UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET
KATEDRA ZA RAČUNARSTVO

Seminarski rad iz : Digitalne forenzike

Forenzika android uređaja povlačenje podataka uz pomoć ADB(Android Debug Bridge)

Student:
Aleksandar Cenić, 1062

S A D R Ž A J

Strana

1. UVOD

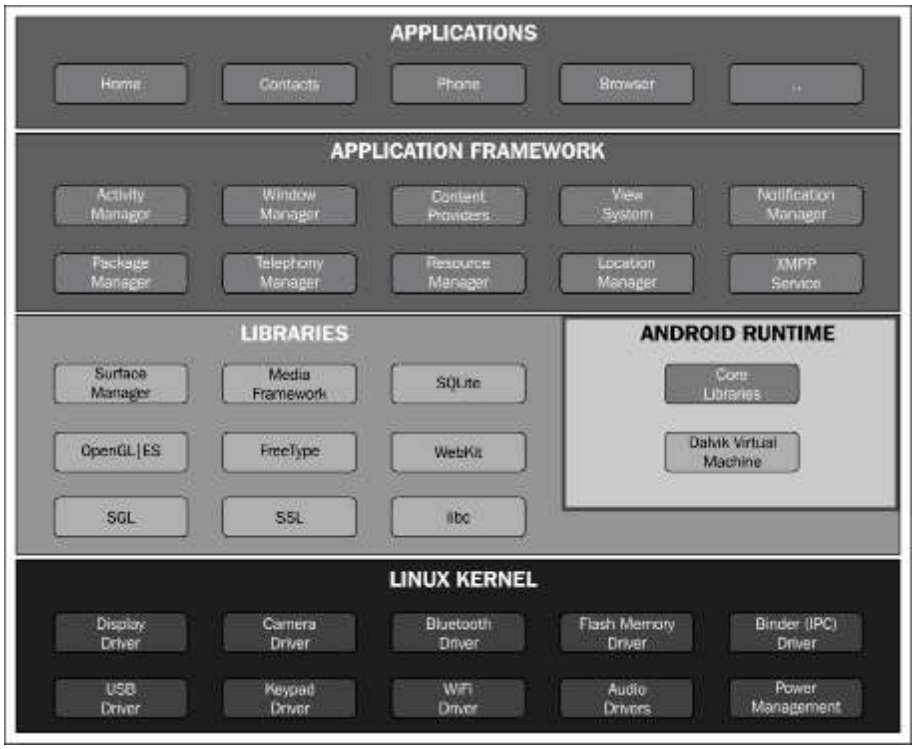
Sve što se dešava u prostoru i vremenu može li se potpuno zaboraviti? Odgovor je ne. Svaki događaj za sobom ostavlja neku vrstu traga. Kako bi smo definisali pojam kao što je trag? Trag bi, po definiciji, bilo sve ono što bi moglo da ukazuje na određeni događaj(radnju) koja se desila u prošlosti. Nauka koja se bavi zbivanjima u prošlosti jeste istorija, koja se temelji na različitim vrstama izvora. Jedan deo izvora predstavljaju različiti tipovi tragova. Tako razne knjige i spisi jesu izvori pisanih tragova koji nam mogu pružiti informacije o sistemu znakova(pismu), administraciji i celokupnom sistemu i društvu, zavisno od sadržaja. Pored pisanih postoje i druge vrste tragova poput bioloških i materijalnih. Danas, se sve više koriste elektronski uređaj koji ostavljaju jednu vrstu savremenog traga koji se naziva digitalni trag. Nauka koja se bavi proučavanjem i pronalaženjem tragova jeste forenzika. Ovaj rad će se baviti digitalnom forenzikom i to konkretno forenzikom Android uređaja.

2. ANDROID OPERATIVNI SISTEM

Android je Google-ov open source operativini sistem baziran na linux-u namenjen za smartphone mobilne telefone. Kako bi smo bolje razumeli forenziku android uređaja, bilo bi nam od pomoći, da imamo predstavu kako izgleda osnovna Android arhitektura. Kao i svaki računar i računarski sistem, koji interaguje sa korisnikom i odrađuje komplikovane zadatke, potreban mu je operativini sistem kako bi te zadatke izvršavao najefikasnije. Operativni sistem kod digitalnih uređaja ima odgovornost za upravljanje resursima celokupnog sistema i obezbeđuje mogućnost aplikacijama da komuniciraju sa hardware-om. Android platforma se sastoji od magacina slojeva koji teku jedan iznad drugog.

2.1. Android arhitektura

Da bi smo razumeli Android-ov ekosistem od esencijalnog je značaja da znamo kakvi su slojevi i šta tačno rade. Arhitektura Android platforme je prikazana na slici 1.



Slika: 1

Svaki od ovih slojeva izvršava nekoliko operacija koje podržavaju specijalne funkcije operativnog sistema. Svaki sloj pruža neophodne servise sloju koji stoji iznad njega.

2.1.1. Linux kernel sloj

Android operativini sistem je izgrađen na osnovu Linux kernela sa nekim arhitekturnim izmenama od strane Google-a. Postoji nekoliko razloga za izbor Linux kernela. Jedan od najvažnih razloga je to što je Linux prenosiva platforma i može se lako iskompajlirati na različitom hardware-u. Kernel se ponaša kao jedan apstraktni sloj između hardware-a i software-a na samom uređaju. Razmotrimo slučaj klika na

kameri. Kada korisnik pritisne dugme za kreiranje fotografije, instrukcija ide na odgovarajući kamera drajver u kernelu, koji šalje određene komande hardware-u kamere.

Linux kernel je odgovoran za upravljanje srži funkcionalnosti Android operativnog sistema kao što su upravlanje procesima, upravljanje memorijom, bezbednost i umrežavanje. Svaka od verzija Android-a ima drugačiju osnovnu verziju Linux kernela.

2.1.2. Sloj biblioteka

Sledeći sloj Android arhitekture sadrži native biblioteke. Biblioteke koje ovaj sloj sadrži napisane su u C i C++ jeziku i pomažu uređaju da rukuje različitim tipovima podataka. Na primer, SQLite biblioteke su korisne za skladištenje i vađenje podataka iz baze podataka. Media Framework biblioteka obezbeđuje servise ostalim osnovnim bibliotekama. WebKit biblioteka pruža učitavanje web stranica u web browser-u, surface manager održava grafiku. U istom sloju se nalazi Android Runtime, koji sadrži Dalvik virtualnu mašinu i biblioteke jezgra. Android runtime je odgovoran za izvršavanje aplikacija na Android uređaju.

2.1.3. Application Framework sloj

Application Framework sloj odgovoran za rukovanje bazičnim funkcijama telefona, kao što je upravljanje resursima, upravljanje pozivima i ostalo. Ovo je blok sa kojem, aplikacije instalirane na uređaju direktno komuniciraju. Neki od najvažnih blokova ovog sloja su:

- **Telephony manager** – Ovaj blok upravlja svim telefonskim pozivima.
- **Content provider –** Ovaj blok upravlja deljivim podacima između različitih aplikacija.
- **Resource manager** – Ovaj blok pomaže oko upravljanja raznim resursima korišćenim u aplikacijama.

2.1.4. Applications sloj

Aplikacioni sloj je najviši sloj u hierarhiji, gde korisnik može direktno da interaguje sa uređajem. Postoje dve vrste aplikacija preinstalirane aplikacije i korisnički-instalirane aplikacije. Preinstalirane aplikacije su aplikacije koje dolaze sa Android operativnim sistemom kao što su osnovni Web Browse-r, Contacts, Calculator i druge. Korisnički-instalirane aplikacije su sve one aplikacije koje korisnik skida sa i instalira sa raznih mesta kao što je Google Play Store, Amazon i drugi. Sloj aplikacija predstavlja sve aplikacije na datom uređaju.

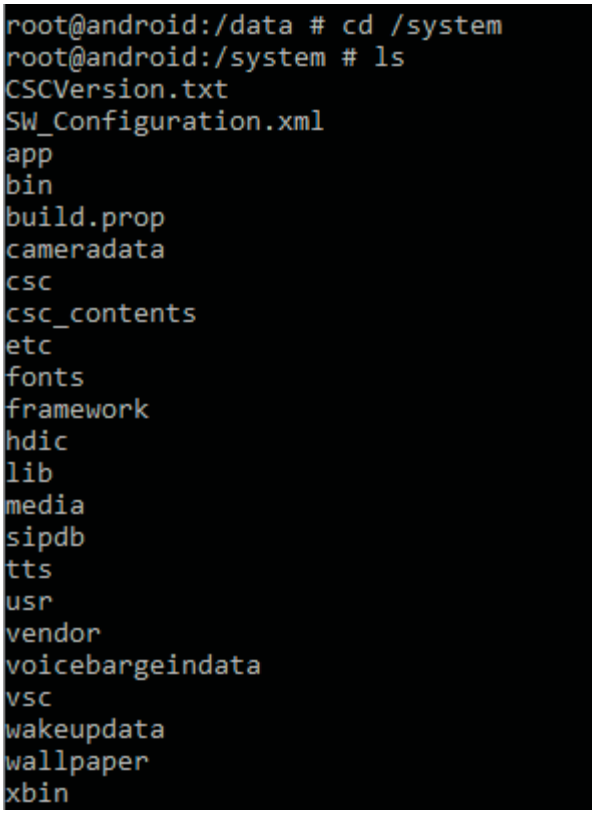
2.2. Android fajl hierarhija

U nameri da odradimo forenzičku analizu bilo kog sistema (računara ili telefona) važno je da razumemo osnovnu fajl hierarhiju. Osnovno razumevanje kako Android organizuje podatke u fajlove i foldere pomaže forenzičkom analitičaru da suzi fajlove i foldere koje treba pretražiti. Vedi još jednom napomenuti da Android koristi Linux kernel. Kako Android koristi Linux kernel, tako i fajl hierarhija je takođe bazirana na Linux hierarhiji sa određenim izmenama. Izmene osnovne Linux hierarhije zavise od proizvođača i od verzije Linux-a koji se koristi.

Prateća lista važnih foldera je ista za većinu Android uređaja. Neki od foldera sa liste su dostupni samo kroz root pristup. Root pristup se može ostvariti procesom koji se zove rutovanje. Proces rutovanja podrazumeva dobijanje privilegija root korisnika kako bi se moglo pristupiti čitavom sistemu. Lista foldera kod Android uređaja je sledeća:

- /boot: Kako samo ime kaže ovo je particija koja ima potrebne informacije i fajlove za pokretanje samog telefona. Boot sadrži kernel i RAM disk. Podaci sadržani u RAM-u jako su bitni za forenziku. Bez ove particije telefon ne bi mogao da izvršava procese.

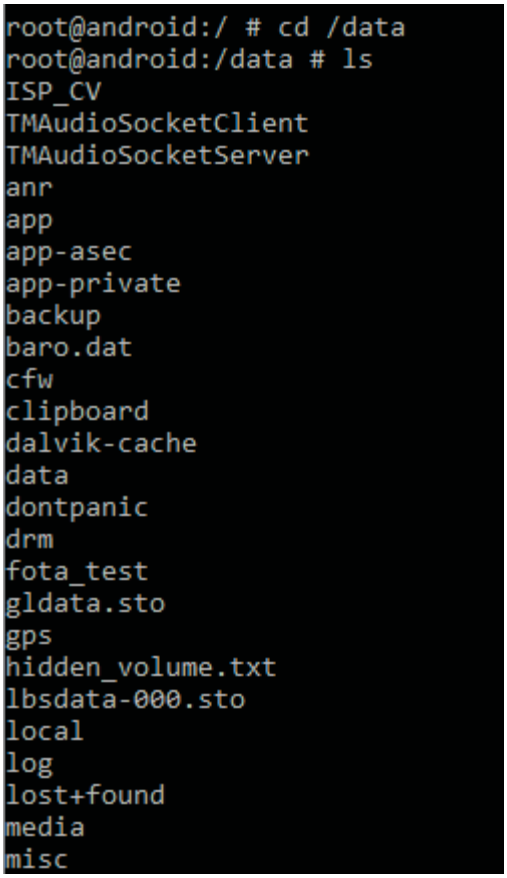
- /sistem: Ova particija sadrži fajlove vezane za sistem osim kernela i RAM diska. Sistem particija ne sme nikada biti izbrisana, jer će to uređaj učiniti neboot-abilnim. Sadržaj ove particije se može videti uz pomoć komande:



Slika: 2

- /recovery: Ova particija je dizajnirana u svrhe backup-a i dozvoljava uređaju da se boot-uje u recovery modu. U recovery modu mogu se naći alati za popravku instalacije sistma.

- /data: Data particija sadrži podatke svih aplikacija na uređaju. Većina podataka pripada korisniku, kao što su kontakti, SMS poruke, pozivani brojevi. Ova particija je veoma važna sa stanovišta forenzike, jer sadrži dragocene podatke. Sadržaj ove particije se može videti uz pomoć komande:



Slika: 3

- /cache: Ovaj folder se koristi kako bi se smestili podaci kojima se frekventno dosta pristupa i od logova kako bi se brže pronašli. Cache particija je takođe jako bitna za forenzičku istragu, jer je moguće da se tu nalaze podaci koji se više ne nalaze u /data partici.

- /misc: Misc particija sadrži informacije vezane za razna podešavanja uređaja. Ta podešavanja u globalu su vezana za hardver i definišu stanje uređaja, da li je nešto uključeno ili isključeno.

- /sdcard: Ova particija sadrži informacije koje se nalaze na SD kartici. Particija je jako ranjiva jer sadrži lične podatke slike, video zapise i dokumente.

U ovom radu najviše će nas zanimati data particija, jer sadrži dragocene podatke korisnika koji su važni za forenzičku analizu.

3. POSTAVLJANJE OKRUŽENJA ZA FORENZIKU

Da bi smo izvršili forenzičku analizu Android uređaja, moramo da instaliramo određene alate kako bi smo postavili potrebno okruženje. Postavljanje okruženja se izvodi u nekoliko koraka:

- Poželjno je da počinjemo sa forenzički čistim računarom. To znači, da ostali podaci koji se nalaze na računaru, ne mogu da kontaminiraju proces istrage.

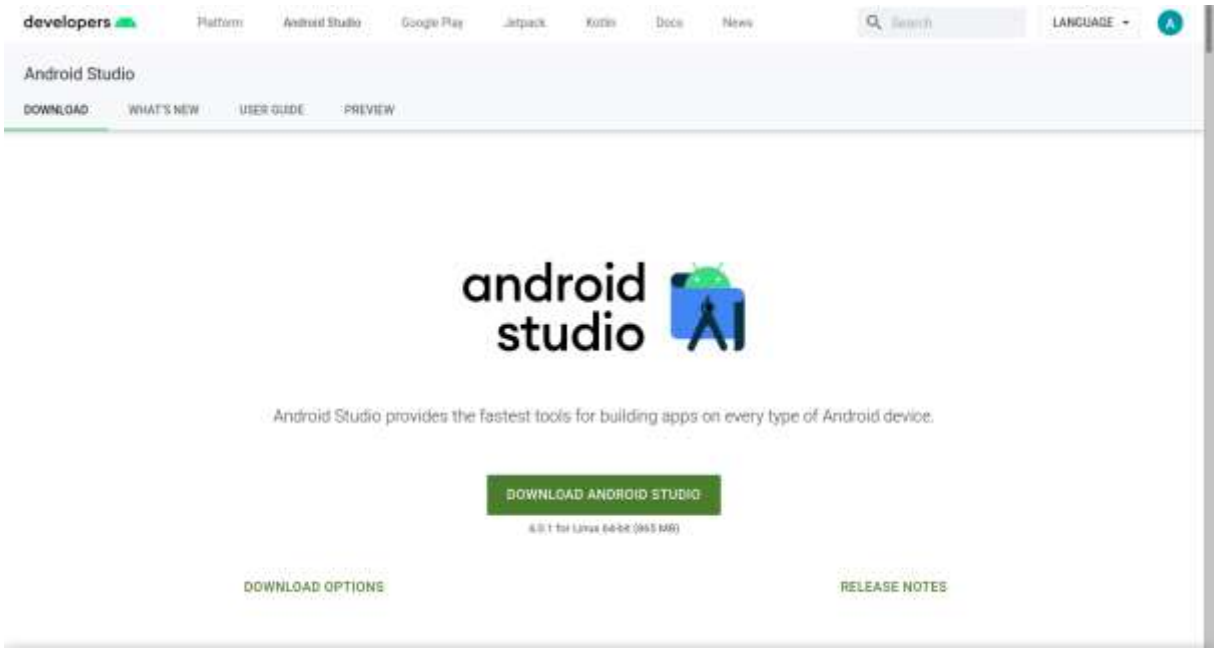
- Instalacija bazičnog softvera potrebnog za konektovanje Android uređaja na računar. Android forenzički alati će raditi na Windows-u, Linux-u i MacOS-u.

- Dobijanje pristupa uređaju. Ispitivač mora biti u mogućnosti da omogući podešavanja ili zaobići ih kako bi se omogućili vađenje podataka sa Android uređaja.

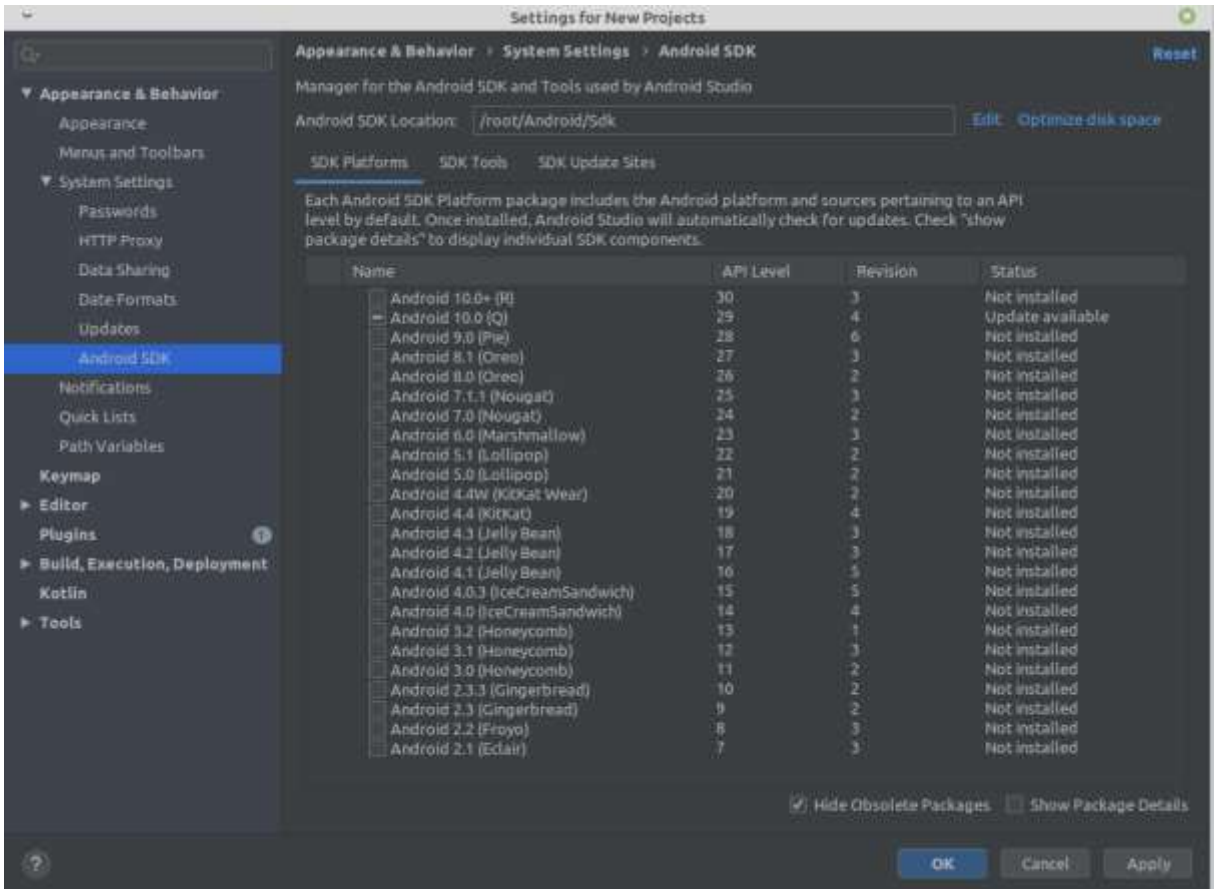
3.1. Android Software Development Kit (SDK)

Android Softvare Development Kit (SDK) pomaže da se izgradi, testira
i debug-uje Android aplikacija. To se postiže korišćenjem potrebnih alata za kreiranje aplikacija. Ali uz to, pruža i dragocenu dokumentaciju i druge alate koji mogu biti od velike pomoći tokom forenzike Android uređaja. A dobro razumevanje Android SDK-a će vam pomoći da se upoznate sa detaljima uređaja i podataka na uređaju.

Pre nego što instalirate Android SDK, proverite da li u vašem sistemu postoji Java JDK.
Java JDK možete naći na: https://jdk.java.net .
Instalacija Android SDK može se obaviti uz pomoć programa Android studio, koji predstavlja razvojno okruženje za Android aplikacije. Program Android studio možete naći na adresi: *https://developer.android.com/studio* .



Slika: 4



Slika: 5

Slika iznad prikazuje Android SDK menager koji je deo Android studija i koji nam omogućava da skinemo određenu verziju Android SDK sa pratećim alatima Najvažnije je instalirati USB driver kako bi računar prepoznao Android uređaj.

Nakon instaliranja Android SDK možemo i napraviti Android virtuelne mašine koje su pogodne za testiranje alata i aplikacija.

Slika: 6

3.2. Android Debug Bridge (adb)

Android Debug Bridge je svestran alat komandne linije koji nam omogućava komunikaciju sa uređajem. Komanda adb olakšava razne akcije uređaja, poput instaliranja i debug-ovanja aplikacija i omogućava pristup Unix shell-u, koga možete koristiti za pokretanje različitih naredbi na Android uređaju. To je klijent-serverski program koji sadrži tri komponente:

- **Klijent** koji šalje komande. Klijent radi na vašoj razvojnoj mašini. Možete pozvati klijenta sa terminala komandne linije izdavanjem adb naredbe.

- **Daemon** (adbd) koji izvršava naredbe na uređaju. Daemon radi kao pozadinski proces na svakom uređaju.

- **Server** koji upravlja komunikacijom između klijenta i daemona. Server se pokreće kao pozadinski proces na vašoj razvojnoj mašini.

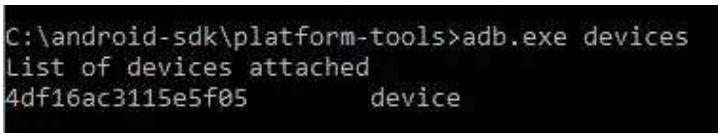
Daemoni (servisi) komuniciraju preko svojih lokalnih host-ova na portovima 5555 do 5585. Kada adbd radne stanice otkrije novi emulator ili uređaj, stvara dve uzastopne veze priključaka. Parni port komunicira s konzolom uređaja dok je neparni port zadužen za adb veze. Program lokalnog adb klijenta koristi port 5037 za komunikaciju s lokalnim adbd.

Da biste koristili adb sa uređajem povezanim preko USB-a, morate da omogućite USB debug-ovanje u sistemskim podešavanjima uređaja, pod opcijama Developer options.

Slika: 7

Jednom kada je podešavanje okruženja završeno i Android uređaj je u USB debug režimu, povežite Android uređaj forenzičku radnu stanicu USB kablom i pokrenite pomoću adb. Sada ćemo navesti nekoliko adb komandi:

Komanda kojom detektujemo uređaje koji su povezani na razvojnoj mašini: adb devices

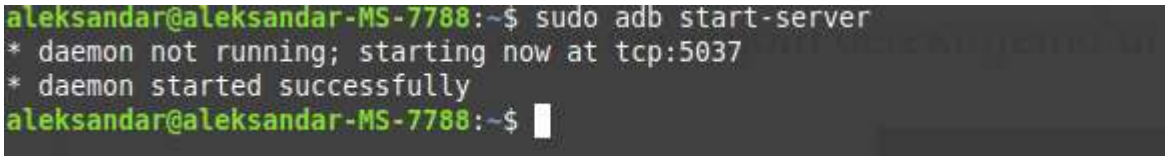


Slika: 8

U rezultatu izvršenja ove komande dobijamo listu konektovanih uređaja kao sa slike iznad.

Kada adb ne prepozna lepo Android uređaj, tada možemo „ubiti“ proces servera i ponovo ga startovati uz pomoć komandi: adb kill-server i adb start-server.

Slika: 9



Slika: 10

Nakon ove komande možemo ponovo izvršiti komandu adb devices kako bi smo videli koji Android uređaj su povezani na razvojnoj mašini.

Kada želimo da pristupimo comandnoj liniji Android uređaja, to možemo učiniti komandom: adb shell.

Slika: 11

Kada želimo da „povučemo“ podatke sa Android uređaja koristimo komandu: adb pull.

```
adb pull $DEVICE_DIR/$file $HOST_DIR/$file;
```

Slika: 12

Možemo, takođe vršiti upite nad bazama podataka koje sadrže korisne podatke, to ćemo učiniti uz pomoć content provider-a.

```
query --uri content://com.android.contacts/data --projection display_name:data1:data4:contact_id'
```

Slika: 13

```
content query --uri content://sms --projection _id,address,body,read,date,type
```

Slika: 14

Uz pomoć adb-a možemo instalirati aplikacije sa razvojne mašine.

```
adb -s 42001551d850b4ed install -t app-debug.apk
```

Slika: 15

U narednom poglavlju ćemo opisati izradu aplikacije koja koristi adb alat u svrhu povlačenja podataka.

4. IZRADA APLIKACIJE ZA POVLAČENJE PODATAKA

U ovom poglavlju ćemo opisati izradu aplikacije, koja uz pomoć adb alata i content-provider-a povlači podatke iz Android uređaja. Aplikacija je, u suštini, korisnik adb alata kao posebnog procesa koji interaguje sa Android uređajem. Aplikacija AndForData je izrađena u C++ programskom jeziku uz pomoć Qt framework-a.

Slika: 16

Verzija Qt frameworka korišćenog za izradu AndForData je 5.15.0 uz MinGW-64bit-ni kompajler. Kako je Qt prenosiva platforma ovaj program se pože iskompajlirati i pokretati na različitim operativnim sistemima kao što su Windows, Linux, MacOS i drugi.

4.1. Ograničenja AndForData aplikacije

Pre detaljnijeg opisa same aplikacije da napomenemo njena ograničenja u pogledu funkcionalnosti. Glavna funcionalnost ove aplikacije jeste povlačenje podataka iz Android uređaja važnih za forenzičku analizu, moramo reći da zbog bezbednosnih ograničenja nije moguće, putem ove aplikacije, pristupiti svim bitnim direktorijuma Android file sistema.

Slika: 17

Kako je Android operativni sistem zasnovan na Linux kernel-u i namenjen različitom hardveru, svaka kompanija ima zasebne načine zaštite osetljivih direktorijuma.

4.1.1. Rutovanje Android uređaja

Za potpunu kontrolu nad Android uređajem potrebno je rutovati taj uređaj
Šta je zapravo predstavlja rutovanje uređaja. Rutovanje uređaja je, zapravo ubacivanje sudo komande u /bin direktorijum. Sa sudo komandom možemo da imamo administratorski pristup Android uređaju. Kako smo ranije naveli, svaki proizvođač ima zasebne načine zaštite od rutovanja tako da proces i uspešnost rutovanja zavisi od verzije Android operativnog sistema i proizvođača uređaja.

Slika: 18

Jedna od poznatih aplikacija za rutovanje Android uređaja je KingORoot.

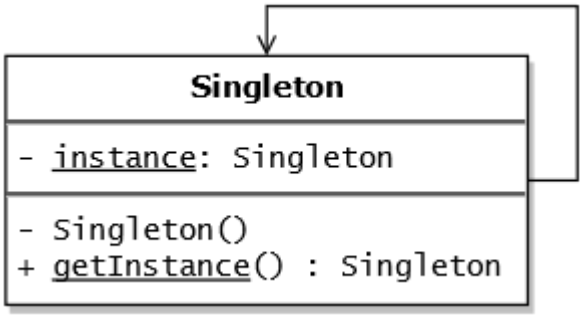
4.2. Arhitektura AndForData aplikacije

Arhitektura AndForData aplikacije je jednostavna, sastoji se od dve glavne klase, a to su:

- MainWindow koja je zadužena prozor i za grafički interfejs aplikacije.
- Command koja je srž aplikacije, napravljena po Singleton pattern-u i sadrži po jednu metodu za svaku adb komandu.

Klasa komand sadrži atribute:

- static Command m_Instance; // > Instanca objekta
- QProcess m_Process; // > Instanca proces objekta
- QString m_Command_Output; // > Izlaz izvrsenja komande
- QString m_Device; // > Selektovani uredjaj

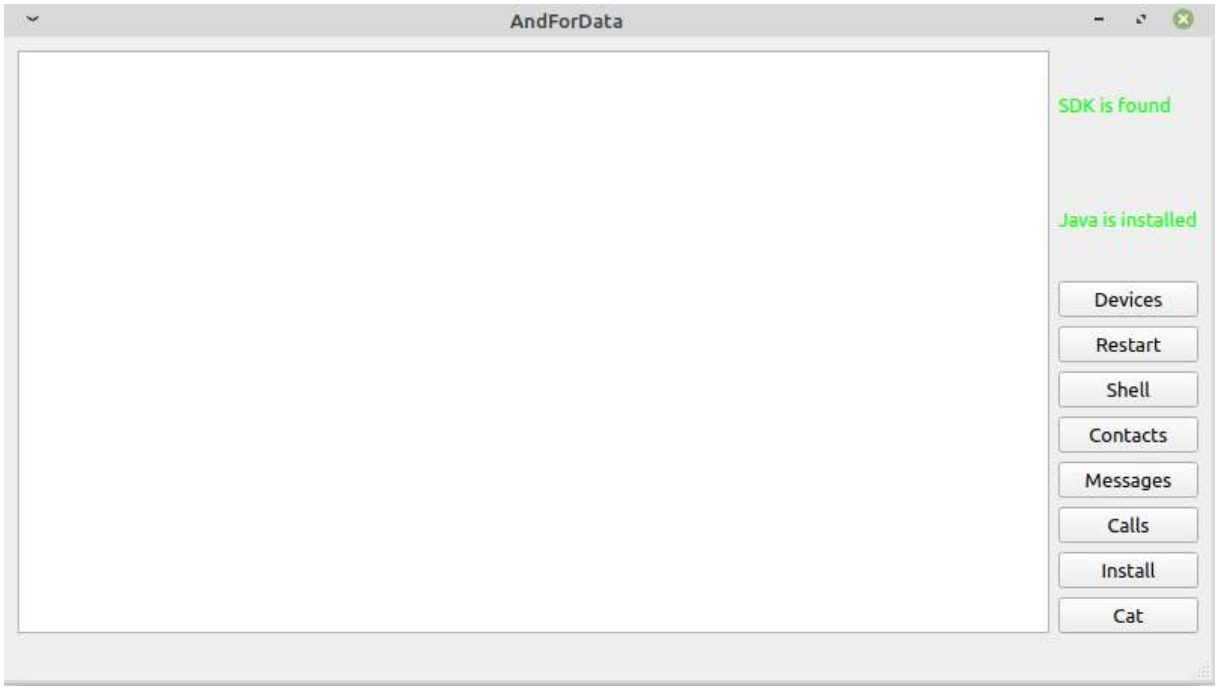


Slika: 19

Komande adb-a se izvršavaju uz pomoć QProcess objekta kao komande komandne linije operativnog sistema.

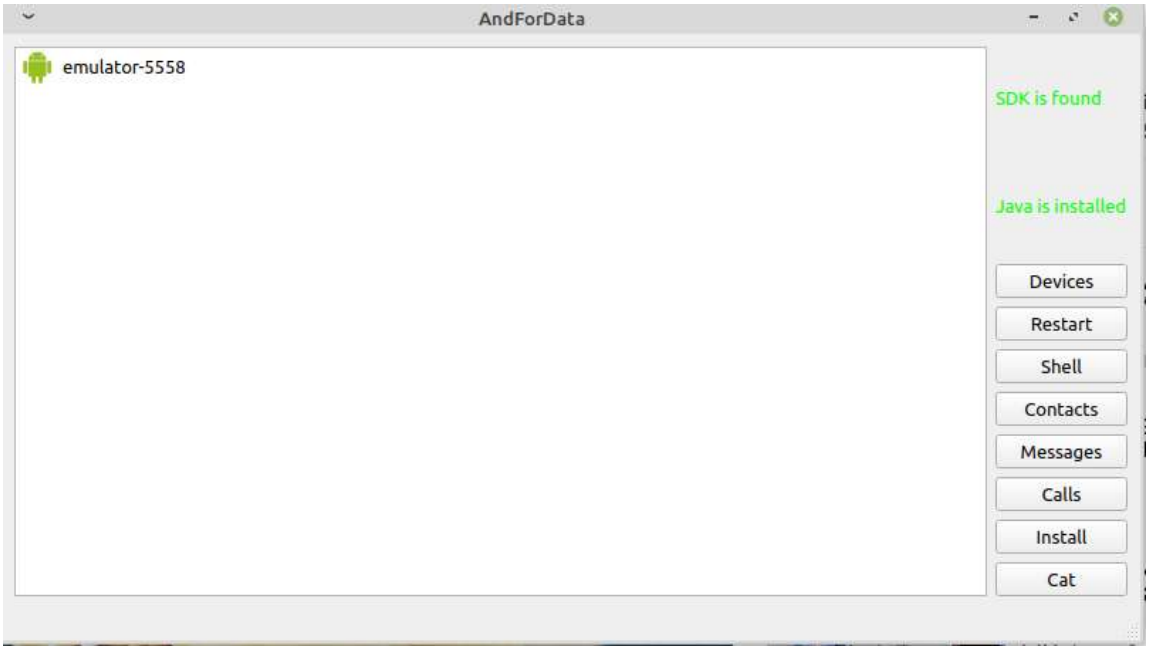
4.3. AndForData aplikacija

Kada pokrenemo AndForData aplikaciju otvori nam se glavni prozor koji izgleda kao na slici ispod.



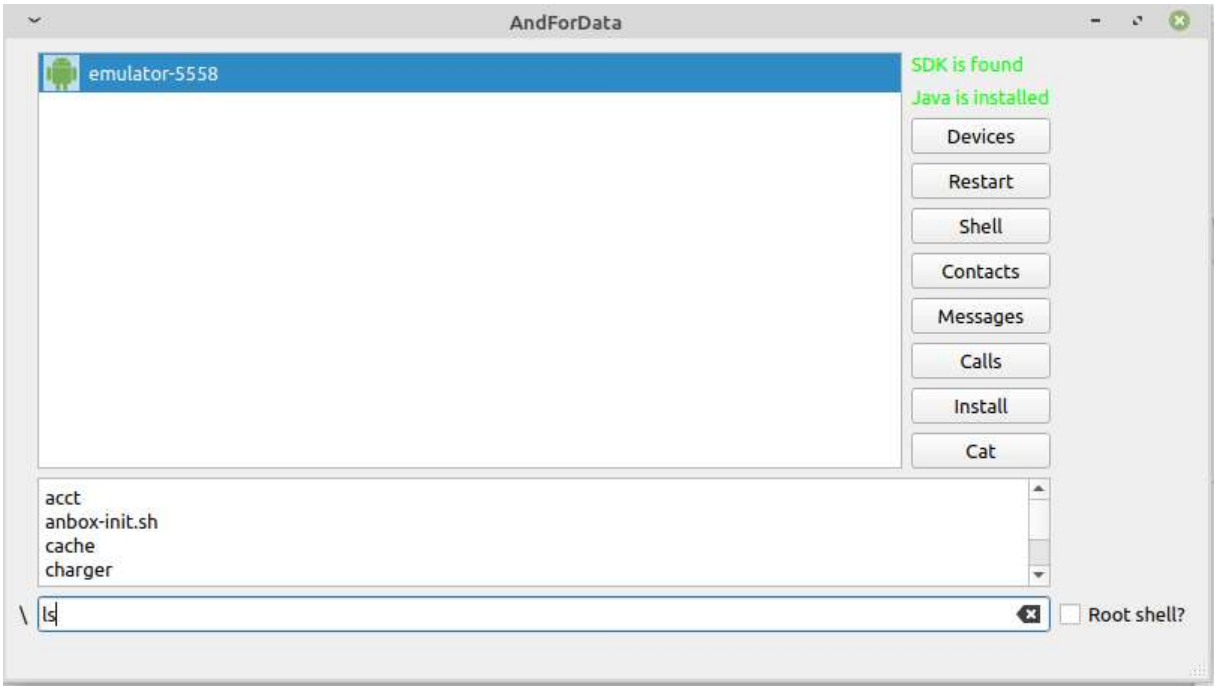
Slika: 20

Sa desne strane nalazi se meni sa komandama. Pritiskom na dugme Device izvršava se komanda adb devices i dobija se lista povezanih uređaja.



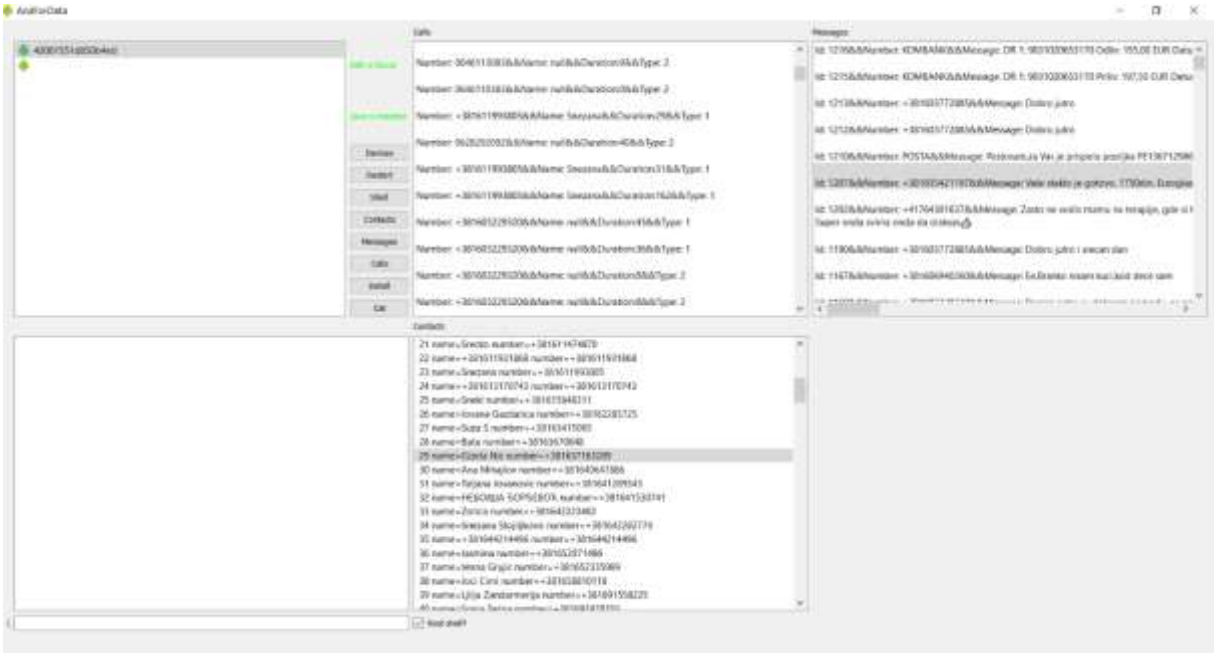
Slika: 21

Klikom na dugme shell dobijaju se kontrolice preko kojih može da se pristupi Android shell-u.



Slika: 22

Dugme Install nam dozvoljava da instaliramo aplikaciju sa razvojne mašine. Sva ostala dugmića sem Reseta služe za pvlačenje određenih vrsta podataka.



Slika: 23

NAPOMENA: Zbog zaštite Android uređaja nije moguće sve korisne podatke izvući preko ADB-a. Zbog toga u prilog ovom radu ide jedna Android aplikacija, koja preko contact provider-a izvlači podatke: sms poruke i pozive i upisuje u fajil koji je moguće uz pomoć AndForData aplikacje privući sa uređaja.

5. ZAKLJUČAK

Kako su mobilni uređaj danas jako zastupljeni, praktično predstavljaju ličnu kartu jedne osobe, jako je bitno za forenzičkog analitičara da zna više tehnika i načina za dobijanje podataka iz samog uređaja, pa čak i ako je potrebno hakovanjem zaobići sistem zaštite. Ovaj rad je bio samo jedan od primera kako se mogu dobiti podaci sa Android uređaja bez zalaženja u duboku analizu razbijanja zaštite.

LITERATURA

[1.] Pratical Mobile Forensics, Second Edition: Poglavlje 9 Understanding Android.

[2.] Pratical Mobile Forensics, Second Edition: Poglavlje 10 Android Forensic Setup and Pre-Data Extraction Techniques.

[3.] Članak dokumentacije Android Debug Bridge, Link: https://developer.android.com/studio/command-line/adb