās, bet novecojušās un aizstātās vēsturiskās atļaujas, Android P atbalstīto atļauju skaits ir vairāk nekā 300 [3, 22]. Agrāk Android atļauju modelis bija labi sadalīts atļaujās, atļaujas tika grupētas atļauju grupās, taču lietotājam nebija nekādas kontroles pār lietotnei piešķirtajām atļaujām vai atļauju grupām. Ja lietotājs nevēlējās lietotnei piešķirt kādu atļauju, tad vienīgais variants bija neuzstādīt

lietotni. Situācija mainījās 2015.gada 5.oktobrī ar Android 6.0 izdošanu, lietotājs tagad var katrai lietotnei piešķirt (vai nepiešķirt) atļauju grupu. Tas gan attiecas tikai uz bīstamā drošības līmeņa atļaujām, jo parastās atļaujas joprojām tiek piešķirtas automātiski, bez atsevišķas lietotāja piekrišanas. Pēc 2018.gada datiem Android 6.0 un jaunākas versijas ir uzstādītas 62.3% ierīcēm, kas strādā Android operētājsistēmā, tas nozīmē, ka 37% ierīču nemaz nevar izmantot jauno un uzlaboto atļauju organizēšanas modeli [9].



1.1. att. **Android un iOS atļauju evolūcijas diagramma**

Attēlā 1.1 attēlotas iOS un Android izdotās versijas uz vienas laika ass. IOS pirmā versija tika izlaista 2007.gada 29. jūnijā. Pirmajā oficiālajā IOS versijā (iPhone OS 1) atļaujas nebija iekļautas vispār. Apple pirmo atļauju pielika 2008.gadā ar iPhone OS 2 versiju. Tāpēc līdz pat 2012. gadam Android atļauju modelis tika saukts par labāku gandrīz visos aspektos, bet 2012.gada 19.septembrī iznāca iOS 6 ar astoņām pieejas atļaujām: atrašanās vietai, kontaktiem, kalendāram,

atgādinājumiem, bildēm, bluetooth, Twitter un Facebook lietotāja kontiem. Jaunā sistēma ļauj katrai lietotnei atsevišķi piešķirt atļaujas un tās mainīt arī pēc uzstādīšanas, kas nozīmē, ka lietotājs var labāk kontrolēt, kādiem datiem var piekļūt trešās puses lietotnes. Tā rezultātā izveidojās uzskats, ka iOS atļauju modelis ir labāks nekā Android piedāvātais. Šis uzskats bija vadošais līdz pat Android 6.0 versijas iznākšanai 2015. gada beigās, kad Android ieviesa atļauju modeli, kas ļauj lietotājam mainīt lietotnēm piešķirtās atļaujas arī pēc to uzstādīšanas, tādējādi Android pietuvinājās iOS rādītajam piemēram [20].

2. ATĻAUJU ORGANIZĒŠANAS MODEĻU SALĪDZINĀJUMS

Nodaļā tiek apskatīti Android un iOS operētājsistēmu atļauju organizēšanas modeļi. Vispirms tiek apkopoti un apskatīti saraksti ar katras operētājsistēmas atļaujām un tad šīs atļaujas tiek salīdzinātas starp abām operētājsistēmām, iekļaujot arī to pārvaldības un pieprasīšanas mehānismu salīdzinājumu.

2.1 Atļauju saraksts

Android un IOS izmantotie atļauju organizēšanas modeļi ir gan savstarpēji līdzīgi, gan atšķirīgi. IOS visas atļaujas tiek pieprasītas automātiski, kad lietotājs vēlas izmantot kādu funkcionalitāti, kamēr Android daļa atļauju ir atsevišķi jāpieprasa, bet daļa tiek jau automātiski piešķirtas. Savukārt abu operētājsistēmu izmantotajām pieejām kopīgs ir tas, ka atļaujas tiek piešķirtas katrai lietotnei nevis lietotājam vai visai sistēmai uzreiz kā tas ir Windows operētājsistēmai. Lietotņu atļaujas modeļi ir veidoti tā, ka atļauju grupa dod plašu pieeju servisiem, bet šie servisi atbild tikai par kādu noteiktu funkcionalitāti. Lietotājiem ir iespējams kontrolēt tikai tad atļaujas grupu vai plašu vienu atļauju. Piemēram, atļauja kameras pieejai tiek parasti veidota kā atļauju grupu vai vienu atļauju, kuru lietotājs var regulēt, bet lietotne iegūst uzreiz iespēju gan uzņemt bildes, gan filmēt ar skaņu.

2.1.1 Android atļaujas

Android atļaujas ir mainītas un papildinātas līdz ar jaunu operētājsistēmas versiju iznākšanu. Sākotnēji Android atļauju modelis bija gaužām vienkāršs. Visas atļaujas bija ar vienādu nozīmi un atļaujas tika pieprasītas lietotājam pirms lietotnes uzstādīšanas bez iespējas tās pārvaldīt. Ar Android versijas 6.0 jeb *Marshmellow (API - 23)* iznākšanu , Android atļaujas tika pārveidotas, kā rezultātā jaunajā atļauju organizēšanas modelī atļaujas tiek sagrupētas trīs drošības līmeņos – parastās (*Normal*), parakstāmās (*Signature*) un bīstamās (*Dangerous*) [3]. Pašas atļauju grupas nav jaunums operētājsistēmā un savus nosaukumus pat nav mainījušas, bet kā tās tika sakārtota, pieprasītas un pārvaldītas. ka var atslēgt daļu no atļaujām.

Parastās atļaujas atbild par pieejām dažādiem datiem vai resursiem ārpus lietotnes, bet šīs pieejas teorētiski nesniedz nozīmīgu informāciju, kas saistīta ar lietotāju privātajiem datiem vai citu lietotņu darbību. Lietotnes deklarācijas dokumentā atļauju sarakstam ir jāpievieno parastās atļaujas un zemā riska dēļ operētājsistēma šis atļaujas izsniedz automātiski, nepieprasot atsevišķu lietotāja piekrišanu.

2.1. tabula

Android atļauju saraksts

Atļaujas kods	Atļaujas apraksts
ACCESS_LOCATION_EXTRA_COMMANDS	Ļauj piekļūt papildus atrašanās vietas servisa komandām
ACCESS_NETWORK_STATE	Ļauj piekļūt informācijai par tīkliem
ACCESS_NOTIFICATION_POLICY	Nodrošina piekļuvi paziņojumu uzstādījumiem
ACCESS_WIFI_STATE	Ļauj piekļūt informācijai par bezvadu tīkliem
BLUETOOTH	Ļaut savienoties ar sapārotajām Bluetooth ierīcēm
BLUETOOTH_ADMIN	Ļauj meklēt un sapārot ar Bluetooth ierīcēm
BROADCAST_STICKY	Ļauj sūtīt Sticky Broadcast
CHANGE_NETWORK_STATE	Ļauj mainīt tīkla savienojuma stāvokli
CHANGE_WIFI_MULTICAST_STATE	Ļaut atvērt Wi-Fi Multicast režīmu
CHANGE_WIFI_STATE	Ļauj mainīt bezvadu tīkla savienojuma stāvokli
DISABLE_KEYGUARD	Ļauj lietotnei izslēgt bloķēšanas ekrānu
EXPAND_STATUS_BAR	Ļauj izvērst vai samazināt statusa joslu
GET_PACKAGE_SIZE	Ļauj noskaidrot pakontēs izmantotās atmiņas vietas daudzumu
INSTALL_SHORTCUT	Ļauj uzstādīt saīsnes lietotņu palaidējā
INTERNET	Ļauj lietot internetu
KILL_BACKGROUND_PROCESSES	Ļauj izbeigt aktivitāti killBackgroundProcesses

Atļaujas kods	Atļaujas apraksts
MANAGE_OWN_CALLS	Ļauj pārvaldīt zvanus, izmantojot ConnectionService API
MODIFY_AUDIO_SETTINGS	Ļauj veikt izmaiņas kopējos audio uzstādījumos
NFC	Ļauj izpildīt ievades/izvades operācijas, izmantojot NFC
NFC_TRANSACTION_EVENT (pieejama sākot ar Android P versiju)	Ļauj saņemt NFC transakciju notikumus
READ_SYNC_SETTINGS	Ļauj nolasīt sinhronizācijas uzstādījumus
READ_SYNC_STATS	Ļauj nolasīt sinhronizācijas statistiku
RECEIVE_BOOT_COMPLETED	Ļauj saņemt notikumu, kas norāda uz iekārtas palaišanās procesa beigām
REORDER_TASKS	Ļauj pārkārtot uzdevumu starp priekšplāna un fona procesiem
REQUEST_COMPANION_RUN_IN_BACKGROUND	Ļauj palīglietotnei darboties fonā
REQUEST_COMPANION_USE_DATA_IN_BACKGROUND	Ļauj fona palīglietotnei izmantot tīkla datus
REQUEST_DELETE_PACKAGES	Ļauj veikt pakotņu dzēšanas pieprasījumus
REQUEST_IGNORE_BATTERY_OPTIMIZATIONS	Nepieciešama, lai izmantotu ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS aktivitāti
REQUEST_INSTALL_PACKAGES	Ļauj veikt pakotņu uzstādīšanas pieprasījumus
SET_ALARM	Ļauj uzsākt modinātāja vai hronometra aktivitāti
SET_WALLPAPER	Ļauj uzstādīt fona attēlu
SET_WALLPAPER_HINTS	Ļauj mainīt fona attēlu lielumu
TRANSMIT_IR	Ļauj izmantot infrasarkano staru raidītāju, ja tāds ir pieejams

Atļaujas kods	Atļaujas apraksts
USE_FINGERPRINT (iznākot Android P versijai, šī atļauja novecojusi un aizvietota ar citu)	Ļauj izmantot pirkstu nospiedumu nolasīšanas sistēmu
USE_BIOMETRIC (pieejama sākot ar Android P versiju)	Ļauj izmantot biometrijas datu sistēmas
FOREGROUND_SERVICE (pieejama sākot ar Android P versiju)	Ļauj pārvietot servisu uz fona procesiem, nepārtraucot servisa darbību
VIBRATE	Ļauj izmantot vibrācijas moduli
WAKE_LOCK	Ļauj novērst iekārtas pāreju miega stāvoklī
WRITE_SYNC_SETTINGS	Ļauj veikt izmaiņas sinhronizācijas uzstādījumos

Tabulā 2.1. apskatāms saraksts, kurā attēlotas visas parastās Android atļaujas [3]. Sarakstā var redzēt arī atļaujas, kuras varētu izmantot kādos no uzbrukumiem, taču lietotnes, kuras šīs atļaujas varētu izmantot ļaunprātīgiem nolūkiem, tiek mēģināts filtrēt jau lietotņu izplatīšanas servisos, piemēram, Google Play Store. Ja nav skaidrs, kāpēc lietotnei nepieciešamas šīs atļaujas, lietotne tiek detaizētāk apskatīta, izmantojot algoritmus vai piesaistot speciālistus. Tomēr visas atļaujas, kas ir apskatāmas tabulā 2.1. nesniedz lietotnēm tik vērā ņemamu pieeju lietotāja datiem, lai par to satrauktos. Lielāka pieeja personīgajiem datiem ir tīkla savienojumu atļaujām. Ja tās ir piešķirtas, rodas iespēja nolasīt IP un MAC adreses un tīkla nosaukumu, kas var palīdzēt identificēt cilvēku.

Parakstāmās atļaujas ir pieejas atļaujas, kas ļauj piekļūt lietotņu vai sistēmu datiem, ja lietotnei ar šo atļauju ir tāds pats sertifikāts kā lietotnei vai sistēmai, kas definē pieprasīto atļauju. Sistēmas noklusētās lietotnes lieto šīs atļaujas, lai nodrošinātu, ka trešās puses lietotnes nevarētu piekļūt sistēmas datiem vai mainīt sistēmas iestatījumus. Trešās puses lietotnes var veidot pašas savas atļaujas šajā grupā, lai ļautu piekļuvi saviem pakalpojumiem citām lietotnēm, bet nevarēs izmantot sistēmas atļaujas. Parakstāmo atļauju sarakstu var aplūkot 2.2. tabulā.

2.2. tabula

Android parakstāmo atļauju saraksts

Atļaujas kods	Atļaujas apraksts
BIND_ACCESSIBILITY_SERVICE	Ļauj piekļūt lietotāju atbalsta sistēmas komandām
BIND_AUTOFILL_SERVICE	Ļauj piekļūt automātiskās formu aizpildīšanas servisa funkcijām
BIND_CARRIER_SERVICES	Ļauj izmantot operatoru lietotņu servisus
BIND_CHOOSER_TARGET_SERVICE	Ļauj izmantot servisu, ar kuru iespējams izvēlēties lietotni noteiktas darbības veikšanai
BIND_CONDITION_PROVIDER_SERVICE	Ļauj izmantot servisu nosacījumu pārraudzībai
BIND_DEVICE_ADMIN	Ļauj izmantot iekārtas administratora servisa komandas
BIND_DREAM_SERVICE	Ļauj piekļūt interaktīvo ekrānsaudzētāju servisa iespējām
BIND_INCALL_SERVICE	Nodrošina piekļuvi zvanu funkciju pārvaldībai
BIND_INPUT_METHOD	Ļauj piekļūt servisam, ar kuru noteikt informācijas ievades metodi
BIND_MIDI_DEVICE_SERVICE	Sniedz piekļuvi virtuālās MIDI iekārtas funkcijām
BIND_NFC_SERVICE	Sniedz piekļuvi NFC sistēmas apstrādes servisam
BIND_NOTIFICATION_LISTENER_SERVICE	Nodrošina piekļuvi servisam, kas uzrauga paziņojumu notikumus
BIND_PRINT_SERVICE	Nodrošina piekļuvi printeru pārvaldības servisam

Atļaujas kods	Atļaujas apraksts
BIND_SCREENING_SERVICE	Atļauj piekļuvi servisam, ar kuru pārvaldīt ienākošos zvanus
BIND_TELECOM_CONNECTION_SERVICE	Sniedz piekļuvi zvanu funkcijas pārvaldības servisam
BIND_TEXT_SERVICE	Nodrošina pieeju teksta pārbaudes funkcijām
BIND_TV_INPUT	Sniedz pieeju video pārraides servisam
BIND_VISUAL_VOICEMAIL_SERVICE	Sniedz pieeju balss pastkastītes pārvaldības servisam
BIND_VOICE_INTERACTION	Atļauj piekļuvi balss komandu servisam
BIND_VPN_SERVICE	Ļauj izmantot VPN veidošanas funkcijas
BIND_VR_LISTENER_SERVICE	Ļauj izmantot virtuālās realitātes režīma funkcijas
BIND_WALLPAPER	Ļauj izmantot fona attēla pārvaldības servisu
CLEAR_APP_CACHE	Ļauj veikt jebkuras lietotnes kešatmiņas tīrīšanu
MANAGE_DOCUMENTS	Sniedz iespēju pārvaldīt pieejas atļaujas iekārtas dokumentiem
READ_VOICEMAIL	Ļauj nolasīt balss pastkastes informāciju
REQUEST_INSTALL_PACKAGES	Ļauj veikt pakotņu uzstādīšanas pieprasījumus
SYSTEM_ALERT_WINDOW	Ļauj veidot jaunus logus virs citām lietotnēm
WRITE_SETTINGS	Ļauj veikt sistēmas uzstādījumu nolasīšanu un labošanu
WRITE_VOICEMAIL	Ļauj veikt balss pastkastes ierakstu labošanu un dzēšanu

Bīstamās atļaujas atbild par pieeju datiem vai resursiem, kas var saturēt lietotāja informāciju vai potenciāli ietekmēt lietotāja datus, vai ietekmēt citu lietotņu darbību. Piemēram, skatīt lietotāja kontaktu grāmatiņu ir bīstama atļauja, jo tā nodrošina piekļuvi lietotāja kontaktiem, kas sniedz informāciju ar kādiem cilvēkiem un numuriem lietotājs ir pazīstams. Bīstamo atļauju gadījumā lietotājam jāapstiprina katra lietotnes pieprasītā atļauja atsevišķi. Kamēr lietotājs nav apstiprinājis atļauju, lietotne nevarēs nodrošināt funkcionalitātes, kas balstās uz šīs atļaujas.

2.3. tabula

Android bīstamo atļauju saraksts

Atļauju grupa	**Atļaujas kods**
Kalendārs	READ_CALENDAR
	WRITE_CALENDAR
Kamera	CAMERA
Kontaktu grāmatiņa	READ_CONTACTS
	WRITE_CONTACTS
	GET_ACCOUNTS
Lokācija	ACCESS_FINE_LOCATION
	ACCESS_COARSE_LOCATION
Mikrofons	RECORD_AUDIO
Telefons	READ_PHONE_STATE
	READ_PHONE_NUMBERS
	CALL_PHONE
	ANSWER_PHONE_CALLS
	READ_CALL_LOG
	WRITE_CALL_LOG
	ADD_VOICEMAIL
	USE_SIP
	PROCESS_OUTGOING_CALLS
	ANSWER_PHONE_CALLS
Sensori	BODY_SENSORS
SMS	SEND_SMS

	RECEIVE_SMS
	READ_SMS
	RECEIVE_WAP_PUSH
	RECEIVE_MMS
Glabāšana	READ_EXTERNAL_STORAGE
	WRITE_EXTERNAL_STORAGE

Tabulā 2.3. attēlotajā bīstamo atļauju sarakstā var redzēt, ka bīstamās atļaujas ir organizētas grupās, kas ir saistītas ar iespējām vai darbībām ar ierīci [3]. Visas atļaujas var būt organizētas grupās neatkarīgi no drošības līmeņa, taču lietotāju atļauju grupas ietekmē tikai, ja atļauja ir bīstama. Lietotājam ir jāapstiprina pieeja atļauju grupām nevis pašām atļaujām.

2.1.2 IOS atļaujas

IOS pilna atļauju sistēma tika pievienota uz iOS6 versiju 2012. Pieeju atrašanās vietai varēja kontrolēt jau krietni agrāk, sākot ar iPhone OS 2 versiju, kuru izdeva 2008.gadā. Atļaujas nav sadalītas atļauju grupās, tādejādi, iegūstot atļauju, tiek iegūta pilna pieeja visai funkcionalitātei un nevar atļaut tikai daļu pieejas.

2.4. tabula

IOS atļauju saraksts

Atļauja	**Atļaujas apraksta kods jeb Cocoa atslēga**	**iOS versijas**
NFC	NFCReaderUsageDescription	iOS 11 un jaunākas
Media	NSAppleMusicUsageDescription	visas
Bluetooth	NSBluetoothPeripheralUsageDescription	iOS 6.0 un jaunākas
Kalendārs	NSCalendarsUsageDescription	iOS 6.0 un jaunākas
Kamera	NSCameraUsageDescription	iOS 7.0 un jaunākas
Kontakti	NSContactsUsageDescription	iOS 6.0 un jaunākas
Face ID	NSFaceIDUsageDescription	iOS 11 un jaunākas
Health Share	NSHealthShareUsageDescription	iOS 8.0 un jaunākas

Atļauja	Atļaujas apraksta kods jeb Cocoa atslēga	iOS versijas
Health Update	NSHealthUpdateUsageDescription	iOS 8.0 un jaunākas
Home Kit	NSHomeKitUsageDescription	visas
Lokācija visu laiku	NSLocationAlwaysUsageDescription	iOS 8.0 un jaunākas
Lokācija	NSLocationUsageDescription	iOS 6.0 un jaunākas, **novecojis** sākot ar iOS 8
Lokācija, kad lieto	NSLocationWhenInUseUsageDescription	iOS 8.0 un jaunākas
Mikrofons	NSMicrophoneUsageDescription	iOS 7.0 un jaunākas
Kustības un akselerometrs	NSMotionUsageDescription	iOS 7.0 un jaunākas
Fotogrāfiju bibliotēkas tikai rakstīšanai	NSPhotoLibraryAddUsageDescription	iOS 11 un jaunākas
Fotogrāfiju bibliotēkas izmantošana	NSPhotoLibraryUsageDescription	iOS 6.0 un jaunākas
Atgādinājumi	NSRemindersUsageDescription	iOS 6.0 un jaunākas
Runas atpazīšana	NSSpeechRecognitionUsageDescription	visas
Sociālo tīklu konti	-	visas
Paziņojumi	-	visas

Tabulā 2.4. attēlotas visas atļaujas, atļauju apraksta kodi un IOS versija no kurām atļauja ir pieejama operētājsistēmā. Atļaujām netiek izmantoti kodi no lietotņu izstrādātāju puses, jo tās tiek izsauktas automātiski. Atļaujas paziņojums lietotājam tiek automātiski pirmo reizi parādīts, kad lietotne mēģina pieslēgties kādai no atļaujas sargātajām pieejām, piemēram, lietotne pēc pogas "Kamera" nospiešanas vēlās izmantot kameru, tad pēc pirmās pogas nospiešanas reizes lietotājam parādīsies logs, kas pieprasīs kameras atļauju. Katrai atļaujai toties ir atļauju apraksta kodi. IOS atļauju apraksta kodi ir daļa no Cocoa atslēgām.

Info.plist failā kopš iOS 10.versijas obligāti ir jānorāda atļaujas apraksts, lai paskaidrotu lietotājam atļaujas izmantošanas iemeslu. Pirms iOS 10.versijas šie apraksti nebija obligāti. 2.1 attēlā attēlotā info.plist piemēra atļaujas apraksta koda fragmentā var apskatīt, kā tiek izmantots info.plist fails un kā tiek norādīts iemesls, kāpēc lietotnei nepieciešama atļauja.

```
…
<key>NSCameraUsageDescription</key>
<string>Lietotnei nepieciešama kamera, lai uzņemtu fotogrāfijas, kuras sūtīt citiem lietotājiem.</string>
…
```

2.1. att. **Info.plist faila piemēra ar atļaujas apraksta koda fragments**

Papildus Cocoa atslēgām, kas ir atļaujas, papildus ir vēl arī iOS atslēgu saraksts, kurā ir ne tik daudz atļaujas, bet drīzāk informatīva informācija par lietotni, piemēram, aizpildot UIBackgroundModes ar vērtību, lietotājs tiek informēts, ka lietotnei nepieciešams turpināt strādāt arī zaudējot fokusu jeb kļūstot par fona procesu.

2.2 Atļauju salīdzinājums

Atļauju organizēšanas modelis kopumā Android un IOS operētājsistēmām ir līdzīgs, tomēr, vairāk iedziļinoties, atšķirības ir diezgan lielas. Abas operētājsistēmas izmanto atšķirīgu pieeju lietotņu darbināšanai nekā tradicionālās sistēmas, kas ir bāzētas uz darbvirsmām. Tradiconālās sistēmas lietotnes strādā zem lietotāja konta, kurš tās palaida, un strādā ar atļaujām, kas bija piešķirtas šim lietotāja kontam. Nav mehānisma, kas sadalītu atļaujas pa lietotnēm. Turklāt visas lietotnes strādā ar tādu pašu pieejas līmeni sistēmas lietojumprogrammas saskarnēm(API) un citiem sistēmas servisiem kā atbilstošā lietotāja kontam piešķirtais. Piemēram, dokumentu pārvaldes lietotnei un VoIP lietotnēm abām ir vienāda pieeja tīklam, jo abas strādā ar vienādu lietotāja identifikatoru pēc noklusējuma. Ja lietotājam ir *root* pieeja, tad visām lietotnēm pēc noklusējuma būs pilna pieeja visiem datiem. Tāds ir pamata pieņēmums tradicionālajā drošības modelī – visas lietotnes no lietotāja vienādi saņem visas privilēģijas un atļaujas [10].

Android un iOS moduļos, katra lietotne strādā tā it kā tai būtu savs lietotāja konts. Rezultātā tiek atdalītas lietotnes un nodrošināts, ka katra lietotne pēc noklusējuma var tikt klāt tikai saviem nevis citu lietotņu datiem. Abās operētājsistēmās papildus tiek nodrošināta arī speciāla atļauju sistēma. Ar atļauju sistēmu lietotnes var piekļūt svešiem servisiem vai datiem, kas ir sensitīvi vai

bīstami, kā piemēram lietotāja personīgie dati vai bluetooth izmantošana. Šādu servisu izmantošanai tiek no lietotāja pieprasīta atsevišķa atļauja katrai lietotnei, kas grib izmantot konkrēto servisu.

Savstarpēji salīdzinot abu operētājsistēmu atļauju modeļus, atšķirības ir ātri pamanāmas. IOS izceļas ar stingro nostāju, ka lietotnēm nevar būt pārāk liela pieeja sistēmas iestatījumiem un servisiem, turklāt lietotnēm nemaz nav iespējams daudzas funkcionalitātes pievienot. Servisiem ir savi oficiālie API un tie izstrādātajiem ir jālieto, lai lietotne tiktu atzīta. Turklāt daudziem servisiem nemaz nav oficiāli API, kā piemēram, iOS nav iespējams mainīt fona attēlus, mainīt zvanu signālu vai mainīt tīkla pieslēgumu no lietotnes, visas šīs darbības ir veicamas tikai ar Apple veidotām lietotnēm vai izmantojot iestatījumu lietotni. IOS atļauju saraksts [2.4.] izskatās krietni īsāks nekākā Android atļauju saraksti [2.1., 2.2., 2.3.], bet tas ir lielo sistēmas ierobežojumu dēļ. Neoficiālas lietojumprogrammatūras saskarnes (API) lietošana vai citādāka piekļuve servisiem, veidojot lietotnes, nav aizliegta, bet lietotnes ar šāda veida pirmkodu tiek filtrētas, netiek atzītas un netiek ievietotas Apple App veikalā. Lai informētu izstrādātājus un novērstu neskaidrības, Apple ir izstrādājis noteikumus, kas ir jāievēro, lai lietotne tiktu apstiprināta Apple App veikalā. Noteikumi skan šādi: “Lietotnes drīkst tikai lietot publiskos API un jāstrādā uz jaunākajām operētājsistēmām. [..] Lietotnēm API un satvari jālieto priekš to paredzētajiem mērķiem un jānorāda to izmantošana lietotnes aprakstā. [..] Lietotnēm vajag būt autonomam savā pakotnē un nedrīkst lasīt vai rakstīt datus ārpus atļautā konteinera apgabala. Kā arī lietotnes nedrīkst lejupielādēt, uzstādīt vai izpildīt kodu, tai skaitā citas lietotnes.” [13].

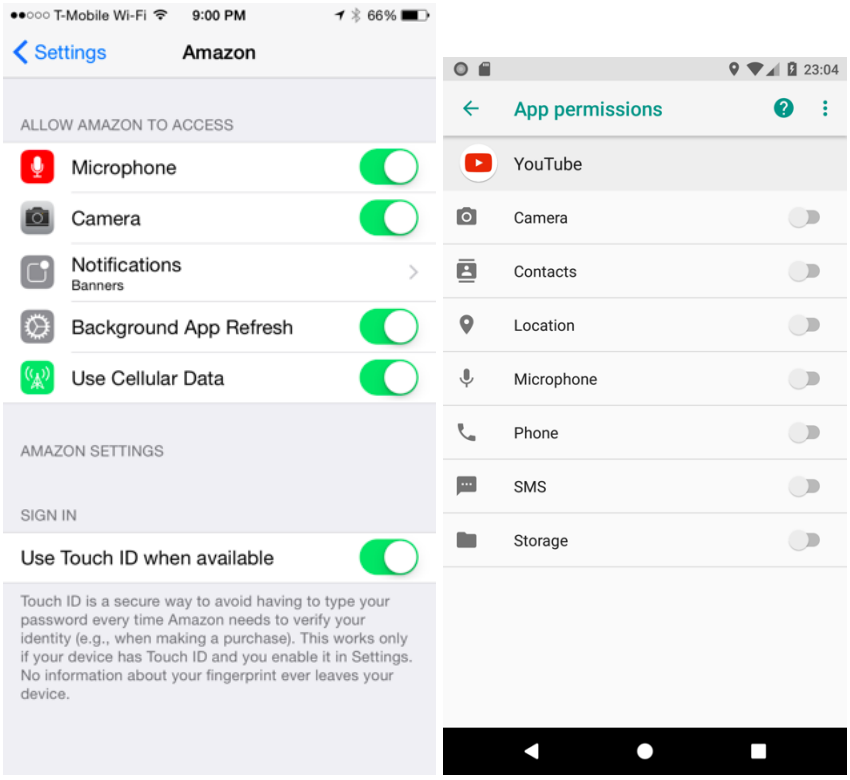
IOS ir informatīvas pazīmes par lietotni, kas sakrīt ar dažām no Android parastā tipa atļaujām, piemēram, ka lietotne strādās fonā vai drīkstēs veidot īsceļu. Zibspuldzes ieslēgšana pieprasa kameras atļauju Android operētājsistēmā, bet iOS ieslēgt zibspuldzi var bez jebkādas atļaujas. Tāpat arī interneta pieeja Android atļaujām ir jānorāda, bet iOS tā tiek sniegta uzreiz.

No lietotņu atļauju skatu punkta Android drošības modelis ir brīvāks nekā iOS tādā ziņā, ka izstrādātājiem ir pieejamas vairāk funkcionalitātes un atļaujas, kas nodrošina pieeju šīm funkcionalitātēm. Rezultātā izstrādātājiem ir vairāk kontroles pār sistēmas darbību, bet lielāka atbildība pret lietotāju. Operētājsistēmas izstrādātājiem tas nozīmē arī vairāk darba, atļaujas pareizi veidojot un pārvaldot. Ar katru jauno versiju Android ir izmaiņas atļauju sistēmā, lai nodrošinātu labāku drošības līmeni lietotāja datiem. Piemēram, ar Android 9.0 versiju vairs nebūs iespējams fona režīmā lietot mikrofonu, kameru vai iegūt sensoru informāciju [14].

Vēsturiski Android atļauju sistēma bija izstrādāta pirms iOS publiskas atļaujas, bet IOS atļauju sistēma bija krietni labāka, kad tā tika izstrādāta, un Android līdzsvaroja abu atļaujas organizēšanas moduļu drošību tikai ar 6. versiju. Vēl Android un iOS atļauju organizēšanas modeļos ir atšķirība, ka Android atļaujas organizē atļauju grupās, t.i. visas atļaujas, kas ir saistītas ar kontaktiem, ir sagrupētas kontaktu atļauju grupā. IOS katra atļauja tomēr ir patstāvīga un netiek grupēta ar citām, pat ja atļaujas atbilst par vienu funkciju apgabalu, piemēram, iOS fotogrāfijas bibliotēkas pieejai ir divas atļaujas: viena rakstīšanai un viena izmantošanai. Android sistēmā tikai bīstamās atļaujas prasa lietotāja apstiprinājumu, kamēr iOS visas atļaujas pieprasa lietotāja apstiprinājumu. IOS atļauju skaits ir krietni mazāks, salīdzinot ar Android, bet ja apzinās, ka tikai bīstamās Android atļaujas ietekmē lietotāju, tad atļauju skaits ir praktiski vienāds.

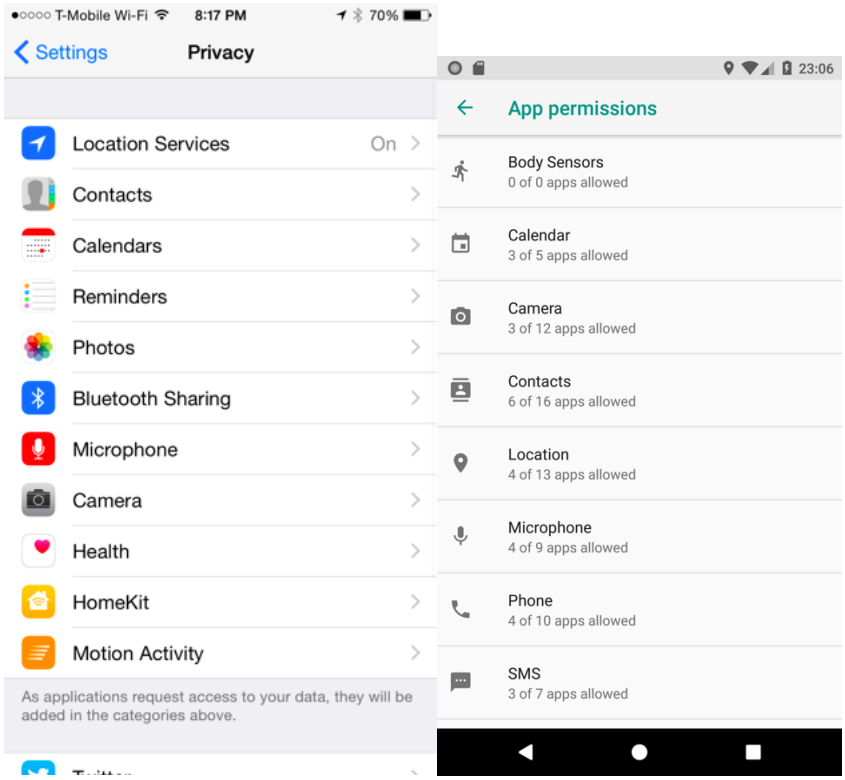
2.3 Atļauju pieprasīšana un pārvaldība

Android un iOS operētājsistēmu atļauju organizēšanas moduļi ir atšķirīgi savā starpā ar to, kā lietotāji un izstrādāji tos var izmantot. Android katra atļauja ir jāpieraksta lietotnes apraksta dokumentā un atsevišķi jāizsauc, ja izmantota ir bīstama drošības līmeņa atļauja. Savukārt iOS atļaujai ir jāpieraksta paskaidrojums un lietotājam paziņojums automātiski parādīsies, kad lietotne vēlas izmantot kādu funkciju, kam ir atsevišķa atļauja. Abu operētājsistēmu atļauju modeļi ir veidoti, domājot par vienkāršību un par lietotāju ērtībām. Piemēram, atļaujas, kurām nepieciešama atsevišķa lietotāja piekrišana, nav pārlieku smalki izdalītas, lai lietotājam nebūtu manuāli jāapstiprina 20 atsevišķas atļaujas, kas lietotājam visticamāk nešķistu ērti. Tāpēc, piemēram, lietotnes pieeja internetam var tik izmantota bez atļaujas pieprasīšanas. Abas operētājsistēmas iesaka lietotņu izstrādātājiem, veidojot savas lietotnes, ņemot vērā, ka atļaujas lietotnei ir iespējams arī liegt. Gadījums, ka lietotne bez kādas atļaujas vairs pilnībā nestrādā, vairs nav tik bieži sastopams.



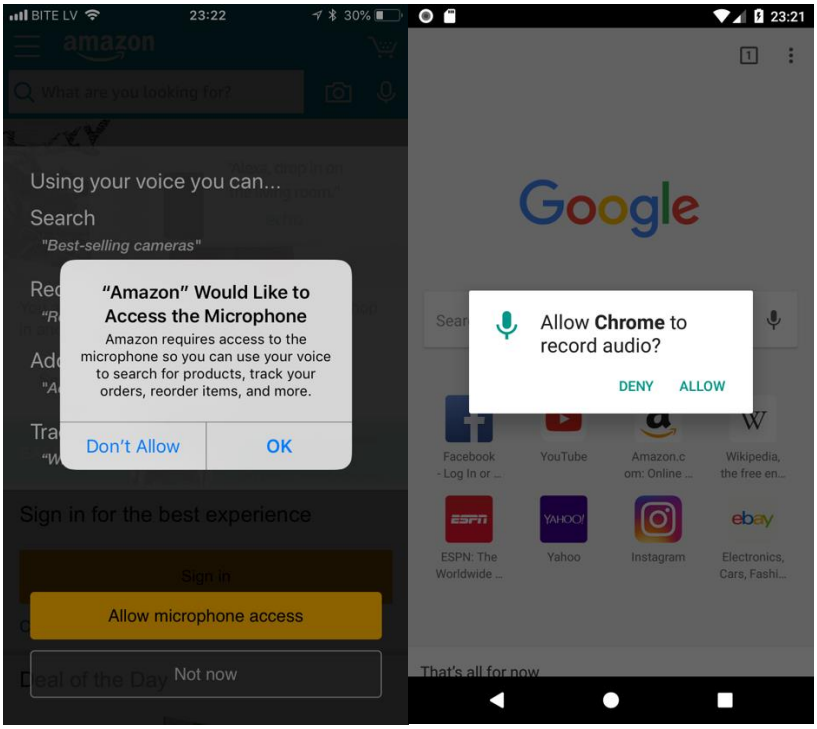
2.2. att. **Lietotnes atļauju pārvaldības skatu ekrānuzņēmumi iOS (pa kreisi) un Android (pa labi) sistēmās**

2.2. attēla redzamajos ekrānuzņēmumos var apskatīt, ka abas operētājsistēmas arī lietotnes atļauju pārvaldību ir veidojušas līdzīgi. Iestatījumu lietotnē ir iespējams atvērt katras lietotnes atļauju pārvaldības skatu, kurā var redzēt visas atļaujas, ko lietotne var izmantot. Lietotnei var noņemt visas nevēlamās vai pielikt visas vēlamās atļaujas. 2.2. attēlā redzamajā Android lietotnes atļauju pārvaldības skata ekrānuzņēmumā (pa labi) var arī izvēlēties iespēju pārskatīt visas atļaujas. Šo iespēju izvēloties, tiek parādītas visas atļaujas saraksta formātā, ieskaitot arī normālās un parakstītās atļaujas. IOS var pārvaldīt visas pieejas šajā skatā, bet Android tikai bīstamās atļaujas ir iespējams atļaut vai neatļaut lietotnei izmantot.



2.3. att. **Atļauju pārvaldības skatu ekrānuzņēmumi iOS (pa kreisi) un Android (pa labi) sistēmās**

2.3. attēlā redzamajos ekrānuzņēmumos var apskatīt, ka abās operētājsistēmās var apskatīt visas atļaujas kā arī apskatīt, kādas lietotnes lieto katru atļauju. No šiem skatiem ir iespējams arī labot lietotnes atļauju. Android šis skats gan ir papildināts ar statistiku, cik lietotnes vēlās izmantot šo atļauju un cik lietotnēm šī atļauja ir sniegta, kas daļai lietotāju var likties saistoša un noderīga informācija.



2.4. att. **Atļauju pieprasīšanas logu ekrānuzņēmumi iOS (pa kreisi) un Android (pa labi) sistēmās**

2.4. attēlā redzamajos ekrānuzņēmumos var apskatīt, kā izskatās atļaujas pieprasīšanas modālais logs katrā no operētājsistēmām. Kopš iOS 11 versijas katrai lietotnei ir jāpievieno savs paskaidrojums, kāpēc pieprasītā atļauja ir nepieciešama, tāpēc 2.4. attēlā redzamajā iOS logs (pa kreisi) satur krietni vairāk informācijas nekā Android logs (pa labi). Abi ekrānuzņēmumos redzamie pieprasījumi tiek attēloti lietotājam pēc lietotnes pieprasījuma izmantot mikrofonu. IOS atļaujas pieprasīšanas paziņojums rādīsies tikai pirmo reizi, vēloties izmantot funkcionalitāti, bet ja tiks noraidīts, tad iespēja piešķirt atļauju būs tikai caur iestatījumu lietotni [11, 12]. Android atļaujas pieprasīšanas paziņojums rādīsies katru reizi līdz netiks atzīmēts, ka lietotājs vairāk nevēlās redzēt šo logu [3]. Pēc atzīmes, ka lietotājs šādu paziņojumu vairs nevēlas redzēt, atļauju iespējams piešķirt tikai no iestatījumu lietotnes.

Daļa Android lietotņu tomēr vēl izmanto veco atļauju modeli, kas paredz atļauju pieprasīšanu pirms lietotne tiek uzstādīta uz ierīces. Atļaujas nevar pārvaldīt un lietotnēm, kas veidotas pirms Android 6.0 versijas, piešķirtās atļaujas nav iespējams noņemt, neatsakoties no pašas lietotnes. Lietotājam nākas izvēlēties vai lietotni pieņemt un dod visas pieprasītās atļaujas vai arī neizmantot vispār, jo tā netiks uzstādīta, ja lietotājs nesniedz visas pieprasītās atļaujas.

3. Uzlabota atļauju organizēšanas modeļa izveide

Nodaļā tiek apskatīts, kādi moderni problēmu risinājumi eksistē un kādi ir paredzēti nākotnē. Tālāk apskatīts, kā var risināt iepriekš nodaļā 1.3 apskatītās problēmas. Tiek izvēlēti risinājumi no iespējamajiem un tiek veidots uzlabots atļauju organizēšanas modelis. Izveidotais modelis tiek aprakstīts un katram izvēlētajam pamatojumam tiek dots pamatojums, kāpēc tas tiek izvēlēts. Pēc tam rezultātu nodaļā 3.3 tiek apskatīts, kā tieši jaunais modelis risina katru no problēmām un īsi tiek salīdzināts ar iOS un Android atļauju modeļiem.

3.1 Iespējamie risinājumi ar atļaujām saistītajām problēmām

Nodaļā 1.3 "Ar atļaujām saistītās problēmas" tika aprakstītas trīs nozīmīgas ar atļaujām saistītas problēmas. Ir dažādi veidi kā šīs problēmas risināt - sākot ar atļauju sadalīšanu detalizētāk līdz pat lietotņu koda pārraudzīšanai pirms lietotne ir apstiprināta izplatītāja veikalā. Vienmēr pirms problēmu risināšanas prātā ir jāpatur arī lietotāja ērtība. Atļauju pieprasījumi nedrīkstētu lietotājam sagādāt pārāk lielas neērtības un lieki aizkavēt no savu mērķu sasniegšanas lietotnē.

Daži piemēri, kas apraksta risinājumus, kas jau tiek pielietoti, lai šīs problēmas novērstu vai vismaz mazinātu, ir:

- Lietotņu pirmkodu analizē pirms akceptēšanas lietotņu veikalos [13, 23];
- Google Play protect sistēma fona režīmā pārrauga un analizē lietotnes un to darbības. Lietotņu analīzi apkopo un aizdomīgās lietotnes tiek apskatītas tuvāk. 2017.gadā 700 tūkstoši nelabvēlīgu lietotņu tika izņemtas no Google Play veikala pateicoties Play-Protect sistēmai [23, 24, 25];
- Lietotnes grupē ar līdzīgām pēc funkcionalitātes ar mašīnmācīšanos algoritmiem un tuvāk apskata tās lietotnes, kas pēc atļaujām nesakrīt ar lielāko daļu no grupas [1];
- Atļaujas pieprasīšanas laikā parāda paskaidrojumu no izstrādātāja, kādēļ atļauja ir nepieciešama vai kādu funkcionalitāti tā dos [28, 5, 3].

Daži piemēri, kas apraksta risinājumus, kas tikai tiks pielietoti vai kas tiek pielietoti, bet vēl nav oficiāli pieņemti:

- Mikrofonu un kameru neļaut izmantot fona režīmā [16,14];

- Lietotājus grupē pēc aptaujas jautājumu atbildēm un iesaka kādām lietotnēm piešķirt atļaujas automātiski pēc grupas, kurā lietotājs ir iešķirots. Personalized Privacy Assistant ir lietotne, kas dara iepriekš minēto, bet diemžēl pieprasa Root lietotāja pieeju ierīcei [26].

Kopumā gan Android, gan iOS pielieto atšķirīgas bet labas metodes, tāpēc autors savā darbā veido teorētisku modeli, kas paņem labāko no Android un iOS pieejamām un citas iegūtās informācijas un pieredzes. Rezultātā tiek radīts iespējams vēl efektīvāks atļauju pārvaldības modelis, kas risina trīs pamata problēmas, ka aprakstītas šajā darbā, un reizē būtu ērts no lietotāju skatu punkta. Ņemot vērā iepriekš aprakstītos risinājumus un dažādās Android un iOS pieejas atļauju organizēšanai, tālāk ir aprakstīta daļa jau eksistējošu risinājumu, kas varētu risināt izvirzītās pamatproblēmas.

Pirmā aprakstītā problēma ir, ka lietotne var piekļūt pārāk daudz datiem, jo atļauja pietiekoši neierobežo pieeju. Šīs problēmas risinājumi tieši saistās ar atļaujām un to realizāciju. Iespējamie risinājumi ir:

- Atļaujas sadalīt smalkāk un grupēt;
- Ieviest iespēju fona režīmā neatļaut pieeju funkcionalitātei vai vismaz to ierobežot;
- Ieviest funkciju moduļus, caur kuriem var piekļūt resursiem vai datiem, bet bez papildus atļaujas neļaut šos datus izmantot ārpus lietotnes, piemēram, neļaut nosūtīt uz serveriem analīzei. Citiem vārdiem ļaut datus tikai lasīt, taču ne pārsūtīt tālāk;
- Lietotāju informēt par mikrofona, sensoru, kameras vai atrašanās vietas noteikšanas izmantošanu;
- Dalīt interneta pieeju lietotnei vai lietotnes reklāmām [27];
- Atļaujas sadalīt smalkāk un veidot hierarhijai līdzīgu modeli [27].

Otrā problēma ir, ka lietotājam var tikt liegta lietotnes izmantošana, ja viņš atsakās piešķirt kādu atļauju. Šo problēmu pilnībā nevar atrisināt, jo ir lietotnes, kuras bez kādas atļaujas nemaz nevar darboties, piemēram, kameru lietotnes, jo viņām vajag atļauju lietot kameru un saglabāt uzņemtos kadrus sistēmas atmiņā. Tomēr lai samazinātu iespēju, ka lietotnes pieprasa pārāk daudz pieejas, var veikt šādas darbības:

- Sodīt lietotnes ar reitinga pazemināšanu meklēšanas rezultātos, ja lietotne pieprasa vairāk atļaujas nekā nepieciešamas pamata funkcionalitātei un lietotājs nevar izmantot lietotni bez to piešķiršanas;

- Informēt lietotāju pirms iegādes, ka lietotne pieprasa visas atļaujas uzreiz. Lietotājam rādīt arī, kādas atļaujas ir obligāti vajadzīgas, lai lietotne strādātu, bet kādas ir izvēles.
- Ekstrēmāks variants būtu neļaut lietotnēm tikt ievietotām lietotņu izplatīšanas veikalos, ja lietotājam nav atļauts izmantot lietotni bez atļauju piešķiršanu, kas nav saistīta ar pamata funkcionalitāti.

Trešā problēma ir, ka lietotājiem ir pārāk maz iespēju pārvaldīt piešķirtās atļaujas vai tas ir sarežģīti. Problēma saistās tieši ar lietotāja un lietotnes atļauju pieprasījumiem. Jāņem vērā, ka risinājums nedrīkst lieku lietotāja apgrūtināšanu ar pārāk daudz vai nesaprotamiem atļauju pieprasījumiem. Daži no risinājumi uzskaitīti zemāk:

- Pievienot saīsni no lietotnes uz lietotnes iestatījumiem, lai lietotājs varētu vieglāk un ātrāk piekļūt atļaujām;
- Ieviest iespēju atļaujas piešķirt uz laiku un tikai pēc regulāras pieejas sniegšanas atļauju piešķirt uz visu laiku;
- Pievienot iespēju ierobežot pieeju internetam;
- Lielo atļauju pieprasīšanu veikt jau pirms uzstādīšanas ar vienu labi pārskatāmu logu, un tālāk pieprasīt, tikai nesniegtās atļaujas, kad nepieciešams.

Nodaļā 3.2 tiks apskatīts, kuri problēmu risinājumi no šajā nodaļā apskatītajiem varētu tikt izvēlēti, lai uzlabotu atļauju organizēšanas modeļus Android un iOS lietotnēs.

3.2 Uzlabotā atļauju modeļa apraksts

Uzlabotā atļauju modeļa galvenais mērķis ir risināt 1.3 nodaļā apskatītās problēmas un tās pēc iespējas labāk novērst. Par pamatu atļauju organizēšanas modelim tiek izmantots Android atļauju modelis, jo Android atļauju saraksts ir krietni plašāks un Android sistēma ir mazāk ierobežota, salīdzinot ar iOS sistēmu, kas palīdzēs apskatīt plašākus problēmu risinājumus.

Atļauju sarakstam par pamatu tiek ņemts Android atļauju saraksts ar atļauju drošības līmeņiem. Tomēr jaunajā modelī ir paredzētas izmaiņas, jo Android atļauju modelī ir atļaujas, kuras ir pārāk plašas un ir nepieciešamas sadalīt smalkāk. Izmaiņas ir apkopotas 3.1 tabulā.

Sīkāk par katru izmaiņu:

- Interneta pieejas atļauja ir sadalīta par divām jaunām atļaujām – lietotnes piekļuve internetam un lietotnes reklāmu piekļuve internetam. Lietotnes piekļuve internetam ir bīstamā atļauja, jo bez tās nevar notikt datu noplūde, jo nevar aizsūtīt lietotāja datus. Lietotnes reklāmu piekļuve internetam ir parasta atļauja, jo tas ir viens no pamata veidiem, kā lietotnes saņem ienākumus. Abas jaunās atļaujas ir grupētas interneta atļauju grupā. Šāda izmaiņa ir nepieciešama, lai lietotājs varētu kontrolēt lietotnes piekļuvi internetam, piemēram, kalkulatoram nav iemesla izmantot internetu, bet bezmaksas kalkulatora lietotnē iespējams būtu reklāmas, kurām vajag internetu. Interneta kontrole nodrošinātu, ka lietotne nevarētu lieki lietot datus lietotājam nevēloties.
- Kameras lietotne ir sadalīta trīs jaunās atļaujās un atļaujas ir sagrupētas kameras atļauju grupā. Jaunās atļaujas ir zibspuldzes darbināšanai, bilžu uzņemšanai un video ierakstīšanai. Bilžu un video atļaujas ir bīstamas, jo uzņemtie kadri ir lietotāja personīgie dati. Zibspuldzes darbināšana atļauja ir parastā līmeņa atļauja, jo ieslēdzot zibspuldzi nevar iegūt nekādus lietotāja datus, vienīgais var lietotāju kaitināt ar zibsnīšanu. Zibspuldzes atdalīšana nodrošinās, ka lietotnes vairs netiks pie kameras izmantošanas, kaut arī lietotnei vajadzētu tikai zibspuldzi. Ne vienmēr lietotnēm vajag gan bilžu uzņemšanas, gan video ierakstīšanas funkcionalitātes, piemēram, svītru kodu lasītāju lietotnēm bieži ir nepieciešamas tikai uzņemt bildi, lai analizētu svītra kodu, tāpēc prātīgi būtu sadalīt šīs funkcionalitātes.
- Atļauja READ_PHONE _STATE aizstāšana ar trīs jaunām atļaujām: atļauja lasīt mobilo sakaru tīkla informāciju, atļauja lasīt tālruņa identifikācijas informāciju un atļauja skatīt zvanīšanas statusu. Visas atļaujas ir bīstamas, jo satur lietotāja datus. Šī izmaiņa ir svarīga, jo daudzas lietotnes pieprasa šo atļauju tikai lai apturētu lietotnes audio signālu, kamēr tālrunis ir zvanīšanas režīmā. Arī zvanītāja numura noteikšanai vai tālruņa zvanīšanas režīma pārraudzībai nebūtu jābūt vienā atļaujā līmenī ar tālruņa identifikācijas informācijas iegūšanu. Tālruņa identifikācija satur ļoti privātu informāciju par tālruni, piemēram, IMEI numuru. IMEI numurs ir lielākoties unikāls numurs, kuru mainīt ir iespējams, taču tas ir pretlikumīgi. Telefonu pēc atrašanas parasti identificē ar šo numuru. Šo iemeslu dēļ zemāk

redzamajā tabulā 3.1. atļaujas zvanīšanas statusa lasīšanai un tālruņa identifikācijas informācijas lasīšanai ir izdalītas atsevišķi.

3.1. tabula

Android atļauju saraksta izmaiņu tabula

Atļauju grupa	Atļaujas kods	Atļaujas apraksts	Drošības līmenis
Internets	APP_INTERNET	Atļauja lietotnei piekļūt internetam.	Bīstama
	AD_INTERNET	Atļauja lietotnes reklāmām piekļūt internetam.	Parasta
Kamera	FLASHLIGHT	Atļauja ieslēgt vai izslēgt zibspuldzi.	Parasta
	PHOTO_TAKING	Atļauja izmantot kameru bilžu uzņemšanai.	Bīstama
	VIDEO_RECORDING	Atļauja izmantot kameru video ierakstīšanai.	Bīstama
Tālrunis	READ_CELLULAR_-NETWORK_INFO	Atļauja lasīt mobilo sakaru tīkla informāciju.	Bīstama
	READ_PHONE_ID	Atļauja lasīt tālruņa identifikācijas informāciju.	Bīstama
	READ _CALL_STATUS	Atļauja skatīt zvanīšanas statusu.	Bīstama

Apdomājot trīs variantus, kā kārtot un veidot atļaujas uzlabotajā atļauju modelī, autors izlēma, ka uzlabotajā modelī atļaujas tiek kārtotas pēc hierarhijas tipa un pagaidām atļauju modelī paredzētas gan atļauju grupas, gan atļaujas, kur atļauja atrodas zem atļauju grupas. Pārējie varianti - atļauju saraksts bez grupām un Android atļauju modelis ar atļaujām un atļauju grupām, tika atzīti par sliktākiem pilnīgai atļauju pārvaldīšanai. Parastā atļauju modelī lietotājam nav iespējas lietotnei piešķirt veselu kopu ar atļaujām, kas samazinātu lietotājam nepieciešamās darbības. Android atļauju modelī ir gan atļauju grupas, gan atļaujas, taču lietotājs var piešķirt tikai atļauju grupu un

detalizētāka pārvaldība nemaz nav iespējama. Tāpēc uzlabotajā atļauju modelī ir gan atļaujas, gan atļauju grupas, bet tās ir iespējams pārvaldīt katru atsevišķi. Atšķirībā no Android atļauju moduļa uzlabotajā modelī bīstamās atļaujas var būt vienā grupā ar parastajām atļaujām, taču atļauju pieprasīšana no drošības līmeņu puses netiks mainīta, tātad lietotājs varēs pārvaldīt tikai bīstamās atļaujas un parastās tiks automātiski piešķirtas lietotnei uzstādīšanas laikā.

Hierarhiskā tipa atļauju modelis ļauj lietotājam pārvaldīt gan atļauju grupas, gan katru atļauju atsevišķi un nodrošina, ka atļaujas var dalīt detalizētākās atļaujās, bet tomēr tās ir pietiekami vienkārši pārvaldīt. Android atļauju modelim ir atļauju grupas un atļaujas, tomēr lietotājs var piešķirt tikai veselu atļauju grupu nevis pārvaldīt katru atļauju atsevišķi, kas uzlabotajā atļauju modelī nav problēma. Turklāt uzlabotajā atļauju modelī tiek risināta arī problēma ar jaunu atļauju pievienošanu lietotnes atjaunināšanā. Android lietotnes atjaunošanā jaunās, pievienotās atļaujas automātiski tiek piešķirtas, ja tās ir atļauju grupā, kas ir jau lietotnei piešķirta. Lietotājam nav nekādas informācijas, ka šādas izmaiņas ir veiktas, jo tās nav obligāti jānorāda izmaiņu aprakstos. Piemēram, lietotājam netiek sniegta informācija, ka lietotne iepriekš izmantoja aptuveno atrašanās vietu, bet pēc atjauninājumiem, lietotne izmanto precīzu lietotāja atrašanās vietu un patērē vairāk bateriju [29]. Šo problēmu atrisinātu hierarhiskā sistēma, jo jaunā atļauja būtu atspējota pēc lietotnes atjaunošanas, un tikai, kad lietotne vēlēsies piekļūt jaunajai funkcionalitātei, lietotājam tiks pieprasīta nesen pievienotā atļauja.

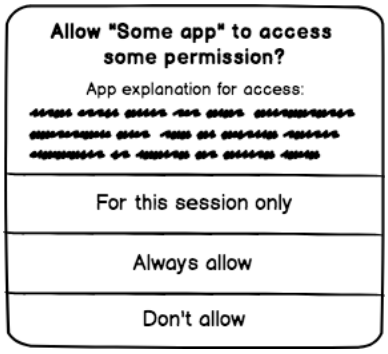
Drošības līmeņi labi sadala atļaujas grupās, kas apraksta, kurām atļaujām ir pieeja lietotāja datiem vai resursiem, kurām atļaujām ir jāpaliek tikai sistēmai izmantojamām un kuras atļaujas var brīvi lietotnei piešķirt, jo nodrošina ērtības funkcijas. Parastās atļaujas tīri teorētiski var aizstāt ar atzīmēm ar karodziņu kā to dara iOS operētājsistēmā, bet tas nav nepieciešamas, tāpēc tās ir atstātas, kā Android atļauju modelī tās ir tikušas realizētas.

Uzlabotajā atļauju modelī tiek saglabāta Android nesen izveidotā sistēma, kas lietotnei neļauj piekļūt kamerai un mikrofonam fona režīmā, jo nav pietiekoši laba pamatojuma, kāpēc lietotnēm vajadzētu piekļūt šiem resursiem fona režīmā. Sensorus vai atrašanās vietu fona režīmā varēs izmanto, jo sensoru dati un atrašanās vieta ir nepieciešams sporta aktivitāšu lietotnēm, kas skaita soļus, veido atrašanās vietu vēsturi un skaita noskrietos kilometrus, lai pēc tam apkopotu lietotāja kopējā treniņa apmērus un parādītu aptuveno kaloriju daudzumu, kas patērēts pēc ievāktajiem datiem. Pienesums, ko radītu papildus fona darbību ierobežojošo atļauju pievienošana modelim, ļoti iespējams neatsvērtu papildus apgrūtinājumu atļauju pārvaldībai vairākumam lietotāju. Ja tiktu nolemts pievienot fona darbību ierobežojošu atļauju kaut vai tikai katrai atļauju

grupai nevis katrai atļaujai, tās jau būtu 12 jaunas atļaujas, kas kopējo atļauju skaitu palielinātu no 35 uz 47. Jau šī brīža atļauju skaits ir liels un varētu traucēt vieglu atļauju uztveramību no lietotāja viedokļa, tāpēc to nevajadzētu lieki palielināt. Vienas atļaujas pievienošana, kas ļautu lietotnei strādāt fonā režīmā, arī nav pietiekoši vērtīga, jo tā visdrīzākais tiktu veidota kā obligātā atļauja un nesniegtu lietotājam iespēju pārvaldīt lietotnes piekļuvi resursiem fona režīmā. Šo iemeslu dēļ tika nolemts uzlabotajā modelī papildus fona darbību ierobežojošo atļauju skaitu nepalielināt.

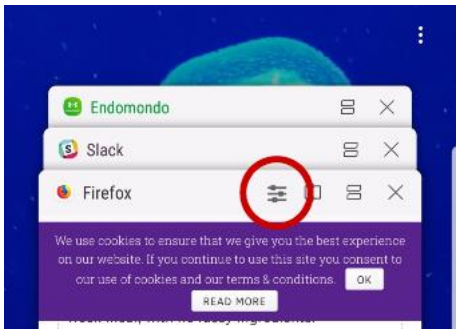
Uzlabotajā atļauju organizēšanas modelī lietotājam ir papildus veidi, kā pārvaldīt atļaujas - piešķirt (vai atspējot) katru bīstamo atļauju atsevišķi vai piešķirt pieeju visai atļauju grupai, dot atļauju uz lietotnes lietošanas sesijas laiku, saīsne uz lietotnes pārvaldības sadaļu un jauns atļauju piešķiršanas logs, kamēr lietotne tiek uzstādīta uz sistēmas. Paziņojumi par resursu izmantošanu, piemēram, paziņojumi par atrašanās vietas izmantošanu netiek ieviesti, jo to sniegtā informācija ir niecīga un ne visi lietotāji pamanīs tos, kā arī lielai daļai lietotāju, tie nebūs būtiski.

Uzlabotajā modelī lietotājam ir iespēja atļauju piešķirt arī uz noteiktu lietotnes izmantošanu laiku. Piemēram, lietotājs vēlēsies lietotnē izmēģināt kādu funkcionalitāti, kas pieprasīs jaunu atļauju, lietotājs varēs to piešķirt tikai uz šo lietotnes dzīves ciklu, tātad pēc lietotnes aizvēršanas, atļauja vairs nebūs piešķirta. Šāds risinājums, iespējams, veicinātu lietotāju vairāk izmēģināt funkcionalitātes, kuras iepriekš nevēlējās izmantot tikai tāpēc, ka tās šķita aizdomīgi prasīgas atļauju ziņā, un arī nodrošinātu lietotājam drošības sajūtu, ka lietotne varēs piekļūt privātiem datiem bez lietotāja ziņas. Attēlā 3.1. ir redzams, kā varētu izskatīties lietotāja atļaujas pieprasīšanas logs. Loga dizains ir bāzēts uz iOS atļauju pieprasīšanas loga, jo tas satur izstrādātāja paskaidrojumu atļaujas pieprasīšanai. Lietotājam ir trīs izvēles varianti: nepiešķirt atļauju, piešķirt vienmēr vai piešķirt tikai uz šo lietošanas sesiju.



3.1. att. **Atļaujas pieprasīšanas loga uzmetums**

Katrai lietotnei ir pievienota saīsne uz lietotnes pārvaldības sadaļu iestatījumu lietotnē. Saīsne atrodas aktīvo lietotņu pārskata skatā blakus lietotnes aizvēršanas pogai kā tas redzams 3.2. attēlā. No lietotāja viedokļa tas atvieglo lietotņu un to atļauju pārvaldīšanu, jo iOS un Android šobrīd ir nepieciešamas vismaz 5 lietotāja darbības ar laikietilpīgu lietotnes meklēšanu starp visām lietotnēm, lai nokļūtu lietotnes pārvaldības skatā. Jaunajā risinājumā lietotājs no aktīvas lietotnes varētu ar divām darbībām nokļūt lietotnes iestatījumu skatā



3.2 att. **Lietotnes iestatījumu saīsnes novietojums aktīvo lietotņu pārskata skatā**

Lietotnēm var būt obligātās atļaujas, kuras tiek piešķirtas automātiski pēc lietotāja iepazīšanās ar lietotnes pieprasītajām atļaujām un atļauju apstiprināšanas. Šī tipa atļaujas lietotājs nevar atspējot un lietotnei šīs atļaujas ir pieejama visu laiku. Lai obligātā tipa atļaujas nevarētu ļaunprātīgi izmantot, tiek ierobežots atļauju skaits, cik atļaujas lietotnei var būt obligātas. Lietotnei varēs būt tikai trīs obligātās atļaujas. Obligātās atļaujas ir domātas, lai nodrošinātu tikai pamata funkcionalitāti. Pareizu obligātā atļauju tipa izmantošanu varēs viegli pārraudzīt pirms lietotņu izplatīšanas veikalos, apskatot lietotnes pirmkodu un līdzīgas lietotnes kategorijā. Obligātās atļaujas tika pievienotas atļauju modelim, lai izstrādāji varētu informēt lietotāju, ka pamata funkcionalitātei ir nepieciešama pieeja tieši šiem resursiem un bez šīm pieejam lietotne nevarēs pat izpildīt pamata funkcijas. Ja šāda risinājuma nebūtu, lietotājs varētu atvērt lietotni, bet bez atļauju piešķiršanas, lietotne rādītos tukša un lietotājam nepildītu nekādu funkciju, tik pat labi lietotājs varētu šo lietotni dzēst.

Lietotājam ir pieejams jauns atļauju pieprasīšanas logs lietotnes uzstādīšanas laikā. Lietotnes uzstādīšana aizņem vismaz 5 sekundes, bet parasti vairāk, un šo laiku lietotājs varētu izmantot, lai pārvaldītu lietotnes pieprasītās atļaujas, papildus nezaudējot savu laiku. 3.3. attēlā var redzēt, kā tas varētu izskatīties. Lietotājam izvēloties uzstādīt lietotni, lietotnes uzstādīšanas laikā, tiek pieprasīts pārskatīt lietotnes atļaujas un tās akceptēt. Lietotājs var skatīt, kādas atļaujas lietotne

pieprasa un kuras atļaujas ir obligātas (3.3. attēlā atļauja ar melnu ieslēgtu slēdzi), kā arī piešķirt atļaujas gan pa vienai, gan visai atļauju grupai uzreiz. Grupu pārskatu varēs samazināt un pēc grupu slēdža krāsas saprast vai visas atļaujas zem šīs grupas ir piešķirtas (zaļš un ieslēgts), vai piešķirta ir tikai daļa atļauju (zils un ieslēgts) vai arī neviena atļauja grupā nav piešķirta (pelēks un izslēgts). Atļauju pieprasīšanas logs risina problēmas ar daudz atsevišķu atļauju pieprasīšanu lietotnes pirmreizējā palaišanā. Turklāt nākotnē šo logu varētu izmantot, lai rādītu rekomendētās atļaujas pēc lietotāja grupēšanas un lietotnes analīzes, līdzīgi, kā dara iepriekš minētā Personalized Privacy Assistant lietotne, bet bez root lietotāja nepieciešamības [26].

3.3. att. **Lietotnes uzstādīšanas laikā atļauju pieprasīšanas loga dizaina skice**

Izmaiņas ir arī lietotņu izstrādātājiem. Izstrādātajiem nav nepieciešams pašiem pierakstīt, kādas atļaujas ir nepieciešamas, jo katra funkcionalitāte ir pieejama caur API (līdzīgi kā iOS) un saraksts ar lietotnei nepieciešamajām atļaujām tiek automātiski ģenerēts pēc pirmkodā

izmantotajām API metodēm. Izstrādājam atliek tikai šajā sarakstā pierakstīt paskaidrojumus, kādēļ atļauja ir nepieciešama, un, ja nepieciešamas, norādīt, kuras ir obligātās atļaujas. Tas atļauju pārvaldīšanu atvieglo arī izstrādātājiem. Tomēr viens papildu sarežģījums izstrādātājiem ir, jo uzlabotajā atļauju modelī ir jādomā par atļauju piešķiršanu tikai uz laiku, kurā lietotājs lietos lietotni jeb atļauju piešķiršanu tikai uz vienu sesiju. Savukārt obligātās atļaujas nodrošina, ka lietotne var rēķināties ar dažām atļaujām, kas būs vienmēr pieejamas, protams paturot prātā, ka ne visas atļaujas var būt obligātas

3.3 Uzlabotā atļauju modeļa rezultāts un salīdzinājums ar citiem modeļiem

Veidojot uzlabotu atļauju organizēšanas modeli, svarīgi ir apskatīt, kā un kādas problēmas tam būtu jārisina. Tikpat svarīgi ir izvērtēt, vai uzlabotais modelis neveido jaunas problēmas, kuras atsver atrisināto problēmu devumu. Iepriekšējā nodaļā aprakstītais atļauju organizēšanas modelis risina visas trīs 1.3 nodaļā aprakstītās problēmas.

Problēma, ka lietotne var piekļūt pārāk daudz datiem, tiek risināta ar hierarhijas atļaujas modeļa ieviešanu, kas atbalsta sīkāku atļauju sadali un atļaujas ļauj pārvaldīt arī katru atsevišķi. Trīs vecās atļaujas no Android atļauju modeļa tika sadalītas sīkākās atļaujās, lai tās nenodrošinātu tik plašu pieeju resursiem kā tās ir tagad. Lietotājs var arī atļauju piešķirt tikai uz sesijas laiku, kas nodrošina, ka lietotne nevarēs izmantot pieeju fona režīmā pēc lietotnes aizvēršanas vai nākamajā palaišanas reizē.

Problēma, ka lietotājam var tikt liegta lietotnes izmantošana, ja viņš atsakās piešķirt kādu atļauju, tiek risināta ar obligātajām atļaujām, obligāto atļauju skaitu ierobežojumu un lietotāja informēšanu, kādas ir obligātās atļaujas ar lietotnes uzstādīšanas laikā pieejamo lietotnes atļauju pieprasīšanas logu. Palīdz arī atļauju detalizētāka pārvaldīšana un izstrādātāju atļauju pieprasīšanas paskaidrojumi.

Problēma, ka lietotājiem ir pārāk maz iespēju pārvaldīt piešķirtās atļaujas, tiek risināta ar jauno atļauju pārvaldīšanas logu, saīsni uz lietotnes iestatījumiem un katras atļaujas atsevišķu pārvaldību no lietotnes iestatījumiem.

Tiek risināta arī Android specifiska problēma ar atļauju automātisku piešķiršanu pēc lietotnes atjauninājumiem, ja atļauja ir grupā, kurai lietotājs jau ir piešķīris atļauju. Šo problēmu risina hierarhiskā tipa atļauju modelis ar iespēju katru atļauju pārvaldīt atsevišķi.

Papildus tiek domāts par lietotāja neapgrūtināšanu ar daudziem papildus atļauju pieprasījumiem, tādēļ tika ieviests atļauju pārvaldības logs uzstādīšanas laikā. Tomēr šis punkts nav tik ļoti pierādāms un to vajadzētu pārbaudīt praktiski ar reālu lietotāju iesaisti.

Visas problēmas netika atrisinātas, taču tika vismaz mazinātas. Piemēram, obligātās atļaujas pilnībā neatrisina problēmu ar lietotnes liegšanu to izmantot, ja lietotājs atsakās piešķirt kādu atļauju. Tomēr problēma ir samazināta, jo lietotājs var redzēt kādas atļaujas ir obligātas un obligātās atļaujas nevar būt vairāk par trīs, kas palielina iespēju, ka lietotne pieprasīs tikai pamatfunkcijām nepieciešamās obligātās atļaujas.

Salīdzinot uzlaboto atļauju modeli ar Android vai iOS atļauju modeļiem, tas ir pārāks ar plašākām lietotāja iespējām pārvaldīt atļaujas un labāku caurredzamību, kādus resursus vai kādas funkcionalitātes lietotne vēlās izmanto. Uzlabotajā atļauju modelī iespējams ir katras atsevišķas atļaujas piešķiršana, kas veidot plašāku iespēju pārvaldīt atļaujas salīdzinot ar Android atļauju modeli, un sīkāk sadalītās atļaujas veido labāku caurredzamību, kādus resursus un kā izmanto lietotne, ko nevarētu teikt par iOS atļaujām, jo katram resursam ir tikai viena vai divas atļaujas.

Tomēr iOS un Android atļauju modeļi mazāk apgrūtina lietotāju ar domāšanu par atļaujām, jo pieprasīto atļauju maksimuma skaits ir līdz 20 atļauju pieprasījumiem, kamēr jaunajā modelī bīstamo jeb pārvaldāmo atļauju skaits ir virs 30. Šī problēma tiek risināta ar mēģinājumu lietotājam pieprasīt apskatīt un pārvaldīt atļaujas kompakti, vienā logā, jau uzstādīšanas laikā, tomēr ne visi lietotāji izmantos šo logu, jo tas prasīs lielāku viņu laika un uzmanības ieguldījumu. Salīdzinot lietotāja ērtības starp iOS, Android un uzlaboto modeli gadījumā, kad lietotājs neapstiprina nevienu lietotnes atļauju pirms tās lietošanas un lietotnei ir visas iespējamās atļaujas, iOS un Android atļauju modeļi ir lietotājam draudzīgāki, jo tie pieprasīs līdz 20 atļauju pieprasījumiem, bet uzlabotais atļauju modelis pieprasīs vismaz 30. Sliktākā gadījuma apskates laikā iegūtais rezultāts gan nenozīmē, ka ikdienā lietotāju atļauju pieprasījumu skaits palielinātos, jo sliktākais gadījums parasti nav vidējais gadījums un šis aspekts būtu jāpārbauda reālos apstākļos ar lietotāju iesaisti. Tomēr, visdrīzākais atļauju pieprasījumu skaits uzlabotājā atļauju modelī palielinātos, jo atļaujas ir sīkāk sadalītas un lietotne pieprasīs katru atļauju atsevišķi, ja tā netiks piešķirta lietotnes uzstādīšanas laikā.

Uzlabotais atļauju modelis krietni palielina prasmīgo lietotāju (*Power users*) iespējas, jo šiem lietotājiem ir svarīga datu drošība un viņi vēlās pārvaldīt visus iestatījumus paši. Parastajiem lietotājiem, kuriem šīs iespējas nav tik aktuālas, šis modelis pagaidām varētu nešķist tik pievilcīgs ar saviem uzlabojumiem. Tomēr šis modelis ir veidots ar mērķi nākotnē pievienot atļauju

rekomendācijas, kas būtu aktuāli tieši parastajiem lietotājiem, jo tas samazinās nepieciešamo lietotāja ieguldījumu atļauju pārvaldībā.

REZULTĀTI

Darba rezultātā autors ir izpildījis visus darbā izvirzītos mērķus, veicis pētījumu par atļauju organizēšanas modeļiem iOS un Android sistēmās, to problēmām un problēmu risinājumiem, kā arī pēc pētījumā iegūtās informācijas aprakstījis, kāds varētu būt uzlabots atļauju organizēšanas modelis pēc iOS un Android atļauju paraugiem.

Autors aprakstījis, kas ir atļaujas, kas ir atļauju organizēšanas modeļi un kāpēc tie ir svarīga mūsdienu operētājsistēmu sastāvdaļa. Tika apskatīts, kādas problēmas ir iOS un Android atļauju organizēšanas modeļos un īsi tika apskatīta atļauju evolūcija Android un iOS sistēmās kopš to pirmsākumiem.

Autors detalizēti aprakstījis Android un iOS atļauju organizēšanas modeļus, kādas ir atļaujas un kā tās tiek grupētas. Viens pret otru tika salīdzināti abu sistēmu izmantotie modeļi un arī abās sistēmās lietotie veidi, kā atļaujas tiek pieprasītas un pārvaldītas. Atļauju apskatā iegūtā informācija tika izmantota, lai veidotu uzlaboto atļauju organizēšanas modeli.

Lai varētu izveidot un aprakstīt uzlabotu atļauju organizēšanas modeli, autors apkopojis ar atļaujām saistīto problēmu risinājumus, kas jau eksistē un kādus nākotnē ir domāts realizēt, kā arī kādi ir citi risinājumi, lai risinātu autora apskatītās problēmas. Pēc risinājumu apkopošanas tika veidots uzlabots atļauju organizēšanas modelis, kam pamatā tika ņemts Android modelis, tas tika uzlabots ar iOS problēmu risinājumiem, avotos atrastiem risinājumiem un autora formulētiem risinājumiem, kas šobrīd netiek izmantoti ne iOS, ne Android. Darba noslēgumā tika apskatīts uzlabotais atļauju organizēšanas modelis un salīdzināts ar jau esošajiem Android un iOS atļauju organizēšanas modeļiem.

SECINĀJUMI

Veicot apskatu par ar atļaujām saistītām problēmām iOS un Android operētājsistēmās, tika secināts, ka abu sistēmu atļauju organizēšanas modeļi ir krietni uzlabojušies laika gaitā, bet vēl nav ne tuvu perfekti un apmierinoši lielai daļai lietotāju, jo daļa no atļaujām lietotājam nav viegli saprotamas un savu būtību neizskaidro, kā arī pietrūkst ērtāka veida, kā atļaujas pārvaldīt.

Darba ietvaros tika secināts, ka, veidojot atļaujas organizēšanas modeļus, tiek staigāts pa iedomātu robežu, kur vienā pusē ir lietotāja ērtums un otrā lietotāja iespējas pārvaldīt savu datu drošību. Lai uzlabotu lietotāja personīgu datu drošību, ir nepieciešams atļaujas veidot ar skaidri saprotamu domu, kādiem resursiem vai datiem tiek dota pieeja, kas parasti nozīmē, ka veidojamo atļauju skaits pieaug. Tomēr, veidojot daudzas atļaujas, tiek pārkāpta iedomātā robeža un tiek apgrūtināta iespēja lietotājam tās ērti un ātri pārvaldīt.

Uzlabotais atļauju modelis labi palīdz lietotājam pārskatīt un pārvaldīt lietotnes pieprasītās atļaujas, kā arī uzlabo to pieprasīšanas iemeslu caurredzamību un samazina pārāk plašo atļauju skaitu. Tomēr uzlabotais modelis arī apgrūtina lietotāju ar papildus uzmanības nepieciešamību lietotņu pārvaldībai.

Nākotnē varētu pētīt, kā pēc iespējas labāk risināt lietotāju personīgo datu un resursu pieejamības problēmas fona režīmā un kā varētu lietotāju pasargāt no lielā skaita atļauju pieprasījuma, izmantojot individuālas atļauju rekomendācijas katram lietotājam.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. Akadēmiskā terminu datubāze AkadTerm [tiešaiste]. Pieejams: http://termini.lza.lv/ [atsauce – 08.05.2018.]
2. James Vincent, Google is using machine learning to sort good apps from bad on the Play Store, *TheVerge*, Jul 12, 2017 [tiešaiste]. Pieejams:https://www.theverge.com/2017/7/12/15958372/google-machine-learning-ai-app-store-malware-security [atsauce - 11.05.2018.]
3. Permissions Overview, *Android* [tiešaiste]. Pieejams: https://developer.android.com/guide/topics/permissions/overview [atsauce - 11.05.2018.]
4. Information Property List Key Reference, Cocoa Keys, *Apple* [tiešaiste]. Pieejams:https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html [atsauce - 13.05.2018.]
5. iOS Security iOS 11, January 2018, *Apple* [tiešaiste]. Pieejams: https://www.apple.com/business/docs/iOS_Security_Guide.pdf [atsauce – 10.05.2018.]
6. Human Interface Guidelines, Requesting Permission, *Apple, 2018* [tiešaiste]. Pieejams: https://developer.apple.com/ios/human-interface-guidelines/app-architecture/requesting-permission/ [atsauce – 10.05.2018.]
7. Do you pay attention to the information used by the apps you use on your smartphone?, *Statista Survey*, 21.05.2017 [tiešaiste]. Pieejams: https://www.statista.com/statistics/714165/us-attention-smartphone-app-permissions/ [atsauce – 09.05.2018.]
8. Kenneth Olmstead un Michelle Atkinson, Apps Permissions in the Google Play Store, *Pew Research Center*, 10.11.2015 [tiešaiste]. Pieejams: http://www.pewinternet.org/2015/11/10/apps-permissions-in-the-google-play-store/ [atsauce – 09.05.2018.]
9. Distribution dashboard, *Android*, 9.05.2018 [tiešaiste]. Pieejams: https://developer.android.com/about/dashboards/ [atsauce – 09.05.2018.]

10. Jeff Six, *Application Security for the Android Platform*, O'Reilly Media, 2012, ISBN: 9781449315078;

11. David Thiel, *iOS Application Security*, No Starch Press, 2016, ISBN: 9781593276010;

12. Dominic Chell, Tyrone Erasmus, Shaun Colley un Ollie Whitehouse, *The Mobile Application Hacker's Handbook,* John Wiley & Sons, 2015, ISBN: 9781118958506;

13. App Store Review Guidelines, *Apple* [tiešaiste]. Pieejams: https://developer.apple.com/app-store/review/guidelines/#software-requirements [atsauce – 15.09.2018.]

14. Chris Welch, Android P won't let apps secretly use your mic or camera in the background, *TheVerge*, 07.03.2018 [tiešaiste]. Pieejams: https://www.theverge.com/2018/3/7/17091104/android-p-prevents-apps-using-mic-camera-idle-background [atsauce - 19.05.2018.]

15. Laurent Simon, Exploring new attack vectors for the exploitation of smartphones, 07.2018, ISSN 1476-2986 [tiešaiste]. Pieejams: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-909.pdf [atsauce - 21.05.2018.]

16. Apple Developer Documentation, *Apple*, 2018 [tiešaiste]. Pieejams: https://developer.apple.com/documentation/avfoundation/avcapturesessioninterruptionreason/avcapturesessioninterruptionreasonvideodevicenotavailableinbackground?preferredLanguage=occ [atsauce - 21.05.2018.]

17. Dave Raggett, Whitepaper: Handling Trust and Permissions in Web Applications, 07.2014 [tiešaiste]. Pieejams: https://www.w3.org/2014/05/wp-trust-permissions/Overview.html [atsauce - 21.05.2018.]

18. Kathy Wain Yee Au, Yi Fan Zhou, Zhen Huang, Phillipa Gill un David Lie, Short Paper: A Look at SmartPhone Permission Models, 2011 [tiešaiste]. Pieejams: http://www.eecg.toronto.edu/~lie/papers/au-spsm2011.pdf [atsauce – 22.05.2018.]

19. Xuetao Wei, Lorenzo Gomez, Iulian Neamtiu un Michalis Faloutsos, Permission Evolution in the Android Ecosystem, 2012, ISBN: 9781450313124 [tiešaiste]. Pieejams: https://web.njit.edu/~ineamtiu/pubs/acsac12wei.pdf [atsauce – 22.05.2018.]

20. Joshua Long, The Evolution of iOS Security and Privacy Features, *Intego*, 29.02.2016 [tiešaiste]. Pieejams: https://www.intego.com/mac-security-blog/the-evolution-of-ios-security-and-privacy-features/ [atsauce – 22.05.2018.]

21. Manifest.permission, *Android*, 2009 [tiešaiste]. Pieejams: http://android.xsoftlab.net/reference/android/Manifest.permission.html [atsauce - 22.05.2018.]

22. Yury Zhauniarovich1 un Olga Gadyatskaya, Small Changes, Big Changes: An Updated View on the Android Permission System, 09.2016, ISBN: 9783319457185 [tiešaiste]. Pieejams: http://hdl.handle.net/10993/28908 [atsauce – 22.05.2018.]

23. Paul Quinn, Play Protect brings Google's app scanning to the foreground in the Play Store, *Android Police*, 17.05.2017 [tiešaiste]. Pieejams: https://www.androidpolice.com/2017/05/17/play-protect-brings-googles-app-scanning-foreground-play-store/ [atsauce – 23.05.2018.]

24. Android – Google Play Protect Overview, Android [tiešaiste]. Pieejams: https://www.android.com/play-protect/ [atsauce – 23.05.2018.]

25. Frederic Lardinois, Google says it removed 700K apps from the Play Store in 2017, up 70% from 2016, *Techcrunch*, 30.01.2018 [tiešaiste]. Pieejams: https://techcrunch.com/2018/01/30/google-says-it-removed-700k-apps-from-the-play-store-in-2017-up-70-from-2016/ [atsauce – 23.05.2018.]

26. Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun Zhang, Norman Sadeh, Alessandro Acquisti un Yuvraj Agarwal, Follow My Recommendations: A Personalized Privacy Assistant for Mobile App Permissions, 22.06.2017, ISBN 978-1-931971-31-7 [tiešaiste]. Pieejams: https://www.usenix.org/system/files/conference/soups2016/soups2016-paper-liu.pdf [atsauce – 24.05.2018.]

27. David Barrera, H. Güne¸s Kayacık, P.C. van Oorschot un Anil Somayaji, A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android, 04.10.2010, ISBN: 978-1-4503-0245-6 [tiešaiste]. Pieejams: https://homeostasis.scs.carleton.ca/~soma/pubs/barrera-ccs-10.pdf [atsauce – 25.05.2018.]

28. Lauren Goode Gear, App Permissions Don't Tell Us Nearly Enough About Our Apps, *Wired*, 14.04.2018 [tiešaiste]. Pieejams: https://www.wired.com/story/app-permissions/ [atsauce – 25.05.2018.]

29. Dan Goodin, Android no longer reveals app permission changes in automatic updates, *Arstechnica*, 11.06.2014 [tiešaiste]. Pieejams: https://arstechnica.com/information-technology/2014/06/android-no-longer-reveals-app-permission-changes-in-automatic-updates/ [atsauce – 26.05.2018.]

30. Natasha Singer, What You Don't Know About How Facebook Uses Your Data, *New York Times*, 11.04.2018 [tiešaiste]. Pieejams: https://www.nytimes.com/2018/04/11/technology/facebook-privacy-hearings.html [atsauce - 27.05.2018.]

31. Mobile Operating System Market Share Worldwide, *Statcounter*, 04.2018 [tiešaiste]. Pieejams: http://gs.statcounter.com/os-market-share/mobile/worldwide [atsauce - 27.05.2018.]

DOKUMENTĀRĀ LAPA

Bakalaura darbs „Atļauju organizēšanas modeļi lietotnēs” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____ Krišjānis Segliņš 28.05.2017.

Rekomendēju/nerekomendēju darbu aizstāvēšanai
Vadītājs: profesors, Dr. dat. Uldis Straujums _____ 28.05.2017.

Recenzents: docents, Dr. sc. administr. Imants Gorbāns

Darbs iesniegts Datorikas fakultātē 28.05.2017.

Dekāna pilnvarotā persona: metodiķe Ārija Sproģe _____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē
___.06.2018. prot. Nr. _____.
Komisijas sekretārs: