UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET

Katedra za računarstvo

Forenzika mobilnih uređaja:

Implementacija alata zasnovanog na ADB alat za forenziku android uređaja

-Digitalna forenzika-

Mentor: prof. dr. Bratislav Predić

Student: Natalija Stamenković, 1258

Niš, 2021.

Sadržaj

1. Uvod

U Prethodnoj dekadi, mobilna tehnologija je doživela enorman rast. Ljudi ih sve više koriste u sledećim sverama: komercijalnim, finansijskim, socijalnim (SMS, MMS, video pozivi, fotografije, mejlovi, veb pretraživanje, pravljenje multimedijalnih zapisa, editovanje video snimaka, dokumentacije, upravljanje i čuvanje ličnih podatala itd..) . Ovaj rast i sve-prisutnost nije dotakao samo korisnike, takođe su zaživeli kompjuterski hakeri, i sajber kriminal je postao sve češći. Takođe broj ukradenih telefona se povećao. Kao što telefoni sadrže osetljive lične podatke, rizično je ako bi se oni ukrali ili izgubili. Forenzička analiza ovih telefona omogućava programerima i korisnicima da postanu svesniji kako i koje podatke da čuvaju na svojim mobilnim uređajima. Čuvanjem ličnih podataka, napadači ih mogu ukrasti i koristiti za lažiranje pravog indetiteta osobe. Takođe kako su mobilni telefoni postali deo svakodnevnog života, tako da se mogu koristiti za izvor enomrne količine podataka forenzičkim istražiteljima u slučajevim istraživanja kriminala. Istražitelji mogu oporaviti obrisane podatke i potencijalno detektovati kriminalne radnje na uređaju. Iz ovih razloga razvila se grana digitalne forenzike – forenzika mobilnih uređaja. Sama forenzika mobilnih uređaja predstavlja granu digitalne forenzike. I odnosi se na oporavak podataka sa mobilnih uređaja i pribavljanje digitalnih dokaza sa istih. Nauka je u stalnom porastu što je prouzrokovano time da se na tržistu stalno pojavljuju novi modeli mobilnih telefona koje postojeći alati ne mogu obraditi.

Mobilna forenzika se najviše razvila i sledećih razloga:

- Korišćenje mobilnih telefona za skladištenje ličnih podataka i korporacijskih informacija
- Korišćenje telefona u onlajn transakcijama
- Kršenje zakona i razvoj kriminala

U ovom radu navedeni su koraci u digitalnoj forenzici mobilnih telefona. Ovi koraci su prikupljanje dokaza (očuvanje istih), nabavka dokaza iz telefona uz očuvanje integriteta podataka, i kao poslednja faza je analiza prikupljenih podataka i sastavljanje izveštaja. Rad se najviše bavi tehnikom prikupljanje podataka sa telefona. Podaci se mogu čitati iz interne ili eksternih memorija telefona. Obrađena je struktura memorija i patricije koje obično telefon sadrži. U radu je najviše pažnje posvećeno ADB alatu koji predstavlja alat za pristup mobilnom telefonu preko računara. Obrađene su neke njegove funkcionalnosti, kao što su pristup *shell-u*, pristup fajl sistemu kao i pristup provajderima sadržaja koji će biti prikazani i kroz praktičnu implementaciju .

U ovom radu kao krajnji biće implementiran jedan alat za ekstrakciju podataka sa android telefona koji funcioniše korišćenjem ADB alata. ADB (*Android debug bridge*) predstavlja klijent-server program koji omogućava prisstup telefonu preko PC-a. Kao što je već ranije pomenuto, aplikacija implementirana u *C#* programskom jeziku kao *WinForms* aplikacija. Korišćena je *AdbSharpClient* biblioteke za korišćenje ADB alata.

2. Forenzika mobilnih telefona

Kao što smo pomenuli već, forenzika mobilnih uređaja je proces analize mobilnog telefona u cilju detektovanja kriminala i kriminalnih aktivnosti koje su izvšenje uz pomoć mobilnih uređaja. Glavni fokus istraživanja kriminala je analiziranje mobilnih internih i eksternih skladišta podataka.

Postoje glavni koraci u procesu digitalne forenzike mobilnih uređaja:

- Zaplena
- Pribavljanje podataka (Ekstrakcija) *(eng. Acquistion)*
- Analiza i prijava

2.1. Zaplena

U krivičnimm predmetima, osoblje za sprovođenje(tehničari) su dužni da obezbede očuvanje dokaza. Dok će u građanskim parnicama obično to radi neki neobučeni radnik. Zakonom je dozvoljena zaplena mobilnih telefona, kao i izdavanje naloga za pretres i zaplenu telefona. U civilne svrhe, kompanija može istražiti svoju opremu bez upozorenja sve dok poštuje ljudska prava i prava zaposlenih. U krivičnim postupcima, pravo na pretres se primenjuje.

Digitalna forenzika funkcioniše tako što dokaz treba biti adekvatno zaštićen, i procesiran i priznat na sudu. Prvo je potrebno konfiskovati mobilne uređaje. Zatim se mobilni uređaj stavlja u faradejevu kutiju, ova kutija i eksterni punjač su glavni tipovi opreme u mobilnoj forenzici. Kutija je namenjena za izolaciju mobilnog uređaja od mrežne komunikacije, takođe pomaže bezbednom transportu telefona do laboratorije. Pre postavljanja telefona u faradejevu kutiju potrebno je diskonektovati telefon sa mreže, isključiti sve mrežne konekcije (WIFI, hotspot, IR itd..) i aktivirati režim letenja u avionu. U svrhu zaštite integriteta dokaza.

Ova faza je veoma bitna u procesu mobilne forentike i glavni cilj je očuvanje dokaza. Ukoliko se ne obavi kako treba dovodi do neuspeha ostalih faza u procesu forenzike (1).

2.2. Pribavljanje podataka

Nakon što je uređaj zaplenjen, spreman je za prikupljanje podataka. Mobilni podaci ili digitalni dokazi mogu biti smešteni u internoj ili eksternoj memoriji na telefonu. Interna ili unutrašnja memorija se obično čuva na samom telefonu, dok se spoljašnja memorija obično čuva na SD karticama ili memorijskim karticama. Ovi podaci mogu biti GPS informacije, podaci sa socijalnih mreža, istorija pretraživanja, poruke, slike, lokacije, mejlovi, beleške itd. Takođe osim što se podaci mogu naći internoj i eksternoj memoriji. Forenzika podataka zavisi i od nivoa prsitupa koji uređaj pruža. *Android* operativni sistem pruža dva sloja kontrole pristupa korisnika koji su *root-ovani* ili *ne-root* pristup. Prvenstveno Android OS ne pruža korisnicima administrativni ili root pristup, pa se uređaji proizvode sa pristupom koji nije *root* [2]. *Root-ovan* uređaj omogućava potpuno izdvajanje korisničkih podataka i pristup sistemskoj particiji. Sistemska particija čuva kompletne podatke aplikacije, ROM i sistemske datoteke. Za korisnika bez *root-a*, particije i

sistemske fascikle ostaju skrivene bez pristupa (2). U ovom radu će biti kansije implementrian *ne-rootovan* pristup.

Već smo pomenuli podelu koja zavisi od vrste pristupa, takođe pribavljanje podataka se može podiliti na način dolaska do tih podataka sa telefona. Postoje četiri glavna oblika pribavljanja (*acquistion*) podataka [1]:

- Ručna nabaka
- Logička nabavka
- Fizička nabavka
- Gruba nabavka

2.2.1. Ručna(Direktna ekstrakcija) podataka

Ručna (Direktna) akvizicija – u ovoj tehnici forenzički analitičar koristi interfejs mobilnog uređaja za istraživanje dostupnog sadržaja. Da bi ova tehnika mogla da se primeni, potrebno je da analitičar telefona zna pin ili šifru telefona koji istražuje. Prilikom ispitivanja uređaja analitičar pravi slike ekrana gde se nalaze traženi podaci. Prednost ove tehnike je da ne zahteva nikakve alate za prikupljanje podataka, takođe nedostatak je da su podaci vidljiivi korisniku na uređaju, ova tehnika oduzima mnogo vremena. Sistemski fajlovi, logovi i patricije ovom tehnikom nije moguće ispitati [1].

2.2.2. Fizička ekstrakcija podataka

Zasniva se na kloniranju podataka sa mobilnog telefona. Kloniranje uključuje i izbrisane podatke i nealocirani memorijski prostor. Ovo kopiranje je bit po bit (*bit-by-bit*). Nakon kloniranja, vrši se inspekcija podataka korišćenjem razno-raznih alata za tu namenu (npr. *FTK image sofware*). Nakon završetka ove ekstrakcije ne ostaju tragovi.

Mogu se razlikovati dve faze:

- *Dumping* - podaci iz memorije se čuvaju u heksadecimalan fajl format.
- *Decoding* - tokom ove faze podaci se konvertuju u čitljiv format.

2.2.3. Logička ekstrakcija podataka

Ova metoda omogućava pristup ne-rootovanim uređajima. Logička nabavka predstavalja nabavljanje sadržaja telefona sa logičkih particija, gde ne spadaju nealocirane memorije telefona. Ovom metodom se podaci sa telefona dobijajaju korišćenjem automatskih alata za sinhronizaciu podataka telefona sa računarom. Može se kopirati skladište podataka, kao i sistemske particije. Na kraju se dobija fajl koji može biti analiziran raznim alatima za forenzičku analizu. *Backup* slika telefona može se smatrati na primer kao logička slika telefona. Primer ovakve forenzike je *ADB* alat. *[1]*.

2.2.4. Nasilna ekstrakcija podataka

Ovaj metod pokušaja i grešaka se koristi za pogađanje odgovarajuće kombinacije šifre i pina. Šalju se serije pinova od 0000 do 9999 dok se ne pogodi odgovarajuća kombinacija na uređaju.

2.2.3. Analiza

Nakon prikupljanja sadržaja fajlova, sadržaji su analizirani u cilju identifikacije dokaza. Tokom ove analize istražitelj obično koristi različite alate (*FTK*) obično počevši od oporavka izbrisanog materijala. Pregledaju se mejlovi, chat logovi, slike, istorija pretraživanja i dokumenti, dokumenti. Podaci mogu biti povraćeni sa pristupačnih lokacija na disku, izbrisanih ili u keš fajlovima operativnog sistema. U slučajevima zlostavljanja dece, na primer, bitna je samo istorija veb pretrage i slike. Kada gledamo samo određene kategorije, u mogućnosti smo da filtriramo druge stvari koje ne moramo da gledamo. Međutim, u velikim slučajevima, gde je mnogo različitih kategorija podataka potencijalno od interesa – konverzacije, slike, kontakti – proces može biti mnogo intenzivniji. Da bi se analizirali takvi podaci potrebno je možda koristiti više tehničkih rešenja. Analitičar mora biti obučen za korišćenje više alata i povezivanje i analizu dokaza.

3. Organizacija eksterne i interne memorije

Android uređaji koriste nekoliko particija da bi organizovali fajlove i foldere na uređaju. Svaka od ovih particija ima drugačiju ulogu u funkcionisanju android uređaja. U nastavku ovog rada vršiće se ekstrakcija podataka sa */data* particije. Dok ćemo u nastavku ovog poglavlja ukratko objasniti čemu su ostale particije namenjene [6].

Standardne particije na uređaju se mogu podeliti u sledeće kategorije:

- interne
- eksterne

3.1. Interna memorija

/boot - ova particija omogućava telefonu da se *boot-uje*, obuhvata dve komponente kernel i ramdisk. Bez ove particije telefon ne bi mogao da se *boot-uje.* Ukoliko se ova particija obriše iz *recovery* moda, i nakon toga uređaj ne sme biti *reboot-ovan* pre instaliranja nove. Nova može biti instalirana insaliranjem *ROM-a* koji uključuje *boot* particiju.

/system - particija sadrži ceo operativni sistem izuzev komponenti koje sadrži *boot* particija, sadrži *UI (user inferface)* kao i sve sistemske aplikacije koje su pre-instalirane na uređaju. Brisanjem ove particije bi se izbrisao Android sa telefona, ali je i dalje moguće ubaciti telefon u *recovery* mod i instalirati novi ROM.

/recovery - može se smatrati još jednom *boot* particijom koja uvodi telefon u *recovery* mode.

/cache - particija skladišti često korišćene korisničke podatke i podatke iz aplikacija, brisanjem ove particije nema efekata i podaci se opet skladište ukoliko se aplikacije opet koriste.

/*data* - takođe se naziva i *userdata* particija, ova particija sadrži korisničke podatke kao što su kontakti, podešavanja, poruke, aplikacije. Brisanjem ove particije telefon se vraća na fabričko podešavanje, vraćajući ga u stanje u kom je bio kad je prvi put *boot-ovan.* Kada se podešavanjima telefona pritisne vraćanje na fabričko podešavanje zapravo se tom akcijom briše ova particija.

/*misc* - particija koja sadrži konfiguraciona podešavanja u formi *off-on*. Ovo je bitna particija i ukoliko dođe do oštećenja neki glavni fičeri neće funkcionisati kako treba.regled

3.2. Eksterna memorija

/sdcard - eksterna particija, bezbedno je izbrisati, ukoliko postoji negde *backup* podataka.

4. Android Debug Bridge (ADB)

ADB je *command-line* alat koji omogućava komunikaciju sa android uređajem. Mogućnosti koje pruža su instalacija i debagovanje aplikacija. Takođe, omogućava pristup *Unix shell*-u koji omogućava pokretanje raznoraznih komandi na uređaju [5].

- Listanje konektovanih uređaja
- Instaliranje i debagovanje aplikacija
- Kopiranje fajlova sa telefona
- Kopiranje fajlova na telefon
- Pravljenje snimaka ekrana, slika, snimanje zvuka ...

Ovaj alat je moguće koristiti u digitalnoj forenzici mobilnih uređaja. ADB je klijent-server program koji uključuje tri sledeće komponente:

- Klijent - ova komponenta omogućava slanje komandi. Klijent se pokreće na host mašini za razvoj. Može se pokrenuti klijent u *command-line* terminalu, tako što će se pokrenuti ABD komanda.
- ADB *deamon* - Komponenta koja pokreće komande na uređaju, on se pokreće kao proces u pozadini na svakom uređaju
- Server - Komponenta koji upravlja komunikacijom između klijenata i ADB demona. Server se takođe pokreće u pozadini na mašini za razvoj. Kao što se može videti na slici, serverski proces se pokreće na host mašini. Može se videti da li je uređaj prikačen ili ne, ili kad emulator se pokrene ili stopira. Takođe određuje stanje svakog od njih.
- Servisi -
    - Host servisi - ovi servisi se pokreću na ADB serveru i ne moraju da komuniciraju sa uređajem uopšte. Tipičan primer je adb devices koji se koristie da vrate listu trenutno poznatih uređaja i nj ihova stanje.
    - Lokalni servisi - ovi servisi se pokreću sa adb demonom ili su startovani od strane adb demona. Adb server se koristi za strimovanje između klijenta i servisa pokrenutih od stane adb demona. U ovom slučaju uloga servera je iniciranje konekcije, a zatim i razmenu podataka između njih.

4.1. Način rada ADB

Kada se pokrene ADB klijent, klijent prvo proverava da li postoji već pokrenut serverski proces. Ukoliko ne postoji klijent ga pokreće. Kada se server startuje on se pokreće na lokalnom TCP portu s brojem 5037, i osluškuje komande poslate od strane ADB klijenata, svi ADB klijenti koriste komunikaciju sa ABD serverom na portu 5037.

Server, zatim, postavlja konekcije sa svim pokrenutim uređajima. Locira emulator skenirajući neparne portove u rangu 5555 – 5585, ovi brojevi se koriste za prve 16 emulatore.

Zatim server pronalazi ADB *deamon* (Adbd), postavlja konekciju na portu. Svaki emulator koristi par sekvenci portova (5).

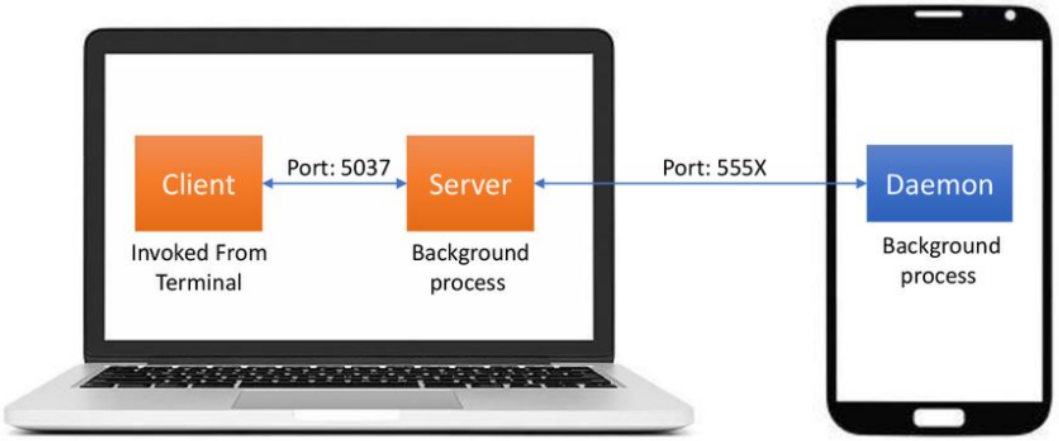
- Parni port za konzolnu konekciju (Konzola emulatora osluškuje na ovom portu)
- Neparni port za adb konekciju

Kada se postave konekcije ka svim uređajima, mogu se koristiti ADB komande za pristup ovim uređajima. Server upravlja konekcijama ka uređajima i upravlja komandama koje šalju više ADB klijenata.

Pre svega je potrebno omogućiti debagovanje na uređaju. Potrebno je uključiti USB *debugging* na uređaju u opcijama za razvoj na uređaju. Da bi se videle opcije za razvoj potrebno je kliknuti na *Build number* sedam puta.

Neke od ADB komandi su sledeće:

- Opcije - adb -d, -e, -s
- Generalne – adb devices (gore pomenta)
- Debug –adb logcat, adb jdwp
- Podaci – adb install, adb push, adb pull
- Portovi i mreže -forward <local> <remote>
- Scripting - adb wait-for-device
- Server – adb start-server, adb kill-server
- Shell - više u nastavku



lustracija konekcije sa ADB

4.2. ADB komande

Neke od bitnih komandi koje biće objašnjenje u sekciiji ispod. Neke su samo opisno objašnjenje, dok će se rezultat nekih videti kroz praktičnu implementaciju u poglavljima koje slede.

- *ADB Install* - se koristi za instaliranje programa na android uređaju. Potrebno je zadati ime željene aplikacije sa apk ekstenizijom *(.apk)* i navesti tačnu putanju. U suprotnom, neće biti uspešno izvršeno. Ukoliko se doda *-r* pre *install*, aplikacija će biti reinstalirana. Dodavanjem *-s* aplikacija će biti instalirala na memorijsoj kartici. Potrebno je proveriri da li je memorijska kartica ubačena na uređaj.
    - *adb install <lokacija_apk_fajla>*

- *ADB Unistall* - ova komanda se koristi za uklanjanje nekorišćenih servisa i aplikacija. Procedura se poklapa sa komandom *install*, koja je objašnjena u poglavlju iznad ovog. Slovo k napisano ispred imena aplikacije će sačuvati korisničke naloge i korisničke podatke brisanjem samo programa.
    - *adb unistall <lokacija_apk_fajla>*

- *ADB Push* - koristi se pomeranje fajlova bez instalacija. Koristi se transfer slika, muzike, video zapisa na uređaj. Bitno je napomenuti da se mora naglasiti putanja fajla i mesto gde želimo prebaciti fajl. Takođe omogućava i prebacivanje celih foldera na uređaj.
    - *adb push <source> <destination>*

- *ADB Pull* - omogućava transfer foldera i fajlova sa telefona na konektovan kompjuter ili laptop. Takođe programer mora navesti tačne putanje sa koje i gde želimo prebaciti folder ili fajl. U suprotnom fajl će se kopirati u folder gde se nalazi *adb tool.*
    - *adb pull <source> <destionation>*

- *ADB Reboo*t - omogućava restartovanje. Koristi se obično da bi prethodno napravljene promene imale efekte. Tako omogućava omogućavanje alternativnih *debbug* modova i *firmware* modova. Ukoliko se u nastavku specificira parametar i ime odgovorajućeg moda.
  Jedan od modova je *recovery* mode. U koji se ulazi komandom *adb reboot recovery.* Ova komanda pokreće uređan na specijalnoj bootabilnoj particiji. Koristi se za ispravljanje grešaka u radu uređaja.
  Takođe postoji *adb reboot bootloader, adb reboot fastboot*

Unosom *adb –help* može se videti spisak svih dostupnih komandi sa parametrima.

4.3. ADB shell

Unosom komande *adb shell* pokreće se *shell* na uređaju. Korišćenjem *adb shell* komandi koje mogu da *reboot-uju* uređaj, preuzmu i kreiraju *backup* i restoruju podatke moguće ostvariti

značajan napredak u digitalnoj forenzici i otkrivanju kriminala.

ADB shell komande rade na mnogo dubljem nivou od običnih komandi. Mogu se koristiti da promene rezoluciju ekrana, deinstaliranju sistemskih aplikacija, omogućavanju i onemogućavanju fičera. Takođe mogu modifikovati sistemske fajlove, i promene njihovu konfiguraciju direktno korišćenjem komadi sa kompjutera. U stvari, pružaju mnogo više mogućnosti od *adb* komandi. *Adb shell* komande omogućavaju kontrolisanje uređaja putem kompjutera. Otvaranje novih prozora, slanje SMS poruka, itd.

```
C:\platform-tools>adb shell am start -a android.intent.action.VIEW
Starting: Intent { act=android.intent.action.VIEW }
```

Primer komande za upravljanje telefonom

Primer komandi:

- *screencap <naziv_fajla>* - screenshot ekrana
- *screenrecord <naziv_fajla>* - snimanje ekrana
- *wm size* - Prikaz informacija o telefonu, rezolucija ekrana.. itd
- *pm* - manipulacija *package manager-om* (aplikacionim paketima instaliranim na uređaju)

Shell komanda koja će biti korišćena u ovom radu je *content*. Ova komanda izvršava direktno Content Provider iz shell-a.

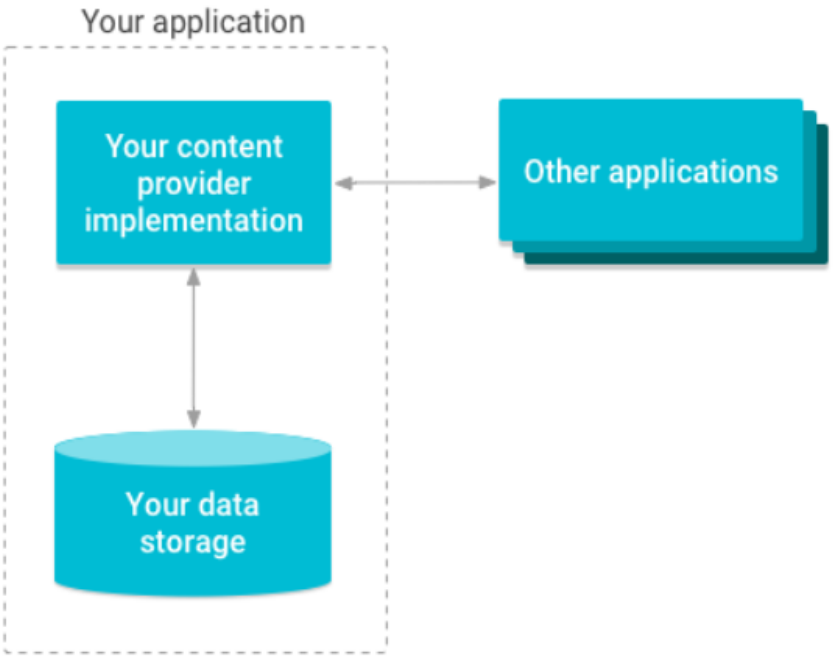
adb shell content query --uri <URI> [--projection <PROJECTION>] [--where <WHERE>] [--sort <SORT_ORDER>]

Biće detaljnije objašnjena u narednoj sekciji.

4.4. Provajderi sadržaja

Provajderi sadržaja omogućavaju aplikaciji da upravlja svojim skladištem podataka, skladištenog od sebe ili drugih aplikacija. Takođe omogućavaju deljenje podataka sa drugim aplikacijama što znači da su provajderi standardni interfejsi koji konektuju podatke između dva procesa.

Provajderi sadržaja enkapsuliraju podatke i pruža mehanizam za definisanje bezbednosti. Odnosno, provajder sadržaja omogućava drugim aplikacijama bezbedan pristup i modifikaciju aplikacionih podataka. Provajdere sadržaja treba koristiti ukoliko je planirano deljenje podataka. Ukoliko, ipak i tad treba koristiti zato što pružaju apstrakciju. Ova apstrakcija omogućava pravljenje modifikacija nad aplikacionim podacima bez uticaja na druge postojeće aplikacije koje se oslanjaju na pristup tim podacima.



Uloga provajdera sadržaja

Kao što smo već rekli, provajderi sadržaja čuvaju podatke i čine ih dostupnim aplikacijama. Provajderi sadržaja koje nudi *Android* platforma obično izlažu podatke kao skup tabela zasnovanih na modelu relacione baze podataka, gde je svaki red zapis, a svaka kolona podaci određenog tipa i značenja. Preko *API-ja* aplikacije i adapteri za sinhronizaciju mogu dobiti pristup za čitanje i pisanje tabela baze podataka koje sadrže podatke korisnika.

4.4.1. Pristup provajderu sadržaja korišćenjem ADB

Svaki kontent provajder izlaže javni URI (upakovan kao Uri objekat) koji jedinstveno identifikuje njegov skup podataka. Provajder sadržaja koji kontroliše više skupova podataka (više tabela) ima poseban URI za svaki od njih. Svi URI-ovi za provajdere počinju nizom content://. Na

ovaj način se identifikuju podaci koje kontroliše provajder sadržaja. Takođe moguće je mnogi provajderi omogućavaju pristup jednom redu, samo dodavanjem ID-a na kraj putanje.

- *Content URIs – je URI* koji identifikuje podatke u provajderu. *Content URI* uključuje simbolično ime provajdera (*authority*) i ime koje referencira tabelu (*path*). Provajder obično ima tabelu za svaku putanju koju referencira.
- *ContentResolver* izdvaja ime provajdera, i koristi ga da resolvuje provajder upoređujući njegov *authority* sa sistemskom tablom provajdera.

Provajderima sadržaja je moguće pristupiti preko *shell-a* koji je spomenut u poglavlju iznad. Pristup je moguć izvšavanjem sledeće komande u ADB cmd klijentu.

adb shell content query --uri <URI> [--projection <PROJECTION>] [--where <WHERE>] [--sort <SORT_ORDER>]

Query() argument	Select keyword/parametar	Komentar
Uri	From table_name	Uri se mapira u tabelu, sa imenom table_name
projection	Col, col, col, ...	Projection je niz kolona koje mogu biti vraćene za svaki red
selection	Where col = value	Kriterijum/filter
selectionArgs		
sortOrder	Order by col, col, . . .	Redosled redova

Značenje paramtera komande kontent provajdera.

Da bi kontent provajder vratio rezultate potrebno je:

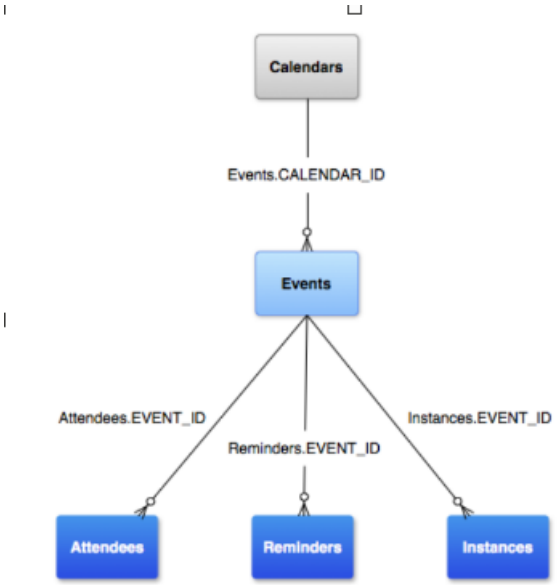
- Zatražiti permisiju za čitanje od kontent provajder
- Definisati upit za priistup kontent provajderu. Komanda koje ispisana iznad tabele.

4.4.2. Kalendar provajder

Kalendar provajder je repozitorijum kalendarskih događaja korisnika. API provajdera kalendara omogućava obavlljivanje sledećih operacija:

- upita
- dodavanja
- ažuriranja
- brisanja kalendara, događaja, posetilaca, podsetnika itd.

Kao što je ranije pomenuto postoji za svaki provajder jedinstveni *URI*, a kada provajder raspolože sa više tabela, provajder kalendara definiše konstante za URI-je za svaku od svojih tabela. Ovi URI-ji imaju format sledeći format: <klasa>.CONTENT_URI. Na primer, Events.CONTENT_URI.



Model kalendar provajdera

Slika prikazuje model podata Kalendar provajdera. Glavni entitet je *Calendars*. *Events* referencira *Calendars* tabelu. Jedan entitet iz *Calendars* može se vezati za više *Events* entiteta. Entitet tabele *Attendees* se vezuje sa *Events* entitet, i veza je jedan na više. Isti slučaj sa tabelama *Reminders* i *Instances* Korisnik može imati više kalendara, oni mogu biti povezani sa različitim nalozima kao što su *Google, Outlook, Xioami [3].*

Tabele kojima kalendar provajder omogućava pristup:

- *Calendar* – Ova tabela čuva informacije o pojedinačnom kalendaru, informacije kao što su ime, boja, informacije o sinhornizaciji. Itd. Neke od kolona su:
    - *name* – ime kalendara
    - *calendar_display_name* – ima kalendara koje se prikazuje korisniku.
    - *visible* – da li će događaji povezani sa kalendarom biti prakazani
    - *sync_events* – 0 – ne sinhronizuju se događaji i ne čuvaju na uređaj, 1- omogucećena sinhronizazacija i čuvanje na uređaju.

- *Events* – tabela koja čuva informacije o svakom pojedinačnom događaju. Informacije kao što su naziv, opis, lokacija, vremenska zona…
    - *calendar_id* – id kalendara kom događaj pripada
    - *organizer* – mejl organizatora događaja.
    - *title* – naslov događaja
    - *event_location* – mesto događaja
    - *description* – opis

- *Attendees* – čuva informacije o gostima događaja. Informacije o tipu gosta i njegov odgovor na događaj.
    - *event_id* – id događa za koji je vezan
    - *attendee_name* – ime

- *attendee_emal* – mejl
- *attendee_relationship* – tip veze, može imati sledeće vrednosti
- *attendee_status*
- *attendee_type*

atendee_relationship	attendee_status	attendee_type
relationship_attendee	attendee_status_accepted	type_required
relationship_none	attendee_status_declinded	type_optional
relationship_organizer	attendee_status_invited	
relationship_performer	attendee_status_none	
relationship_speaker	attendee_status_tetative	

Moguće vrednosti za kolone iznas

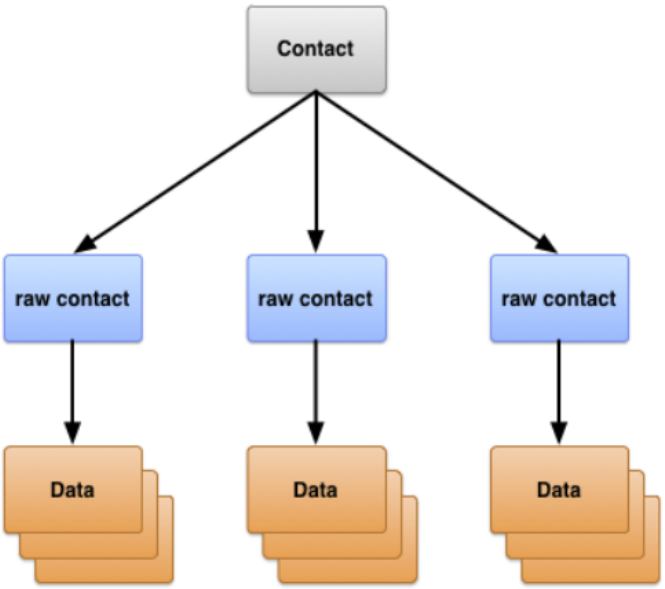
- *Reminders* – Svaki red predstavlja jedan podsetnik na događaj. Može se specificirati maksimala broj podsetnik za jedan događaj. Sadrži kolone koje označavaju početak i kraj.
  - *event_id*
  - *minutes*
  - *method - tip*
    - *method_alert*
    - *method_default*
    - *method_email*
    - *method_sms -*

- *Instances* – ova tabele čuva informacije o svakom pojedinačnom pojavljivanju jednom događaja, ukoliko se događaj pojavljuje samo jednom onda je relacija 1-1, ukoliko ne, onda je 1- n. Informacije su početak i kraj svakog pojedinačkog pojavljivanja događaja.

Za pristup ovom kontent provajderu potrebne su permsije. Međutim ukoliko se ne koristi za izgradnju aplikacije za sinhronizaciju. Permsiije nisu potrebne.

4.4.2. Kontakt provajder

Kontakt provajder održava tri tipa podata o osobi, gde svaki od njih odgovara jednoj tabeli ponuđenoj od strane provajdera.



Model kontakt provajdera

Tabele Kontakt provajdera su: *Contacts* – redovi predstavljajuu različite osobe, bazirano na agregaciji sirovih redova. *RawContacts* – predstavljaju sveobuhvate lične podatke, specifične korisničkim nalozima i tipovima. *Data* – sadrže detalje za *rawcontacts*, kao što su mejlovi i brojevi telefona. *Raw contact* - predstavlja kontakte koji dolazi od jednog naloga i jednog tipa naloga, zato što sve više provajdera omogućava više servisa kao izvor kontakta. Takođe ovaj provajder omogućava više redova za jednu osobu. Većina ovih podataka je smeštena u *data* tabeli. Najbitnije kolone u tabeli su *account_type* i *account_name*. Možemo posmatrati da korisnik ima dva naloga i oba ih koristi za komunikaciju sa istim kontaktom, tada će ova tabela sadržati dva reda za jedan kontakt. *Data* - u ovoj tabeli se čuvaju različiti tipovi podataka. Redovi sa prikazanim imenom, brojem telefona i mejlom, poštanskom adresom, slikom, detaljima o veb lokacijama. Da bi se čuvali različiti tipovi podataka, ova tabele ima kolone sa opisnim imenima, dok ima i neke sa generičim imenima. Sadržaj kolone opisnog imena ima isto značenje bez obzira na vrstu podatka u redu, dok sadržaj kolone sa generičkim imenom ima različita značenja u zavisnosti od tipa podataka [4].

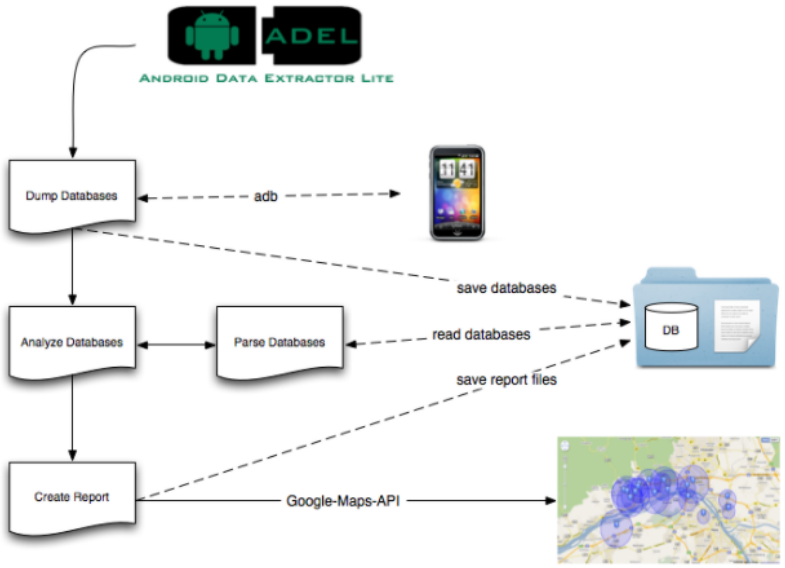
Postoje 15 generičkih kolona pod nazivom DATA1 do DATA15 koje su generalno dostupne i dodatne četiri kolone SYNC1 do SYNC4 koje koriste adapteri za sinhornizaciju. Generičke kolone uvek imaju isti sadržaj bez obzira na tip podataka koji red sadrži. *Contacts* – Kombinuje sve kontakte sa svih naloga i tipova i formira *Contact.* Kombinuje sve podatke koje korisnik sakupio o osobi.

5. Alati za mobilnu forenziku

5.1. Android Data Extractor Lite (ADEL)

ADEL alat razvijen u pajtonu. Specijalizovan za android uređaje, kao što i sam naziv kaže. Dozvoljava forenzičarima da prikupe podatke iz baza mobilnih uređaja. Mobilni telefon mora biti rutovan ili da ima instaliran lični alat za oporavak. Ovaj alat ekstraktuje *dump* i sve bitne *SQLite* baze podataka konektovane na adroid telefon lokalno na disk i analizira te fajlove. Nakon toga se kreira *.xml* dokument. Na ovaj način, omogućava sigurnost da nijedan fajl neće biti modifikovan tokom ovog procesa. *ADEL* je ekstenzibalan alat i njegova ekstenzibilnost se zasniva da ima dva odovjena modula za analizu i izveštaje i svaki od njih se može nadograditi posebno. Alat omogućava istražitelju da ima sledeće podatke: Informacije o telefonu i SIM kartici (npr. IMSI i serijski broj), telefonski imenik i liste poziva, unosi u kalendar, SMS poruke, GPS lokacije iz različitih izvora na pametnom telefonu.

ADEL koristi adb alat za pristup telefonu.



5.2. Oxygen Forensics Detective

Mobilni forenzički alat sa ugrađenom analitkom i *cloud* ekstraktorom. Veoma ga je lako koristiti. Imao prijateljski U/I za pretraživanje, filtriranje i analiziranje ekstraktovanih podataka Sposoban je da prikuplja podatke sa više od 10 000 različitih modela telefona.

Neke od mogućnosti *Oxygen forensic detective:*

- Pronalazi šifre, backupove i slike
- Pristupa *cloud* servisima i skladištima
- Kolekcionira korisničke podatke

- Podržava import i analazu istorije poziva

Oxygen Forensic Detective je napredni alat u forenzici mobilnih telefona. Može da dekodira i ekstraktuje podatke iz širokog sprektra digitalnih resursa. *Oxygen Forensic Detective* može da ekstraktuje podatke i fajlove sa android uređaja *i* ajfon uređaja, kao i sa ostalih mobilnih modela, čak i istoriju letova sa dronova. Omogućava istražiteljima da generišu i eksportuju izveštaje u fajlove različitih formata koji uključuju xml, pdg, xls.,

5.3. MOBILedit

MOBILedit platforma koja radi sa različitim telefonima i smartfonima i istražuje njihov sadržaj kroz MS *outlook* folder strutkture. Omogućava bekap informacija na telefonu čuvajući ih na računaru. Koristi IR port ili *bluetooth* vezu, wi-fi ili U/I. Nakon uspostavljenja konekcije, model je identifikovan svojim proizvođačem, brojem modela, i serijskim brojem i odgovarajućom slikom telefona. Podaci koju su ekstraktovani sa telefonu čuvaju se u fajlovima formata *.med*. Nakon završene ekstrakcije, ekstraktovani informacije o uređaju, imenik, SIM imenik, propušteni pozivi, pozivi, primljeni pozivi, upućenni pozivi, poruke, draftovi, folderi sa fajlovima.

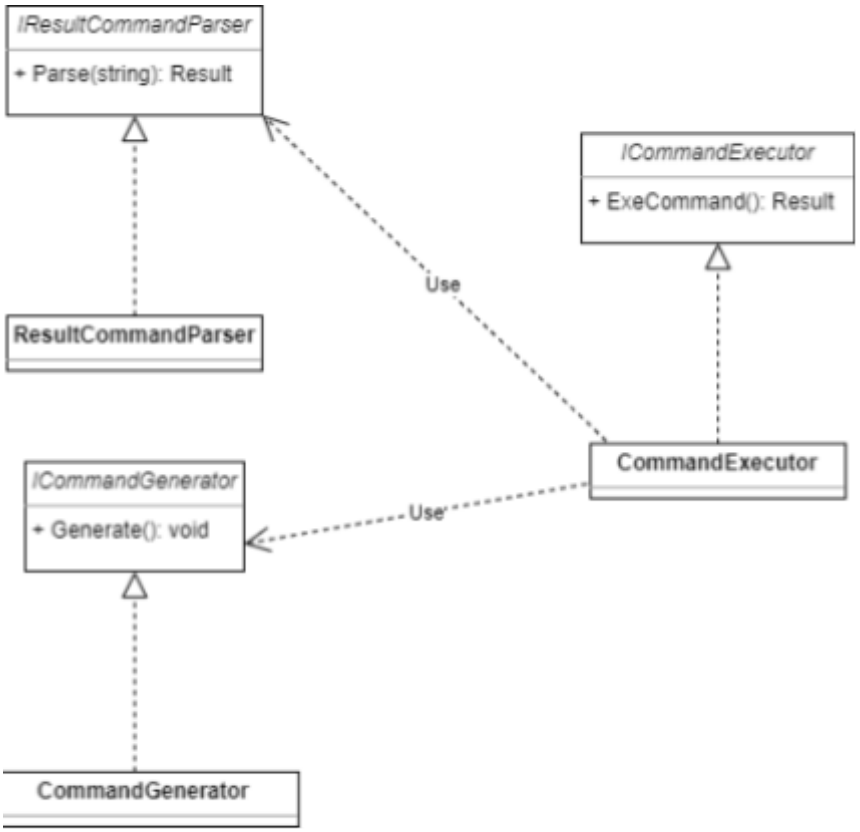
5.4. FTK Imager

FTK pruža fičere u realnom vremenu koje pomažu timovaima da upravljaju masivnim skupovima podataka, odvajajući kritične podatke od trivijalnih detalja, *FTK Imager* je alatka za pregled podataka i slikanje koja se koristi za prikupljanje podataka (dokaza) na forenzički ispravan način kreiranjem kopija podataka bez menjanja originalnih dokaza. Nakon što napravite sliku podataka, korise se drugi za analizu i izveštaj.

- Kreirajte forenzičke slike lokalnih hard diskova, CD-ova i DVD-ova, fleš diskova ili drugih USB uređaja, celih foldera ili pojedinačnih fajlova sa različitih mesta u mediju.
- Pregled foldera i fajlova na lokalnim hard diskovima, mrežnim diskovima, CD-ovima i DVD-ovima, fleš diskovima ili drugim USB uređajima.
- Pregled sadržaj forenzičkih slika uskladištenih na lokalnoj mašini ili na mrežnom disku.
- Montiranje slike za prikaz samo za čitanje koji koristi *Window expoerer* da bi se prikazao baš kako i u orginalnoj verziiji.
- Eksporovanje fajlova i foldera.
- Pogledajte i oporavite datoteke koje su izbrisane iz korpe za otpatke, ali još uvek nisu prepisane na disku.

6. Implementacija

U praktičnom delu implementiran je alat koji korišćenjem ADB alata ekstraktuje podatke sa android uređaja. Alat je implementiram u programskom jeziku *C#* kao *WinForms* aplikacija.



UML dijagram aplikacije

SharpAdbClient je .NET biblioteka koja dozvoljava .NET aplikacijama da komuniciraju sa Android uređajima. Pruža .NET implementaciju adb protokola. Ovaj paket je moguće instalirati preko *NugetPackage Manager-a.*

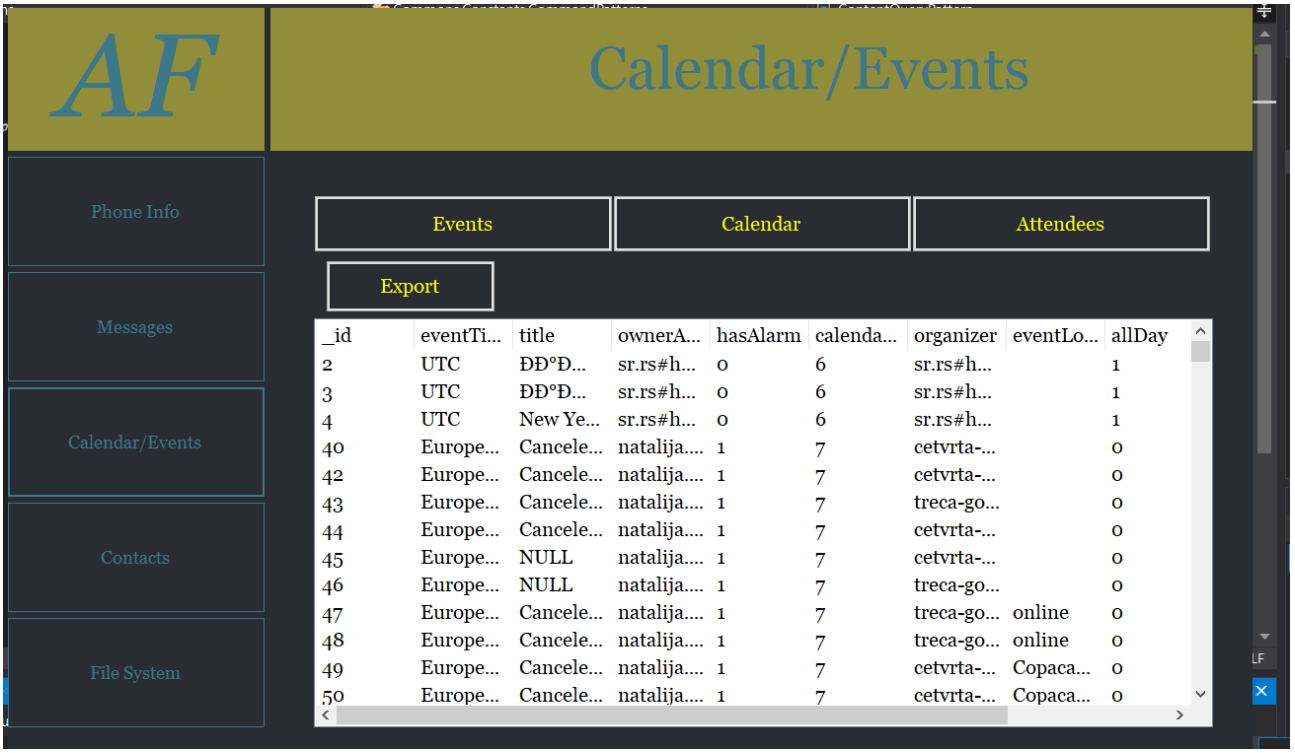
Aplikacija sadrži četiri glavne komponente:

- pristup porukama – projekat *Messages*
- pristup kalendarima i događajima – projekat *Calendar*
- pristupi kontaktima – projekat *Contacts*
- pristup fajl sistemu- projekat *FileSystem*

Svaki od ovih projekata (biblioteka) sadrži dva glavna dela:

- *CommandGenerators* – klase koje se nalaze u ovom folderu implementiraju interfejs *ICommandGenerator* i njegovu metodu *Generate(),* ove klase postoje za svaku komandu koja se koristi za interakciju sa Anroid uređajem. I ovako generisane komande se prosleđuje dalje *SharpAdbClient*-u na izvršenje.

- *CommandResultsParsers* – u ovom folderu se nalaze klase koje implementiraju *ICommandResultParser* interfejs i njegovu metodu *Parse*(). Takođe postoje za sve komande koje se koristi za pristup android uređaju. Ove klase sadrže metode koje izvadajaju rezultat koji je dobijen izvšavanje komandi koje su generisane metodama koje imlementiraju gore naveden interfejs i koje izvršava *SharpAdbClient*.



U/I aplikacije

Komponenta *ICommandExecutor* sadrži metodu *ExeCommand*() i ova komanda sadrži reference na odgovarajuće klase koje imeplementiraju interfejse *ICommandGenerator* i *ICommandResultParser,* na osnovu prosleđenih implementacija izvršava se odgovarajuća komanda i parsira rezultat

Komponenta *IClient* sadrži konačno metode koje su dostupne *AdbTool* aplikaciji i koje se mogu pozvati i prikazati interface. Svrha postojanja ovog interfejsa je da ne bi došlo do nepoklapanja određenih generatora i parsera, tada metoda ne bi bila uspešno izvšena.

U projektu *Commons* nalaze se zajedniče klase koje su potrebne svim modulima i svaki od gore navedenih modula referncira ovaj projekat. U ovom projektu se nalaze komande i šabloni komandi koje su korišćenje od strane klasa koje implementiraju *ICommandGenerator,* takođe namespace *Commons.Enums* sadrži kolone nad kojim je vršena projekcija prilikom pristupa provajderima sadržaja. Ovde se nalazi i zajedniči objekti *Result* koji se koristi za izdvajanje rezultata komande i kolona nad kojima je vršena projekcija.

Phone info sadrži informacije o konektovanom telefonu, ukoliko nema konektovanog telefona na ekranu se pojavljuje not connected, i ukoliko je se u međuvremenu poveže telefon, potrebno je kliknuti refresh i data sve opcije za pristup telefonu postaju dostupne.

6.1. Pristup porukama

U ovom delu aplikacije pristupa se provajderu sadržaja za sms poruke, ADB shell komanda za pristup koja je korišćena je:

adb shell content query --uri content://sms --projection column1:column2:..

Kolone nad kojima je izvršena projekcija mogu se naći u projektu *Commons.Enums.Messages.* Takođe da bi se isfiltrirale samo Inbox poruke korišćena je sledeća komanda:

adb shell content query --uri content://sms-inbox --projection column1:column2:..

6.2. Pristup kontaktima

U ovom delu aplikacije pristupa se provajderu kontaka, preko shell-a android uređaja. Komande za pristup koje su korišćene u radu su:

adb shell content query --uri content://com.android.contacts/data --projection column1:column2:..

adb shell content query --uri content://com.android.contacts/groups --projection column1:column2:..

Kolone nad kojima se vrši projekcija kao i u prethodnom poglavlju su podesive. Definisane su u *Commons.Enums.Contacts*

6.3. Pristup kalendarima i događajima

adb shell content query --uri content://com.android.calendar/calendars --projection column1:column2:..

adb shell content query --uri content://com.android.calendar/events --projection column1:column2:..

adb shell content query --uri content://com.android.contacts/reminders --projection column1:column2:..

Kolone nad kojima se vrši projekcija kao i u prethodnom poglavlju su podesive. I definisane su *Commons.Enums.Calenadar*

6.4. Pristup fajl sistemu

Izlistavanje fajlova komandom:

adb shell ls path -1

Zatim preuzimanje fajla komandom:

adb pull devicePath pcPah

7. Zaključak

Digitalna forenzika mobilnih uređaja predstavlja intesentanu granu digitalne forenzike. Veliki izazov za istraživanje kriminala predstavlja analiza podataka sa ovih mobilnih uređaja. Kroz ovaj rad videli smo da se podaci sa telefona nalaze u internim, i eksternim skladištima, koje takođe možemo podeliti na sistemske, interne, eskterne i skladišta mobilnih aplikacija. Danas postoji veliki broj alata na tržištu koji omogućavaju forenzičarima pristup podacima sa telefona neki od njih su obrađeni u ovom radu. Detaljno je objašnjen *adb* alat jedan od praktičnih sredstava za prikupljanje podataka sa ovih uređaja, pruža nam širok spektar komandi i pristup shell-u mobilnih telefona i aplikacionim podacima preko provajdera sadržaja.

Adb tool nam omogućava ekstrakciju pristup telefonu preko računara. A mobilnoj forenzici omogućava dobavljanje podataka s telefona u cilju njihove analize. U radu je implementrina aplikacija koja korsiti adb alat za pristup mobilnom telefonu Xiaomi redmi note 8 pro. Implementiran je pristup i ekstrakcija podataka Kalendar, Kontakt i Poruka provajdera. Takođe implementiran je pristup fajl sistemu i preuzimanje njegovog sadržaja, takođe moguće je preuzeti sadržaj aplikacija kao *.csv* fajl.

Literatura

1. https://resources.infosecinstitute.com/topic/android-forensic-logical-acquisition/
2. https://medium.com/@lucideus/android-forensic-acquisition-techniques-lucideus-forensics-e7671dbac984
3. https://developer.android.com/guide/topics/providers/calendar-provider
4. https://developer.android.com/guide/topics/providers/contacts-provider
5. https://developer.android.com/studio/command-line/adb
6. https://www.addictivetips.com/mobile/android-partitions-explained-boot-system-recovery-data-cache-misc/
7. https://medium.com/@lucideus/android-forensic-acquisition-techniques-lucideus-forensics-e7671dbac984
8. https://www.milwforensics.com/PrivateInvestigatorReviews-logicalPhysicalExtraction