VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERIO IR DUOMENŲ MODELIAVIMO KATEDRA

Baigiamasis bakalauro darbas

Teisminė ekspertizė pagal apibrėžtus atvejus Android įrenginiuose

Atliko: 4 kurso, 3 grupės studentė
Aurėja Klevinskaitė (parašas)

Darbo vadovas:
lekt. Virgilijus Krinickij

Vilnius
2020

Turinys

Sutartinis terminų žodynas

- „**ADB**“ – „Android Debug Bridge“ – komandinės eilutės įrankis, kuris leidžia valdyti įrenginį per USB iš kompiuterio, kopijuoti failus į priekį ir atgal, įdiegti ir pašalinti programas, paleisti „Shell“ komandas ir dar daugiau. Tai yra sujungimas tarp mobiliojo įrenginio bei kompiuterio.

- **„Bash“** – „Unix“ apvalkalo komandų kalba.

- „**Boot**“ – kompiuterio paleidimas. Tai gali inicijuoti aparatūra, tokia kaip mygtuko paspaudimas arba programinė įranga.

- „**Device driver**“ – tvarkyklė – specialios paskirties programa, sukurianti tam tikrą abstrakcijos lygmenį, vykdant tam tikrus veiksmus, dažniausiai – kreipiantis į įrenginius ar duomenis.

- „**Hash**“ – maišos kodas **–** algoritmo gautas kodas, kuris yra unikalus kiekvienam failui, tai reiškia, kad kiekvienas failas turi savo unikalų maišos kodą.

- **„Loop device“** - įprastas failas arba įrenginys, kuris yra sumontuotas (angl. „mounted“) kaip failų sistema.

- „**Root**“ **teisės** - suteikiamos visos teisės prie sistemos, t.y., priėjimas prie sisteminių failų, prie kurių paprastai nebūtų galima prieiti; failai, kurie yra pasiekiami **/system/data** particijose.

- „**Shell script**“ – kompiuterio programa, skirta valdyti „Unix shell“, komandinės eilutės interpretatorių.

- **Particija** – atvaizdas (*.img failas).

- **Skriptas** - programa, sudaryta iš interpretavimui skirtų komandų. Diske skriptas laikomas pirminiu tekstu, kurį, prireikus, vykdo interpretatorius . Skriptų būna įvairių rūšių (šiame darbe naudojami: „Shell Script“, „Python“).

Santrauka

Išmanieji telefonai yra vieni iš sudėtingesnių technologijų, egzistuojančių šiandien, ir nuolat populiarėjančių dėl geresnio sujungiamumo, funkcionalumo bei produktyvumo. Vis didėjantis šių išmaniųjų telefonų technologinis sudėtingumas sukelia naują grėsmių lygį. „Android“ yra viena iš naujesnių operacinių sistemų, pagrįstų išmaniaisiais telefonais, kuri tampa reikšminga jėga labai konkurencingoje išmaniųjų telefonų rinkoje. „Android“ telefonai kaupia didžiulį kiekį duomenų, kuriuos galima saugoti lokaliai arba nuotoliniu būdu, ir leidžia teisminės ekspertizės analitikams pasiekti patikimus duomenis, renkant šią vertingą informaciją, susijusią su pačiu tyrimu. Rinkoje yra daugybė teismo ekspertizės įrankių - tiek atviruųjų, tiek uždaruųjų šaltinių, siekiant išgauti duomenis iš „Android“ įrenginių.

Šio darbo pagrindinis tikslas yra sukurti dinamišką įrankį duomenims ištraukti, analizuoti bei paruošti teisminei ekspertizei. Tam įvykdyti, pradėta nuo jau sukurtų įrankių, palaikančių „Android“ telefonus, analizės, t.y., apžvelgti skaitmeninės kriminalistikos srities įrankiai ir duomenų gavimo metodai bei aptarti kriterijai ir metodika, ko reikia, siekiant reikšmingos turimų duomenų analizės. Įrankis sukurtas, atsižvelgiant į tam tikras išsiaiškintas išvadas, pvz., kad dauguma sukurtų įrankių yra tinkami ne visiems mobiliesiems „Android“ įrenginiams. Įrankis suprogramuotas „Shell Script“ bei „Python“ kalbomis, kurias naudojant komandos buvo automatizuotos bei suskirstytos į skriptus, atliekančius skirtingas funkcijas, kas palengvina teisminės ekspertizės analizę. Įrankis veikia „Android“ įrenginių bei „Linux“ operacinės sistemos technologijų kombinacija, o patiems duomenims pasiekti naudotas „ADB Shell“. Skambučių žurnalo istorija, SMS žinutės, naršyklės istorija, nuotraukos ir kiti failai – duomenys, į kuriuos buvo labiausiai atsižvelgta, turint omenyje, kad tai skirta teisminei ekspertizei. Dėl minėtos priežasties duomenys yra suskirstyti į aplankus, kad būtų našesnis darbas, analizuojant duomenis detaliau.

Tyrimo išvada atsižvelgia į gautus rezultatus pagal gautus duomenis bei jų analizę – pagrindinis tikslas įvykdytas - sukurti dinamišką teisminės ekspertizės įrankį, kuris ištrauktų, analizuotų bei sudėliotų duomenis, galinčius palengvinti bylų sprendimus.

Summary

Forensics on Specific Cases in Android Devices

Smartphones are one of the most advanced technologies that exist today and are constantly gaining popularity for better connectivity, functionality and productivity. The increasing technological complexity of these smartphones poses a new level of threat. Android is one of the newer smartphone-based operating systems, which is becoming a significant force in the highly competitive smartphone market. Android phones store vast amounts of data that can be stored locally or remotely, and allow forensics examination analysts to access reliable data by gathering this valuable information related to the investigation itself. There are many forensics tools on the market - both open and closed source in order to extract data from “Android“ devices.

The main goal of this work is to create a dynamic tool for data extraction, analysis and preparation for forensics. To accomplish this, it was started with an analysis of tools already developed to support “Android“ phones, i.e., review tools and data mining tools in the field of digital forensics and discuss the criteria and methodology needed to make meaningful analysis of available data. The tool has been developed based on certain clarifications such as that most of the tools developed are not suitable for all “Android“ mobile devices. The tool is programmed in “Shell Script“ and “Python“ languages, which have been used to automate commands and divide them into scripts that perform different functions, in order to facilitate forensics analysis. The tool works on a combination of “Android“ devices and “Linux“ operating system technologies, and “ADB Shell“ is used to access the data itself. Calls history, SMS messages, browser history, photos and other files are the data that have been taken into account the most given, that this information is for forensics purposes. For this reason, the data is divided into folders for more productive work by analyzing the data in detail.

The conclusion of the study takes into account the results obtained based on the received data and their analysis - the main goal fulfilled - to create a dynamic forensics tool that would extract, analyze and filter the data that can facilitate the resolution of cases.

Įvadas

Darbo aktualumas. Mobiliosios platformos per pastaruosius kelerius metus patobulėjo dėl geresnio pasiekiamumo, funkcionalumo ir produktyvumo. Ne paslaptis, kad mobilieji įrenginiai yra įtraukti į kriminalinius veiksmus, pvz., paslėptų kreditinių kortelių rinkimą. Dėl šios priežasties nuolatinis tobulėjimas, deja, bet tuo pačiu kelia ir naują grėsmės lygį [10], todėl pažeidžiamumai ir kenkėjiškos programos akivaizdžiai paveikė naują mobilųjį kraštovaizdį. Atsižvelgiant į šį faktą, mobiliosios aplikacijos atlieka reikšmingą vaidmenį teisminėje analizėje, kai labai svarbu yra surinkti įrodymus prieš nusikaltėlius. Minėtas informacijos ištraukimas iš įrenginių ir yra vadinamas teisminės ekspertizės tyrimu.

Darbo problema. Tradiciniai kriminalistiniai metodai, kurie pagrįsti rankiniu tyrimu, kai tyrėjas, nenaudodamas jokių automatizuotų įrankių, tyrimą atlieka rankiniu būdu, sunkiai arba nėra pritaikomi prie daugelio mobiliųjų programų, beje, užtrunka pakankamai daug laiko, nes didžioji dalis procesų yra pasikartojantys ir tikrai galėtų būti automatizuoti, tačiau yra ir kiti darbai, pvz., konkrečių įkalčių rinkimas, jų įrodymas bei nusikaltimo tinklo kūrimas, kurie kiekvieną kartą yra unikalūs procesai ir sunkiai galėtų būti automatizuoti, nebent reikėtų pritaikyti semantinius tinklus [40] bei jų algoritmus. Kita vertus, dinamišką analizę sunku automatizuoti dar ir dėl naštos nustatyti tinkamą darbo aplinką, kad būtų galima pritaikyti skirtingus modelius bei operacines sistemas, priklausomas bibliotekas, kad būtų galima aktyvuoti visus galimus programos variantus, o tai yra iššūkis, nes skirtingi įrenginiai dažniausiai skiriasi savo architektūra. Panašu, kad tai tik dar vienas atvejis technologijose, kai sunku rasti geriausią bendrą sprendimą, tinkantį skirtingoms iškeltoms problemoms. Siekiant reaguoti į šį naują grėsmių rinkinį, svarbu, kad dabartinio saugumo technikos ir priemonės prisitaikytų prie naujos dabartinės situacijos [24]. Šių laikų technikos, įrankiai ir procesai, skirti tinklų ir sistemų kriminalistinei analizei atlikti, liečia ir pačias, jau minėtas, mobiliąsias platformas [29].

„Android“ yra viena iš naujesnių operacinių sistemų, paremtų išmaniaisiais telefonais, dėl to jų teisminė ekspertizė atliekama dažniausiai. Jie saugo didžiulį kiekį duomenų, kurie gali būti saugomi vietoje arba nuotoliniu būdu. Vertingą informaciją bei įrodymus leidžia įsigyti teisminės ekspertizės analizei. Yra pakankamai daug įrankių (atvirų ir uždarų šaltinių), siekiant išgauti duomenis iš „Android“ įrenginių [17].

Darbo objektas. Sukūrus įrankį, kuris automatizuotų pirmąją teisminės ekspertizės tyrimo dalį, t.y., įrenginio atvaizdo išgavimą, montavimą, reikiamos informacijos ištraukimą bei jos filtravimą, tyrėjo darbas taptų kokybiškesnis bei taupesnis laiko atžvilgiu. Įrankio veikimo principas teoriškai yra paprastas – kompiuteryje turi būti įdiegtas įrankis, kuris, prijungus įrenginį, turi ištraukti informaciją, esančią viduje, bei išfiltruoti taip, kad sutalpintų bet kokius rastus įkalčius, juos suformatuoti ir sutvarkytą informaciją pateikti tolimesniems tyrimams. Svarbu paminėti, kad įrankio veikimo metu duomenų formatavimas turėtų būti vientisos tvarkos, nes bet koks duomenų vientisumo pažeidimas tyrimo metu sugadina rezultatų validumą.

Darbo tikslas. Darbo tikslas yra sukurti įrankį, kuris automatizacijos būdu ištirtų „Android“ įrenginius arba bent palengvintų šį tyrimo procesą. Iškeltam tikslui pasiekti, pirmiausia bus analizuojami susiję darbai, tada vizualizuosime modelį ir galiausiai bus prieita prie pagrindinės darbo dalies – implementacijos (duomenų ištraukimas, filtravimas, formatavimas) bei testavimo. Šis darbas yra apie tai, kaip galima atlikti teisminę analizę „Android“ įrenginiuose.

Darbo uždaviniai:

- Padaryti susijusių darbų analizę, apžvelgti metodus.
- Vizualizuoti įrankio veikimo modelį.
- Sukurti įrankį, kuris galėtų analizuoti įkalčių ekspertizę.
- Testuoti įrankį.
- Aprašyti ateities tyrimų planą.

1. Susijusių darbų analizė

Šiais laikais visose technologijose yra nuveikta tikrai daug, todėl kompiuterių bei mobiliųjų įrenginių teisminės ekspertizės srityje taip pat yra patobulėta. Šioje srityje yra sukurti programinės įrangos įrankiai, kurie, kaip minėta įvade, palengvina tyrėjų darbą, automatizuodami dalį procesų. Iš karto svarbu paminėti, kad su „Android“ ir apskritai su mobiliaisiais įrenginiais situacija yra sudėtingesnė, nes unikalių ir veikiančių visuose įrenginiuose teisminės ekspertizės įrankių tikrai būtų sunku rasti arba jų kol kas tiesiog nėra. Taip yra dėl įvade paminėtų priežasčių – skirtingų „Android“ mobiliųjų įrenginių modelių, jų operacinių sistemų, ir dėl skirtingų architektūrų. Šios priežastys lemia daugelio įrenginių atsiradimą, kurie skiriasi savo turimomis operacinių sistemų versijomis, modeliais (tarp kai kurių modelių ir pačios operacinės sistemos skiriasi radikaliai, o kai kur kaip tik minimaliai arba visiškai nesiskiria), todėl automatizuoto modelio dar niekam nepavyko sukurti (yra pilna automatizuotų įrankių, tačiau visada atsiranda kokia nors spraga, nes, jeigu toks įrankis būtų sukurtas, tai jis turėtų implementuoti visų šiuo metu esamų „Android“ OS versijų bei visų esamų įrenginių teisminės ekspertizės veikiančius metodus), kiek leidžia žinoti turimi šaltiniai [38] [16].

Nors kompiuterių architektūra yra unikali, dauguma operacinių sistemų yra tokios pačios (gali ir skirtis), dėl to, sukurti automatizuotus įrankius, paminėti aspektai nesutrukdė.

Kompiuterinėje teisminėje ekspertizėje sukurti ne tik automatizuoti įrankiai, bet ir yra apibrėžta teorinė metodika, kurios principais sekant ir yra atliekami tyrimai. Kadangi darbas orientuotas į „Android“ įrenginius, šaltiniuose rašoma, kad jų teisminė ekspertizė konkrečių apibrėžtų tyrimo metodikų neturi, nors bandymų jas sukurti yra [34].

1.1. Metodai

Šio tyrimo tikslais teismo ekspertizės analizė sutelkta į keturis pagrindinius sluoksnius, kuriuos svarbu suprasti: įrenginių žurnalus, tinklo srautą, failų sistemą ir atmintį. Kiekviename etape surinkti įrodymai gali būti koreliuojami su informacija, gauta kituose žingsniuose, kad būtų visiškai aiškus vaizdas [6].

1.1.1. Įrenginių žurnalai

„Android“ teikia visą žurnalo analizės įrankių rinkinį, aktualiausias yra „Logcat“. Naudojant šį įrankį galima stebėti įvairių rūšių žurnalus, pvz., radijo sukurtus įvykius. Ši informacija gali būti labai naudinga aptikti gaunamus ir siunčiamus SMS/MMS pranešimus. Pavyzdžiui, galima pastebėti kenkėjiškas programas, kurios fone siunčia SMS ir ištrina pranešimus.

1.1.2. Tinklo srautas

Tinklo srauto analizė yra labai svarbi kai kuriuose teismo ekspertizės tyrimuose, todėl yra svarbu, kad būtų galima tinkamai stebėti srautą. Įmonėse gali būti įrankiai, pvz., įgaliojimai (angl. „proxies“) arba ugniasienės (angl. „firewalls“), kuriose būtų stebimas srautas. Kitas metodas, kurio esmė yra nuotoliniu būdu stebėti tinklo srautą, yra aprašytas autoriaus Alonso Parrizas: „Monitoring network Traffic for Android Devices“ [7].

1.1.3. Failų sistema

Mobiliuose įrenginiuose failų sistemoje saugomi duomenys gali būti surinkti keliais būdais, kaip yra aprašyta autorių S. Bommisetty, R. Tamma ir H. Mahalik knygoje: „Practical Mobile Forensics“: rankinis ištraukimas, loginis ištraukimas, fizinis ištraukimas (angl. „Hex dump“), lusto išjungimas ir skaitymas [31].

1.1.4. Atmintis

Analizuojant atmintį sistemoje, yra daug naudingų įrodymų ir informacijos, pvz., kriptografiniai raktai arba nešifruoti duomenys.

Pirmas žingsnis visada turi būti tikslios sistemos atminties nukreipimas. Tai gali būti padaryta su kai kuriomis esamomis priemonėmis, pvz., „LiME“, „Linux Memory Extractor“, kurie gali būti sukompiliuoti „Android“.

Be to, egzistuoja atminties analizės įrankiai, tokie kaip „Volatilty“, kuris leidžia analizuoti atminties sąvartynus. Šiam tikslui reikia sukurti profilį, pritaikytą tam tikram „Android“ įrenginiui.

Verta paminėti, kad yra ir kitų būdų analizuoti atmintį „Android“, pavyzdžiui, naudojant monitoriaus įrankį (angl. „monitor tool“), kurį teikia „Android“. Tačiau šis metodas leidžia tik stebėti vieną procesą kiekvieną kartą, o ne analizuoti visą sistemos atmintį.

1.2. „Android“ įrenginių teisminės ekspertizės įrankiai

Informacijos ištraukimo įrankiai gali būti aparatinės arba programinės įrangos, priklausomai nuo to, kaip duomenys yra išgauti iš mobiliojo įrenginio. Šiandien rinkoje yra nemažai ištraukimo įrankių, o naujesnės priemonės atsiranda su kai kuriomis novatoriškomis priemonėmis. Nustatyta, kad beveik visi įrankiai yra komerciniai ir keletas yra atviro kodo įrankių. Deja, bet šių priemonių įsigijimas labai sunkus, nes tai yra susiję su privatumu ir saugumu, taip pat įtaką daro išlaidos. Fizinis informacijos ištraukimas šiame darbe nebus aptariamas, todėl pagrindinius įrankius, tokius kaip „Cellebrite UFED“ ir „XRY“ praleisime. Kitus pirmaujančius įrankius, tokius kaip „Access Data Mobile Phone Examiner Plus (MPE+)“ ir „viaForensics ViaExtract“, taip pat sunku įsigyti.

1.2.1. „Andriller“

„Andriller“ - tai programinė įranga, turinti išmaniųjų telefonų teisminės ekspertizės priemonių rinkinį. Jis atlieka tokias funkcijas - turi „Lockscreen“ krekingo (angl. „cracking“) modelį, kuris išgauna PIN kodą arba slaptažodį; turi pritaikytą dekoderį, skirtą „Android“ (kai kurie pritaikyti „Apple iOS“ ir „Windows“) duomenų bazėms, skirtoms ryšių dekodavimui. Ištraukta informacija pateikiama ataskaitose „HTML“ ir „Excel“ formatuose.

Kalbant apie pagrindinę sąranką, „Andriller“ yra „Microsoft Windows“ ir „Ubuntu Linux“ tarpvalstybinė programa. „Windows“ diegimo įrenginys reikalauja įdiegti tik „Microsoft Visual C ++ 2010“ perskirstomąjį paketą (x86), „Android“ įrenginio USB tvarkykles ir interneto naršyklę, skirtą rezultatams peržiūrėti. „Ubuntu“ versijai reikia įdiegti „android-tools-adb“ paketą.

Šį įrankį sudaro toks funkcionalumas:

- automatizuotas duomenų ištraukimas ir jų dešifravimas;
- duomenų ištraukimas iš „Android“ telefonų be „root“ teisių (tinka tik „Android 4.x“ operacines sistemas turintiems įrenginiams);
- duomenų ištraukimas su „root“ teisėmis (per „ADB daemon“, „CWM“ atstatymo režimą arba „SU binary“);
- duomenų analizavimas ir dešifravimas pagal aplanko (angl. „folder“) struktūrą;
- individualių duomenų bazių dekoderių pasirinkimas „Android“ ir „Apple“ įrenginiams;
- „WhatsApp“ duomenų bazių dešifravimas;
- ekrano užrakinimo šablono, PIN kodo, slaptažodžio nulaužimas;
- „Android“ atsarginių duomenų ištraukimas;
- įrenginio ekrano kopijų atlikimas.

Funkcijos tikrai nėra pritaikytos visiems „Android“ įrenginiams ir visoms jų operacinių sistemų versijoms.

„Andriller“ siūlo dar vieną funkciją – duomenų bazės dekoderius. Ši funkcija leidžia importuoti atskirus programėlių duomenų bazės failus automatizuotam duomenų analizavimui. Sėkmingai

dekodavus, ataskaitos rodomos naršyklėje. Duomenų bazės gali būti eksportuojamos iš pagrindinių teismo ekspertizės priemonių, tokių kaip „XRY“, „UFED Cellebrite“, „Oxygen Forensic“ (apie visus juos užsiminta anksčiau), ir importuojamos į „Andriller“ individualiam dekodavimui.

Norint ištraukti duomenis iš „Android“, reikia prijungti „Android“ įrenginį USB kabeliu, įgalinti USB „Debugging“ (reikia įsitikinti, kad įdiegosi įrenginio tvarkyklės (angl. „drivers“)).

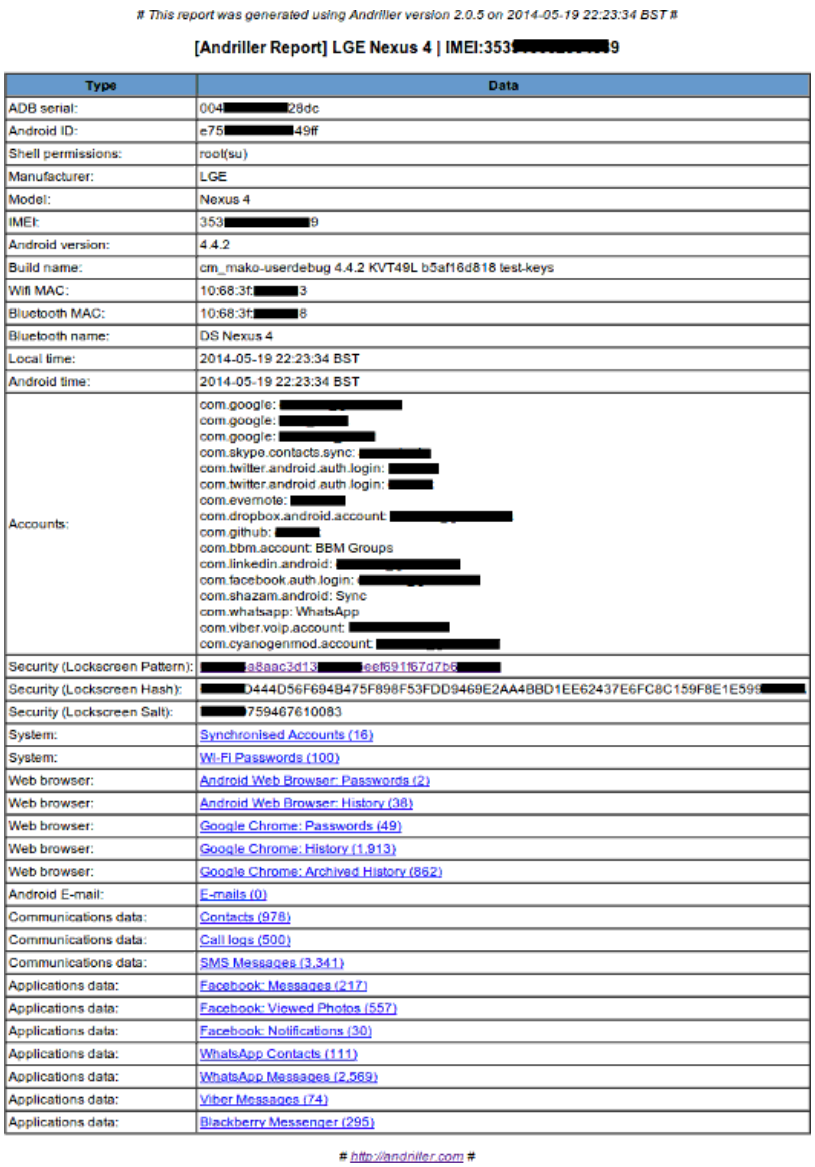
Pirmiausia reikia pasirinkti [Output] katalogą, kuriame norime išsaugoti ištrauktus duomenis. Antra, reikia spausti [Check], kad pamatytumėme, ar „Andriller“ aptiko prijungtą įrenginį. Norint pradėti duomenų ištraukimą, reikia paspausti [Go!] mygtuką. „Andriller“ turėtų pradėti veikti-atsisiųsti bet kokius duomenis ir iš karto iššifruoti.

„Andriller“ turi labai paprastą ir aiškią grafinę vartotojo sąsają. Įrankis akcentuoja aplankalo struktūrą (išanalizuoja pagal „Android“ failų sistemas ir paruošia „Andriller“ stiliaus ataskaitą. Ji gali būti eksportuota arba pagal failo sistemos neapdoroto vaizdo failą, arba iš „adb pull/data“, arba neišplėsto *.tar failo), „Tarball“ failus (išanalizuoja ir dekoduoja atsargines kopijas, pvz., „data.tar“ ir taip pat paruošia šio įrankio stiliaus ataskaitą) ir „Android“ atsarginius failus (analizuoja ir dekoduoja „backup.ab“ failus ir vėl sukuria ataskaitą)).

Svarbiausia dalis yra duomenų ataskaita, nes jos, teisminės ekspertizės atveju, būtų pateikiamos kaip analizės rezultatai, įrodymai ir pan., todėl svarbi įrankio funkcija – automatinis „HTML“ bei *.xlsx failų generavimas. Šie failai (duotas pavyzdys yra „REPORT.html“ ar „REPORT.xlsx“) yra naudojami kaip ataskaitos, skirtos rezultatams pateikti. Šio įrankio ataskaitos privalumas yra tas, kad visa gauta informacija bei rezultatai yra parodomi tvarkingai bei sistemingai (1 pav.).

1 pav. matome, kad yra daug informacijos, pvz., yra parašyta apie patį įrenginio modelį bei operacinės sistemos versiją, „IMEI“ (angl. „International Mobile Equipment Identity“), „Android“ ID, „Wifi“ ir „Bluetooth“ MAC adresus. Plačiausiame bloke matomi visų programėlių paskyros, kurios buvo sukurtos telefone, pvz., „Skype“, „LinkedIn“, „Facebook“ ir t.t. Taip pat yra nulaužtos ekrano atrakinimo sistemos (angl. „Lockscreen Pattern“), slaptažodžiai, prisijungimai, naršyklės istorija, SMS ir skambučių žurnalai. Paskutinėje ataskaitos dalyje yra daug nuorodų į kitas ataskaitas, pvz., „Call logs“ (liet. skambučių žurnalas).

„Andriller“ - kokybiškas įrankis dėl to, kad yra aiški ir plataus funkcionalumo vartotojo grafinė sąsaja. Beje, rašoma, kad „Andriller“ kol kas yra vienintelis kokybiškas įrankis šių dienų rinkoje teisminei ekspertizei atlikti. Visgi šis įrankis nėra unikalus dėl to, kad netinka visiems „Android“ įrenginiams bei jų OS versijoms, nors yra aktyviai vykdomi pakeitimai (jų puslapyje yra atnaujinimų žurnalas, pvz., paskutinis atnaujinimas matomas 2019 m. gegužės mėn. viduryje (3.1.0 versija) [4]).



 # This report was generated using Andriller version 2.0.5 on 2014-05-19 22:23:34 BST #

[Andriller Report] LGE Nexus 4 | IMEI:353[redacted]9

Type	Data
ADB serial:	004[redacted]28dc
Android ID:	e75[redacted]49ff
Shell permissions:	root(su)
Manufacturer:	LGE
Model:	Nexus 4
IMEI:	353[redacted]9
Android version:	4.4.2
Build name:	cm_mako-userdebug 4.4.2 KVT49L b5af16d818 test-keys
Wifi MAC:	10:68:3f:[redacted]3
Bluetooth MAC:	10:68:3f:[redacted]8
Bluetooth name:	DS Nexus 4
Local time:	2014-05-19 22:23:34 BST
Android time:	2014-05-19 22:23:34 BST
Accounts:	com.google: [redacted] com.google: [redacted] com.google: [redacted] com.skype.contacts.sync: [redacted] com.twitter.android.auth.login: [redacted] com.twitter.android.auth.login: [redacted] com.evernote: [redacted] com.dropbox.android.account: [redacted] com.github: [redacted] com.bbm.account: BBM Groups com.linkedin.android: [redacted] com.facebook.auth.login: [redacted] com.shazam.android: Sync com.whatsapp: WhatsApp com.viber.voip.account: [redacted] com.cyanogenmod.account: [redacted]
Security (Lockscreen Pattern):	[redacted]a8aac3d13[redacted]eef691f67d7b8[redacted]
Security (Lockscreen Hash):	[redacted]D444D56F694B475F898F53FDD9469E2AA4BBD1EE62437E6FC8C159F8E1E59[redacted]
Security (Lockscreen Salt):	[redacted]759467610083
System:	Synchronised Accounts (16)
System:	Wi-Fi Passwords (100)
Web browser:	Android Web Browser: Passwords (2)
Web browser:	Android Web Browser: History (38)
Web browser:	Google Chrome: Passwords (49)
Web browser:	Google Chrome: History (1,913)
Web browser:	Google Chrome: Archived History (862)
Android E-mail:	E-mails (0)
Communications data:	Contacts (978)
Communications data:	Call logs (500)
Communications data:	SMS Messages (3,341)
Applications data:	Facebook: Messages (217)
Applications data:	Facebook: Viewed Photos (557)
Applications data:	Facebook: Notifications (30)
Applications data:	WhatsApp Contacts (111)
Applications data:	WhatsApp Messages (2,569)
Applications data:	Viber Messages (74)
Applications data:	Blackberry Messenger (295)

 # http://andriller.com #

1 pav. „Andriller“ stiliaus ataskaita

Nors įrankis periodiškai yra naujinamas, „Android“ įrenginiai naujinami ne tik dažniau, bet ir greičiau, tuo labiau, kad leidžiamos ne tik naujos įrenginių versijos, bet ir sukuriami patys nauji įrenginiai. Išvada - nors „Andriller“ įrenginys yra gausus funkcionalumo, įrankis bet kokiu atveju nėra universalus.

1.2.2. „ViaForensics AFlogical OSE“

„Aflogical“ yra atviro kodo įrankis duomenims ištraukti. Programa naudojama komandinėje eilutėje. Įrankis suteikia funkciją „Androidadb“ - galimybę komunikuoti su kompiuteriu. Šaltiniuose [15], [14] pavyzdžiai yra atlikti naudojant „Santoku Linux“ operacinę sistemą, todėl paminėti pavyzdžiai bus ir išnagrinėti.

Paleidimas – reikia „Santoku Linux“ aplinkoje paleisti „Android SDK manager“, kuriame reikia pasirinkti įrankius, kuriuos žadama naudoti (svarbu atkreipti dėmesį į versijas) [26]. Įdiegus, galima kurti „Android“ virtualų diską (angl. „AVD“), kur pasirenkamas tas įrenginys, kurio reikia teisminei ekspertizei, taip pat galima parinkti tokius nustatymus kaip „CPU“, apvalkalą (angl. „skin“), „SD“ kortelės dydį ir pan. Parinkus visus nustatymus, matome „Android“ telefono aplinką kompiuterio ekrane (aplinka yra draugiška vartotojo sąsajai (2 pav.)), kurioje galima sukurti įrašus, pvz., galima įvesti kontaktą, paskambinti, parašyti SMS žinutę ir pan.

2 pav. „AFLogical“ aplinka

Sukūrus kelis įrašus, paleidžiamas įrankis „AF Logical OSE“. Komandinės eilutės pagalba prijungiame įrenginį ir matome tokius pasirinkimus kaip skambučių žurnalas, kontaktai, MMS, SMS ir pan. Iš minėto sąrašo reikia pasirinkti duomenis, kuriuos norime ištraukti, paspausti mygtuką: „Capture“ (liet. gauti), komandinėje eilutėje parašyti: „Enter“ ir jau yra sukurti atskiri aplankalai su skambučių žurnalu, kontaktais ir kitais duomenimis, o jie išvesti formate, kuris yra panašus į „Excel“.

Pagrindinis šio įrankio privalumas – draugiška vartotojo sąsaja, kas tikrai yra labai svarbu ir kelia konkurenciją kitiems įrankiams.

1.2.3. „Oxygen Forensics“

„Oxygen Forensic“ yra taip pat vienas iš pirmaujančių įrankių teismo ekspertizės srityje, teikiantis pagalbą telefonu, kaip rašo V. Vijayan darbe: „Android Forensic Capability and Evaluation of Extraction Tools“ [37]. Įrankis ištraukia didžiąją dalį informacijos, turi aiškiai apibrėžtą ataskaitų teikimo sistemą, kad egzaminuotojas galėtų perskaityti ir patikrinti išsamią surinktų įrodymų informaciją, o tam yra galimybė pasirinkti tokius formatus kaip „PDF“, „XLS“, „RTF“, „XML“ ir „HTML“. Priežastis, kodėl šis įrankis yra paminėtas prie „Android“ įrenginių – „Android Jet Imager“ funkcija, kuri sukuria pilnus sandėlius (angl. „dumps“) būtent iš „Android“ įrenginių 25% greičiau negu reguliarūs metodai.

„Oxygen Forensic“ siūlo šias funkcijas [27]:

- kontaktų informacijos nagrinėjimą – parodomi vardai, pavardės, el. pašto adresai ir daug kitos informacijos iš įvairių įrenginio šaltinių;
- programas – ištraukti, iššifruoti ir išanalizuoti vartotojo duomenis iš populiariausių programėlių;
- atsarginių kopijų kūrimą ir atvaizdų importą – importuoti ir išnagrinėti įvairias atsargines kopijas ir atvaizdus iš šių dienų „iOS“, „Android“ taip pat kaip ir iš kitų teisminės ekspertizės įrankių, pvz., „Cellebrite“ ir „MSAB“;
- „CDR“ (angl. „Call Detailed Record“) analizę – vykdyti ir analizuoti „CDR“ įrašus, gautus iš belaidžio ryšio tiekėjų. Galima vizualizuoti koordinates žemėlapyje ir identifikuoti nuorodas tarp skambinančių asmenų;
- Duomenų pasiekiamumą debesyje – gauti prieigą prie debesies paslaugų, pvz., „WhatsApp“, „Telegram“, „iCloud“, „Google“, „Samsung“, „Micosoft“, „Facebook“, „Instagram“, „Twitter“ ir dar daug kitų socialinių medijų, turinčių debesies paslaugas;
- kompiuterio įgaliojimų rinkimą – rinkti įgaliojimus ir slaptažodžius kompiuteryje, naudojant „Oxygen Forensic“ aprašus;
- duomenų aprašymą – atnaujinti tiesioginio prenumeratoriaus informaciją, susijusią su telefono numeriu, įtraukiant adresą, telefono transporterį (angl. „phone carrier“) ir kitus prieinamus duomenis;
- duomenų paiešką – pasaulinė paieška viename renginyje, daug įrenginių arba visiems atvejams.

Aukščiau aprašytos ne visos įrankio galimybės, nes „Oxygen Forensic“ taip pat turi tokias funkcijas kaip ištrintų duomenų atstatymas, informacijos apie įrenginį peržiūra, greita užšifruotų atsarginių kopijų peržiūra, neįvykusių skambučių ar kitų įvykių peržiūra, veido atpažinimas, failų paieška, koordinačių ištraukimas iš įvairių šaltinių, pagrindinių įrodymų pasižymėjimas, „IoT“ (angl. „Internet of Things“) įrenginių palaikymas, raktinių žodžių paieška, nuorodų analizė, tiesioginis (angl. „live“) duomenų ištraukimas, vietų vizualizacija, žinutės, kalendoriaus įrašų peržiūra, slaptažodžių nulaužimas, visa informacija iš telefono žurnalo, „PLIST“ (angl. „Property List XML File Troubleshoot“) ir „SQLite“ peržiūros, užrakto panaikinimas, laiko juosta, internetiniai prisijungimai ir t.t. Visos įrankio galimybės pakankamos atlikti kokybišką teisminę ekspertizę. Vienas iš trūkumų - įrankis tinka ne visiems mobiliesiems įrenginiams, taip pat reikia užfiksuoti faktą, kad tai yra kompiuterinis įrankis, dėl ko kyla didesnė statistinė rizika, kad egzaminuojamame telefone gali būti virusas ar kenkėjiška programa. Beje, „Oxygen Forensic“ naudoja „brute force“ techniką, kuri užima daug laiko, kad atliktų tam tikrus procesus.

Šio įrenginio aprašymas susiejamas su autorių M. Al-Hadadi, A. AlShidhani darbu: „Smartphone Forensics Analysis: A Case Study“ [19], kuriame buvo išbandyti du teisminės ekspertizės įrankiai „Oxygen Forensic“ ir „UFED Physical Analyzer Cellebrite Finding“ dėl to, kad teisminės ekspertizės ekspertai rekomenduoja naudoti daugiau negu vieną įrankį, nes vienas įrankis niekada nebus pakankamai įtikinamas.

Minėto darbo autorių išvada, palyginus įrankius - nors „Oxygen“ turi ribotą prisijungimą prie įrenginių, suteikia daugiau informacijos, pvz., iš „WhatsApp“ negu įrankis „UFED“. Duomenų radimo palyginimas rodo, kad kiekvienas įrankis išskiria specifinę informaciją, naudingą kriminalistinei analizei. Tokia informacija kaip atvaizdai arba kontaktai geriau ištraukiami su „UFED“ įrankiu, bet „Oxygen“ kokybiškiau ištraukia duomenis iš „WhatsApp“ programėlės.

1.2.4. „UFED Physical Analyzer Cellebrite Finding“

Įrankio funkcijos yra panašios kaip ir kitų aprašytų įrankių – įrenginio atrakinimas, duomenų ištraukimas, šifruotų duomenų konvertavimas į veiksmingą intelektą (jie siūlo šias dešifravimo galimybes: „SQLite Wizard“, „Virtual Analyzer“, „Python Scripting“ ir „Hex Highlighting“), duomenų suvienodinimas, kad galėtume gauti išsamesnę apžvalgą, turinio analizavimas daugiau negu 70 kalbų (ši funkcija nebuvo siūlyta anksčiau aprašytuose įrankiuose) ir ataskaitos. Trūkumas - nėra tinkamas visiems įrenginiams.

1.2.5. „MOBILedit Forensic“

Šis įrankis taip pat yra aprašytas V. Vijayan [37]. „MOBILedit“ leidžia egzaminuotojams nagrinėti logiškai išdėstytą informaciją. Įrankis naudoja daugialypį sujungimo mechanizmą, todėl programinės įrangos pakanka, kad gautume telefono sistemos informaciją, kontaktus ir žinučių sąrašus. Atstatomi ištrinti duomenys, vykdoma naujų įrašų tiesioginė analizė, kitų įrankių integracija - funkcijos kartojasi. Puslapyje [22] yra pateiktas vaizdo įrašas, kuriame rodoma, kaip naudotis jų įrankiu, kuris yra draugiškas vartotojui, o funkcijų pavadinimai aiškiai nusako, ką kiekviena iš jų daro.

1.2.6. „Android“ įrenginių teisminės ekspertizės įrankių apibendrinimas

Apibendrinant „Android“ teisminės ekspertizės įrankius, svarbu yra įžvelgti pagrindinius skirtumus, kurie surašyti lentelėje (1 lentelė). Yra šeši išskirti aspektai, pagal kuriuos lyginami anksčiau aprašyti įrankiai.

	Tinka visiems įrenginiams	Ataskaita	Galima pasiekti duomenis iš programėlių?	Atstato duomenis?	Ar galima gyvai stebėti atnaujinimus?	Nemokamas
„Andriller“	-	+	+	+	-	-
„ViaForensic AFlogical OSE“	-	+	-	-	-	+
„Oxygen Forensic“	-	+	+	-	-	-
„UFED Physical Analyzer Cellebrite Finding“	-	+	-	-	-	+
„MOBILedit Forensic“	-	+	-	+	+	+

1 lentelė. „Android“ įrenginių teisminės ekspertizės įrankių palyginimas

1 lentelėje matoma, kad nėra nei vieno įrankio, kuris būtų universalus, dinamiškas. Priešingai negu pirmąjį kriterijų, antrąjį - atitinka visi aprašyti įrankiai, nes ataskaita yra pagrindinis dalykas teisminėje ekspertizėje, kuris vaizduoja reikalingus duomenis rezultate. Kiti aspektai rodo, kad įrankiai skiriasi ir turi skirtingas funkcijas, pavyzdžiui, jeigu „Andriller“ turi galimybę pasiekti duomenis iš programėlių, tai trūkumas yra tas, kad negalima gyvai stebėti atnaujinimų, priešingai negu yra su „MOBILedit Forensic“.

Išvada - nėra universalaus „Android“ įrenginių teisminės ekspertizės įrankio, kuris tiktų visiems įrenginių modeliams bei atitiktų visus aspektus, nepaisant to, kad apžvelgti tik keli iš jų.

1.3. Kompiuterių teisminės ekspertizės įrankiai

Šiame darbe aktualiausi yra „Android“ įrenginių teisminės ekspertizės įrankiai, tačiau ne mažiau svarbu yra nagrinėti ir kompiuterių teisminės ekspertizės įrankius.

1.3.1. „Autopsy“

„Autopsy“ yra atviro kodo programinė įranga, skirta teisėsaugai, kariuomenei tirti, kas įvyko kompiuteryje. Įrankis skirtas kompiuterinei teisminei ekspertizei, su kuriuo galima ištraukti nuotraukas iš kameros atminties kortelės.

Įrankis yra lengvai naudojamas - programa, paremta „GUI“, kuri veikia ant „Windows“, „Linux“ bei „MAC“ operacinių sistemų. 2016 ir 2017 metais buvo išrinktas geriausiu atviro kodo kompiuterinės teisminės ekspertizės įrankiu. Jis skirtas kietojo vidinio disko (HDD ar SSD) tyrimui bei nagrinėjimui.

Pagrindinis „Autopsy“ skirtumas nuo kitų kompiuterinių teisminės analizės įrankių yra tas, kad jame galima implementuoti trečiųjų šalių (angl. „Third – parties“) modulius. Moduliai suteikia šias funkcijas:

- laiko juostos analizę (pažangi grafinė įvykių peržiūros sąsaja);
- maišų (angl. „Hash“) filtravimą (blogų ir gerų failų filtravimas pagal maišos funkcijų rezultatus);
- paiešką pagal raktinius žodžius (raktažodžiai padeda surasti failus pagal tam tikrus terminus);

- failų atgavimą (atstato ištrintus failus, naudojant „PhotoRec“);
- multimediją (ištraukia „EXIF“ (angl. „Exchangeable Image File Format“ iš nuotraukų ir vaizdo įrašų));
- naršyklių analizę (istorija, prisijungimai, sausainiai (angl. „Cookies“)).

Šiais laikais duomenis norima gauti kuo greičiau, todėl svarbu pastebėti, kad šis įrankis vykdo užslėptas užduotis lygiagrečiai naudodamas kelis branduolius, kad pateiktų rezultatus kuo greičiau. Įrankis yra nemokamas.

Šalia „Autopsy“ yra naudojamas „The Sleuth Kit“ įrankis (komandinių eilučių kolekcija ir C biblioteka, kuri leidžia analizuoti disko vaizdus ir atgauti duomenis iš jų) [9] [3]. Yra ir dar viena svarbi funkcija – galimybė sudaryti duomenų populiacijos laiko juostą (angl. „Populating Timeline Data“). Yra aiškiai matoma duomenų laiko juosta, o naudotojas gali pasirinkti net tris duomenų atvaizdavimo būdus – skaičiavimo (angl. „Counts“), detalių (angl. „Details“) ir sąrašo (angl. „List“). Teisminės ekspertizės metu tam tikrais atvejais vaizdinė duomenų išklotinė svarbi funkcija.

1.3.2. „CAINE“

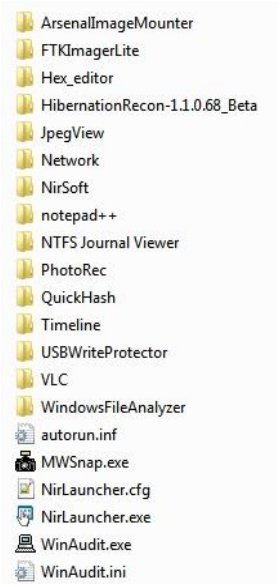
„CAINE“ (angl. „Computer Aided INvestigative Environment“) - įrankių visuma ir/ar operacinė sistema, skirta kompiuterių teisminės ekspertizės atlikimui, todėl tai yra pagrindinis skirtumas nuo kitų įrankių. Italų sukurta operacinė sistema taip pat yra atviro kodo, „GNU/Linux“ tiesioginio (angl. „live“) tipo distribucija. Ji yra pasiekiama tiesiai iš USB įrenginio, nerašant jos į kietąjį diską, dėl to yra privalumas – mobilumas (gali būti prijungtas bet kuriame kompiuteryje) ir kietojo disko nemodifikavimas.

Pagrindiniai projektavimo tikslai, kuriuos siekta užtikrinti, yra šie:

- operacinė sistema, talpinanti dešimtis įrankių, skirtų kompiuterinės teisminės analizės atlikimui;
- paprasta grafinė vartotojo sąsaja (angl. „user-friendly“), tuo siekta sukurti lengvai perprantamą naudojimą.

Kaip ir daugelis išnagrinėtų įrankių, taip ir ši įrankių sistema pati sugeneruoja ataskaitas teisminės ekspertizės tyrimo rezultatams analizuoti. Pačius „CAINE“ OS teisminės ekspertizės įrankius („CAINE“ turi „Windows IR/Live“ teisminės ekspertizės įrankius (3 pav.)) skirsto į kelias pagrindines kategorijas:

- kietojo disko analizę;
- duomenų bazės analizę;
- maišos funkcijas, kenksmingos programinės įrangos analizę;
- RAM atminties, mobiliųjų įrenginių bei tinklų analizę.



3 pav. „Windows“ įrankiai

Įrankis paskutinį kartą buvo atnaujintas 2018 m. gruodžio mėn., todėl įrankis nėra periodiškai naujinamas. Atnaujinimus galima stebėti jų puslapyje [2].

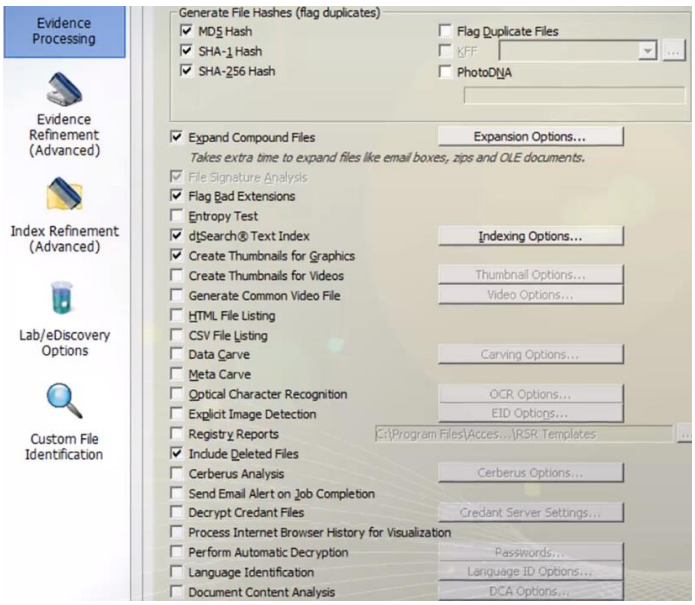
Dar vienas sistemos minusas yra tas, kad, nors „CAINE“ turi mobiliųjų įrenginių teisminės ekspertizės įrankių skiltį, ji nėra taip stipriai išvystyta kaip kompiuterinės analizės skiltis. Ši operacinė sistema yra orientuota į kompiuterių ir tinklų teisminę analizę, todėl mobiliųjų įrenginių analizė nėra stiprioji jos vieta.

1.3.3. „FTK Imager“

„FTK Imager“ – vienas populiariausių teisminės ekspertizės įrankių. Jis skirtas daryti analizuojamų diskų kopijoms (atvaizdams), nepaliečiant originalaus disko (išlaikant duomenų vientisumą). Įrankis tinka kompiuterinei įrangai su x86 architektūra, (mobiliems įrenginiams nepritaikytas, todėl daryti „Android“ įrenginių atminties kopijų negalima) [20].

Programinės įrangos pagrindinės funkcijos (4 pav.):

- sukuria tiriamojo kompiuterio kietojo disko atvaizdą (kopiją);
- parodo diske esančius failus bei aplankus;
- parodo diske esančių failų turinį;
- montuoja atvaizdą tik skaitymo režimu;
- eksportuoja reikalingus failus;
- atstato jau ištrintus failus iš šiukšlių dėžės, tačiau dar neperrašytus, disko atmintyje;
- generuoja failų maišos kodus (angl. „hashcode“);
- generuoja failų maišos kodų ataskaitas teisminei ekspertizei;
- leidžia nulaužti slaptažodžius;
- generuoja ataskaitas.



4 pav. Iš anksto apibrėžiami apdorojami profiliai

Įrankis naudingas, nes:

- siūlo greitį bei stabilumą - „FTK“ naudoja paskirstytą apdorojimą ir yra vienintelis teisminės ekspertizės sprendimas, leidžiantis pilnai panaudoti kelių gijų (angl. „multi-thread“) kompiuterius. Kol kiti įrankiai išvaisto potencialą modernios aparatūros įrangos sprendimams, „FTK“ naudoja pilnai savo resursus, taip padėdami tyrėjams greičiau rasti reikalingus įrodymus;
- siūlo greitesnę paiešką – kai indeksavimas atliekamas iš anksto, filtravimas ir paieška yra užbaigiami efektyviau negu naudojant kažkokį kitą sprendimą. Nesvarbu, ar tiriami, ar žiūrimi dokumentai, šio įrankio pagalba yra bendrinamas failas, todėl nereikia kurti naujų failų ar jų kopijų;
- siūlo duomenų bazės vadovą – „FTK“ yra pilnai valdoma vienos bendros duomenų bazės. Visi duomenys laikomi saugiai ir centralizuotai, todėl komandos gali naudoti tuos pačius duomenis. Visa tai sumažina duomenų kūrimo sąnaudas ir sudėtingumą.

„FTK“ yra naudingas įrankis darant kompiuterių teisminės ekspertizės tyrimus, nes jis yra sukurtas sąveikai su mobiliaisiais įrenginiais ir elektroninių atradimų sprendimais, padeda greičiau rasti svarbius įrodymus, pagreitina analizę ir sumažina atsilikimą. Tai yra vienintelis sprendimas, kuris naudoja vieną bendrą duomenų bazę ir sukuria aiškų įvykio (angl. „event“) vaizdą, tačiau „Android“ įrenginiams jis visiškai netinka, nes radikaliai skiriasi sistemų architektūra.

1.3.4. „Linux dd“

Tai yra vienas iš seniausių vaizdavimo (angl. „imaging“) įrankių, kuris vis dar yra naudojamas. Funkcionalumas yra pakankamas ir išnaudoja mažai išteklių, bet, kad būtų plačiau naudojamas, trūksta tam tikrų naudojamų funkcijų, kurios yra kituose modernesniuose įrankiuose, pvz., metaduomenų rinkimo, klaidų taisymo, šifravimo ir draugiškos vartotojo sąsajos, nes šis įrankis yra programa komandinėje eilutėje, pritaikyta „Unix“ ir šio tipo operacinėms sistemoms, kuri generuoja neapdorotų vaizdų failus bei gali perskaityti daugelis kitų programų. Įrankio pagrindinis tikslas yra konvertuoti ir kopijuoti failus.

„Unix“ aplinkoje įrankis gali rašyti ir/arba skaityti iš/į failus tuo atveju, jeigu funkcija bus įdiegta tam tikrose tvarkyklėse. Dėl to „Linux dd“ galima naudoti tokioms užduotims kaip standžiojo disko „boot“ sektoriaus atsarginių kopijų kūrimui ir fiksuotų atsitiktinių duomenų kiekio gavimui. Programa gali atlikti duomenų konvertavimą, kai duomenys yra kopijuojami,

įskaitant baitų keitimą ir konvertavimą į „ASCII“ („American Standard Code for Information Interchange“) bei „EBCDIC“ („Extended Binary Coded Decimal Interchange Code“) kodavimus.

„Linux dd“ yra dalis „GNU Coreutils“ paketo, kuris buvo perkeltas į daugelį operacinių sistemų. Yra kelios šakos (angl. „forks“), skirtos teismo ekspertizėms, įskaitant „dcfldd“, „sdd“, „dd_rescue“, „dccidd“ ir „Microsoft Windows“ versijas, palaikančias fizinės atminties skaitymą.

Įrankis siūlo reikalingas funkcijas, kurios leidžia įvykdyti teisminę ekspertizę.

1.3.5. „Paladin“

„Paladin“ yra modifikuotas tiesioginis (angl. „live“) „Linux“ paskirstytojas, pritaikytas „Ubuntu“, taip pat palengvinantis įvairias teismo ekspertizės užduotis, siūlydamas funkcijas. Šiuo metu yra 64 ir 32 bitų versijos. Pagal paskutinius duomenis, šis įrankis teisėsaugos, kariuomenės, federalinių, valstybinių bei korporacinių agentūrų yra renkamasis pasaulyje pirmu numeriu.

Taip yra ne veltui – programos įrankiai sujungia kelių, teismo patikrintų, atvirojo kodo įrankių gausybę į paprastą sąsają, kuria gali visi naudotis. Vartotojas gali lengvai ir greitai išbandyti įvairias funkcijas, pradedant antivirusinės, duomenų bazės, programavimo, iššifravimo, diferencijuotų failų, failų sistemų, šifravimo, vaizdavimo, ataskaitos, stenografijos įrankiais, baigiant aparatinės įrangos, interneto, logaritmo, elektroninio pašto, kenkėjiškų programų, atminties, metaduomenų, mobiliųjų įrenginių, tinklo, nuotraukų, išmestų duomenų, socialinės medijos, miniatiūrų (angl. „thumbnail“), laiko juostos analizėmis, o taip pat yra galimybė naudoti kriminalistikos komplektą, „Hex“ redaktorių, „Messenger“ teisminę ekspertizę, slaptažodžių aptikimą, virtualias mašinas, „Windows“ registrus.

Visos „Paladin“ versijos turi nuotolinio aptarnavimo režimą (angl. „Remote Service Mode“), kuris aktyvuojamas „boot“ režime, todėl tyrėjai turi galimybę atlikti įvairias paslaugas nuotoliniu būdu bet kurioje pasaulio vietoje, o tai sumažina išlaidas. „Paladin“ siūlo patobulintas funkcijas, kurių nėra kituose teisminės ekspertizės įrankių rinkiniuose, kurie, beje, kainuoja didelius pinigus.

„Autopsy“ ir „Paladin“ kombinacija leidžia vartotojui atlikti teismo ekspertizę nuo pradžios iki pabaigos, beje, įrankiai gali būti naudojami ir „Mac“, ir „Windows“, ir „Linux“, ir „Android“ failų sistemose, todėl šis įrankis yra vienas iš universaliausių, lyginant su kitais. [35]

1.3.6. Kompiuterių teisminės ekspertizės įrankių apibendrinimas

Anksčiau šiame darbe apžvelgėme „Android“ teisminės ekspertizės įrankius, todėl lygiagrečiai, bet atskirai palyginame ir apžvelgiame ir kompiuterių teisminės ekspertizės įrankius, pagal tuos pačius kriterijus, kad būtų aiškūs skirtumai tarp įrankių (2 lentelė).

	Tinka visiems įrenginiams	**Ataskaita**	**Galima pasiekti duomenis iš programėlių?**	**Atstato duomenis?**	**Ar galima gyvai stebėti atnaujinimus?**	**Nemokamas**
„Autopsy“	-	+	+	+	-	+
„CAINE“	-	+	-	-	-	+
„FTK Imager“	-	+	+	+	-	+
„Linux dd“	-	-	+	+	-	-
„Paladin“	-	+	+	+	+	-

2 lentelė. Kompiuterių teisminės ekspertizės įrankių palyginimas

Pirmasis kriterijus pasižymi tokia pačia tendencija - nėra dinamiško net ir kompiuterių teisminės ekspertizės įrankio, kuris būtų pritaikytas visiems įrenginių modeliams. Priešingai, negu buvo su „Android“ įrenginių įrankiais, kompiuterių ne visi turi ataskaitos formavimo įrankį, nes „Linux dd“ yra komandinės eilutės principais paremtas įrankis, todėl tokioje aplinkoje ataskaita būtų ne visai patogus variantas.

Galimybe gyvai stebėti atnaujinimus pasižymi tik vienas iš analizuotų įrankių.

Išvada, kaip ir apie „Android“ įrenginių ekspertizės įrankius, apie kompiuterių įrankius išlieka ta pati – nėra universalaus įrankio, kuris tiktų visiems modeliams bei turėtų visus apžvelgtus funkcionalumus, neatsižvelgiant į tai, kad funkcijų yra daug.

2. Modelio vizualizavimas

Norint sukurti dinamišką teisminės ekspertizės įrankį, svarbu suprasti modelio vizualizavimą, numatyti galimas problemas, kurios galėtų iškilti įrankio kūrimo bei testavimo metu, apžvelgti architektūrą, įrenginių versijas bei suvokti bendrą įrankio sistemos veikimą.

2.1. Architektūra

Darbas yra orientuotas į „Android“ įrenginius, todėl juos svarbu apžvelgti. Iš aprašytos įrankių analizės akivaizdu, kad „Android“ įrenginių teisminės ekspertizės įrankiai skiriasi nuo kompiuterinių įrenginių įrankių, todėl ir architektūros turi skirtumų [25].

Architektūrų skirtingumas prasideda nuo naudojamų skirtingų aparatinių įrangų (angl. „hardware“) iki skirtingų operacinių sistemų (šiame darbe kompiuteryje naudota „Linux“ operacinė sistema, kuri skiriasi nuo „Android“, nors ir yra iš tos pačios „Unix“ šeimos) [1].

Dėl paminėto fakto skiriasi ne tik įrenginiai, bet automatiškai atsiranda skirtumai ir tarp sukurtų įrankių - būtent dėl to, kad beveik visi kompiuterinės teisminės ekspertizės įrankiai yra paremti gan senais metodais, kurie nebuvo naujinami, priešingai negu buvo daroma su „Android“ įrankiais [28]. Nepaisant to, kad pastarieji įrenginiai buvo naujinami, čia taip pat galima susidurti su nesklandumais teisminėje ekspertizėje, pvz., turimų „Android“ įrenginių modeliai bei versijos gali neatitikti tam tikrų naudojamų įrankių (dėl šios priežasties siekta įgyvendinti dinamiškumą).

2.2. Versijos

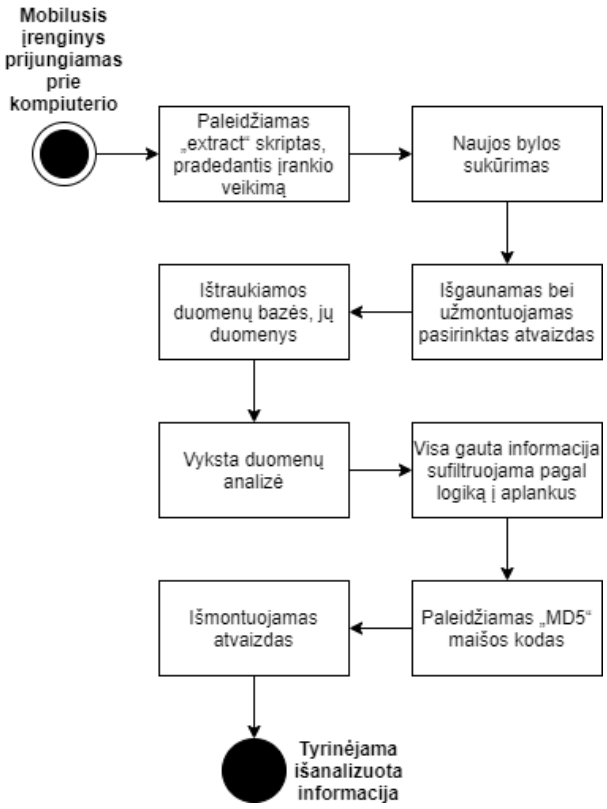
Kaip ir minėta praeitame poskyryje apie architektūrą, svarbu atkreipti dėmesį į turimų įrenginių, norimų naudoti teisminei ekspertizei, esančias versijas, nes, joms nesutampant, nebus galimybės ne tik atlikti duomenų analizės, bet ir pasiekti duomenų.

„Android“ savo mobiliųjų operacinių sistemų versijų istoriją pradėjo tada, kai viešai paskelbė „Android beta“ versiją 2007 m., o „Android 1.0“ išleista 2008 m. Jie ir toliau sėkmingai tobulina bei naujina versijas – šiai dienai jau yra pasiekiama „11.0“ versija [32]. Dėl minėtos situacijos, svarbu patikrinti turimų mobiliųjų įrenginių versijas, prieš pradedant rengti, testuoti teisminę ekspertizę atliekantį įrankį [5].

Tam, kad sėkmingai būtų naudojamas teisminės ekspertizės įrankis, vien atitinkamo mobiliojo telefono modelio bei versijos neužtenka – reikia galimybės mobiliajam įrenginiui suteikti „root“ teises. Jos reikalingos tam, kad galėtume pasiekti bet kokius reikalingus duomenis, esančius mobiliajame telefone.

2.3. Įrankio veikimas

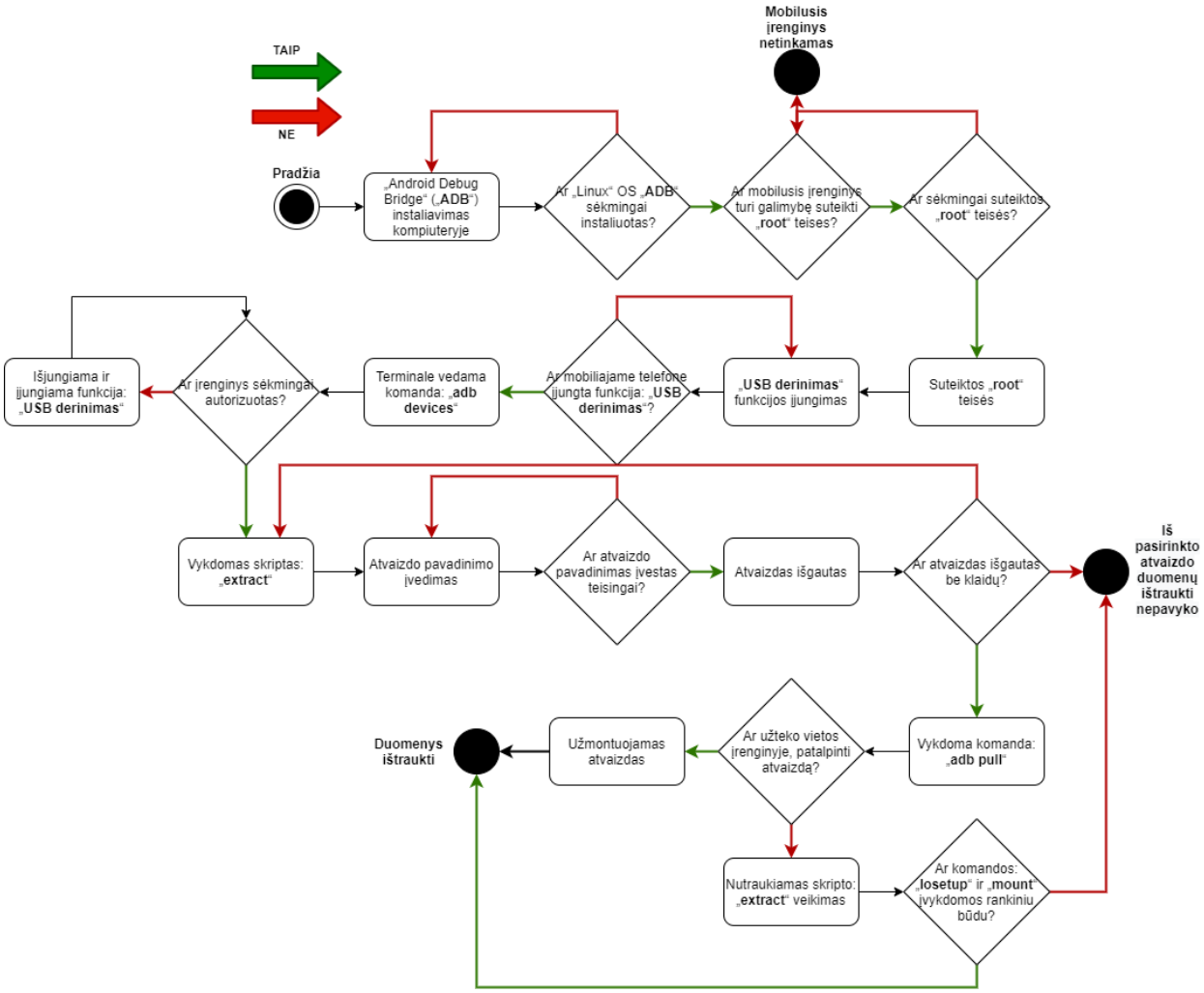
Žinant svarbiausius architektūrinius momentus, galima pradėti vizualizuoti modelį. Jo vizualizavimas padeda suprasti kuriamo bei testuojamo įrankio veikimo principą, kuris pavaizduotas diagramoje (1 diagrama).



1 diagrama. Įrankio veikimo eiga

1 diagrama supažindina su modelio vizualizacijos esme bei paaiškina įrankio veikimą. Yra matomi pagrindiniai aspektai, kuriuos įvykdo teisminės ekspertizės įrankis.

Norint, kad įrankis galėtų veikti, visų pirma, reikia prijungti patį įrenginį prie kompiuterio. Jeigu atsižvelgiama į aprašytus architektūrinius momentus, mobilusis įrenginys sėkmingai pajungiamas. Kadangi vizualinis modelis yra pritaikytas būtent šio darbo sukurtam įrankiui, iš karto žinoma, kad įrankį pradeda/paleidžia skriptas: „**extract**“. Įrankio paleidimą seka tolimesni veiksmai, tokie kaip naujos bylos sukūrimas, vartotojui įvedant reikiamą informaciją, išgaunamas bei užmontuojamas reikalingas atvaizdas sukurtai bylai, tada vyksta duomenų bazių bei jose esančios informacijos ištraukimas, vykdoma jos analizė. Tam, kad būtų aiškesnis duomenų ištraukimo procesas, pateikiama tai reprezentuojanti diagrama (2 diagrama).



2 diagrama. Duomenų ištraukimo procesas

Po pagrindinės dalies (duomenų analizės) svarbu sufiltruoti visą turimą informaciją logine tvarka į aplankus, kad būtų patogu tyrinėti išanalizuotą informaciją. Kai filtravimas įvykdytas, paleidžiamas „**MD5**“ maišos kodas, kuris reikalingas saugumui (duomenų vientisumas yra svarbus momentas šiame darbe). Įrankio darbas užbaigiamas užmontuoto atvaizdo išmontavimu. Paruošiama informacija tolimesniam tyrimui.

Tolimesnis bei detalesnis įrankio veikimas aprašomas šiame darbe, atkreipiant dėmesį į įvairias situacijas, su kuriomis buvo susidurta įrankio kūrimo bei testavimo metu.

3. Implementacija

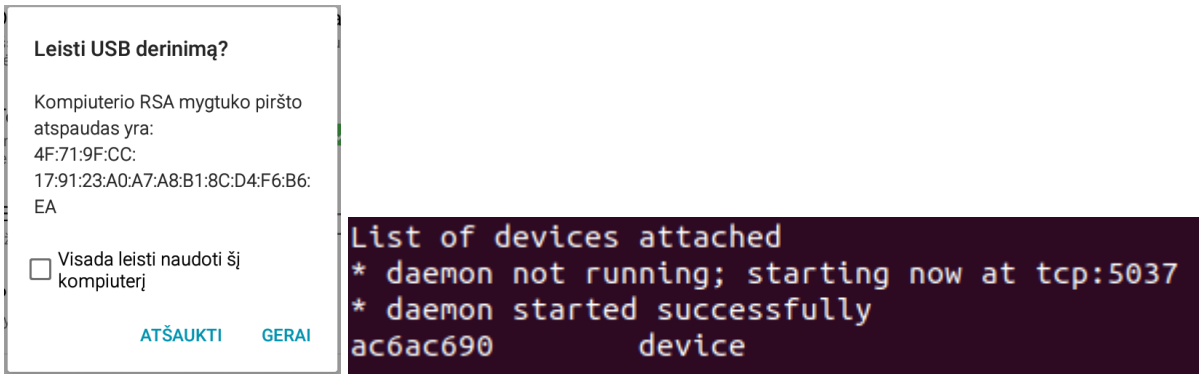
Darbo pagrindinis tikslas - sukurti kuo universalesnį bei dinamiškesnį įrankį, kuris tiktų įvairiems „Android“ įrenginių modeliams, todėl šiame skyriuje apžvelgiamas įgyvendinimas, kuriuo pasiekta įrankio implementacija.

3.1. Įranga ir jos paruošimas

Praktinei daliai turėjau du mobiliuosius įrenginius – „**Samsung Galaxy A3**“ („Android“ versija 5.0.2) ir „LG G4s“ („Android“ versija 6.0). Pastarasis pagrindiniams numatytiems praktinės dalies veiksmams nebuvo naudojamas, nes kilo problema su vienu iš pirmųjų veiksmų – „root“ teisių suteikimu – „LG G4s“ telefonas neatitinka „KingoRoot“ siūlomų versijų, o šis įrankis buvo pasirinktas dėl to, kad yra vienas dažniausiai bei sėkmingiausiai naudojamų. Anot šaltinio [36], yra siūlomas „root“ teisių suteikimas „Android“ įrenginių versijoms nuo 1.5 iki 5.0. Ieškoti ir kiti būdai. „Root“ teisių suteikimui yra ir kitos programėlės, atliekančios tą pačią funkciją reikiamai „Android“ versijai, pvz., tokios kaip „Titanium Backup“ arba „Adblock Plus“, tačiau šiam darbui jų nenaudojau, nes programėlės yra mokamos. Dėl šios priežasties darbą atlikau su „**Samsung Galaxy A3**“ mobiliuoju telefonu, kuriam tiko „KingoRoot“ siūlomos versijos. „Root“ teisių suteikimas reikalingas tam, kad būtų pasiekiama visa reikalinga informacija, esanti mobiliajame įrenginyje, todėl ir ši funkcija yra įtraukta. „KingoRoot“ – vieno paspaudimo programinė įranga, skirta „Android“ mobiliesiems įrenginiams, pasižyminti dideliu sėkmingumo procentu. Tai yra viena duomenų ištraukimo iš nusikaltėlio įrenginio dalis.

Taip pat reikia kompiuterio. Duomenų ištraukimas gali vykti tiek „Windows“, tiek „Linux“ operacinėse sistemose. „Windows“ atveju reikia instaliuoti „Android Studio“ – oficialią integruotą „Google“ „Android“ operacinės sistemos kūrimo aplinką, sukurtą pagal „JetBrains“ „IntelliJ IDEA“ programinę įrangą ir sukurtą specialiai „Android“ plėtrai, kuris turi „**ADB**“ (angl. „Android Debug Bridge“) įrankį – sujungimą tarp mobiliojo įrenginio bei kompiuterio. „Linux“ atveju minėto įrankio nereikia, bet principas išlieka panašus tiek vienos, tiek kitos operacinės sistemos atveju. Šiam darbui pasirinkau „Linux“ operacinę sistemą dėl veiksmingesnių techninių galimybių, kurios bus aprašytos toliau šiame skyriuje.

Mobiliajame įrenginyje, nustatymų pagalba, turi būti įjungtas „Debug Mode“. Taip pat turi būti įrašytas „KingoRoot“, kuris suteikia teisę pasiekti duomenis mobiliajame įrenginyje. Mobiliajame telefone, esančiose kūrėjų parinktyse, reikia aktyvuoti „USB derinimo“ funkciją. Ją įjungus, mobilusis įrenginys klausia, ar leisti USB derinimą. Patvirtinus, „Linux“ operacinės sistemos darbalaukyje matome atsiradusią mobiliojo telefono ikoną (šio darbo atveju – „SAMSUNG Android“), tai reiškia, kad „root“ teisės yra sėkmingai suteiktos, mobilusis telefonas prijungtas ir paruoštas tolimesniems veiksmams. Kai mobilusis įrenginys jau yra su „root“ teisėmis – mobilųjį įrenginį reikia prijungti prie „ADB Shell“. Tam, kad patikrintume, ar mobilusis įrenginys yra tikrai prijungtas, vedame komandą **„adb devices“**. Telefone pasirodo įspėjamasis pranešimas, prašantis patvirtinti USB derinimą (5 pav. kairėje).



5 pav. USB derinimas

Patvirtinus derinimą, terminale matome mobiliojo įrenginio identifikacinį numerį (5 pav. dešinėje). Gali būti situacija, kai įrenginys bus neautorizuotas (angl. „unauthorized“), tokiu atveju mobiliajame įrenginyje kūrėjų parinktyse reikia atšaukti USB derinimo patvirtinimus bei vėl paleisti USB derinimo funkciją, trumpai tariant – perkrauti/atnaujinti derinimą. Apsidraudžiant, rekomenduojama įjungti ir funkciją: „Toliau veikti budrumo režimu“, kad nepaveiktų/nenutrauktų tam tikrų procesų ir kad viso įrankio procesas įvyktų be nenumatytų klaidų, kurias gali sukelti mobiliojo įrenginio „užmigimas“.

3.2. Įrankio paleidimas

Sukurtas aplankalas: „**Project**“, kuriame yra dar vienas aplankalas: „**Cases**“ (čia bus sudėtos įrankio pagalba sukurtos bylos) bei 17 (ir dar vienas skriptas: „**extract**“, paleidžiantis visus likusius) skirtingų skriptų („Shell“ bei „Python“ – įrankis parengtas šių dviejų tipų skriptų pagalba), kurių funkcionalumą iš esmės nusako kiekvieno pavadinimas. Esantį aplanką atidarome terminale (angl. „Open in Terminal“) ir paleidžiame skriptą: „**extract**“ (vedame komandą: „**./extract**“), kuris paleidžia patį įrankį. Šiame skripte yra visas įrankio įgyvendinimas.

3.3. Įrenginio atvaizdo gavimas

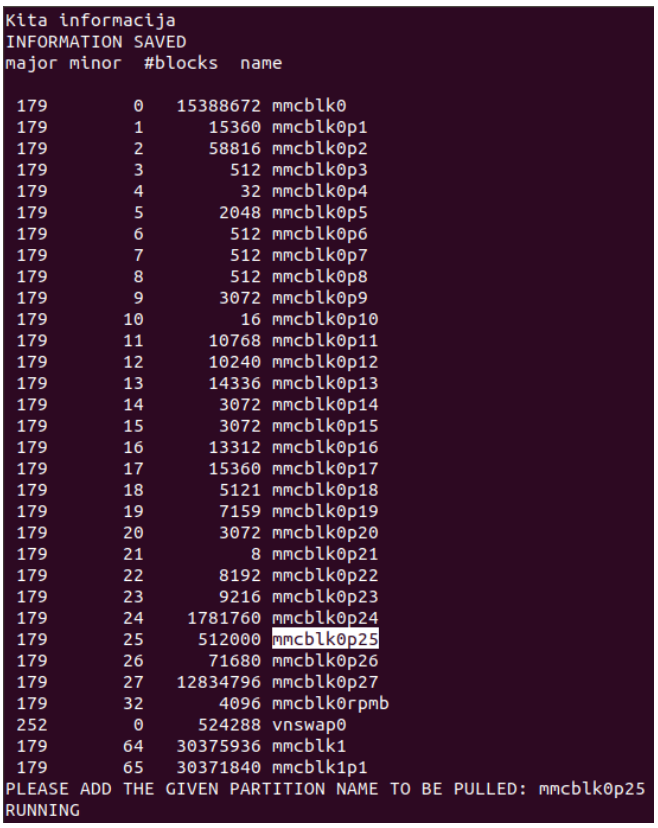
Kai USB derinimas įvyksta sėkmingai bei pats įrankis yra paleidžiamas sėkmingai, įrankis prisistato, kad yra skirtas teisminei ekspertizei. Vartotojas yra kviečiamas įvesti bylos pavadinimą, pvz.: „CaseOne“. Pavadinimui sukurti nėra jokių ribojimų tam, kad bylas būtų galima naudoti pagal reikiamą poreikį. Jeigu byla, vartotojo įvestu pavadinimu jau yra užimta, gaunamas įspėjimas apie tai ir prašoma įvesti kitą bylos pavadinimą (tas pats tikrinimo principas kaip ir particijos įvedime, apie kurį skaitysite toliau). Kai bylos pavadinimas bus tinkamas, bus sukuriama tokia struktūra: sukurtos bylos aplankas (esantis aplanke: „**Cases**“) su įvestu žmogaus pavadinimu (pvz., „CaseOne“), viduje matysime tam tikrą aplankų išdėstymo struktūrą, apie kurią plačiau skaitysite skyriuje apie testavimą.

Dar viena įrankio paskirtis yra atskirti, pvz., particijas (*.img) nuo kitų duomenų failų (nuotraukų failus (*.jpg) atskirti nuo dokumentų (*.txt) ar kitų duomenų failų), todėl, sukurtos bylos viduje matysime šiuos aplankalus: „**Analysis**“ (viduje yra šie aplankalai: „**Browser**“, „**Calls**“, „**DatabasesForAnalysis**“, „**Files**“, „**Report**“ „**SMS**“), „**Databases**“ (*.db), „**Documents**“ (*.txt), „**Photos**“ (*.jpg), „**Videos**“ (*.mp4). Komandinės eilutės pavyzdys, ieškant failų pagal *.jpg šabloną:

„**sudo find /mnt/disk -iname *.jpg -print0 | xargs -I{} -0 cp -v {} ~/Desktop/Project/Cases/$title/Evidence/Photos**“.

Taip pat svarbus yra duomenų, rastų pasirinktame bloke, atskyrimas nuo pačios analizės (jau minėtas aplankas: „**Analysis**“), nes aplankale yra daug svarbios informacijos, reikalingos teisminei ekspertizei atlikti.

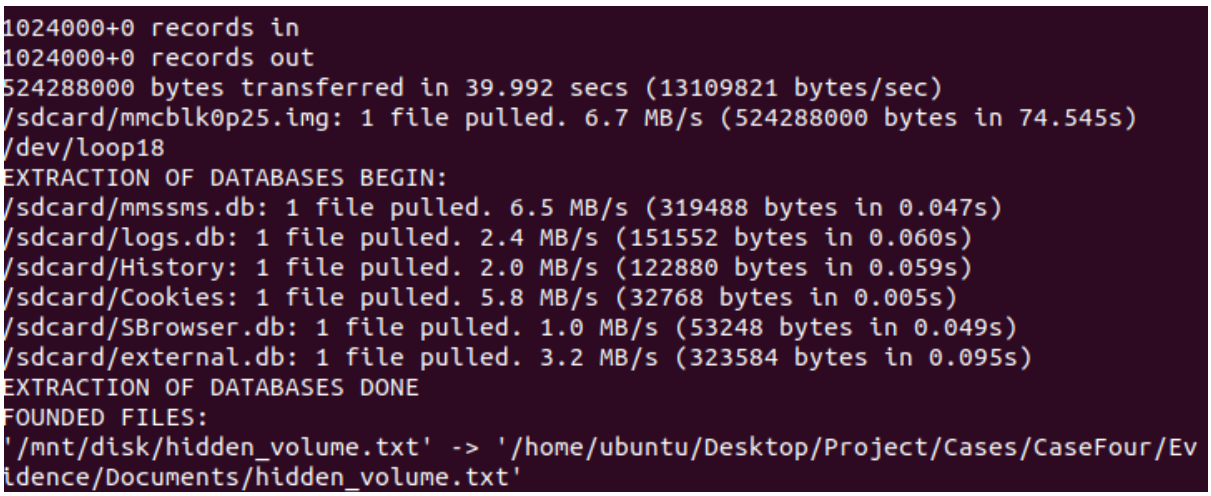
Viena pirmųjų komandų yra „**cat /proc/partitions**“ ir, jai įvykus, sustabdomas veikimas, kad žmogus matytų galimas particijas (angl. „partitions“) (6 pav.) [39].



6 pav. Particijos

Veikimas pratęsiamas tuo, kad prašoma įvesti pasirinktos particijos pavadinimą (tikrinama, kad pavadinimas atitiktų „mmcblk“ šabloną, pvz., „**mmcblk0p25**“; jeigu pavadinimas yra įvestas klaidingai, atitinkamai yra įspėjama apie tai ir prašoma įvesti particijos pavadinimą tol, kol tai atitinka minėtą šabloną – tokiu atveju veikimas vyksta toliau). Svarbu pabrėžti, kad aprašytas šablonas netinka būtent: „vnswap0“ atvaizdui, todėl jis buvo testuojamas rankiniu būdu.

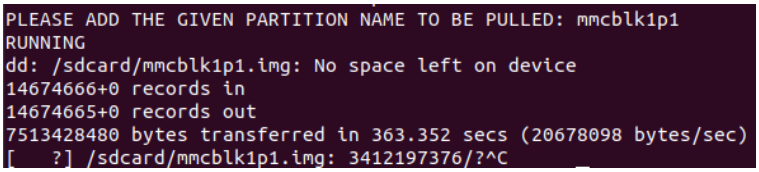
Pačiuose aplankaluose bus talpinami tam tikri duomenys, kuriuos ištrauksime iš mobiliojo įrenginio. Prie jų be klaidų prieisime tuo atveju, kai terminale teisingai įvesime aplankalo (nebus egzistuojančio) bei particijos pavadinimus. Minėti duomenys pasiekiami „**ADB Shell**“ vykdomomis „**su**“ (ši komanda naudojama vykdyti komandoms su kitos vartotojo paskyros privilegijomis (kai ši komanda paleidžiama, ji iškviečia „shell“, nekeisdama dabartinio darbo katalogo ar vartotojo aplinkos)) su „**DD**“ (priešingai negu komanda „**exec-out tar**“, kuria ne failų archyvą, o visą particijų lentelę, kurią sudeda į norimą išvestį) komandomis, po jų – „**adb pull**“ (kopijuoja failus arba katalogus bei jo pakatalogius (angl. „sub-directories“) iš „Android“ įrenginio), „**losetup**“ (nustato ir kontroliuoja „loop“ įrenginius) (iš pradžių vietoje šios komandos buvo naudojamos komandos: „**exFAT**“ bei „**mkfs**“, tačiau su jomis kontroliuoti „loop“ įrenginių nepavyko, nes jie siūlomi tik labai ribotiems duomenims, todėl galutiniam įrankio įvykdymui naudota būtent „**losetup**“) bei „**mount**“ (montuoja diską) funkcionalumų komandos. Sėkmingai užmontavus atvaizdą, 7 pav. matome ne tik paminėtų komandų funkcionalumų eiliškumą terminale, bet ir dar dvi reikalingas funkcijas – duomenų bazių ištraukimą bei rastus failus, kurie yra išskirstomi į kiekvienam jų priklausomus aplankus (terminale matoma vieta, kur patalpinta (7 pav.)).



7 pav. Komandų funkcionalumas terminale

Svarbu atkreipti dėmesį į komandos: „**adb pull**“ dalį – kai traukiamam blokui (pvz., „**mmcblk1p1**“, kuriame didžiąją dalį užima nuotraukos, esančios mobiliajame įrenginyje) neužtenka vietos, įrankio veikimas nutraukiamas, tai reiškia, kad neįvykdytas toliau aprašytas funkcionalumas (8 pav.).

Šią problemą galima išspręsti vietoj komandos: „**adb pull**“ (po kurios ir sutrikdomas įrankio veikimas) vedant komandas: „**adb exec-out tar c**“ (duomenis archyvuoja į *.tar formatą) bei „**tar -xvf**“ (x – išskleidžia archyvą; v – žodinis (angl. „verbal“), pateikia informaciją apie archyvą, jį archyvuojant; f – naudoja archyvo failą), tačiau kompiuteryje, su kuriuo atliktas darbas, nesuveikė – gaunamas pranešimas, kad *.tar failas nėra atpažintas (reikalingi failai yra instaliuoti). Bandyta testuoti tas pačias komandas virtualioje „Kali“ mašinoje, tačiau ten nebuvo galimybės persiųsti atvaizdą, kuris užėmė per daug vietos – persiuntimas užstrigdavo.



8 pav. „adb pull“ komandos veikimas terminale, norint išgauti „mmcblk1p1“ atvaizdą

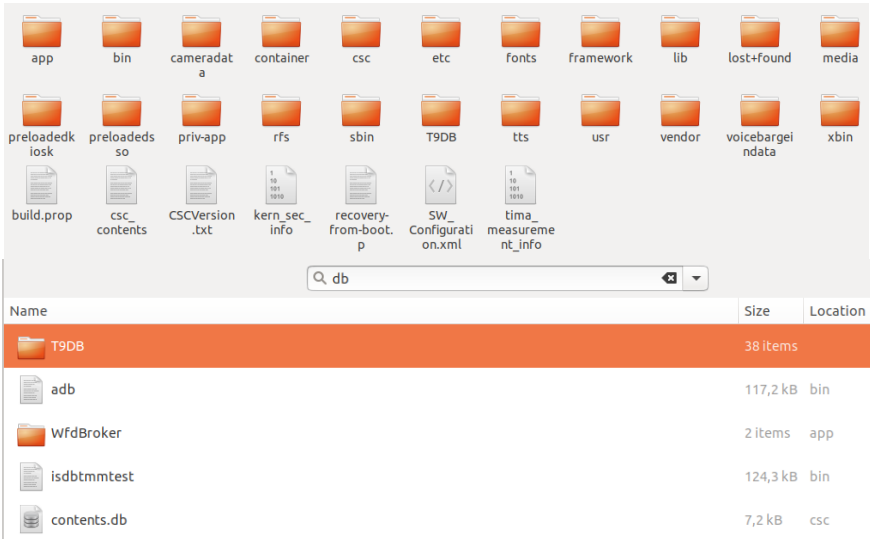
Kadangi veikimas po komandos užstringa (sukasi skaičiai dėl vietos trūkumo), įrankis yra išjungiamas „CTRL+C“ kombinacija. Nepaisant to, duomenis (šiuo atveju - nuotraukas) vis tiek galima ištraukti, surenkant reikalingas komandas („**losetup**“ bei „**mount**“) rankiniu būdu.

Tęsiant analizuoti 7 pav. esantį funkcionalumą, apačioje matome: „FOUNDED FILES“ – rodo rastus failus pagal įvestus kriterijus, kurie yra paimti iš „**/mnt/disk**“ katalogo (9 pav. viršuje).

Po reikalingų duomenų bazių ištraukimo, jos bus perkeliamos į katalogą: „**Analysis**“, kuriame bus reikalingos analizei atlikti, o po duomenų skirstymo dalies, *.img failas (atvaizdas), kadangi tolimesniam funkcionalumui nebebus reikalingas, yra perkeliamas į jam skirtą aplanką: „**Partitions**“.

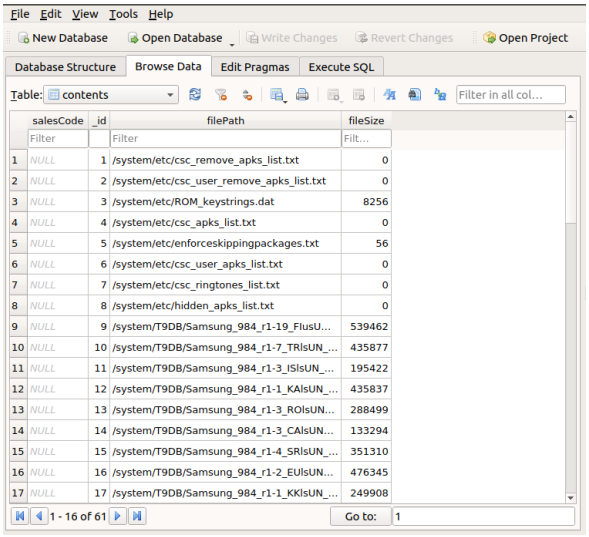
3.4. Duomenų bazės

Darbas su duomenų bazėmis šiame darbe taip pat yra reikšmingas, todėl *.db tipo failai yra atitinkamai aktualūs. 9 pav. apačioje parodyta, kad yra galimybė ir rankiniu būdu patikrinti, pvz., ar yra *.db tipo failų, įvedant raktažodį: „db“ į paieškos laukelį.



9 pav. Užmontuotas įrenginio atvaizdas

Pradinis duomenų bazių analizavimas buvo atliktas įrankio: „Autopsy“ pagalba. Yra galimybė analizuoti *.db tipo failus ir detaliau (10 pav.) „DB Browser for SQLite“ įrankio pagalba (tai yra aukštos kokybės, vizualus, atvirojo kodo įrankis, skirtas kurti, projektuoti ir redaguoti duomenų bazės failus, suderinamus su „SQLite“).



10 pav. Duomenų analizė

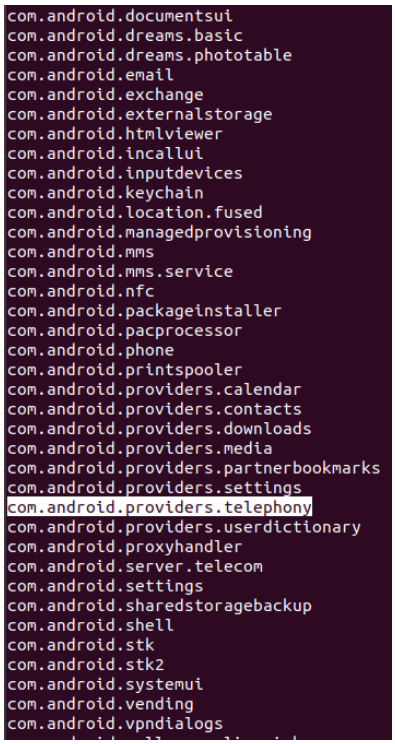
Pagrindinis dėmesys skirtas toms duomenų bazėms, kurios turi didžiausią tikimybę, kad pravers teisminėje ekspertizėje, todėl pagrinde darbas vyko su SMS žinučių, skambučių žurnalo istorija, naršyklės istorija, „sausainiais“, žymėmis, kitais failais bei kitų failų duomenų bazėmis.

Šiame darbe buvo ištraukti ir išanalizuoti visų particijų duomenys dėl testavimo (plačiau parašyta 4 skyriuje), todėl svarbu paminėti, kad ne visos duomenų bazės gali būti pasiekiamos particijų ištraukimo būdu (pvz., gauname tokias klaidas kaip: „NTFS signature is missing“ [18] [8] arba „Wrong fs type, bad option, bad superblock“. Abi šios klaidos būna montavimo proceso metu), tačiau tam yra sprendimo būdas. Gali būti, kad tam tikros duomenų bazės kažkur užšifruotos kaip kitokie failai su kitais pavadinimais, tačiau šio darbo atveju pavadinimai atitiko standartinius planinius rodiklius (angl. „targets“) ar, kitaip tariant, vietas, kur talpinamos duomenų bazės:

- SMS žinutės:
  /data/data/com.android.providers.telephony/databases/mmssms.db

- skambučių žurnalo istorija:
  /data/data/com.sec.android.provider.logsprovider/databases/logs.db
- naršyklės istorija:
  /data/data/com.android.chrome/app_chrome/Default/History
- „sausainiai“ (angl. „cookies“):
  /data/data/com.android.chrome/app_chrome/Default/Cookies
- žymės (angl. „bookmarks“):
  /data/data/com.sec.android.app.sbrowser/databases/SBrowser.db
- įvairūs failai:
  /data/data/com.android.providers.media/databases/external.db

Tam, kad galėtume patikrinti šių ar kitų duomenų bazių vietas bei pačius duomenis, visų pirma, reikia prijungti mobilųjį įrenginį bei patikrinti, ar telefonas yra prijungtas, **„adb devices“** komandos pagalba. Prisijungiame „**ADB Shell**“ komandinės eilutės pagalba bei suteikiame „root“ teises (komanda **„su“**). Einame į „**/data/data**“ lokaciją, o joje, įvedę komandą **„ls“** matome duomenų paketų (angl. „packages“) sąrašą (11 pav.).

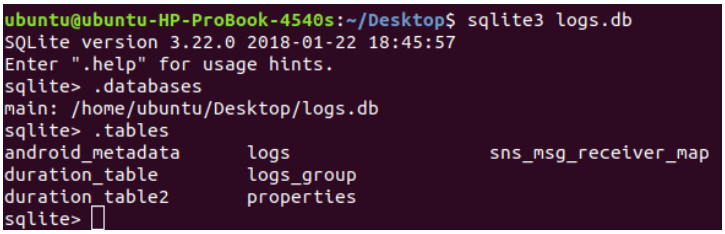


11 pav. Duomenų paketai

11 pav. baltai yra pažymėtas vienas iš paketų, kuris aktualus teisminei ekspertizei. Šioje situacijoje, kai jau žinome duomenų bazės patalpinimo vietą, norėdami pažiūrėti duomenis pačiame terminale, rašome komandą, pvz.: **„cat /data/data/com.android.provider.logsprovider/databases/logs.db“** ir matome duomenis, pavaizduotus 12 pav.

12 pav. „logs.db“ aplanko turinys terminale

Terminale matomi telefono numeriai, žinučių turinys, tačiau tai nėra patogiausias būdas analizuoti duomenis, todėl iš pradžių galima patikrinti duomenis struktūrizuotai jau minėto įrankio „DB Browser for SQLite“ pagalba. Tokiu atveju minėtos komandos pabaigoje pridedame, pvz.: **„> /sdcard/logs.db“** ir tada iš telefono duomenų bazę perkeliame į kompiuterį. Pačias lenteles bei kitas funkcijas taip pat dar galima peržiūrėti ir „SQLite3“ pagalba (13 pav.).



13 pav. Skambučių žurnalo istorijos duomenų bazės lentelės

Komanda „**sqlite3 logs.db**“ paleidžia pačią duomenų bazę minėtoje aplinkoje. Tada galime naudoti įvairias funkcijas būtent tai duomenų bazei, pavyzdyje yra naudojama komanda **„.tables“**, kuri parodo lenteles, kurias turi duomenų bazė. Duomenis lygiagrečiai galima analizuoti „SQLite3“ (13 pav.) bei „DB Browser for SQLite“ (10 pav.) pagalba, nes funkcionalumo esmė yra panaši.

Tam, kad būtų aiški naudojamų duomenų bazių struktūra, 3 lentelėje matome „SQL“ užklausoms atrinktas duomenų bazes, lenteles bei jose esančius duomenis, informacijai analizuoti:

Informacija	Duomenų bazė	Lentelės	Naudoti duomenys
SMS žinutės	mmssms.db	sms	date body address
Skambučių žurnalo istorija	logs.db	logs	m_content number type duration type
Naršyklės istorija	History	urls	url title
Naršyklės „sausainiai“	Cookies	cookies	host_key name is_secure creation_utc expires_utc last_access_utc

			priority
Naršyklės žymės	Sbrowser.db	BOOKMARKS	TITLE
Nuotraukos bei kiti failai	external.db	images, files	_display_name datetaken bucket_display_name _data

3 lentelė. „SQL“ užklausoms atrinkta informacija, duomenų bazės ir lentelės

3.5. Duomenų analizė

Prieš tai aprašytos funkcijos įvykdytos, todėl po jų visi skriptai yra nukopijuojami į sukurtąją bylą, kad būtų galima vykdyti duomenų analizę norimai bylai. Po to įrankio funkcionalumas tęsiasi „**Analysis**“ kataloge, kadangi analizei reikalingos duomenų bazės yra pasiektos.

Iš pradžių, prieš pradedant duomenų analizę, kad būtų aiškuu, kokios duomenų bazės buvo reikalingos tam tikrai analizei, pateikiama lentelė, kurioje yra matoma, kokios duomenų bazės naudojamos tam tikruose skriptuose (4 lentelė).

Duomenų bazė	**Skriptai**
mmssms.db	SMSDate.py, wordMarking
logs.db	CallsAnalysis.py SMSAnalysis.py
History	URLs.py
Cookies	CookiesAnalysis.py CookiesDate.py
SBrowser.db	Bookmarks.py
external.db	DCIM.py Files.py

4 lentelė. Skriptai, kurie naudoja duomenų bazes

Iš lentelėje esančių duomenų bazių bei skriptų pavadinimų galima suprasti, į ką buvo kreiptas pagrindinis dėmesys, atliekant duomenų analizę.

„Shell“ skripte: „**extract**“ paleidžiamas kitas skriptas: „**run.sh**“ – jis atlieka pagrindinę įrankio funkciją – duomenų analizę. Yra paleidžiamas septyniolikos skirtingų skriptų rinkinys.

Jeigu skriptų rinkinys paleidžiamas sėkmingai, terminale kiekvieno skripto metu pranešama, kad duomenų bazė buvo sėkmingai pajungta bei atjungta ir pranešama, kad tam tikro skripto failas yra taip pat sėkmingai sukurtas. Kitu atveju, jeigu testuojant su kitu mobiliuoju įrenginiu atsirastų klaida, pranešama, kad nepavyko perskaityti lentelių, dėl to kažkurių duomenų analizė gali būti neįvykdyta.

3.5.1. Grafas

Semantinis tinklas (angl. „Semantic Network“) – terminas, kuris reiškia ateities interneto tinklą kaip globalią duomenų bazę. Semantinio tinklo infrastruktūra leistų tiek žmonėms, tiek mašinoms daryti sprendimus, kaip suskirstyti turimą informaciją ir ją analizuoti [33]. Vienas iš duomenų analizės tipų yra tam tikrų duomenų atvaizdavimas grafe, pasitelkiant semantinius tinklus. Tam naudotas įrankis: „Graphviz“, kuris yra atvirojo kodo grafikos vizualizacijos programinė įranga. Grafiko vizualizacija yra būdas atvaizduoti struktūrinę informaciją kaip abstrakčių diagramų ir tinklų diagramas [12].

Svarbu atkreipti dėmesį, kad, norint sėkmingai paleisti skriptus, kuriuose naudojamas „Graphviz“, būtina suvesti šias komandas:

„**sudo apt-get install python**“,

„**sudo apt-get install graphviz**“,
„**pip install graphviz**“.

Tam, kad būtų aiškiau suprasti duomenų atvaizdavimą grafo pagalba, šiame darbe bus aptarti pavyzdžiai, vienas iš jų – „**SMSCriminals.py**“ (šio skripto paleidimo komanda: „**python SMSCriminals.py**“). Skripto pagrindinis tikslas - ištraukus duomenis apie žinutes (ją rašiusių asmenų mobiliuosius numerius, tipus (žinutė siųsta ar gauta) bei skaičiuojant tam tikrų tipų žinučių kiekį), išsiaiškinti, ar nėra daugiau įtariamųjų be pagrindinio įtariamojo. Algoritmas, leidžiantis išsiaiškinti, ar yra daugiau įtariamųjų, veikia taip:

```
all_rows = cursor.execute('SELECT number, type, count(type) FROM logs where type <> 3 and number is not null and
g1 = gv.Digraph(format='png')
who = ['Main Suspect']
for row in all_rows:
    g1.attr(label='Other Suspects')
    g1.node(str("Other Suspect"), color='#ff0040', style='filled', shape='rarrow')
    g1.node(str("Not a Suspect"), color='#00ffbf', style='filled', shape='rarrow')
    if(row[2] > 30):
        g1.edge(str(row[0]), who[0], label= str(row[2]), color='#ff0040', style='filled')
    else:
        g1.edge(who[0], str(row[0]), label= str(row[2]), color='#00ffbf', style='filled')
```

1 kodas. Algoritmas implementacijoje: „SMSCriminals.py“

Šio algoritmo atveju yra primetama sąlyga, kad kito įtariamojo tikimybė yra tada, jeigu asmuo iš vieno numerio siuntė daugiau negu 30 žinučių.

Kitas pavyzdys – skriptas: „**SMSAnalysis.py**“, kuris analizuoja žinučių skaičių, gautų iš skirtingų mobiliųjų telefonų numerių bei išsiųstų skirtingiems gavėjams (2 kodas).

```
all_rows = cursor.execute('SELECT number, type, count(type) FROM logs where type <> 3 and number
g1 = gv.Digraph(format='png')
who = ['Suspect']
for row in all_rows:
    g1.attr(label='Number of SMS')
    g1.node(str("Received"), color='#8a0045', style='filled', shape='rarrow')
    g1.node(str("Sent"), color='#008a45', style='filled', shape='rarrow')
    if(row[1] == 1):
        g1.edge(str(row[0]), who[0], label= str(row[2]), color='#8a0045', style='filled')
    else:
        g1.edge(who[0], str(row[0]), label= str(row[2]), color='#008a45', style='filled')
```

2 kodas. Algoritmas implementacijoje: „SMSAnalysis.py“

Tam tikrais atvejais vizualus analizuojamų duomenų būdas (šiuo atveju - grafai) yra naudingas teisminei ekspertizei. Šis duomenų analizės tipas taip pat buvo panaudotas ir dar keliuose skriptuose: „**CallsAnalysis.py**“, „**CookiesAnalysis.py**“.

3.5.2. Žinučių filtravimas

Kadangi žinutės sudaro gana didelę dalį teisminės ekspertizės, joms yra sukurtas skriptas: „**wordMarking**“, kurio esmė tokia – vartotojo prašoma įvesti tam tikrą raktažodį ar raktažodžius, pagal kuriuos tikisi atrasti tam tikras žinutes, o įrankis filtruoja tas žinutes. Šis funkcionalumas yra įvykdytas (3 kodas) algoritmo pagalba:

```
function sms ()
{
    echo "ENTER KEYWORDS (PRESS CTRL+D WHEN DONE)"
    while read line
    do
    words=("${words[@]}" $line)
    done
    for i in "${words[@]}"
    do
    :
    sudo sqlite3 mmssms.db "SELECT body, address from sms where body LIKE '%$i%';" >> ~/Desktop/Project/Cases/filteredsms.tx
    done

    for i in "${words[@]}"
    do
    :
    sed -i "s/${i}/∞${i}∞/Ig" ~/Desktop/Project/Cases/filteredsms.txt
    done
}
sms
```

3 kodas. Algoritmas, filtruojantis žinutes pagal įvestą raktinį žodį

Skripto (3 kodas), skirto žinučių filtravimui pagal vartotojo įvestą raktinį žodį, funkcionalumas – pagal įvestą raktinį žodį yra atrenkamos visos žinutės bei išvedamos į „**filteredSMS.txt**“ failą. Išskiriami tie žodžiai, kuriuos įvedė vartotojas, o, šalia jų, nurodomas mobiliojo telefono numeris.

3.5.3. Duomenų išvedimas

Vienas iš pagrindinių būdų duomenims analizuoti yra pačių duomenų išvedimas, šiuo atveju, į *.txt failą. Šis duomenų analizavimas yra vienas paprasčiausių ir aiškiausių sukurtame įrankyje, dėl to, kad patogu analizuoti reikiamą informaciją, tačiau šiame darbe noriu atkreipti dėmesį į skriptą: „**CookiesAnalysis.py**“. Jame iš duomenų bazės: „**Cookies**“ analizuojami tokie duomenys kaip: „**creation_utc**“, „**expires_utc**“ bei „**last_access_utc**“, kurių formatas duomenų bazėje yra, pvz., 13, žmogui nesuprantamų, skaitmenų, todėl šis skriptas konvertuoja duomenų bazėse esantį datos formatą į žmogui suprantamą formatą.

```
sqlite_select_query = """SELECT DISTINCT name, datetime(creation_utc/10000000, "unixepoch", "localtime"),
datetime(expires_utc/10000000, "unixepoch", "localtime"),
datetime(last_access_utc/10000000, "unixepoch", "localtime"), priority FROM cookies group by name"""
cursor.execute(sqlite_select_query)
records = cursor.fetchall()
with open('CookiesDate.txt', 'w') as f:
    for row in records:
        f.write("Name: %s\n" % str(row[0]))
        f.write("Creation UTC: %s\n" % str(row[1]))
        f.write("Expires UTC: %s\n" % str(row[2]))
        f.write("Last access UTC: %s\n" % str(row[3]))
        f.write("Priority: %s\n" % str(row[4]))
        f.write("%s\n" % str("\n"))

cursor.close()
```

4 kodas. Algoritmas, konvertuojantis datą į žmogui suprantamą formatą

Algoritmas konvertuoja ištrauktus duomenis (kiekvienus atskirai) į aiškų datos formatą bei sukuria failą: „**CookiesDate.txt**“, kuriame yra išvedami jau teisminei ekspertizei paruošti duomenys.

Dažniausiai šiame darbe duomenų bazėse datos buvo pateiktos milisekundėmis arba nanosekundėmis – tiek vienas, tiek kitas pavyko konvertuoti į aiškų datos formatą. Šis duomenų analizavimas naudotas ir kituose skriptuose, pvz.: „**Bookmarks.py**“, „**URLs.py**“, „**DCIM.py**“, „**Files.py**“, „**SMSDate.py**“.

3.6. Filtravimas

Sėkmingai ištraukus duomenis bei įvykdžius informacijos analizę, prieinama prie taip pat svarbios dalies – filtravimo. Filtravimas svarbus tam, kad teisminės ekspertizės metu būtų greitai ir lengvai matoma bei pasiekiama bylai reikalinga informacija bei jos analizė.

Viena paskutiniųjų skripto: „**extract**“ komandinių eilučių atlieka pagrindinį filtravimo darbą. Kadangi išanalizuota informacija viename aplanke atrodytų tikrai nestruktūrizuotai, kiekvienas failas (nebuvo automatizuota, pvz., skirstant pagal *.py arba pagal *.txt failus, nes skirtingi *.py failai pagal logiką priklauso skirtingoms analizėms) buvo nukopijuotas iš bendrojo aplanko: „**Analysis**“ bei nukopijuotas į kiekvienam skirtą vietą (pvz.: „**sudo mv /home/ubuntu/Desktop/Project/Cases/$title/Evidence/Analysis/SMSAnalysis.png /home/ubuntu/Desktop/Project/Cases/$title/Evidence/Analysis/SMS/SMSAnalysis.png**“). Atitinkamai padaryta ir su duomenų bazėmis, kurios buvo naudotos skriptuose.

Išskirsčius visus išanalizuotus duomenis logiška prasme, informacija praktiškai yra paruošta tyrinėti tolimesniems darbams.

3.7. Ataskaita

Gautus rezultatus suformatuoja skriptas: „**reportPDF.sh**“, kuris savyje turi „Python“ skriptą: „**dataPDF.py**“. „Shell“ skriptas kviečia egzaminuotoją parašyti tam tikrus komentarus apie tam tikrus išvestus duomenis („**CookiesDate.txt**“, „**SMSDate.txt**“, „**Files.txt**“, o parašyti komentarai apie kiekvieną iš jų bus atitinkamai išvesti šiuose failuose: „**cookiesDateInfo.txt**“, „**SMSDateInfo.txt**“, „**filesInfo.txt**“). Po komentarų suvedimo bei jų išsaugojimo, kviečiamas kitas „Python“ skriptas, kuris sukuria galutinę ataskaitą apie visus tekstinius failus (5 kodas).

```
pdf.cell(190, 10, txt = "THIS IS A REPORT OF ALL GENERATED TXT FILES", ln = 1, align = 'C')

f = open("cookiesDateInfo.txt", "r")
for x in f:
    pdf.cell(190, 10, txt = "COOKIES DATE: " + x, ln = 1, align = 'C', border = 1)
f = open("CookiesDate.txt", "r")
for x in f:
    pdf.cell(190, 10, txt = x, ln = 1, align = 'L')
```

5 kodas. Algoritmas implementacijai: „reportPDF.sh“

Algoritmas (5 kodas) patalpina egzaminuotojo suvestus komentarus apie minėtus išvestus duomenis, pačius duomenis bei likusių tekstinių failų informaciją. Ši visa informacija išvedama į bendrą galutinę ataskaitą: „**REPORT.pdf**“, kuri bus perkelta į katalogą: „**Report**“.

Analizės metu duomenys bus išvedami ne tik *.txt, bet ir *.png formatu, todėl pastariesiems failams sukurtas kitas algoritmas: „**graphsPDF.py**“ (6 kodas).

```
from PIL import Image

image1 = Image.open(r'CallsAnalysis.png')
image2 = Image.open(r'CookiesAnalysis.png')
image3 = Image.open(r'SMSAnalysis.png')
image4 = Image.open(r'SMSCriminals.png')

im1 = image1.convert('RGB')
im2 = image2.convert('RGB')
im3 = image3.convert('RGB')
im4 = image4.convert('RGB')

imagelist = [im1,im2,im3,im4]

im1.save(r'Graphs.pdf',save_all=True, append_images=imagelist)
```

6 kodas. Algoritmas implementacijai: „graphsPDF.py“

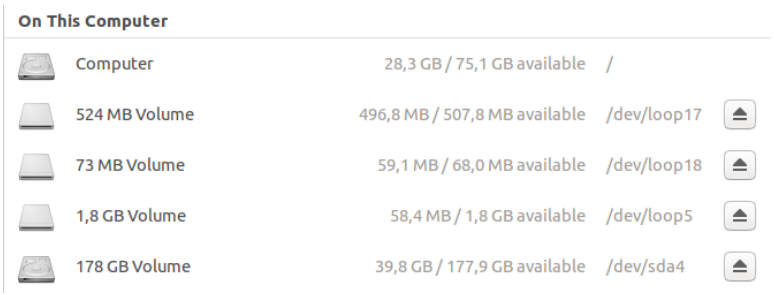
Algoritmas (6 kodas), skirtas *.png failų išvedimui į ataskaitą, išveda duomenis į *.pdf formato failą: „**Graphs.pdf**“.

3.8. „Hash“

„**MD5.py**“ skriptas sukuria manifesto failą (manifesto failas – failas, kuris atsako už visą failinės medžio struktūros, kurioje guli failai, ir tikrina jų „CRUD“ operacijas): „**hashes.txt**“. Yra išlaikomas duomenų vientisumas, konfidencialumas bei prieinamumas (angl. „CIA – Confidentiality, Integrity, Availability“ [30]). „MD5“ maišos kodas sukuria reikalingą šifruotę tam, kad apsaugotume duomenis bei jie nebūtų taip lengvai atsekami. Ši funkcija yra paleidžiama viena iš paskutiniųjų bendrame įrankyje, kai jau visi failai yra sukurti, kad visi jie ir būtų užšifruoti [13].

3.9. Įrenginio atvaizdo šalinimas

Pats paskutinis įvykdomas skriptas įrankyje - **„detach.sh“**, kurio funkcija yra išmontavimas. Pirma jo komanda atlieka „**unmount**“ (išmontavimo) veiksmą. Po jo vykdoma **„losetup grep“**, kurio pagalba matome užkabintą „loop“ pavadinimą, o tada „loop“ naikiname su komanda **„sudo losetup -D“** (komanda naikina visus „loop“, kurie kabo, todėl nereikia naikinti jų po vieną). 14 pav. matome, kaip atrodo užkabinti „loop“ (taip neturi būti).



14 pav. Užkabinti „loop“

Visos minėtos komandos, išmontuoja atvaizdą. Terminale parodomas „loop“, kuris išmontuojamas, pvz., **„/dev/loop16“**.

Įrankį užbaigia pranešimas: „DONE“, tai reiškia, kad įrankio teisminė ekspertizė baigta ir galima tikrinti duomenis.

3.10. Įrankio pagrindinė esmė

Apibendrinant įrankio veikimą, kuris analizuoja įkalčių ekspertizę, svarbu suprasti, kad, visų pirma, vyksta duomenų ištraukimas iš nusikaltėlio įrenginio. Kadangi naudojau „Linux“ operacinę sistemą, įrankio, kuris sujungtų mobiliųjį įrenginį su kompiuteriu, papildomai siųsti nereikėjo. Telefone per nustatymus turi būti įjungtas „Debug Mode“, turi būti įrašytas „KingoRoot“, kuris suteikia teisę atsisiųsti disko atvaizdus (angl. „images“) iš „Android“ telefono. Ištraukus atvaizdus, naudoju „dd“ bei „adb pull“ komandas kopijavimui iš telefono į SD kortelę (mobiliajame įrenginyje bendroji įrenginio talpa yra 16 GB, todėl naudojau 8 GB, kad tilptų bent pusė įrenginyje esančių duomenų). Turint duomenis, yra vykdomi visi procesai, aprašyti aukščiau, kurie sudaro teisminės ekspertizės esmę.

Vienas iš pagrindinių šio įrankio privalumų – teisminė ekspertizė įvyksta labai greitai, nepaisant to, kad funkcijų yra ne viena ir ne dvi. Tai priklauso nuo to, kokio dydžio atvaizdas yra išgaunamas, pvz.: „**mmcblk1p1**“ tikrai užtruks ilgiau negu atvaizdas: „**mmcblk0p13**“, bet esmė išlieka ta pati – greitis bet kokiu atveju yra svarbus [23].

4. Testavimas

Kaip jau minėta anksčiau, šio darbo pagrindinis reikalingas resursas yra mobilusis įrenginys. Konkrečiai šiam darbui buvo naudotas būtent „**Samsung Galaxy A3**“ („Android“ versija 5.0.2), su šiuo mobiliuoju įrenginiu atliktos bei testuotos sukurtos funkcijos, kurios sudaro parengtąjį įrankį. Taip pat turėtas ir dar vienas mobilusis telefonas „LG G4s“ („Android“ versija 6.0), bet šis įrenginys nebuvo tinkamas rengti bei testuoti įrankį, nes neatitinka versijų, kas reiškia, kad nebuvo galima suteikti „root“ teisių, kad būtų galima daryti tolimesnius veiksmus, tokius kaip atvaizdo išgavimas, duomenų analizavimas. Galimybė kurti bei testuoti įrankį tik su vienu mobiliuoju telefonu kėlė tam tikrą iššūkį, nes nebuvo galimybės aptikti klaidų, testuojant su kitais mobiliaisiais įrenginiais, kad jas būtų galima išspręsti, siekiant sukurti kuo universalesnį įrankį. Nepaisant to, nors ir turint galimybę testavimą atlikti tik su vienu įrenginiu, vis tiek tikslas buvo sukurti kuo dinamiškesnį įrankį, kad tiktų įvairių „Android“ įrenginių modeliams, todėl, atsižvelgiant į šį tikslą, įrankis taip ir buvo sukurtas.

4.1. Atvaizdo išgavimas

Svarbu paminėti, kad buvo naudota 8 GB SD kortelė (pirkta būtent tokio dydžio, nes dvigubai tiek pat užima duomenys mobiliajame įrenginyje – įėję į mobiliojo telefono nustatymus, matome, kad visa įrenginio vieta (16 GB) yra užimta). Pagrinde SD kortelė buvo reikalinga tam, kad būtų vietos ištrauktiems atvaizdams patalpinti. Žinoma, 8 GB dydžio kortelė nėra užtektina visiems atvaizdams sutalpinti, pvz., „**mmcblk1p1**“, kuriame didžioji informacijos dalis yra nuotraukų pavidalu, nufotografuotos mobiliuoju telefonu – užima gerokai daugiau vietos. Vienas iš to sprendimų galėtų būti didesnės talpos SD kortelė. Pats atvaizdo ištraukimo bei jo užmontavimo principas yra aiškus bei paprasto veikimo, bet tam išpildyti, kad veiktų be jokių klaidų, pareikalavo, kuriant įrankį, gana daug pakeitimų, tobulinimų bei testavimo, įsitikinant, kad galutinis rezultatas veiktų taip kaip siekiama sukurti.

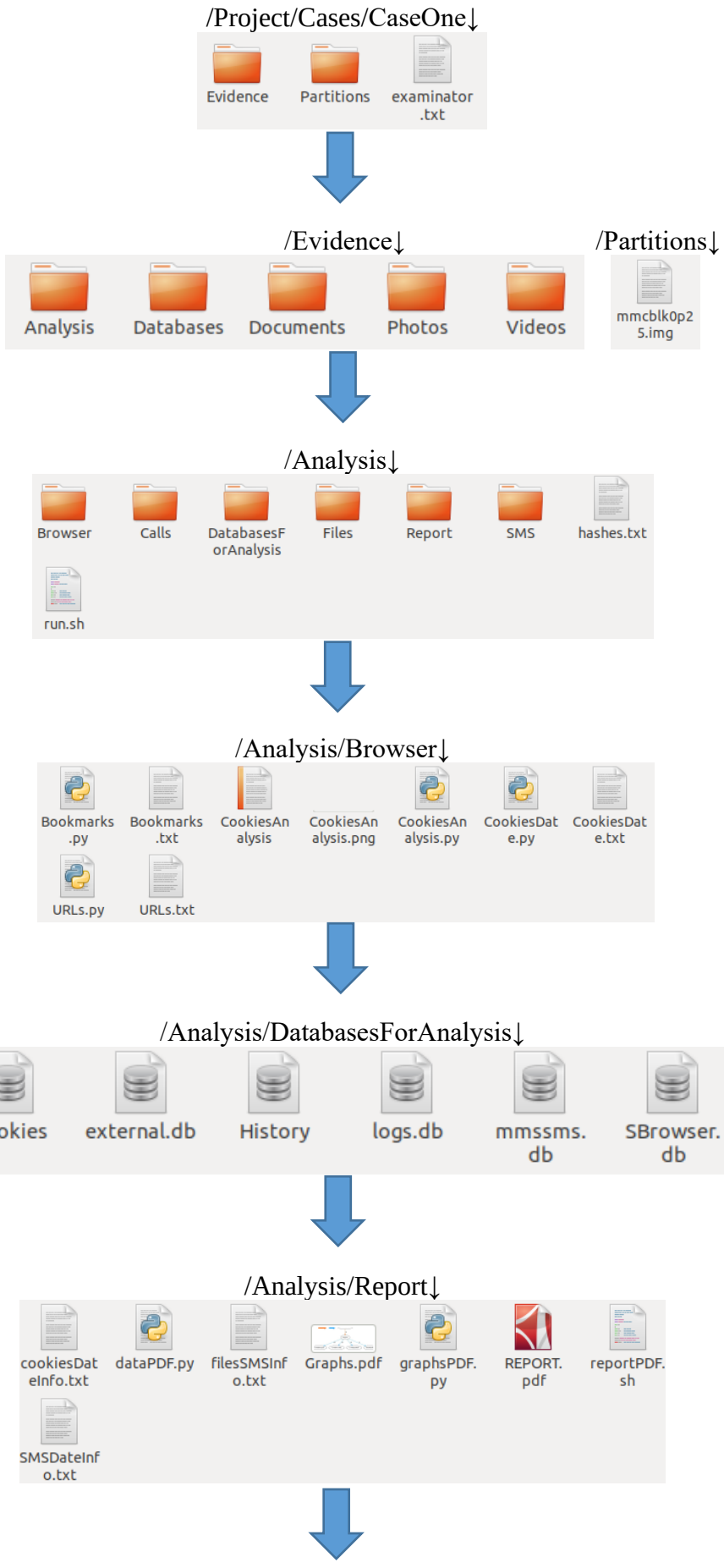
Ne visus atvaizdus pavyko ištraukti visiškai be klaidų, todėl buvo ištestuotos visos esamos particijos nuo „**mmcblk0**“ iki „**mmcblk1p1**“. Tų atvaizdų, kurių nepavyko išgauti automatizuotu būdu, pvz., buvo užfiksuotos klaidos: „NTFS signature is missing“ ir „bad option error“, taip pat pusiau klaida galima laikyti ir vietos trūkumą mobiliajame įrenginyje), tačiau tam tikros klaidos buvo išspręstos, ištraukiant atvaizdus rankiniu būdu (pvz., „**mmcblk0**“, „**mmcblk1**“), taip patikrinant esančią informaciją. Įprastai, jeigu automatizuotu būdu atvaizdo išgavimas buvo sėkmingas (pavyko su daugeliu particijų, pvz.: „**mmcblk0p13**“, „**mmcblk0p23**“, „**mmcblk0p24**“, „**mmcblk0p25**“ ir t.t.), tai užmontavimas – taip pat, kitu atveju, norint išanalizuoti duomenis, tam tikros komandos vykdomos rankiniu būdu.

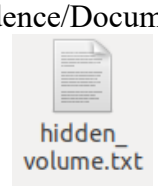
4.2. Duomenų bazės

Dar vienas svarbus momentas, išgaunant bei montuojant atvaizdus – duomenų bazės, esančios atvaizduose. Kaip ir rašyta ankstesniame poskyryje, buvo ištestuotos visos particijos, kiekvienoje jų atkreiptas dėmesys į išgaunamas duomenų bazes. Visų gautų informacijos bazių buvo per mažai, todėl buvo ieškotos papildomos rankiniu būdu, įvedus komandas (jos taip pat automatizuotos): „**adb shell**“, „**su**“ bei ieškant: „**/data/data**“ direktorijose, esančiose mobiliajame įrenginyje. Rasti duomenų bazes padėjo raktiniai žodžiai.

Nepaisant to, kad yra rastas būdas pasiekti bet kokią duomenų bazę - visos jos nebuvo užtektinai užpildytos bei prasmingos analizei, pvz., duomenų bazė: „**contacts2.db**“, nors ir buvo rankiniu būdu suvesti kontaktai į žurnalą, testuotame įrenginyje buvo tuščia, todėl bendroje analizėje ji nebuvo naudojama, nes nebūtų loginės prasmės. Tuščios duomenų bazės priežastis yra netinkamai išsaugota kontaktų informacija. Kitos duomenų bazės atidžiai išanalizuotos bei panaudotos įrankyje taip, kad būtų galima iš to daryti prasmingą teisminę ekspertizę, duodančią

naudingos informacijos tyrimui. Skriptas: „**extract**“ sukūrė logišką aplankalų struktūrą, kuri pavaizduota 15 pav., kad būtų lengviau suprasti.





15 pav. Aplankalų struktūra

Direktorijoje: **„/Evidence/Documents**“ sukelti *.txt formato failai. Išgavus atvaizdą: „**mmcblk0p25.img**“, ištraukiamas failas: „**hidden_volume.txt**“, kuriame yra išvesta užslėpta informacija. **„/Evidence/Photos**“ aplankale sukelti *.jpg formato failai, kurie, kaip pavaizduota 16 pav., išgauti iš atvaizdo: „**mmcblk0p26.img**“.

16 pav. *.jpg failai

Su vaizdo įrašais, atitinkamai, yra tokia pati situacija – į aplankalą **„/Evidence/Videos**“ sukelti failai su *.mp4 formatu.

Aplankalų struktūroje (15 pav.) logine prasme išdėstyta ištraukta bei išanalizuota informacija, kurios tikslas - teisminė ekspertizė. Kadangi kiekvienas atvaizdas savyje turi skirtingą informaciją, tai ir duomenys kataloguose: „**Documents**“, „**Photos**“, „**Videos**“ visada bus skirtingi arba gali išvis jų nebūti.

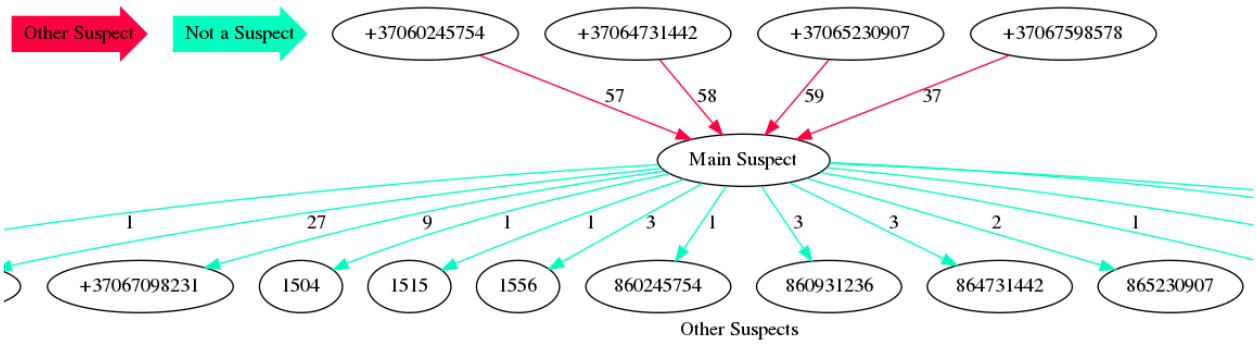
4.3. Duomenų analizė

Kai atvaizdas jau sėkmingai užmontuotas, prasideda pagrindinė testavimo dalis – duomenų analizavimas. Pagrinde rezultatai buvo išvedami grafų (*.png) bei *.txt formatais, kurie pasirinkti kaip vieni aiškiausių. Jeigu atvaizdai bei jose esančios duomenų bazės yra sėkmingai pasiekiamos, tai ir minėti rezultatai bet kokiu atveju bus atvaizduoti, nes tai jau nėra tiesiogiai priklausoma nuo to, ar mobilusis įrenginys pajungtas, ar – ne.

Taip pat ištrauktos particijos, duomenų bazės, skriptai, išvesti rezultatai buvo logiška seka suskirstyti į skirtingus aplankus, siekiant teisminę ekspertizę padaryti kuo aiškesnę bei patogesnę.

4.3.1. Grafas

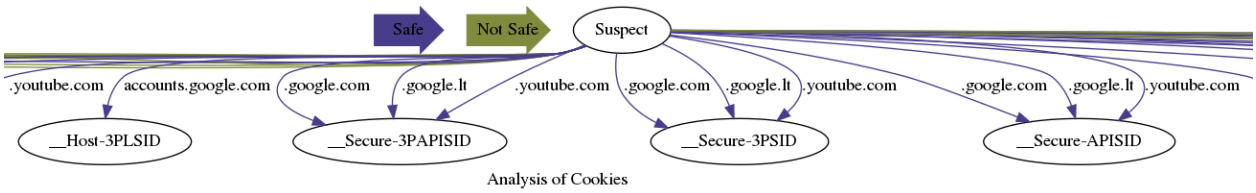
Grafai suteikia galimybę analizuoti duomenis vizualiu būdu. Vienas iš tokių variantų panaudotas skripte: „**SMSCriminals.py**“, kuris sukūrė failą: „**SMSCriminals.png**“ (17 pav.).



17 pav. „SMSCriminals.png“ algoritmo: „SMSCriminals.py“ testavimo rezultatas

17 pav. matome tris semantinių tinklų architektūrinius elementus – semantiką (elementų pavadinimai), struktūrą (elementų hierarchija) ir sintaksę (bendravimas). Paleisto skripto rezultate matome tripletą (angl. „triplet“) – trijų elementų struktūrą, kurią, atsižvelgiant į paimtus duomenis, sudaro mobilieji telefono numeriai, žinučių skaičius bei tipas (kitas įtariamasis ar neįtariamasis). Kadangi sąlyga buvo, kad turi būti parašyta daugiau negu 30 žinučių, nors 17 pav. matomas ne pilnas *.png failas, iš jo aišku, kad yra galimi kiti 4 įtariamieji (šio testavimo atveju). Teisminės ekspertizės metu tai yra svarbus momentas, kuris gali pakeisti iškeltos bylos situaciją.

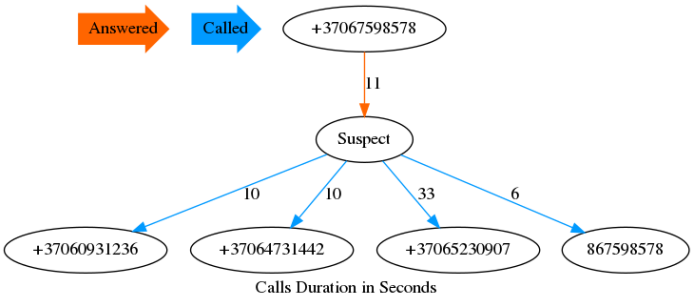
Kitas grafu atvaizduotas rezultatas: „**CookiesAnalysis.png**“ iš skripto: „**CookiesAnalysis.py**“. Matoma tokia informacija – pagrindiniai raktai, „sausainių“ pavadinimai bei saugumas („Safe“, „Not Safe“).



18 pav. „CookiesAnalysis.png“ algoritmo: „CookiesAnalysis.py“ testavimo rezultatas

Rezultatas (18 pav.) taip pat yra matomas ne visas, tačiau minėta informacija išdėstyta taip pat semantine prasme.

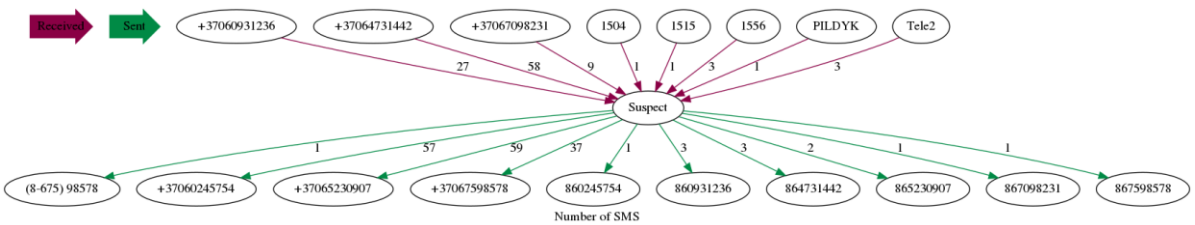
Dar vienas rezultatas: „**CallsAnalysis.png**“, kuris išgautas iš algoritmo: „**CallsAnalysis.py**“, sukūręs 19 pav. rodomą rezultatą.



19 pav. „CallsAnalysis.png“ algoritmo: „CallsAnalysis.py“ testavimo rezultatas

Algoritmas rodo tinklą, kuris nusako, kas ir kiek kartų atsiliepė bei kiek kartų skambino pats įtariamasis.

Ketvirtasis išgautas grafas: „SMSAnalysis.png“ yra iš algoritmo: „SMSAnalysis.py“ (20 pav.).

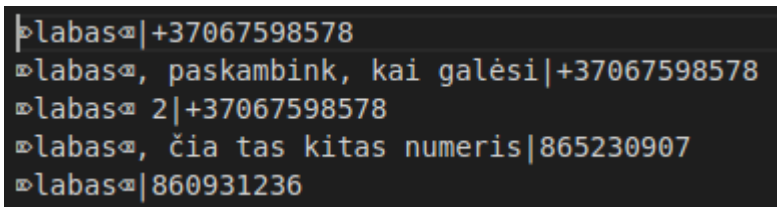


20 pav. „SMSAnalysis.png“ algoritmo: „SMSAnalysis.py“ testavimo rezultatas

Paskutinysis grafas išveda informaciją apie įtariamojo išsiųstas bei jo gautas žinutes iš kitų telefono numerių. Taip pat yra suskaičiuotas žinučių kiekis.

4.3.2. Žinučių filtravimas

Algoritmas: „**markingWords**“, kuris buvo aprašytas anksčiau, sukuria failą: „**filteredSMS.txt**“ (21 pav.).

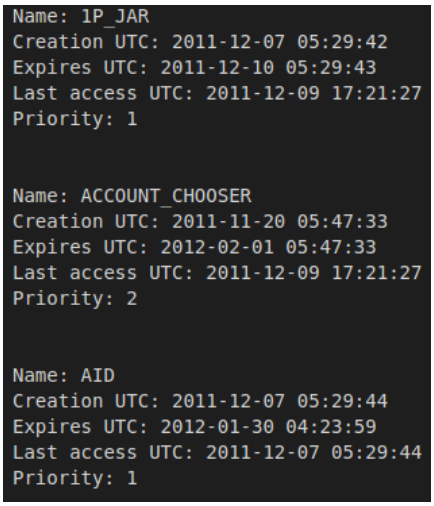


21 pav. Filtruotos žinutės

21 pav. šalia kiekvienos rastos žinutės matomi telefono numeriai, kurie prideda naudingos informacijos, atliekant teisminę ekspertizę. Jeigu norima tai pačiai bylai dar kartą filtruoti žinutes pagal kitus raktinius žodžius – galima paleisti skriptą: „**markingWords**“ atskirai - tame pačiame faile: „**filteredSMS.txt**“ bus išvedamos žinutės pagal visus raktažodžius – ir pagal senuosius, ir pagal naujai įvestus, bei atitinkamai yra atskiriami įvesti raktiniai žodžiai.

4.3.3. Duomenų išvedimas

Kadangi duomenų išvedimas į tekstinius failus šiame darbe buvo vienas dažniausiai naudojamų, tai ir esmė visų yra panaši, tačiau ir vėl norima atkreipti dėmesį būtent į skripto: „**CookiesDate.py**“ sukurtą failą: „**CookiesDate.txt**“, kur algoritmas konvertavo 13 skaitmenų datos formatą į žmogui suprantamą formatą (22 pav.).



22 pav. Konvertuotos datos į žmogui suprantamą formatą

Atkreipiant dėmesį ir į kitus duomenis, šiuo atveju, pvz., pavadinimas: „AID“ yra priskiriamas „sausainiui“: „Google“ („AID“ „sausainis“ skirtas užfiksuoti tiems, kurie lankosi puslapyje, užfiksuoja lokaciją, veiklą bei nuostatas), o informacija: „Priority“ reiškia tam tikro „sausainio“ reikšmingumą (1 yra reikšmingiausias, 2 – mažiau reikšmingas).

4.4. „Hash“

Egzaminuotojas negali koreguoti teisminės ekspertizės metu išgautų duomenų, todėl, kad būtų įrodytas atliktos analizės duomenų vientisumas, naudojamas skriptas: „**MD5.py**“, sukuriantis manifesto failą: „**hashes.txt**“ (23 pav.).

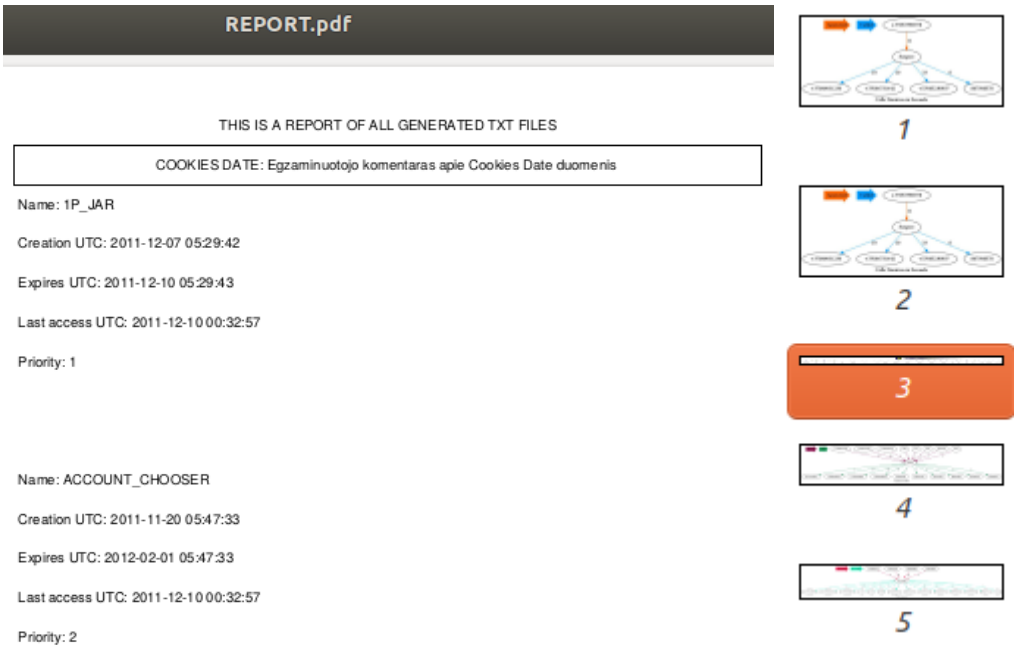
```
<open file '/home/ubuntu/Desktop/Project/reportPDF.sh', mode 'r' at 0x7ff7a0065d20>ae479584b4fc186127466e48473815c3
<open file '/home/ubuntu/Desktop/Project/SMSAnalysis.py', mode 'r' at 0x7ff7a0065db0>027a6120eab087672bf95da8b92ba4d3
<open file '/home/ubuntu/Desktop/Project/URLs.py', mode 'r' at 0x7ff7a0065d20>29e2b906197d00ffdb1239a2df1d3e29
<open file '/home/ubuntu/Desktop/Project/dataPDF.py', mode 'r' at 0x7ff7a0065db0>be1ce9dc5c6937a0564acb8be288672a
<open file '/home/ubuntu/Desktop/Project/Bookmarks.py', mode 'r' at 0x7ff7a0065d20>33ce0fbd7f04d77224c77250a6751a91
<open file '/home/ubuntu/Desktop/Project/MD5.py', mode 'r' at 0x7ff7a0065db0>a41f229361b099061541dcb164ea966c
<open file '/home/ubuntu/Desktop/Project/hashes.txt', mode 'r' at 0x7ff7a0065d20>94afa4508df9c713b8ae125c8037e641
<open file '/home/ubuntu/Desktop/Project/CallsAnalysis.py', mode 'r' at 0x7ff7a0065db0>0dea4449658413952 76c94475a7d2bb2
<open file '/home/ubuntu/Desktop/Project/detach.sh', mode 'r' at 0x7ff7a0065d20>132b588f57131734d347597d44c84c43
<open file '/home/ubuntu/Desktop/Project/wordMarking', mode 'r' at 0x7ff7a0065db0>bdb434fd23013d0e0f728e5ef4ec2cf8
<open file '/home/ubuntu/Desktop/Project/DCIM.py', mode 'r' at 0x7ff7a0065d20>01b84664365f055f2cc63416446ab8df
<open file '/home/ubuntu/Desktop/Project/extract', mode 'r' at 0x7ff7a0065db0>cc75b53f45f03539195e399ebdead9c1
<open file '/home/ubuntu/Desktop/Project/Files.py', mode 'r' at 0x7ff7a0065d20>c5b33af1670511022e665672b6a6ae2c
<open file '/home/ubuntu/Desktop/Project/SMSDate.py', mode 'r' at 0x7ff7a0065db0>05ce7a0b160fd0d82afe910cc14744f1
<open file '/home/ubuntu/Desktop/Project/SMSCriminals.py', mode 'r' at 0x7ff7a0065d20>b8eabdc8a5d61ef98713c9eaba8efaa5
<open file '/home/ubuntu/Desktop/Project/CookiesAnalysis.py', mode 'r' at 0x7ff7a0065db0>04bca8b34417434abbf6012a63804a74
<open file '/home/ubuntu/Desktop/Project/run.sh', mode 'r' at 0x7ff7a0065d20>c6cbe444710115aee16a89ae028a0b9a
<open file '/home/ubuntu/Desktop/Project/CookiesDate.py', mode 'r' at 0x7ff7a0065db0>1344f8cd0538494bf6f8bd1fbdd9d964
<open file '/home/ubuntu/Desktop/Project/Cases/filteredsms.txt', mode 'r' at 0x7ff7a0065d20>8e0272cd1414214ffe6b8b84d2af4826
<open file '/home/ubuntu/Desktop/Project/Cases/CaseReport/examinator.txt', mode 'r' at 0x7ff7a0065db0>416c3674994de23bd2d0fc4b566b68a4
<open file '/home/ubuntu/Desktop/Project/Cases/CaseReport/Evidence/Analysis/reportPDF.sh', mode 'r' at
```

23 pav. „hashes.txt“ turinys

Manifesto faile (23 pav.) matome užšifruotą informaciją – ilgas atsitiktinių simbolių kombinacijas, kurios, kaip minėta anksčiau, yra unikalios kiekvienam užšifruotam failui (taip galima teigti tik praktiškai, nes MD5 maišos kodas yra fiksuoto ilgio, todėl jis negali būti unikalus absoliučiai visoms įmanomoms įvestims, bet ši maišos funkcija suprojektuota taip, kad būtų kuo mažesnė tikimybė susidurti su situacija, kad du atskiri failai turėtų tą pačią maišos vertę [21]). Jeigu failas būtų modifikuojamas, atitinkamai ir manifesto failas turės kitokias atsitiktinių skaičių kombinacijas.

4.5. Ataskaita

Visa bylos tekstinė informacija („**filteredSMS.txt**“ bei „**hashes.txt**“ nėra išvedami į bendrą ataskaitą, nes pritrūko resursų) yra išvedama į bendrą ataskaitos dokumentą: „**REPORT.pdf**“, kuriame matomi egzaminuotojo įvesti komentarai apie kelis duomenų failus bei likusi tekstinė informacija (24 pav. kairėje).



24 pav. Ištraukos iš sugeneruotų ataskaitų

Dauguma rastų ataskaitos pavyzdžių yra panašaus principo, kuris ir panaudotas šiame darbe, nes, nepaisant duomenų aiškios struktūros svarbos, pagrindinis momentas – duomenys, kuriuos svarbu aiškiai išdėstyti, neapkraunant kitais niuansais [11].

Atitinkamai įrankis sugeneruoja ir antrą ataskaitą: „**Graphs.pdf**“, kurioje yra išvedami įrankio sugeneruoti grafai (24 pav. dešinėje). Šioje ataskaitoje nėra papildomų komentarų, nes grafai yra vizualus teisminės ekspertizės rezultatas, kurie patys iš savęs suteikia reikalingą informaciją.

4.6. Apibendrinimas

Apibendrinus visą testavimą, tiek atvaizdo išgavimą, jo montavimą, tiek informacijos analizavimą bei kitas funkcijas, buvo testuota pakankamai kartų, kol buvo pasiektas norimas rezultatas, kuris atitiktų teisminės ekspertizės logiką. Kadangi šis darbas, kaip ir minėta anksčiau, testuotas tik su vienu mobiliuoju įrenginiu – tikėtina, kad skirtinguose „Android“ įrenginiuose bei jų modeliuose gali skirtis, pvz., duomenų bazių pavadinimai, todėl gali atsirasti tam tikrų klaidų testuojant visą įrankį.

Nepaisant to, kad informacija gali skirtis bei būti kitaip pateikiama, šio darbo testavimas su minėta įranga atitinka iššikeltus rezultatus, kurie yra tinkami teisminei ekspertizei.

Išvados ir rekomendacijos

Pagrindinis šio darbo tikslas buvo sukurti kuo universalesnį, dinamiškesnį įrankį, skirtą teisminei mobiliųjų įrenginių ekspertizei, todėl suprantama, kad darbas pradėtas nuo atskirų veiksmų, pavienių komandų, kurios galiausiai sudarė automatizuoto įrankio visumą. Žinoma, kuriant įrankį, keblumų kilo jau pačioje pradžioje, pvz., pradėjus aiškintis, kaip prijungti mobilųjį įrenginį, kaip įvykdyti tam tikras funkcijas, pasitelkiant pavienes komandas, bet būtent tai ir yra aiškinimosi procesas, siekiant išspręsti kilusias problemas, norint įgyvendinti visą numatytą funkcionalumą bei, svarbiausia, jį automatizuojant. Teisminės ekspertizės įrankis gali turėti galybę įvairiausių funkcijų, kaip ir aprašyta šiame darbe, tačiau šis įrankis parengtas, atsižvelgiant į reikalingiausias funkcijas bei logiškiausią duomenų analizę, kad nebūtų beprasmiškų duomenų bei jų analizės. Kadangi tikslas įvykdytas, svarbu paminėti, kad įrankis veikia iš tiesų labai greitai, kas reiškia, kad, nepaisant laiko, kuris reikalingas atvaizdui išgauti, teisminė ekspertizė su reikalingiausiomis funkcijomis bei duomenų analize yra pasiekiama neužtrunkant daug laiko.

Kaip ir minėta anksčiau, įrankis sukurtas, testuojant komandą po komandos, ieškant logiškiausio bei aiškiausio varianto galutiniam variantui, kuris pateisintų lūkesčius. Nepaisant to, jog buvo daug koregavimų bei kad ne visos problemos buvo išspręstos taip kaip buvo užsibrėžta iš pradžių, visiems atvejams yra rasti alternatyvūs sprendimai, todėl galutinis tikslas, numačius jį prieš pradedant rengti darbą, išpildytas.

Išvada trumpai – problemų sprendimo dėka, su kuriomis buvo susiduriama nuo kūrimo pradžios,
pavyko įgyvendinti dinamišką įrankį, skirtą teisminei ekspertizei, kuris atitiktų pradinius lūkesčius.

Rekomendacija - turėti tikrai daugiau negu vieną mobilųjį įrenginį, kuriam
būtų galima suteikti „root“ teises. Tam, kad būtų galima aptikti kuo daugiau galimų klaidų bei jas išspręsti, kad įrankis būtų kuo labiau universalesnis, reikėtų skirtingų modelių mobiliųjų telefonų. Su vienu mobiliuoju įrenginiu, kaip daryta šiame darbe, parengti bei testuoti įrankį tikrai nėra patikimiausias ir protingiausias variantas, tačiau, nepaisant prastos resursų situacijos, įrankis vis tiek buvo kuriamas atsižvelgiant į pagrindinį tikslą – parengti kuo dinamiškesnį įrankį.

Ateities tyrimų planas

Suvokiant faktą, kad technologijas visada įmanoma tobulinti (kas ir yra įprastai daroma), šį įrankį, skirtą teisminei mobiliųjų įrenginių ekspertizei, taip pat galima pagerinti daugelyje sričių.

Nepaisant to, kad įrankis yra sukurtas, siekiant įgyvendinti pagrindinį tikslą - dinamiškumą, jį galima puoselėti bei daryti įrankį dar labiau universalesnį.

Šiuo metu darbo įrankis yra paremtas skriptais bei juose esančiais algoritmais, kuriuose įprastai yra nustatomos tam tikros sąlygos, pvz., SMS žinučių analizės metu, siekiant išsiaiškinti, ar galbūt yra kitų įtariamųjų šalia pagrindinio nusikaltėlio, yra sukuriamas semantinis tinklas, kuriame matome tuos mobiliuosius telefonų numerius, iš kurių buvo išsiųsta daugiau žinučių negu tam tikras fiksuotas skaičius minėtame algoritme. Šioje vietoje įrankio dinamiškumą būtų galima patobulinti, leidžiant pačiam įrankio naudotojui įvesti tam tikrą fiksuotą žinučių skaičių ar net jų amplitudę. Būtų dar geriau, jeigu būtų įgyvendintas skriptas, kuris turėtų algoritmą, kuris pats apskaičiuotų ir nustatytų tam tikrą SMS žinučių skaičių, kurių kiekis jau galėtų kelti įtarimą dėl kitų galimų įtariamųjų. Panašų algoritmą būtų galima įgyvendinti taip pat ir su skambučių žurnalo istorija, naršyklės istorija ir pan.

Dar daugiau dinamiškumo sukurtam įrankiui galėtų pridėti galimybė tyrėjui pasirinkti ir ištraukti papildomus reikalingus duomenis, nors duomenis turinčios duomenų bazės iš tiesų ir būtų labai minimalios ar net tuščios. Duomenų nebuvimas – taip pat svarbi analizės dalis, nes galbūt būtų galima įtarti, kad tie duomenys yra paslėpti/panaikinti, kuriuos norėta pasiekti.

Dirbtinio intelekto panaudojimas įrankyje - papildoma galimybė, kuri taip pat pagerintų įrankio kokybę. Kaip pavyzdys, DI galėtų būti naudojamas žinučių atpažinimui bei atfiltravimui pagal dirbtinį intelektą. Paprastai tariant, tyrėjui įvedus raktinius žodžius, pagal kuriuos norima filtruoti žinutes, dirbtinis intelektas galėtų rasti žinutes ne tik pagal konkrečiai įvestus raktažodžius, bet ir papildyti žinutėmis, kurios turi panašios prasmės žodžius. Dar aukštesnio lygio įrankio pagerinimas, pasitelkiant DI, galėtų būti toks, kad minėta technologija pati, analizuodama žinutes, iš konteksto atpažintų pagal algoritmą, kurios žinutės galėtų pasitarnauti kaip turinčios įkalčių ir/ar kitos naudingos informacijos. Panašus principas taip pat galėtų būti taikomas ir atpažįstant informaciją iš pokalbių, jeigu būtų įmanoma pasiekti pačius pokalbius, bet tam, žinoma, taip pat reikėtų išsamesnio tyrimo, kurio metu būtų išsiaiškinta, kaip įvykdyti balso atpažinimą bei pasiekti, analizuoti pokalbius.

Trumpai tariant – atlikus darbą pagal išsikeltus tikslus, įprastai atsiranda idėjų, kaip galima tobulinti jau sukurtą darbą, kuris jau atlieka savo numatytas funkcijas.

Literatūros šaltiniai

[1] A. Hoog. *Android Forensics. Investigation, Analysis, and Mobile Security for Google Android.*

[2] A. Recon. CAINE. *URL:* https://www.caine-live.net/.

[3] A. Tabona. Top 20 Free Digital Forensic Investigation Tools for SysAdmins – 2019 Update. *URL: https://techtalk.gfi.com/top-20-free-digital-forensic-investigation-tools-for-sysadmins/*. 2019.

[4] Andriller. *URL:* https://www.andriller.com/download/. 2019.

[5] Android version history. *URL: https://en.wikipedia.org/wiki/Android_version_history*.

[6] Angel A. Parrizas. *Forensic Analysis On Android: A Practical Case*. 2015.

[7] Angel A. Parrizas. *Monitoring network Traffic for Android Devices*. 2013.

[8] B. Carrier. *File System Forensic Analysis.*

[9] B. Carrier. The Sleuth Kit. *URL:* https://www.sleuthkit.org/sleuthkit/. 2003-2019.

[10] D. Forte. *Dealing with Forensic Software Vulnerabilities: is Anti-forensics a Real Danger?* 2008.

[11] Forensic Case Report Examples. *URL: https://ncssm.instructure.com/courses/1674/pages/forensic-case-report-examples.* 2016-2017.

[12] Graphviz. *URL: https://www.graphviz.org/.*

[13] H. Baier, F. Breitinger. *Security Aspects of Piecewise Hashing in Computer Forensics.* 2011.

[14] How to Forensically Examine an Android Device with AFlogical OSE on Santoku Linux. *URL: http://digital-forensics-science.blogspot.com/2015/08/how-to-forensically-examine-android.html*. 2019.

[15] HOWTO: Use AFLogical OSE for Logical Forensics of an Android Device. *URL: https://santoku-linux.com/howto/howto-use-aflogical-ose-logical-forensics-android/.*

[16] INFOSEC. Practical Android Phone Forensics. *URL: https://resources.infosecinstitute.com/practical-android-phone-forensics/#gref*. 2017.

[17] J. Lessard, Gary. C. Kessler. *Android Forensics: Simplifying Cell Phone Examinations*. 2010.

[18] M. Alazab, S. Venkatraman, P. Watters. *Effective Digital Forensic Analysis of the NTFS Disk Image.* 2009.

[19] M. Al-Hadadi, A.AlShidhani. *Smartphone Forensics Analysis: A Case Study*. 2013.

[20] M. Lohrum. Obtaining All Files in the Data Partition Without a Physical Image. *URL: http://freeandroidforensics.blogspot.com/.*

[21] MD5. *URL: https://en.wikipedia.org/wiki/MD5*. 2020.

[22] MOBILedit. *URL: https://www.mobiledit.com/forensic-express*.

[23] MSAB. The Need for Speed in Mobile Forensics. *URL: https://www.msab.com/2019/12/13/the-need-for-speed-in-mobile-forensics/.*

[24] M-Trends 2020. Fireeye Mandiant Services. Special Report. 2020.

[25] N. Santos. *Mobile Forensics: Android. Part II.A. Techniques and Tools: Computer Forensics.* 2015.

[26] NowSecure. List of Mobile Incident Response Tools. *URL: https://books.nowsecure.com/mobile-incident-response/en/tools/open-source-ir-tools.html.*

[27] Oxygen Forensic Detective Features. *URL: https://www.oxygen-forensic.com/en/products/oxygen-forensic-detective.*

[28] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, Manoj S. Gaur, M. Conti, M. Rajarajan. *Android Security: A Survey of Issues, Malware Penetration, and Defenses*. 2014.

[29] R. Ahmed, Rajiv V. Dharaskar. *Mobile Forensics: An Introduction from Indian Law Enforcement Perspective*. 2009.

[30] Ricci S. C. Ieong. *FORZA – Digital Forensics Investigation Framework That Incorporate Legal Issues.* 2006.

[31] S. Bommisetty, R. Tamma, H. Mahalik. *Practical Mobile Forensics*. Packt Publishing – ebooks Account, 2014.

[32] S. Valle. *Android Forensics and Security Testing*. 2012.

[33] Semantic Network. *URL: https://en.wikipedia.org/wiki/Semantic_network*.

[34] Simson L. Garfinkel. *Digital Forensics Research: The Next 10 Years*. 2010.

[35] Sumuri. *URL: https://sumuri.com/software/paladin/*.

[36] Uptodown. *URL: https://kingo-root.en.uptodown.com/android/versions*.

[37] V. Vijayan. *Android Forensic Capability and Evaluation of Extraction Tools*. 2012.

[38] X. Lin, T. Chen, T. Zhu, K. Yang, F. Wei. *Automated Forensic Analysis of Mobile Applications on Android Devices.* 2018.

[39] XDA Developers. Making Dump Files Out of Android Device Partitions. *URL: https://forum.xda-developers.com/showthread.php?t=2450045.*

[40] XML Schema for Digital Forensics XML. *URL: https://github.com/dfxml-working-group/dfxml_schema*.