Analiza primjene postupaka jailbreak i root u svrhu forenzičkog ispitivanja mobilnih uređaja

Kovač, Mateo

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Mateo Kovač

ANALIZA PRIMJENE POSTUPAKA JAILBREAK I ROOT U SVRHU FORENZIČKOG ISPITIVANJA MOBILNIH UREĐAJA

DIPLOMSKI RAD

Zagreb, 2020.

Sveučilište u Zagrebu

Fakultet prometnih znanosti

Diplomski rad

ANALIZA PRIMJENE POSTUPAKA JAILBREAK I ROOT U SVRHU FORENZIČKOG ISPITIVANJA MOBILNIH UREĐAJA

ANALYSIS OF APPLICATION OF JAILBREAK AND ROOT PROCEDURES IN FORENSIC TESTING OF MOBILE DEVICES

Nastavnik: dr. sc. Siniša Husnjak

Student: Mateo Kovač
JMBAG: 0246055362

Zagreb, rujan 2020.

SAŽETAK

Zbog sve boljih sigurnosnih modela implementiranih u operativnim sustavima pametnih telefona, postalo je puno teže uspješno provoditi forenzičke postupke prikupljanja i analiziranja podataka. Eskalacija privilegija je jedna od metoda koja može olakšati provođenje tih postupaka, a primjeri takve metode su *root* i *jailbreak* postupci korišteni na Android i iOS operativnim sustavima. Iako je primarni cilj navedenih postupaka jednak, zahtjevi ali i procedure obavljanja istih razlikuju se ovisno o značajkama sustava. Nedostatak ovih metoda je taj što oni mogu utjecati na integritet prikupljenih podataka i time narušiti jedan od glavnih zahtjeva forenzike, tj. zahtjev forenzičke ispravnosti. Iz tog razloga važno je detaljno dokumentirati način provođenja tih postupaka, te odrediti isplativost provođenja postupaka u svrhu forenzičke istrage.

KLJUČNE RIJEČI: jailbreak; root; podaci; forenzika; pametni telefon

SUMMARY

Due to ever-improving security models implemented in smartphone operating systems, it has become much more difficult to successfully conduct forensic data collection and analysis procedures. Privilege escalation is one of the methods that can be used to perform these procedures more easily and some of the examples are root and jailbreak procedures used on Android and iOS operating systems. Although the primary goal of these procedures is the same, the requirements and procedures for performing them may differ depending on the characteristics of the system. The disadvantage of these methods is that they can affect the integrity of the collected data and thus violate one of the main forensics requirements, ie forensic soundness. For this reason, it is important to document in detail the manner in which these procedures are conducted and to determine the effectiveness of conducting the procedures for the purpose of forensic investigation.

KEYWORDS: jailbreak; root; data; forensics; smartphone

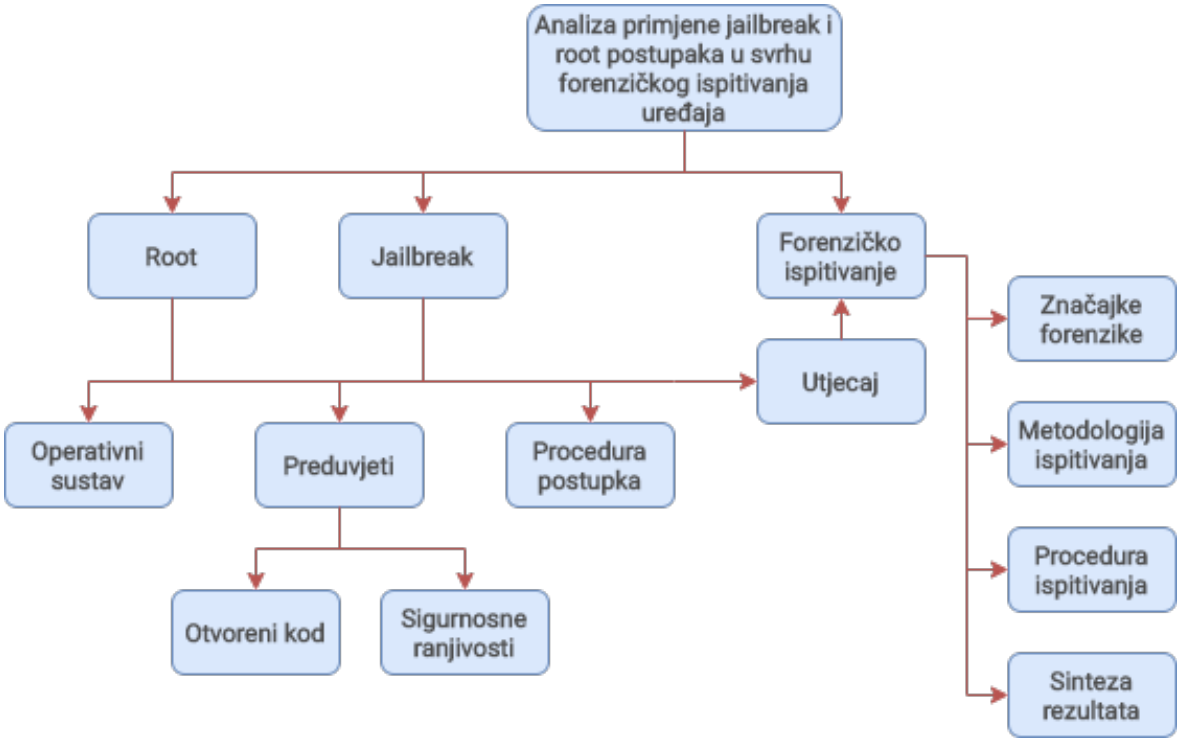
Sadržaj

1. Uvod

Pametni telefoni postali su neophodan alat u svakodnevnom životu. Oni korisnicima omogućuju skladištenje i pristupanje velikoj količini podataka, ali i obavljanje mnogih funkcija koje je u prošlosti bilo moguće obaviti isključivo pomoću stolnih računala. U tom pogledu, zaštita privatnosti i sigurnost podataka postali su jedni od glavnih zahtjeva koje mnogi proizvođači pametnih telefona nastoje osigurati kako bi zadovoljili potrebe svojih kupaca. Međutim, sigurnost uređaja u velikoj mjeri također ovisi i o implementiranim operativnim sustavima.

Nakon pojave prvih pametnih telefona pa do dana pisanja diplomskog rada, na tržištu su opstala samo dva operativna sustava: Android i iOS. Pristup razvoju navedenim sustavima je u najvećoj mjeri moguće razlikovati po otvorenosti koda koja može biti ključna za sigurnost uređaja. U pogledu Android OS-a, izvorni kod sustava je moguće u potpunosti preuzeti besplatno te ga izmijeniti u vlastite svrhe što većini proizvođača pruža viši stupanj slobode prilikom proizvodnje pametnih telefona. S druge strane, većina izvornog koda iOS-a je u potpunosti zatvorena te je kao takav zamišljen u svrhu korištenja na uređajima jednog proizvođača. Sukladno pristupu razvoja, svaki sustav zahtjeva različite metode pomoću kojih bi bilo moguće zaobići implementiranu sigurnosnu zaštitu.

Osim u slučaju kriminalnih radnji, zaobilaženje zaštite uređaja u svrhu prikupljanja i analize skladištenih podataka je također osnovno područje rada digitalne forenzike. Kako bi se takve radnje što bolje prilagodile dinamičnoj prirodi pametnih telefona, uspostavljena je zasebna grana digitalne forenzike odnosno forenzika pametnih telefona. Uspješnost obavljanja takvih postupaka u velikoj mjeri ovisi o sposobnosti ispitivača da osigura potpuni pristup svim funkcijama uređaja. Takva vrsta pristupa se najčešće ostvaruje procedurama eskalacije privilegije odnosno *root* i *jailbreak* postupcima. Cilj diplomskog rada je istražiti utjecaj takvih postupaka na forenzičko ispitivanje, pri čemu je naslov diplomskog rada: Analiza primjene postupaka *jailbreak* i *root* u svrhu forenzičkog ispitivanja mobilnih uređaja. Nadalje, pristup razradi teme prikazan je slikom 1 kojom su naznačeni svi elementi važni za tematike rada.



Slika 1. Pristup obradi tematike diplomskog rada

Rad je podijeljen u sedam cjelina:

1. Uvod
2. Značajke operativnih sustava pametnih telefona
3. Sigurnosne ranjivosti pametnih telefona
4. Značajke i procedura eskalacije privilegija
5. Veza forenzičke analize uređaja i *root*/*jailbreak* postupaka
6. Postupci ekstrakcije i analize podataka
7. Zaključak

Navedenim poglavljima bit će detaljnije objašnjeni svi osnovni elementi važni za tematiku rada, ali i praktični primjeri ispitivanja pametnih telefona kako bi se prikazala razlika u količini prikupljenih podataka. Ispitivanje će biti provedeno na dva uređaja, odnosno jedan temeljen na Android operativnom sustavu, a drugi temeljen na iOS operativnom sustavu. Provedeni postupci će pri tome biti u potpunosti dokumentirani zajedno s dobivenim rezultatima.

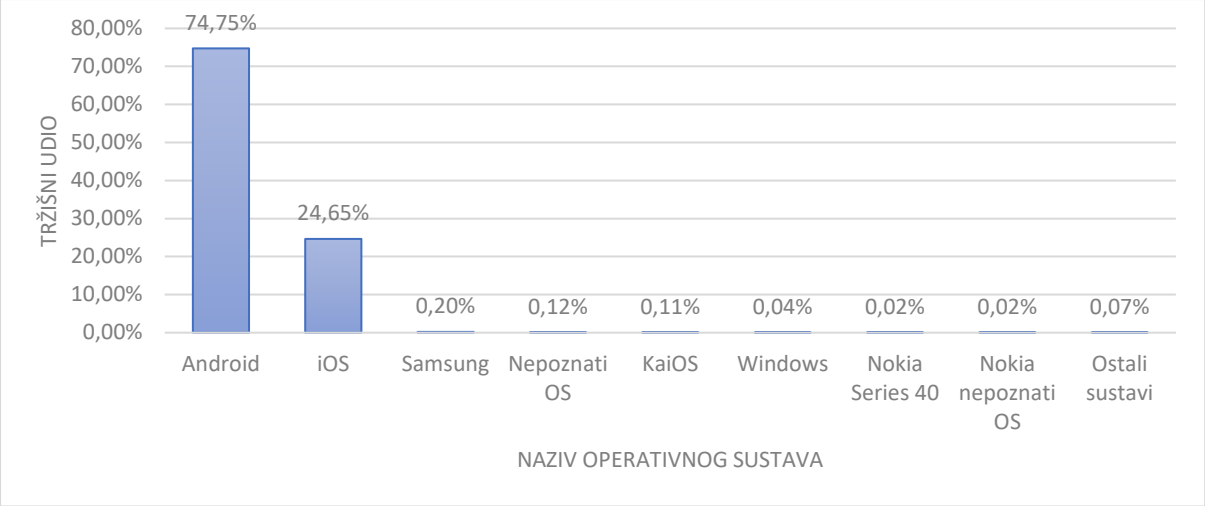
2. Značajke operativnih sustava pametnih telefona

Operativni sustav (eng. *operating system* – OS) je primarni softver koji upravlja memorijom uređaja, pokrenutim procesima, hardverom i drugim softverom uređaja. Krajnjem korisniku omogućuje jednostavno i pregledno korisničko okruženje pomoću kojeg može upravljati radom uređajem bez potrebe poznavanja programskog jezika. Uobičajeno je sastavljen od softverskih komponenti kao što su upravljački programi (eng. *drivers*) i aplikacijska programska sučelja (eng. *Application programming interface* – API) pomoću kojih koordinira radom svim funkcija uređaja. U odnosu na operativne sustave stolnih računala, operativni sustavi pametnih telefona su posebno prilagođeni radu na uređajima s ograničenim hardverom. U prošlosti je bio korišten veliki broj takvih operativnih sustava, međutim danas dva najpoznatija OS pametnih telefona su Android i iOS, [1].

2.1. Značajke Android operativnog sustava

Android operativni sustav razvila je tvrtka Android, Inc. koju je 2005. godine kupila tvrtka Google. Sustav je prvobitno zamišljen kao operativni sustav digitalnih fotoaparata, međutim daljnjim razvojem je prenamijenjen za uporabu na pametnim telefonima zbog porasta njihove popularnosti. Prvi Android pametni telefon pušten je u prodaju 2008. godine, kada je također osnovan Open Handset Alliance konzorcij čija je svrha daljnji razvoj sustava te standardizacija prijenosnih uređaja, [2].

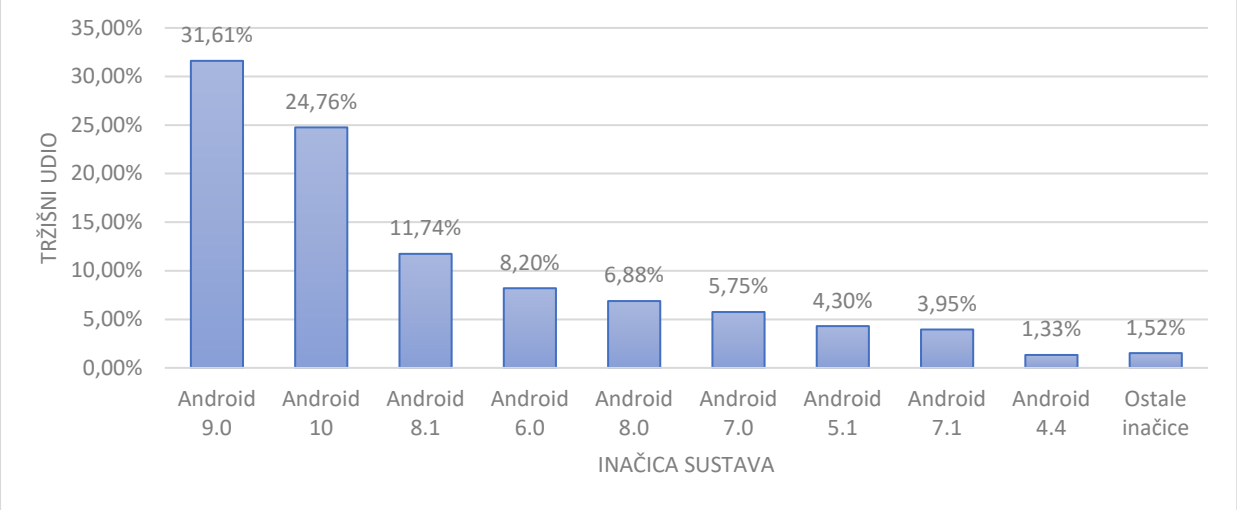
Kao što je prikazano grafikonom 1, u srpnju 2020. godine, Android OS je bio najzastupljeniji operativni sustav na tržištu pametnih telefona sa 74,75% tržišnog udjela. Danas ga koriste gotovi svi proizvođači pametnih telefona, ali i ostalih pametnih uređaja poput televizora, ručnih satova te tableta. Do datuma pisanja rada, izdano je 17 inačica Android OS-a kojima su uvedene brojne nove funkcije, ali i optimizirane već postojeće. Sve do 2019. godine Android inačice dobivale su nazive prema poznatim slasticama kao što je Android KitKat u slučaju inačice Android 4.4, međutim počevši od najnovije inačice Android 10, Google je u potpunosti napustio navedeni način imenovanja u svrhu bolje potrošačke razumljivosti, [3].



Grafikon 1. Tržišni udio operativnih sustava pametnih telefona

Izvor: [4]

Prema statistici iz srpnja 2020. godine, prikazanom grafikonom 2, najzastupljenija inačica Android OS-a je Android 9 sa 31,61% tržišnog udjela dok je najnovija Android 10 druga po redu s tržišnim udjelom od 24,76%.



Grafikon 2. Statistika raspodijele tržišne zastupljenosti Android inačica

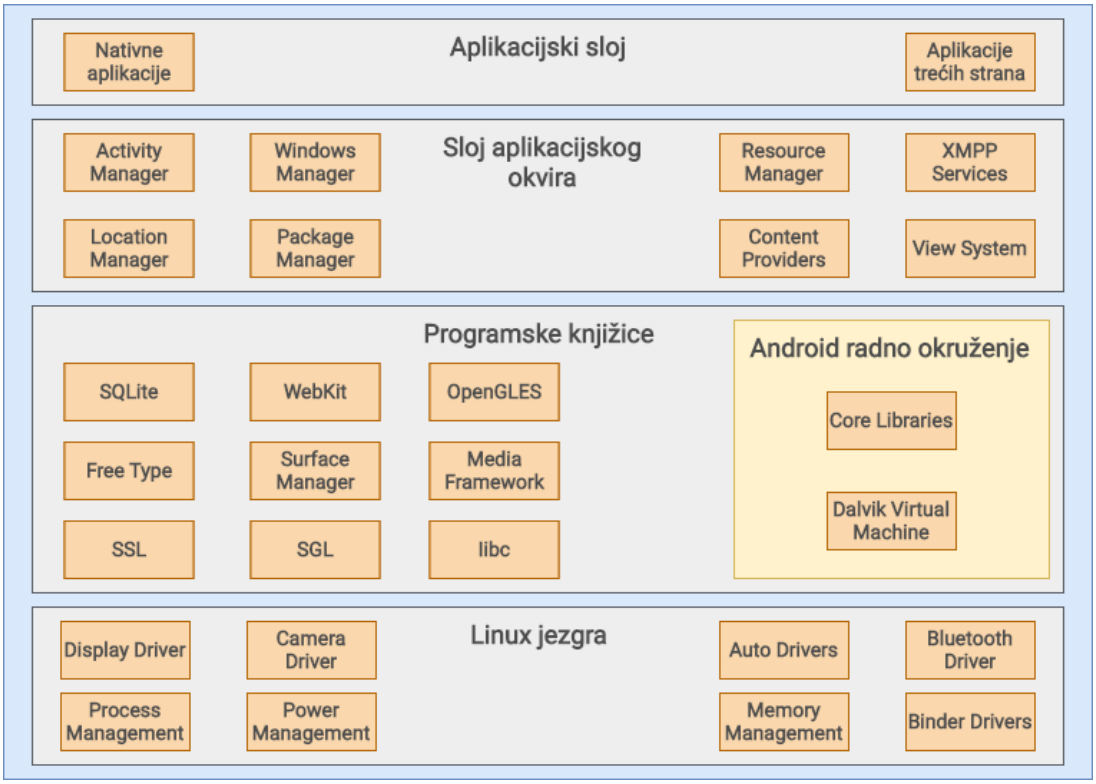
Izvor: [5]

U lipnju i srpnju 2020. godine, Google je omogućio javni pristup testnim inačicama OS-a Android 11. U odnosu na prijašnje inačice, Android 11 je usmjereniji na poboljšanje stabilnosti rada sustava uz manje izmjene korisničkog sučelja. Neke od do sada prikazanih izmjena su: dorađenije upravljanje obavijestima, implementacija „mjehurića“ za razgovor, dorađeniji izbornik napajanja te veća kontrola nad dozvolama. S obzirom na to da je navedena

inačica OS-a još u fazi razvoja, do kraja 2020. godine biti će objavljena još jedna testna verzija pri čemu bi konačna verzija trebala postati dostupna tijekom trećeg tromjesečja 2020. godine, [2].

2.1.1. Arhitektura sustava

Arhitektura Android OS-a strukturirana je slojevito kao što je to prikazano slikom 2, a potpuna funkcionalnost svakog sloja postiže se zajedničkim radom svih komponenti pojedinog sloja. Slojevi nižih razina pružaju usluge slojevima viših razina, a kao temelj arhitekture odabrana je Linux jezgra karakteristična po otvorenosti svog koda. S obzirom na to da jezgra sadrži upravljačke programe hardvera, dostupnost potpunog koda jezgre omogućava proizvođačima lakšu prilagodbu operativnog sustava različitom spektru uređaja.



Slika 2. Arhitektura Android sustava

Izvor: [6]

Sloj programskih knjižica sadrži komponente koje se koriste u svrhu dohvaćanja i obrade različitih vrsta podataka. U sklopu navedenog sloja također se nalazi i sloj Android radnog okruženja, čije komponente, kao što je virtualni stroj *dalvik* (eng. *Dalvik Virtual Machine* – DVM), pokreću aplikacije instalirane na uređaju. Kako bi pokrenute aplikacije

mogle koristiti osnovne funkcionalnosti uređaja, implementiran je sloj aplikacijskog okvira. Navedeni sloj je također jedini sloj Android arhitekture s kojim aplikacije mogu izravno komunicirati. Na samom vrhu arhitekture nalaze se aplikacijski sloj sačinjen od aplikacija prethodno ugrađenih u operativni sustav te aplikacija naknadno preuzetih putem Google Play trgovine, [6]. Ispod svih navedenih slojeva također se nalazi *U-Boot* sloj koji, iako nije dio Android arhitekture, se koristi u svrhu učitavanja podataka s različitih logičkih particija uređaja.

2.1.2. Struktura logičkih particija

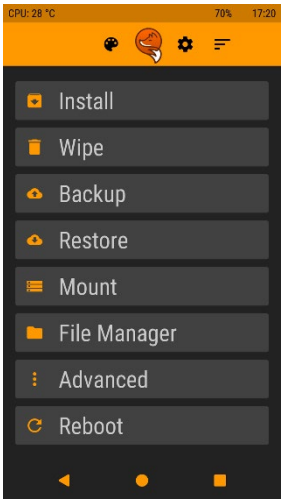
Prema [7], particije su logičke memorijske jedinice izrađene unutar trajne memorije uređaja, a koriste se kako bi se raspoloživi prostor raspodijelio na odjeljke kojima će softver zasebno pristupati prilikom pokretanja operativnog sustava. Struktura particija Android pametnih telefona može se razlikovati ovisno o proizvođaču uređaja i inačici sustava, međutim uobičajene particije koje se nalaze u većini uređaja su:

- */boot*
- */system*
- */recovery*
- */data*
- */cache*
- */mics*
- */sdcard*
- */sd-ext*.

Prva navedena *boot* particija sadrži sve podatke potrebne za podizanje sustava uređaja što uključuje Linux jezgru i *ramdisk* koji je korišten u svrhu pokretanja inicijalnih procesa. Druga navedena *system* particija sadrži isključivo podatke operativnog sustava te sve unaprijed instalirane sistemske aplikacije. S obzirom na to da ona ne sadrži korisničke podatke, njezinim brisanjem uklanja se samo operativni sustav kojeg je moguće ponovno instalirati koristeći *recovery* particiju.

Recovery particija sadrži odvojeni operativni sustav čije se funkcionalnosti izvode s *root* privilegijama, a omogućuju instalaciju novog operativnog sustava te ažuriranje postojećeg operativnog sustava. S obzirom na to da je tvornička inačica tog softvera ograničena brojem

funkcija, danas su također učestale modificirane inačice poput OrangeFox Recovery Projecta čije je sučelje prikazano slikom 3. Takve inačice omogućuju korisnicima prošireni zbir funkcionalnosti poput instalacije novog Android sustava, pravljenje sigurnosnih kopija podataka i pregledavanja datotečnog sustava uređaja.



Slika 3. Sučelje OrangeFox Recovery Project particije za oporavak, [8]

Particija *data*, također poznata kao particija korisničkih podataka (eng. *userdata partition*), sadrži podatke kao što su kontakti, poruke, povijest poziva, zapisi imenika, te svi ostali podaci koji mogu nastati korištenjem pametnog telefona. Svaka izmjena ove particije može utjecati na pohranjene korisničke podatke, te bi se njezinim brisanjem uređaj vratio na tvorničke postavke odnosno stanje uređaja u kojem je bio prilikom prvog pokretanja.

Cache particija pohranjuje podatke i komponente aplikacija kojima korisnik često pristupa. Podaci navedene particije automatski se obnavljaju korištenjem uređaja, te se njihovim brisanjem ne čine izmjene nad korisničkim podacima. Unatoč tome, takvi podaci mogu sadržavati informacije kao što su korisničke zaporke.

Particija *mics* sadrži sistemske informacije poput identifikacijskih brojeva operatora i regija, ali i informacije o postavkama hardvera poput konfiguracije univerzalne serijske sabirnice (eng. *Universal Serial Bus* – USB). Oštećenje ili potpuno uklanjanje ove particije može utjecati na ispravnost rada uređaja.

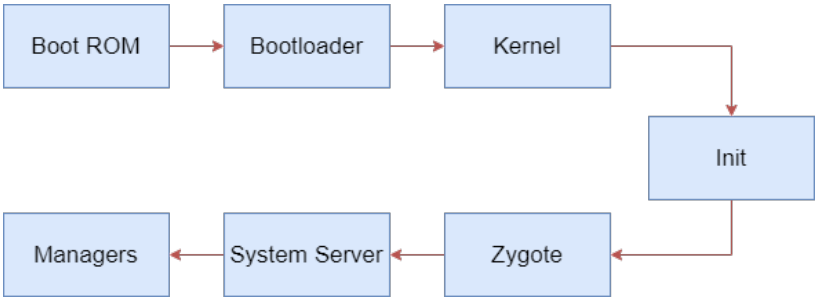
Particija *sdcard* predstavlja vanjsku pohranu uređaja na koju je moguće pohranjivati različite vrste multimedijskih datoteka, dokumenata te drugih vrsta podataka. Brisanje ove

particije ne utječe na rad uređaja, ali može dovesti do gubitka pohranjenih korisničkih podataka.

Iako se u slučaju tvorničkih inačica Android operativnih sustava rijetko koristi, također je učestala i *sd-ext* particija koju je moguće pronaći na uređajima koji koriste prilagođene inačice Android OS-a kao što je LineageOS. Riječ je o dodatnoj particiji koja koristi vanjsku pohranu u svrhu proširivanja unutarnje pohrane uređaja. Android OS uobičajeno sve aplikacije instalira isključivo na unutarnjoj memoriji kako bi se osigurala brzina njihovog pokretanja. Korištenjem *sd-ext* particije, aplikacije se u potpunosti pohranjuju na vanjsku pohranu u svrhu očuvanja slobodnog prostora unutarnje pohrane. S obzirom na to da navedena particija može sadržavati podatke instaliranih aplikacija, njezino brisanje bi imalo jednak utjecaj na uređaj kao i brisanje *data* particije, [7]. Ispravnost većine navedenih particija ključna je u svrhu uspješnog pokretanja Android pametnih telefona.

2.1.3. Tijek pokretanja uređaja

Uključivanjem Android uređaja pokreće se tijek podizanja Android operativnog sustava (eng. *Android booting sequence*) čiji je redoslijed aktiviranja komponenata prikazan slikom 4. Prvom fazom pokreće se *BootROM* ugrađen u centralnu procesorsku jedinicu (eng. *Central Processing Unit* – CPU), a koristi se u svrhu učitavanja programskog koda *bootloadera* u pohranu s nasumičnim pristupom (eng. *Random Access Memory* - RAM). *Bootloader* je dio programskog koda kojim se provjerava integritet *recovery* i *boot* particija, a također se koristi u svrhu postavljanja minimalnog okruženja u kojem će OS biti pokrenut. Nadalje, *bootloader* sadrži *init.s* i *main.s* podkomponente kojima se pokreće hardver uređaja te osnovni procesi operativnog sustava, [9].



Slika 4. Grafički prikaz tijeka pokretanja Android uređaja

Mijenjanje prethodno spomenutih logičkih particija, kao što je u slučaju *root* postupka, zahtjeva prethodno izvođenje postupka otključavanja *bootloadera* odnosno isključivanje

provjere integriteta *recovery* i *boot* particija. Većina proizvođača pametnih telefona svojim korisnicima omogućuje otključavanje *bootloadera* uz uvjet preuzimanja pune odgovornosti za svaku nastalu štetnu posljedicu poput gubitka ili krađe privatnih podataka, te mehaničkog kvara uređaja, [9].

Idućom fazom očitava se jezgra sustava koja pokreće:

- pohranu podataka
- predmemoriju uređaja
- ulazno/izlazne jedinice
- upravljačke programe hardvera uređaja
- *init* pokretački (eng. *initialization*) proces.

Init je prvi proces koji se pokreće prilikom postavljanja korisničkog prostora. Njime se pokreću *daemon* procesi koji postavljaju *dev*, *sys* i *proc* direktorije te upravljaju radom hardvera niže razine. Takvi procesi ne posjeduju vlastito programsko sučelje, pa se stoga izvršavaju isključivo u pozadini operativnog sustava. Jedan od prvih procesa koje *init* pokreće prilikom podizanja sustava je *Zygote*.

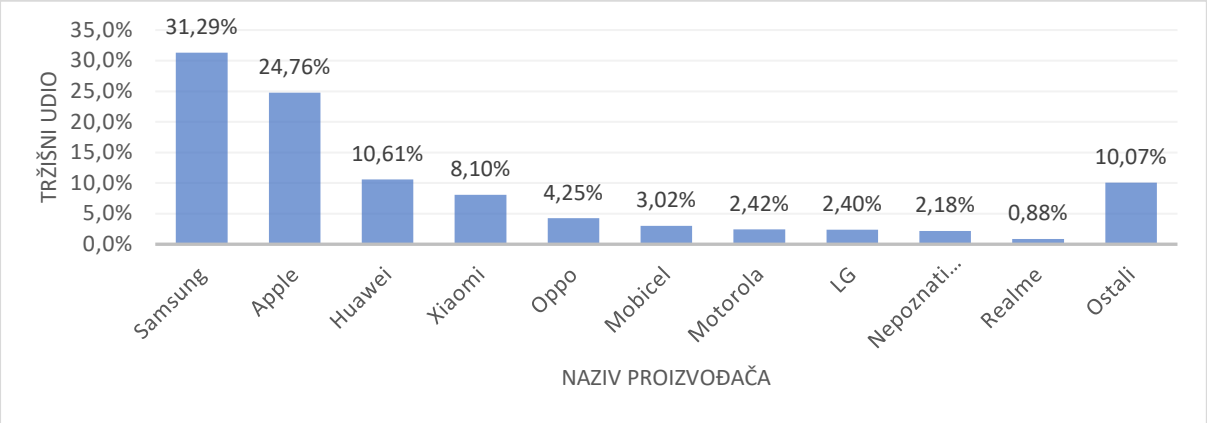
U svrhu bržeg pokretanja aplikacija, *Zygote* proces unaprijed očitava sve sistemske resurse i klase koje koristi Android radni okvir. Također pokreće virtualni stroj *dalvik* i poslužitelj sustava (eng. *System Server*) koji evidentira i pokreće osnovne funkcionalnosti uređaja kao što je korisničko sučelje. Sve pokrenute aktivnosti evidentiraju se u upravitelju aktivnosti (eng. *Activity Manager*) koji upravlja životnim vijekom svakog pokrenutog procesa. Kraj podizanja operativnog sustava završava emitiranjem *ACTION_BOOT_COMPLETED* poruke, [9].

2.2. Značajke iOS operativnog sustava

Operativni sustav iOS razvila je tvrtka Apple za sve svoje prenosive uređaje kao što su iPhone, iPod Touch te sve donedavno iPad. Sustav je predstavljen u siječnju 2007. godine na Macworld konferenciji gdje je također predstavljen i prvi iPhone uređaj. Prvo neslužbeno ime sustava bilo je OS X, međutim nekoliko mjeseci kasnije, izlaskom iPhone telefona na tržište, sustav je dobio svoje prvo službeno ime, iPhone OS. Apple je koristio to ime sve do 2010.

godine kada je na tržište izašlo prvo iPad tablet računalo nakon čega je naziv sustava preimenovan u iOS, [10].

Kao što je prikazano grafikonom 1 u potpoglavlju 2.1, iOS je drugi najpopularniji operativni sustav pametnih telefona nakon Android OS-a. Unatoč visokoj poziciji, razlika u tržišnom udjelu operativnih sustava pametnih telefona iznosi gotovo 50%. Do velike razlike u postotku tržišnog udjela dovela je otvorenost koda Android OS-a, ali i nedostatak alternativnih operativni sustava koje proizvođači mogu implementirati u svoje uređaje. Unatoč tome, prema statistici prikazanom grafikonom 3, Appleovi pametni telefoni su drugi najkorišteniji uređaji na tržištu pametnih telefona sa 24,76% tržišnog udjela.



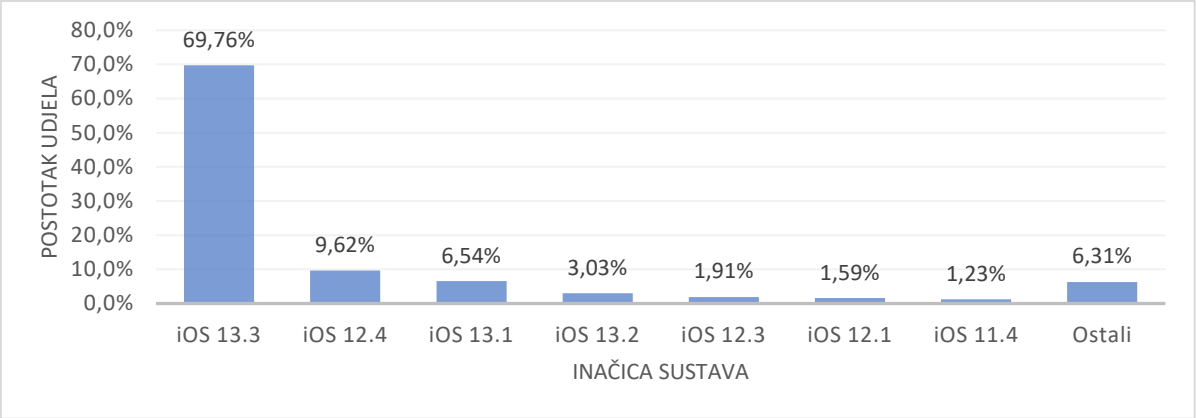
Grafikon 3. Prikaz tržišnog udjela proizvođača na tržištu pametnih telefona u siječnju 2020. godine

Izvor: [11]

Od 2007. godine pa do datuma pisanja rada, Apple je na tržište izdao sveukupno 13 glavnih inačica sustava, od kojih je zadnja iOS 13 inačica izašla u lipnju 2019. godine. Iste godine je Apple također predstavio operativni sustav namijenjen iPad uređajima imena iPadOS. U svrhu ažuriranja sustava, ranije inačice sustava koristile su iTunes aplikaciju koja je preuzimala kopiju iOS sustava te ju instalirala na uređaj. Nedostatak ovakvog oblika ažuriranja je bio taj što je svaka nadogradnja sustava zahtijevala preuzimanje potpunog instalacijskog paketa operativnog sustava. Iz tog razloga uvedena je *Over-the-air* (OTA) metoda ažuriranja kojom su se preuzimale samo one komponente sustava koje je bilo potrebno ažurirati, [12].

S obzirom na zastarjelost hardvera, neki uređaji su već izgubili podršku za novijim inačicama što ih čini sigurnosno najugroženijima. U vrijeme pisanja rada, najstariji uređaji koji

još uvijek podržavaju posljednju iOS 13 inačicu su iPhone 6S i iPhone 6S Plus. Kao što je vidljivo na grafikonu 4, u siječnju 2020. godine, iOS 13 je bio korišten na skoro 70% svih aktivnih iPhone pametnih telefona u svijetu, pri čemu su starije inačice korištene na puno manjem broju iPhone telefona.



Grafikon 4. Zastupljenost iOS inačica u svijetu

Izvor: [13]

2.2.1. Arhitektura sustava

Operativni sustav iOS nastao je na temelju macOS operativnog sustava kojem je dodatno prilagođena jezgra sustava u svrhu učinkovitijeg funkcioniranja na uređajima s ograničenim hardverskim komponentama. Ključne razlike ova dva sustava su:

- iOS uređaji koriste *Advanced RISC Machine* (ARM) arhitekturu umjesto Intel x86_64 arhitekture
- zatvorenost iOS jezgre u odnosu na otvorenost macOS jezgre
- zabranjen pristup temeljnim API-ima iOS sustava
- macOS koristi AppKit grafičko korisničko sučelje, a iOS koristi UIKit grafičko korisničko sučelje.

S obzirom na to da aplikacije ne mogu izravno komunicirati s hardverom uređaja, operativni sustav služi kao posrednik između hardvera i softvera uređaja. Sučelja korištena u tu svrhu jasno su definirana strukturom prikazanom slikom 5. Arhitektura se sastoji od četiri sloja pri čemu svaki sloj sadržava određene radne okvire potrebne za ispravno funkcioniranje uređaja, [6].

Slika 5. Arhitektura iOS operativnog sustava

Izvor: [6]

Cocoa touch sloj sadrži radne okvire potrebne za pružanje vizualnog sučelja iOS aplikacija. Radni okviri ovog sloja omogućuju korištenje ključnih funkcionalnosti kao što su višezadaćnost, unos temeljen na dodiru te sve ostale usluge više razine. Sljedećim *media* slojem definira se cjelokupna multimedijska arhitektura iOS uređaja što uključuje audio i video radne okvire, [6].

Core services sloj pruža osnovne sistemske usluge kao što su funkcionalnost lokacije, funkcionalnost iCloud pohrane, mogućnost povezivanja na društvene mreže i mnoge druge. Aplikacije pri tome ne koriste sve funkcionalnosti sloja, već isključivo one potrebne za njihov ispravan rad. Posljednji, *Core OS* sloj sadrži jezgru iOS operativnog sustava koja pruža sučelje prema hardveru uređaja, a odgovorna je za upravljanje:

- pohranom uređaja
- životnim ciklusom procesa
- međuprocesnom komunikacijom
- umrežavanjem
- pristupom vanjskoj opremi
- strukturom datotečnog sustava, [6].

2.2.2. Struktura logičkih particija

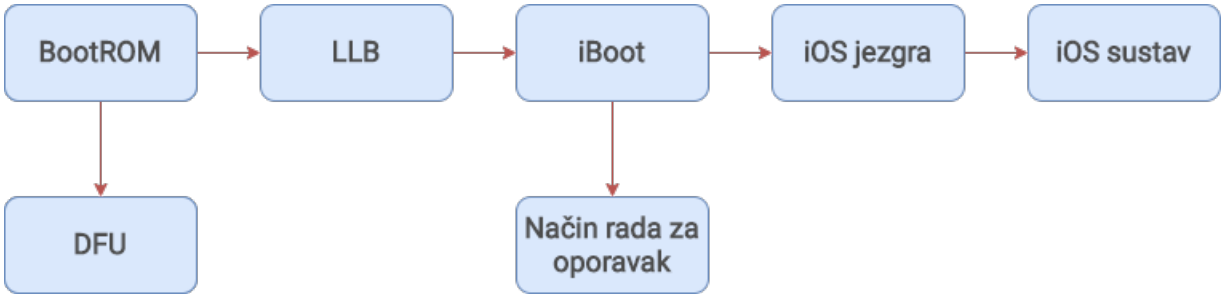
Strukturu datotečnog sustava iOS-a moguće je podijeliti na dvije logičke particije: sistemsku particiju i particiju korisničkih podataka. Sistemska particija sadrži operativni sustav i sve unaprijed instalirane aplikacije. Prema zadanim postavkama, ona je očitana kao *read-only*

particija te zauzima mali dio skladišnog prostora koji uobičajeno iznosi između 0.9 GB i 2.7 GB. Korisnik ju ne može obrisati ili izmijeniti osim u slučaju postupka potpune nadogradnje operativnog sustava. S obzirom na to da je sistemska particija odvojena od particije korisničkih podataka, nadogradnja sustava ne utječe na skladištene korisničke podatke. U pogledu forenzičke analize, sistemska particija ne sadrži potencijalne dokazne podatke, ali je pomoću nje moguće utvrditi jesu li nad uređajem učinjene modifikacije kao što je *jailbreak* postupak.

Particija korisničkih podataka sadrži sve podatke koje je korisnik stvorio poput zapisa kontakata, preuzetih ili stvorenih datoteke različitih formata, podataka aplikacija i ostale vrste podataka koje mogu imati dokaznu vrijednost u slučaju forenzičke analize, [6].

2.2.3. Tijek pokretanja uređaja

Tijek pokretanja iOS uređaja, također poznat kao sigurnosni lanac podizanja sustava (eng. *Secure Boot Chain*), sastoji se od više faza čije su osnovne komponente prikazane slikom 6. Sve komponente su kriptografski potpisane u svrhu osiguravanja integriteta operativnog sustava. Iz tog razloga, svaku iduću komponentu je moguće pokrenuti tek onda kada se prethodnoj komponenti potvrdi njezina autentičnost, [12].



Slika 6. Tijek pokretanja iOS uređaja

Prva komponenta koja se pokreće prilikom podizanja sustava je *BootROM*. Riječ je o kodu koji je, prilikom proizvodnje, hardverski implementiran u *read-only* pohranu procesora. On sadrži javni Apple Root certifikat kojim se provjerava integritet iduće komponente, a budući da se radi o prvom kodu kojeg uređaj pokreće, njegova ranjivost može ugroziti svaku sljedeću fazu pokretanja iOS uređaja, [12].

Ako, tijekom pokretanja uređaja, *BootROM* ne može potvrditi autentičnost iduće komponente, na starijim modelima se pokreće način rada za nadogradnju upravljačkog softvera (eng. *Device Firmware Upgrade* – DFU), dok se na novijim pokreće način rada za oporavak

(eng. R*ecovery mode*), [12]. Pomoću DFU načina rada moguće je ažurirati ili zamijeniti *firmware* iOS uređaja, a najčešće se koristi u svrhu ispravljanja grešaka nastalih ažuriranjem sustava ili vračanja prethodnih inačica operativnog sustava.

Pokretanjem DFU načina, uređaj prolazi kroz drugačiji tijek pokretanja prikazan slikom 7. Prvom fazom, *BootROM* kod pokreće iBSS (eng. *iBoot Single Stage*) i iBEC (eng. *iBoot Epoch Change*) koji su zaduženi za pokretanje i provjeru integriteta jezgre sustava. Pokrenuta jezgra provjerava *ramdisk* te ga očitava u memoriju, nakon čega *ramdisk* očitava instalacijsku datoteku OS-a, [6].

Slika 7. Alternativni tijek pokretanja uređaja

Izvor: [6]

U slučaju da je *BootROM* uspješno potvrdio sljedeću komponentu, idućom fazom pokreće se *bootloader* sačinjen od *bootloadera* niže razine (eng. *Low Level Bootloader* - LLB) te iBoot komponente. *Bootloader* niže razine pokreće rutinske operacije kao što je pronalazak iBoot particije unutar *flash* memorije te provjera njezine autentičnosti. U slučaju da nije moguće potvrditi autentičnost iBoot particije, pokreće se način rada za oporavak.

Način rada za oporavak je vrsta sigurnosnog mehanizma kojom se pokreće softverski paket pomoću kojeg se provode postupci ažuriranja i popravka operativnog sustava. Osnovna razlika načina rada za oporavak u odnosu na DFU način rada jest ta što DFU ne pokreće iBoot komponentu koja sprječava instalaciju starijih inačica sustava. Međutim, u oba slučaja je uređaj potrebno povezati s računalom pomoću USB kabla kako bi putem iTunes programa bilo moguće pokrenuti postupke oporavka i ažuriranja sustava, [12], [14].

Druga, iBoot komponenta provjerava autentičnost iOS jezgre, te ju pokreće zajedno s NAND *flash* memorijom. S obzirom na to da je implementirana softverski, sve sigurnosne ranjivosti moguće je ispraviti softverskim ažuriranjem. Kraj tijeka pokretanja operativnog sustava završava pokretanjem jezgre sustava te *ramdisk* komponente, nakon čega se korisniku po prvi puta pruža pristup korisničkom sučelju uređaja, [15].

3. Sigurnosne ranjivosti pametnih telefona

Uzevši u obzir svestranost funkcija, lakoću prenosivosti ali i mogućnost pohrane velike količine podataka, sigurnost telefona postala je ključan zahtjev kojeg svaki proizvođač pametnih uređaja nastoji udovoljiti. Iako analiza pametnih telefona može pripomoći legitimnim radnjama kao što su kriminalne istrage, njih je također moguće koristiti u svrhu brojnih zlonamjernih radnji poput nedozvoljene upotrebe privatnih podataka. U svrhu sprječavanja takvih radnji implementira se sigurnosno okruženje segmentirano na softverske i hardverske komponente čiji se rad međusobno mora nadopunjavati.

S gledišta softverskih komponenti, Android i iOS operativni sustavi koriste, na prvi pogled, slične sigurnosne modele. Najvažnije značajke svakog sigurnosnog modela prikazane su tablicom 1, pri čemu je naglasak stavljen na sigurnost skladištenih podataka.

Tablica 1. Usporedba sigurnosnih značajki iOS i Android operativnih sustava

	Android	**iOS**
Izolacija aplikacija	Različit model izolacije za svaku aplikaciju	Identičan model izolacije za sve aplikacije
Potpisivanje koda aplikacija	Upotreba vlastitih certifikata provjere koda	Upotreba iOS certifikata provjere koda
Način šifriranja podataka	Šifriranje cijelog diska ili šifriranje na osnovi datoteka, ovisno o inačici sustava	Hardverski implementirana mehanika šifriranja na osnovi datoteka
Vrsta pohrane podataka	Unutarnja pohrana uz mogućnost korištenja vanjske pohrane na određenim uređajima.	Unutarnja pohrana
Mogućnost zaključavanja zaslona	Zaključavanje temeljeno na uzorku, pinu/pristupnom kodu, otisku prsta, licu	Face ID, TouchID, Zaključavanje temeljeno na pinu

Izvor: [6], [16]

Jedan od važnijih elemenata sigurnosnih modela je mehanika šifriranja podataka kojom se nastoji očuvati sigurnost podataka čak i u slučajevima gdje je podatke bilo uspješno oporaviti s uređaja. Google je inačicom Android 3 uveo mehaniku šifriranja cijelog diska (eng. *Full-disk encryption* – FDE) koja, prilikom prvog pokretanja uređaja, stvara jedinstveni 128-bitni ključ pomoću kojeg se šifriraju svi podaci prije pohranjivanja na disk uređaja. Android inačicom 7.0. uvedeno je šifriranje na osnovni datoteka (eng. *File-based encryption* – FBE) koje je u potpunosti zamijenilo prethodnu FDE metodu na novijim inačicama Android OS-a. FBE metodom svaka se datoteka šifrira pomoću jedinstvenog ključa što otežava probijanje zaštite.

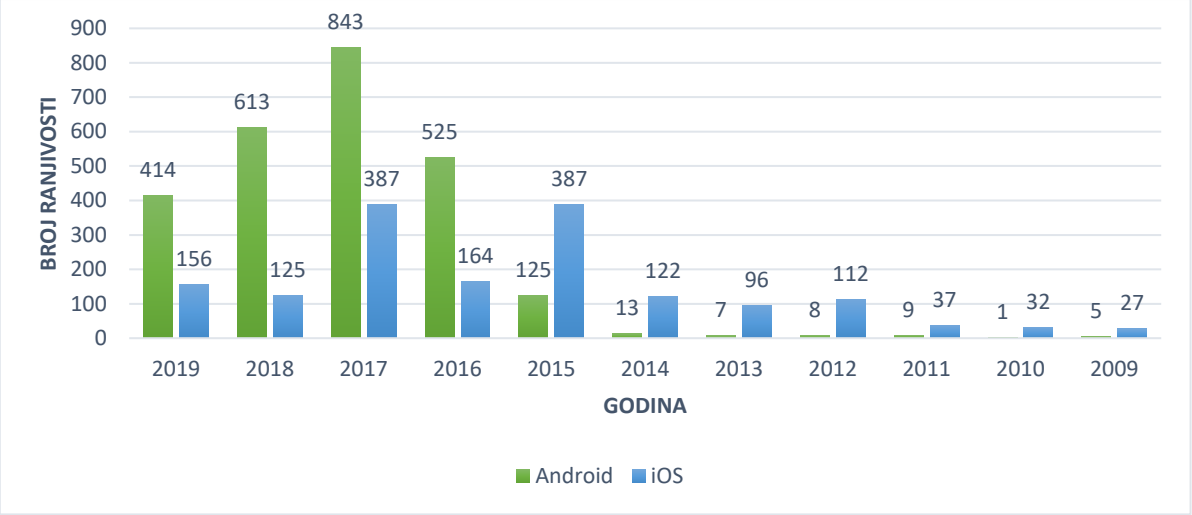
S druge strane, iOS operativni sustavi koristi mehanizam hardverskog šifriranje na osnovi datoteka. Ključevi, korišteni u svrhu šifriranja podataka, se spremaju unutar sigurnosne enklave, odnosno hardverske komponente Apple A serije procesora. S obzirom na to da se rukovanje ključevima odvija isključivo unutar sigurnosne enklave, koja je odvojena od operativnog sustava, bilo kakav pokušaj dešifriranja podataka je teško izvediv čak i u slučaju prethodnog ostvarivanja pristupa unutarnjoj pohrani telefona, [6], [16] [12].

S aspekta hardverskih komponenata, pametni telefoni temeljeni na Android operativnom sustavu nisu ograničeni na jednog proizvođača kao što je u slučaju iOS operativnog sustava. Upravo zbog toga se sigurnost svakog modela Android telefona razlikuje ovisno o korištenim hardverskim komponentima koje mogu sadržavati različite sigurnosne ranjivosti. U svijetu cyber sigurnosti, sigurnosne ranjivosti su nenamjerne mane računalnih sustava, programa te hardvera koje narušavaju sigurnost uređaja, a nastaju greškom programskog koda ili nepravilnom računalnom i sigurnosnom konfiguracijom. Osnovna obilježja sigurnosnih ranjivosti moguće je definirati prema:

- utjecaju kojeg sigurnosne ranjivosti mogu imati na sigurnost uređaja
- podložnosti određenog modela uređaja na otkrivenu sigurnosnu ranjivost
- ispravljivosti sigurnosnih ranjivosti pri čemu je neke ranjivosti moguće ispraviti softverskim ažuriranjem sustava ili isključivo revizijom hardverskih komponenti
- upoznatosti šire javnost s postojanjem sigurnosne ranjivosti
- preduvjetima iskorištavanja otkrivene sigurnosne ranjivosti npr. isključivo uz izravan pristup uređaju ili uz mogućnost udaljenog pristupa uređaju, [17].

Same po sebi, sigurnosne ranjivosti ne mogu naštetiti uređaju, ali mogu biti iskorištene u svrhu izvršavanja zlonamjernog koda. Računalni softver koji iskorištava sigurnosne ranjivosti u svrhu izazivanja nenamjernog ili nepredviđenog ponašanja naziva se *exploit*, a moguće ga je podijeliti na poznate *exploitove* i nepoznate *exploitove*. Poznati *exploitovi* su već otprije poznati proizvođačima hardvera ili softvera te su najčešće ispravljeni u vrlo kratkom vremenskom roku. Nepoznati *exploitovi*, također poznati kao *zero day exploitovi*, sadrže kritičnu ranjivost sustava ili uređaja s kojom proizvođači još uvijek nisu upoznati, pa se stoga najčešće koriste u svrhu izvođenja zlonamjernog softvera, [18].

Kao što je prikazanom grafikonom 5, na Android operativnom sustavu je od 2009. godine do 2019. godine otkriveno 2563 sigurnosnih ranjivosti, dok je na iOS sustavu za isto razdoblje otkriveno 1645 sigurnosnih ranjivosti. Nadalje, prema bazi podataka sigurnosnih ranjivosti Nacionalnog instituta za standarde i tehnologiju (eng. *National Institute of Standards and Technology* – NIST), Android je 2019. godine također bio operativni sustav s najvećim brojem otkrivenih sigurnosnih ranjivosti pri čemu su također uzeti u obzir i ostali operativnih sustavi drugih elektroničkih uređaja poput stolnih računala.



Grafikon 5. Statistika sigurnosnih ranjivosti

Izvor: [19], [20].

Unatoč velikoj razlici u brojevima, sigurnost operativnog sustava ne može biti temeljena isključivo na broju otkrivenih sigurnosnih ranjivosti. Razlog tome je velika raznovrsnost hardvera Android uređaja, ali i razlika u utjecaju sigurnosne ranjivosti na cjelokupni sustav uređaja. U svrhu sprječavanja onih sigurnosnih ranjivosti koje mogu imati

veći utjecaj na sigurnosti model uređaja, potrebno ih je pravovremeno otkriti prije nego što one budu iskorištene od strane zlonamjernih pojedinaca.

3.1. Otkrivanje sigurnosnih ranjivosti

Za pravovremeno otkrivanje sigurnosnih ranjivosti zaduženi su stručnjaci u hardverskom i softverskom području. Ovisno o broju stručnjaka koji međusobno surađuju moguće ih je podijeliti na:

- pojedince
- timove
- korporacije.

Stručnjaci slobodnjaci također poznati kao hakeri bijelih šešira (eng. *white hat hackers*) ispituju ranjivosti hardverskih i softverskih sustava u suradnji s proizvođačima hardvera ili softvera kojeg testiraju. Jedan primjer takve suradnje je Apple Security Bounty program osmišljen u svrhu prijavljivanja sigurnosnih ranjivosti javno dostupnih verzija Apple operativnih sustava kao što su iOS-a, iPadOS-a, macOS, tvOS te watchOS. Glavna misija programa je otkrivanje sigurnosnih prijetnji te nagrađivanje pojedinaca zaslužnih za njihovo otkrivanje. Novčana nagrada izdaje se samo za one sigurnosne ranjivosti koje su prethodno bile nepoznate javnosti, pri čemu je iznos određen kategorijom sigurnosne ranjivosti te kvalitetom zaprimljenog izvještaja. Kategorije uključuju područja kao što su: iCloud pohrana, hardverske ranjivosti, softverske ranjivosti te mrežne ranjivosti. Izvješće se smatra potpunim samo onda kada sadrži potpuni eksploatacijski lanac uz detaljan opis problemima, preduvjete te korake repliciranja sigurnosne ranjivosti, [21].

S druge strane postoje timovi oformljeni od strane većih kompanija kao što je Project Zero tim. Project Zero tim oformio je Google 2014. godine s ciljem otkrivanja sigurnosnih ranjivosti svih hardverskih komponenti i operativnih sustava neovisno o proizvođaču. Svaka otkrivena sigurnosna ranjivost pohranjuje se u javnu bazu podataka. Kao sigurnosnu mjeru, informacije o sigurnosnim ranjivostima ne sadržavaju potpuni eksploatacijski lanac kako ne bi mogle biti iskorištene u zlonamjerne svrhe. Također, proizvođači proizvoda zahvaćenih sigurnosnom ranjivosti, obavještavaju se 90 dana prije javnog objavljivanja kako bi ranjivosti mogle na vrijeme biti ispravljene. Transparentnost javnog objavljivanja pronađenih

sigurnosnih ranjivosti omogućuje pokretanje javnih rasprava u svrhu poboljšavanja softverske i hardverske sigurnosti ali i edukacije novih istraživača sigurnosne zajednice, [22].

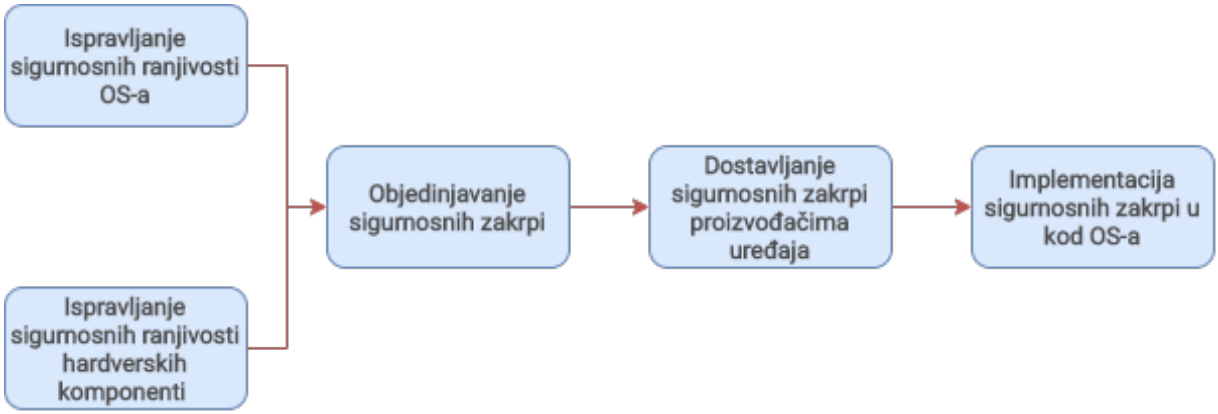
Osim spomenutih grupacija i pojedinaca slobodnjaka, također postoje i sigurnosne tvrtke poput Zerodiuma čija je glavna djelatnost otkrivanje i pribavljanje *zero-day* sigurnosnih ranjivosti u svrhu daljnje prodaje. Kupci otkrivenih sigurnosnih ranjivosti uglavnom su organizacija kao što su vladine organizacije ili velike korporacije u obrambenom, tehnološkom te financijskom sektoru.

Izvješća poslovanja tvrtki kao što je Zerodium najčešće se koriste u svrhu definiranja vrijednosti određenih kategorija sigurnosnih ranjivosti. Kao primjer tome, može se navesti izvješće iz 2019. godine, prema kojem je vrijednost Android sigurnosnih ranjivosti premašila vrijednost iOS sigurnosnih ranjivosti. Kao razlog tome navodi se velika raznolikost hardverskih komponenti Android uređaja ali veći broj stručnjaka koji svojim zajedničkim radom nastoje unaprijediti sigurnost sustava, [23]. Neovisno o načinu otkrivanja sigurnosnih ranjivosti, kako sigurnost uređaja ne bi bila ugrožena, iste je potrebno ispraviti u što kraćem roku putem sigurnosnih zakrpi.

3.2. Implementacija sigurnosnih zakrpi

Sigurnosne zakrpe su oblik ažuriranja operativnog sustava kojim se prave manje izmjene u svrhu ispravljanja sigurnosnih ranjivosti. S obzirom na izvedivost *root* postupka, a naročito *jailbreak* postupka, učestalost izdavanja sigurnosnih zakrpi predstavlja važan faktor za uspješnost provođenja postupaka.

U pogledu Androida, Google sigurnosne zakrpe isporučuje svaki mjesec za sve glavne inačice sustava. Međutim, zbog kompleksnosti koda Android OS-a, ali i raznolikosti uređaja koji ga koriste, postupak implementacije sigurnosnih zakrpi zahtjeva korake prikazane slikom 8, [24].



Slika 8. Koraci implementacije sigurnosnih zakrpi na Android pametnim telefonima

Sigurnosne ranjivosti Android operativnog sustava ispravlja Google, dok sve ostale, hardverski povezane sigurnosne ranjivosti, ispravlja proizvođač hardvera. U slučaju hardverski uzrokovanih ranjivosti, svaku sigurnosnu zakrpu je također potrebno proslijediti Googleu kako bi ona mogla biti distribuirana na odgovarajuće uređaje. S obzirom na to da Android OS koriste uređaji koji nisu nužno proizvedeni od strane Google-a, svaka sigurnosna zakrpa mora biti dostavljena proizvođačima pametnih telefona. Riječ je o proizvođačima kao što su Samsung, Xiaomi, Sony, Lg te čak i Google u pogledu njihove Pixel serije pametnih telefona.

Svaki od navedenih proizvođača koristi vlastitu modificiranu verziju operativnog sustava u kombinaciji s različitim hardverskim komponentama. Iz tog razloga, implementacija sigurnosnih zakrpi također zahtjeva odstranjivanje nepotrebnih ili već implementiranih zakrpi. Prije nego što konačna skupina sigurnosnih zakrpi bude puštena krajnjim korisnicima, istu je potrebno testirati kako bi se utvrdili mogući nedostaci.

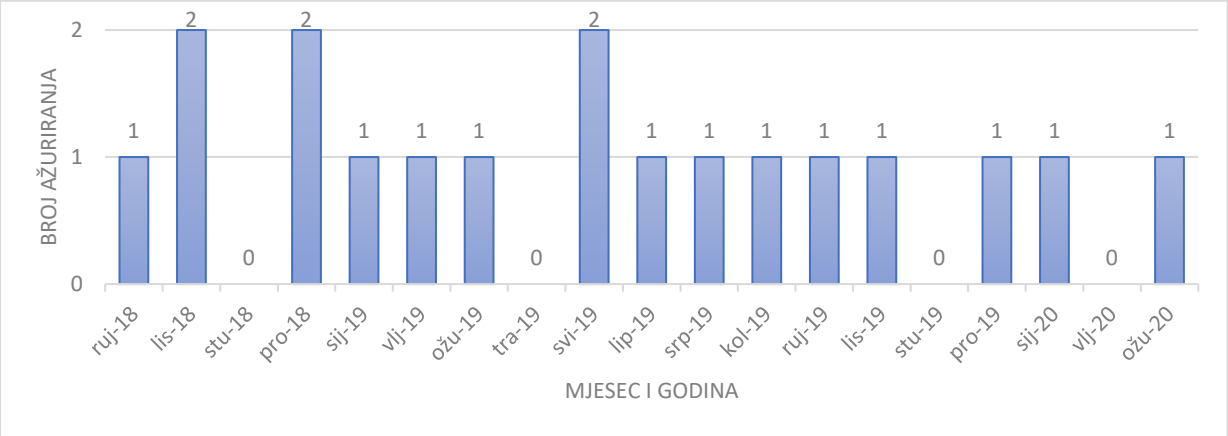
Posljednjom fazom se sigurnosne zakrpe dostavljaju korisničkim uređajima putem Interneta. U tu svrhu koristi se OTA softver koji preuzima, provjerava te instalira sve datoteke sigurnosne zakrpe. U slučajevima kada se izmjenjuju trenutno aktivni procesi sustava, uređaj je također potrebno ponovno pokrenuti kako bi sigurnosne zakrpe bile ispravno instalirane i primijenjene. Nadalje, u svrhu dodatnih mjera opreza, ali i s ciljem sprječavanja zagušenja servera, sigurnosne zakrpe dostavljaju se na korisničke uređaje u nekoliko navrata sve dok svi uređaji ne budu zahvaćeni, [25].

Kompleksnost postupka i manja zainteresiranost proizvođača u tom pogledu dovela je do problema neredovitog implementiranja sigurnosnih zakrpi. Iz tog razloga, Google je obvezao proizvođače Android pametnih telefona na obvezno izdavanje sigurnosnih zakrpi

najmanje dvije godine od početka prodaje uređaja. Točnije, tijekom prve godine proizvođač je obvezan osigurati najmanje četiri sigurnosna ažuriranja s maksimalnim vremenskim razmakom od 90 dana, dok za drugu godinu broj sigurnosnih zakrpi nije definiran. U slučaju nepoštivanja navedenih uvjeta ugovora, Google zadržava pravo uskratiti proizvođaču licencu korištenja Google servisa na budućim modelima, [26].

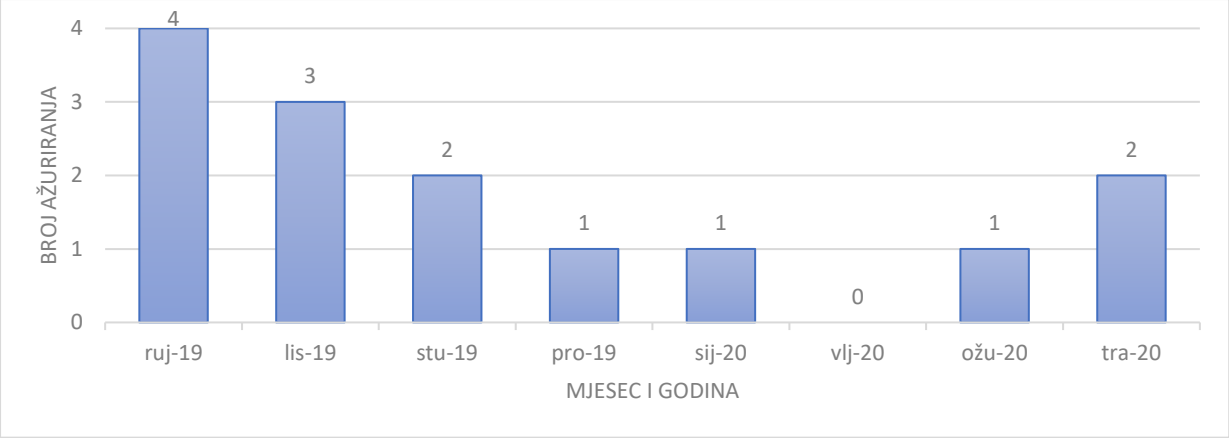
U odnosu na Android, postupak implementacije sigurnosnih zakrpi na iOS sustavu je mnogo jednostavniji. Glavni razlog tome je manja raznovrsnost uređaja koji koriste iOS sustav. Zbog toga što iOS koriste isključivo oni uređaji proizvedeni od strane Applea, ispravci se izdaju u obliku unificiranih sigurnosnih zakrpi diljem svih podržanih iPhone modela, ali i drugih Appleovih uređaja kao što su iPad touch te starije inačice iPad uređaja.

Od službenog izdavanja iPhone 4S modela, svaki idući model podržavao je minimalno četiri veće nadogradnje sustava, nakon čega se podrška nastavila u pogledu sigurnosnih ažuriranja. Iako učestalost sigurnosnih ažuriranja nije jasno navedena od strane Applea, iz grafikona 6 i 7 vidljivo je kako su zadnje dvije inačice sustava dobivale minimalno jedno sigurnosno ažuriranje gotovo svaki mjesec od datuma izdavanja.



Grafikon 6. Broj sigurnosnih ažuriranja iOS 12 inačice izdanih nakon prvog datuma izdavanja

Izvor: [27]



Grafikon 7. Broj sigurnosnih ažuriranja iOS 13 inačice izdanih nakon prvog datuma izdavanja

Izvor: [27]

S obzirom na to da je iOS 12 inačica dobila 5 sigurnosnih zakrpi nakon izlaska iOS 13 inačice, također je moguće zaključiti kako starije inačice nastavljaju dobivati sigurnosna ažuriranja čak i nakon izlaska novih inačica, iako u puno manjem broju.

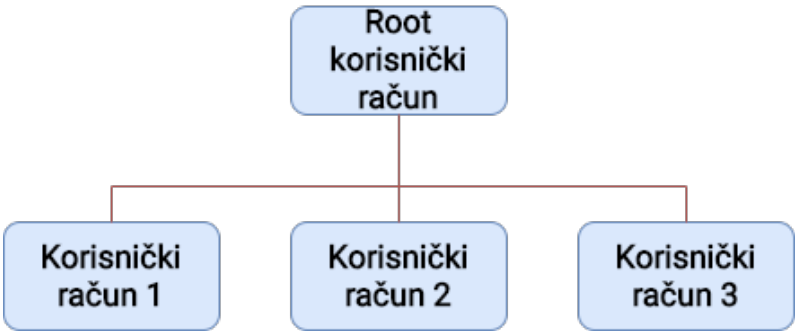
4. Značajke i procedura eskalacije privilegija

Kako bi aplikacije mogle izvršavati vlastiti programski kod moraju od operativnog sustava zadobiti određene privilegije. Privilegije predstavljaju skupove sigurnosnih značajki, a koriste se u svrhu ograničavanja pristupa određenim podacima ili procesima. Programeri uobičajenim korisničkim računima, odnosno aplikacijama, pridružuju najmanji skup privilegija kako bi spriječili:

- nenamjernu izmjenu osjetljivog programskog koda
- nenamjerno brisanje datoteka ključnih za funkcioniranje sustava
- krađu podataka
- nedozvoljeno nadziranje korisničkih aktivnosti.

Pristup pregledavanju i izmjeni osjetljivih podataka ima isključivo administrativni račun ili *root* račun u slučaju Unix operativnih sustava. Iz tog razloga provodi se postupak eskalacije privilegija, najčešće iskorištavanjem nenamjernih grešaka softvera ili hardvera, u svrhu osiguravanja većeg skupa privilegija. Slika 9 predstavlja uobičajeno stablo raspodjele privilegija korisničkih računa na operativnim sustavima. Dvije osnovne vrste eskalacije privilegija su:

- Horizontalna eskalacija privilegija, korisnik s osnovnim skupom privilegija ostvaruje pristup podacima drugih korisničkih računa osnovnih privilegija npr. Korisnički račun 1 može ostvariti pristup podacima Korisničkog računa 2.
- Vertikalna eskalacija privilegija, korisnik s osnovnim skupom privilegija ostvaruje privilegije Root korisničkog računa, [28], [29].



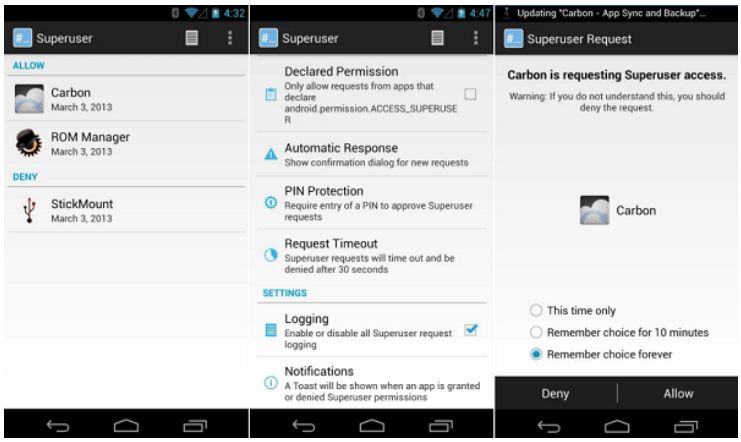
Slika 9. Stablo raspodjele privilegija korisničkih računa

Root i *jailbreak* postupci su primjeri vertikalne eskalacije privilegija na Android i iOS operativnim sustavima, a njihove osnovne značajke i procedure detaljnije su objašnjene idućim potpoglavljima.

4.1. Značajke *root* postupka

Operativni sustavi temeljeni na Linux jezgri koriste naziv *root* kako bi definirali korisnički račun koji ima mogućnost izvršavanja svih naredbi sustava i pristupa svim datotekama uređaja. U praksi također mogu biti korišteni nazivi kao što su *root account*, *root user* te *superuser*. Proizvođači Android pametnih telefona korisnicima ne dopuštaju upotrebu *root* korisničkog računa što je dovelo do popularnosti *root* postupka, [30].

Root postupak temelji se na *su* (eng. *Switch user*) izvršnoj binarnoj datoteci koja procesima aplikacija omogućuje izmjenu privilegija korisničkih računa. Uz *su* datoteku uobičajeno dolazi i popratna aplikacija s definiranim grafičkim sučeljem unutar kojeg se obavlja kontrola danih i odbijenih privilegija. Kao što je vidljivo sa slike 10, korisnik ima mogućnost odobriti ili uskratiti pristup *root* računu, uz mogućnost pamćenja odabira za svaki sljedeći upit, [31].



Slika 10. Sučelje Superuser aplikacije, [32]

Samo neke od prednosti obavljanja *root* postupka su:

- pokretanja ili zaustavljanja bilo kojeg sistemskog procesa
- uklanjanje neželjenog predinstaliranog softvera (eng. *bloatware*)
- uređivanje ili brisanje bilo koje datoteke uređaja

- modificiranje privilegija bilo kojeg korisnika odnosno aplikacije
- stvaranje sigurnosne kopije potpunog sustava zajedno s svim pohranjenim korisničkim podacima
- instalacija modificiranih inačica operativnog sustava, [31].

Unatoč tome, uređaji nad kojima je proveden *root* postupak su također podložniji sigurnosnim prijetnjama s obzirom na to da aplikacije mogu iskoristiti *root* privilegije u svrhu zlonamjernih radnji kao što su krađa podataka ili nedozvoljeno nadziranje korisnika.

4.1.1. Vrste *root* postupaka

Root postupke moguće je segmentirati prema:

- trajnosti postupka
  - *soft root*
  - *hard root*
- načinu izvođenja postupka
  - *system root*
  - *systemless root*.

Soft root metoda je u potpunosti softverski temeljena metoda te se postiže iskorištavanjem ranjivosti jezgre uređaja. Rezultat ove metode je privremen, a najčešće se koristi u zlonamjerne svrhe. S druge strane, hard *root* metoda zahtijeva fizičku interakciju s uređajem. Ona se postiže pokretanjem *recovery* načina rada unutar kojeg se pokreće instalacija *su* datoteke ili potpuna zamjena operativnog sustava s onim koji već u sebi ima ugrađenu *su* datoteku. Za razliku od *soft roota*, *hard root* nije privremena metoda no postupak provođenja može biti otežan te u nekim slučajevima i potpunosti onemogućen npr. nemogućnost izvođenja postupka otključavanja *bootloadera*, [31].

System root metodom se prave izmjene u sustavu kako bi se omogućila prethodno uklonjena s*uperuser* Linux funkcija, a koristi se na starijim inačicama Android OS-a. Pokretanjem *su* binarne datoteke, aplikacije od korisnika mogu zatražiti privilegije koje kao obični korisnički račun ne posjeduju. S obzirom na to da se u ovom slučaju izmjenjuje sustav uređaja, uklanjanjem *su* datoteke nije moguće prikriti tragove prethodno obavljenog postupaka.

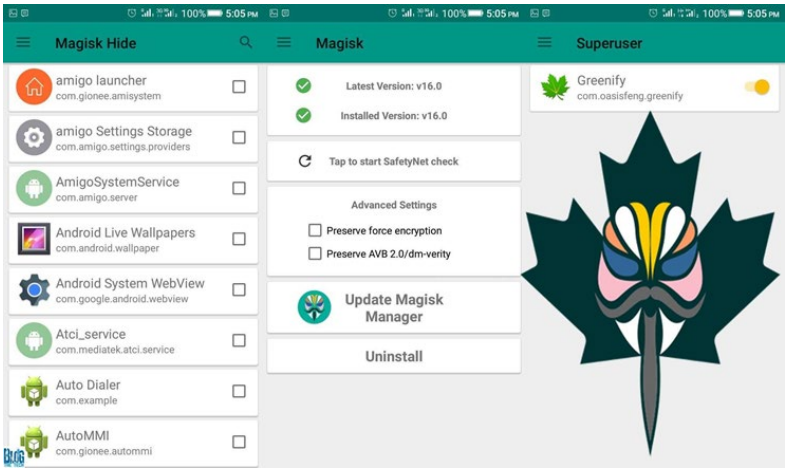
Također, provođenjem ove metode postoji opasnost od kreiranja izmjena koje mogu dovesti do nemogućnosti pokretanja operativnog sustava, [33].

Novijim inačicama Android operativnog sustava, Google je uveo sigurnosna poboljšanja u pogledu SELinux modela pružanja privilegija. Sve inačice Androida starije od Androida 4.3 koristile su diskrecijsku kontrolu pristupa (eng. *Discretionary Access Control* - DAC) prema kojoj je jedan korisnički račun mogao proslijediti svoje privilegije drugim korisničkim računima. SELinux modelom je uvedena obvezna kontrola pristupa (eng. *Mandatory Access Control* - MAC) prema kojoj se za svaki korisnički račun definiraju sve moguće radnje koje se mogu izvoditi bez mogućnosti prosljeđivanja i izmjenjivanja privilegija, [34].

To je rezultiralo stvaranjem *systemless root* metode kojom se izmjenjuje *boot* particija u svrhu pokretanja *superuser* procesa prilikom podizanja sustava. Primarni uvjet ove metode je otključan *bootloader*, iako postoje izuzetci kod kojih je moguće iskoristiti sigurnosne nedostatke uređaja ili sustava, [33].

Primjer *systemless root* metode je Magisk *root* čiji programski kod izmjenjuje inicijalizacijski proces. Nakon pokretanja instalacijske skripte putem izmijenjene particije za oporavak, *init* proces dodjeljuje Magisk *daemon* procesu *root* privilegije potrebne za njegovo ispravno funkcioniranje. S obzirom na to da je *init* datoteka sadržana unutar *boot* particije, sustav uređaja nije potrebno modificirati od čega je i proizašlo ime *systemless root*.

Svaki put kada neka aplikacija zatraži *root* privilegije, izvršava se Magisk *su* binarna datoteka koja će od Magisk *daemon* procesa zatražiti potrebne privilegije. Kako bi korisnik odobrio ili odbio takav zahtjev, koristi se aplikacija Magisk upravitelja čije je sučelje prikazano slikom 11. Kao što je vidljivo na slici, jedna od pogodnosti koje Magisk također pruža je sakrivanje *su* binarne datoteke i ostalih povezanih Magisk datoteka, kako aplikacije poput Google Playa ne bi mogle detektirati *root*, [35].



Slika 11. Sučelje Magisk Manager aplikacije, [36]

Aplikacija Magisk upravitelja je otvorenog koda te omogućuje upotrebu modifikacijskih modula. Modifikacijski moduli koriste se u svrhu kreiranja dodatnih izmjena nad sustavom. U tu svrhu koristi se Magic Mount procedura pomoću koje originalne sistemske datoteke ostaju netaknute, a procesi i aplikacije se usmjeravaju na lokaciju modificiranih sistemskih datoteka.

Otvoreni kod aplikacije i postupka ključan je za transparentnost provođenja *root* postupka, a samim time i veću sigurnost uređaja, [35]. Pogodnosti koje omogućuje Magisk *systemless root* su:

- mogućnost daljnjeg ažuriranja operativnog sustava
- mogućnost potpunog uklanjanja *root* mogućnosti s uređaja
- manja opasnost kvara uređaja
- mogućnost skrivanja *root* privilegija od sustava i aplikacija kao što je Google SafetyNet i bankovne aplikacije, [33].

4.1.2. Iskorištavanje sigurnosnih ranjivosti u svrhu *root* postupka

Uobičajeno, izvođenje *root* postupka zahtjeva prethodno otključavanje *bootloadera* tj. namjerno onemogućavanje sigurnosne provjere *boot* particije. Metode otključavanja *bootloadera* mogu se razlikovati ovisno o proizvođaču uređaja, a bez prethodno provedenog postupka nije moguće unijeti *root* programski kod. Iako većina proizvođača Android pametnih telefona omogućuje otključavanje *bootloadera* uređaja, u nekim slučajevima je izvođenje postupka otežano ili u potpunosti onemogućeno kao što je u slučaju Amazon uređaja. Iz tog

razloga potrebno je koristiti sigurnosne ranjivosti koje omogućuju obavljanje *root* postupka bez prethodnog otključavanja *bootloadera*, [37].

S obzirom na veliku raznolikost hardverskih komponenti i inačica Android sustava, *root* postupci temeljeni na sigurnosnim ranjivostima moraju biti prilagođeni za svaki pojedinačni model uređaja. Iz tog razloga, uobičajeno je provođenje dviju faza: faza provjere okoline i faza provjere rada. U prvoj fazi provjerava se vrsta uređaja, model uređaja te inačica korištenog operativnog sustava kako bi se identificirala odgovarajuća sigurnosna ranjivost uređaja koja će biti iskorištena. Jedna od najnovijih sigurnosnih ranjivosti koja je zahvatila veliki spektar uređaja bila je CVE-2020-0069 sigurnosna ranjivost otkrivena u MediaTek mikroprocesoru.

Navedena sigurnosna ranjivost otkrivena je u travnju 2019. godine, a ispravljena je tek u ožujku 2020. godine. Ona je omogućila pokretanje programskog koda u svrhu eskalacije privilegija na većem broju uređaja što uključuje određene modele proizvođača kao što su Alcatel, Amazon, Huawei, Sony te Xiaomi. Zbog različite učestalosti implementacije sigurnosnih zakrpi provodi se i druga faza provjere rada. Navedenom fazom se provjerava izvedivost *root* postupka odnosno dali se iskorištavanjem odabrane sigurnosne ranjivosti postižu željeni rezultati, [37], [38].

Ovakvu metodu *root* postupka primjenjuju *one-click root* aplikacije. Riječ je o aplikacijama koje izvode *root* postupak koristeći veći broj sigurnosnih ranjivosti kako bi time podržali što veći broj modela pametnih telefona. Godine 2015. izviješteno je kako prosječna komercijalna *one-click root* aplikacija koristi 59 vrsta sigurnosnih ranjivosti. Samo neki od primjera ovakvih alata za *rootanje* su sljedeće aplikacije:

- CF-Auto-Root
- Framaroot
- KingoRoot
- OneClickRoot
- Root Master, [38], [39].

Međutim, aplikacije pomoću kojih se izvode ovakve metode *rootanja* također mogu praviti dodatne izmjene u sustavu bez znanja korisnika. Iako nisu sve učinjene izmjene sustava zlonamjerne, kao što je u slučaju izmjena u svrhu prikazivanja oglasa, njihova uporaba nije preporučljiva kako integritet i povjerljivost skladištenih podataka ne bi bio narušen, [40]. Iz

tog razloga, daljnjim poglavljem rada bit će opisana isključivo Magisk metoda *rootanja* čije izvođenje ovisi isključivo o postupku otključavanja *bootloadera*.

4.1.3. Procedura *root* postupka

U svrhu provođenja *root* postupka korišten je Redmi Note 7 pametni telefon prikazan slikom 12. Xiaomi Redmi serija, prepoznatljiva po uređajima srednjeg cjenovnog razreda, 2019. godine odvojena je od matične firme Xiaomi kao nezavisan brend. Odvajanje je obilježeno puštanjem u prodaju spomenutog uređaja, a prihvatljiv omjer cijene i karakteristika učinio ga je jednim od najpopularnijih modela uređaja 2019. godine.

Slika 12. Redmi Note 7, [41]

Prema tvorničkim postavkama, na uređaju je predinstalirana MIUI 11 modifikacija Android OS-a temeljena na Android inačici 9, ali također podržavana službenu nadogradnju na MIUI 12 OS koji je temeljen na Android inačici 10. S obzirom na to da je u prodaju pušteno više inačica navedenog modela, karakteristike prikazane tablicom 2 opisuju globalnu inačicu korištenu za potrebe diplomskog rada.

Tablica 2. Specifikacije Redmi Note 7 uređaja

Podržane mreže	GSM, HSPA, LTE, WiFi, A-GPS, Bluetooth 5.0, IR, FM Radio
Zaslon	IPS LCD zaslon, 16 milijuna boja, 6.3 incha, 1080 x 2340 pixela
OS	Android 9.0, MIUI 11

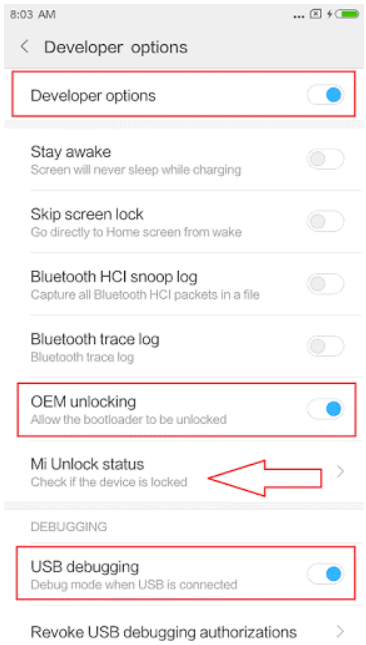
SoC	Qualcomm SDM660 Snapdragon 660
CPU	Octa-core (4x2.2 GHz Kryo 260 Gold & 4x1.8 GHz Kryo 260 Silver)
GPU	Adreno 512
Pohrana	32 GB / 3 GB RAM, 64 GB / 4 GB RAM, 128 GB / 6 GB RAM, eksterna pohrana podržana
Stražnja kamera	48 MP, f/1.8, (wide), ½.0“, 0.8µm, PDAF, 5 MP, f/2.2,
Prednja kamera	13 MP
Baterija	4000 mAh

Izvor: [42]

Prema zadanim postavkama, *bootloader* uređaja nije otključan iz sigurnosnih razloga, pa je stoga potpunu *root* proceduru moguće podijeliti na sljedeće korake:

1. Otključavanje *bootloadera*
2. Instalaciju prilagođene particije za oporavak
3. Instalacija Magisk *root* skripte i pripadajuće aplikacije Magisk upravitelja, [43].

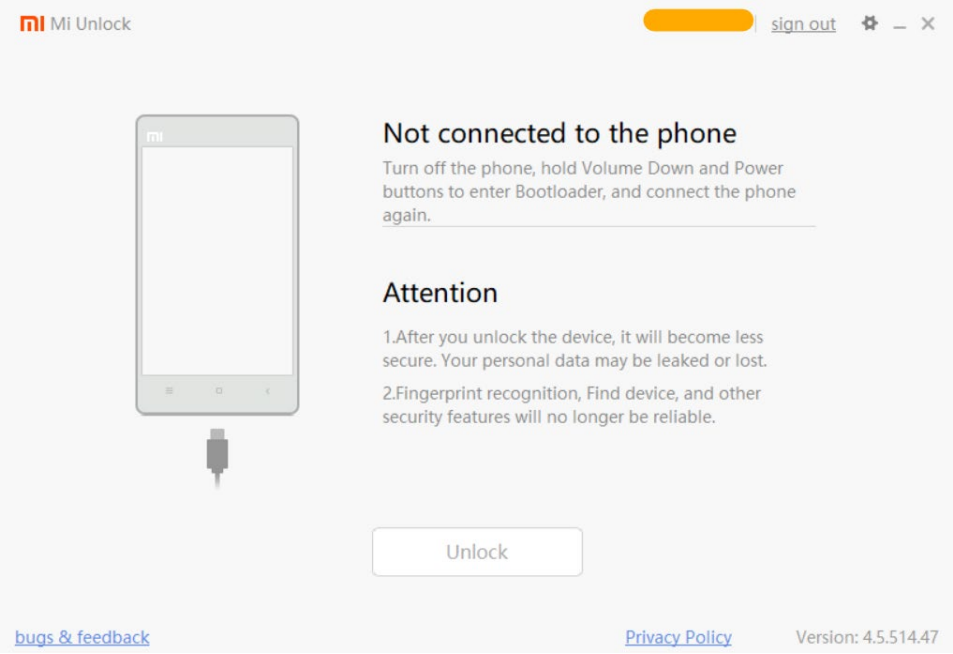
U odnosu na druge proizvođače pametnih telefona kao što je Samsung, Xiaomi ne omogućuje svojim korisnicima otključavanje *bootloadera* izravno unutar postavki uređaja. Iz sigurnosnih razloga, dozvolu za provođenje postupka otključavanja *bootloadera* je potrebno prethodno zatražiti putem web stranice [44] koristeći Xiaomi Mi račun. Time se sprječava daljnja preprodaja pametnih telefona s instaliranim zlonamjernim softverom. Nadalje, prije pokretanja postupka je također potrebno konfigurirati postavke namijenjene razvojnim programerima na način prikazan slikom 13, [43], [45].



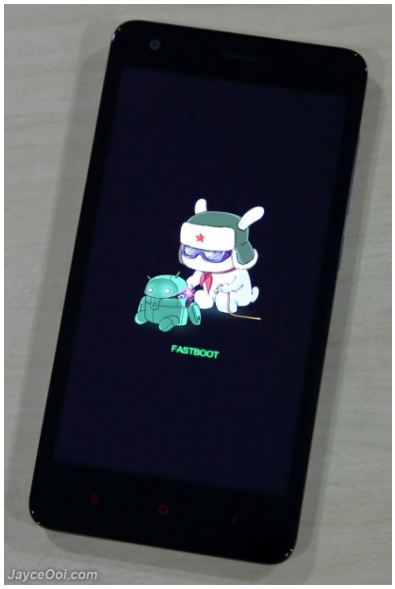
Slika 13. Prikaz konfiguracije postavki za razvojne programere, [46]

Nakon uspješno osiguranog odobrenja, procedura otključavanja *bootloadera* obavlja se pomoću Mi Unlock programa preuzetog s izvora [44] te instaliranog na računalu s kojim će pametni telefon biti povezan. Pokretanjem programa prikazuje se početni zaslon na kojem je potrebno prijaviti se pomoću istog Mi računa koji je bio korišten prilikom traženja dozvole otključavanja bootloadera.

Nakon uspješne prijave, prikazuje se prozor prikazan slikom 14. Kao što je vidljivo na slici, pametni telefon je, prije povezivanja na računalo, potrebno ponovno pokrenuti unutar *fastboot* načina rada koje omogućuje preslikavanje logičkih particija uređaja. Sučelje fastboot protokola Xiaomi uređaja prikazano je slikom 15. Iz sigurnosnih razloga, provođenjem postupka otključavanja *bootloadera* brišu se svi korisnički podaci skladišteni na uređaju kako postupak ne bi mogao biti korišten u svrhu nedozvoljenog pristupa osjetljivim podacima.



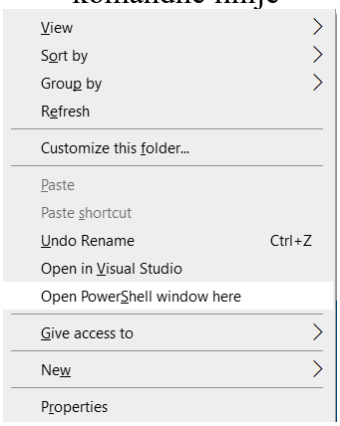
Slika 14. Sučelje Mi Unlock programa



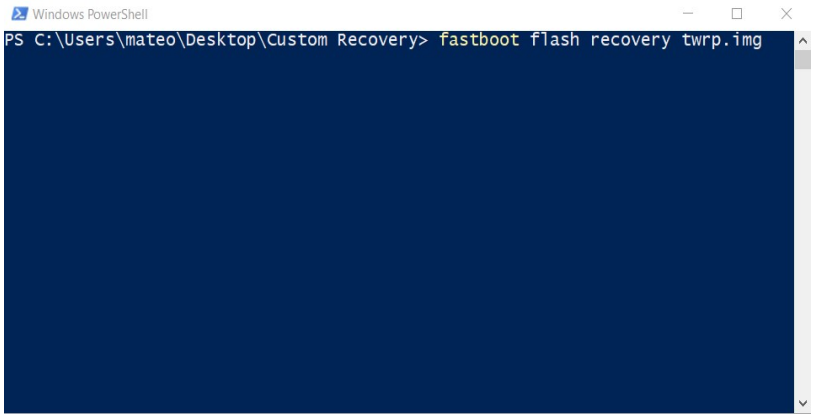
Slika 15. Prikaz fastboot načina rada, [47]

Nakon uspješno provedenog postupka otključavanja *bootloadera*, idućim korakom je na pametni telefon instalirana prilagođena particija za oporavak *Team Win Recovery Project* (TWRP), preuzeta putem web stranice [48]. Kao i u prethodnom koraku, uređaj je prije spajanja na računalo, pokrenut unutar *fastboot* načina rada. Na računalu je, unutar direktorija preuzete datoteke, pokrenut tumač naredbenih linija PowerShell u kojeg je upisana naredba prikazana slikom 16. Nakon uspješne instalacije, particija za oporavak pokrenuta je istovremenim pritiskom gumba za aktiviranje pametnog telefona te gumba za pojačavanje zvuka.

a) Koraci pokretanja komandne linije

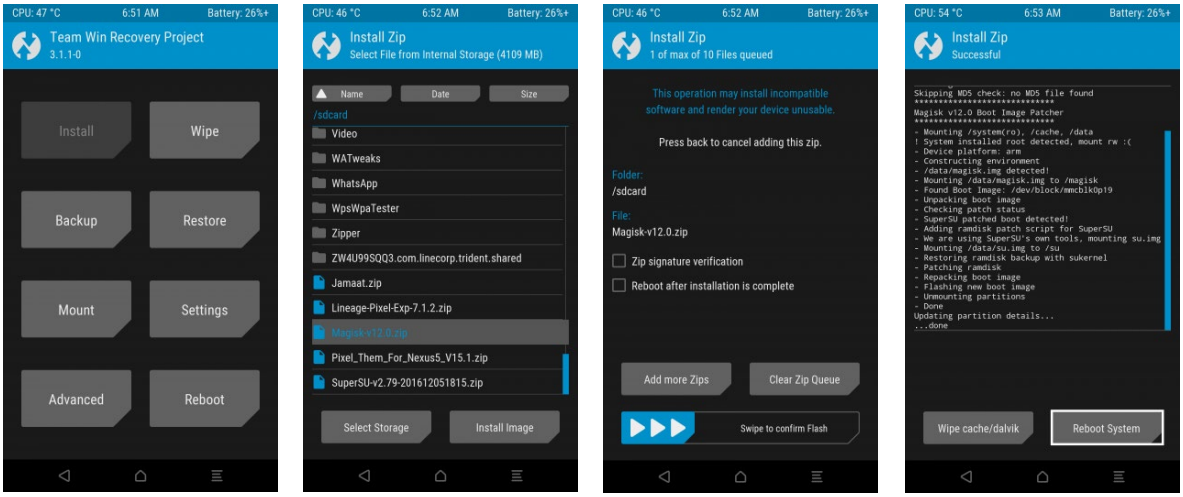


b) Sučelje komandne linije



Slika 16. Grafičko sučelje za pokretanje komandne linije (slika 16-a) i sučelje komandne linije (slika 16-b)

Posljednjim korakom *root* procedure, pokrenuta je instalacijska skripta Magisk *roota* preuzeta s stranice [49]. U tu svrhu korištena je TWRP particija za oporavak instalirana u prošlom koraku. Kao što je prikazano slikom 17, unutar sučelja za oporavak prvo je odabrana naredba *Install,* nakon čega je unutar direktorija pohrane odabrana Magisk zip instalacijska datoteka. Instalaciju je na kraju također bilo potrebno potvrditi.



Slika 17. Koraci instalacije Magisk *Root* skripte koristeći TWRP particiju za oporavak, [49]

Nakon uspješne instalacije Magisk zip datoteke, uređaj je bilo potrebno ponovno pokrenuti pritiskom tipke „Ponovno pokreni sustav“ (eng. *Reboot System*). Time je završen *root* proces, te će svakim idućim pokretanjem sustava također biti pokrenut i proces koji omogućava Magisk aplikaciji dodjeljivanje *root* privilegija. Magisk aplikaciju nije bilo

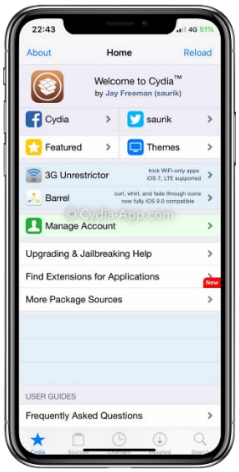
potrebno ručno preuzeti s obzirom na to da je ona također instalirana prilikom instalacije Magisk zip datoteke.

4.2. Značajke *jailbreak* postupak

Jailbreak je postupak eskalacije privilegija na uređajima koji koriste Appleove operativne sustave kao što su iOS i iPadOS. Zbog zatvorenosti operativnih sustava, krajnji korisnik nema pristup *root* mogućnostima koje bi mu omogućile potpunu kontrolu nad uređajem. U tu svrhu, postupkom se mijenja sistemska particija uređaja kako bi uklonila softverska ograničenja i time omogućile mogućnosti kao što su:

- instalacija, pokretanje i brisanje bilo kojeg izvršnog softvera
- prilagođavanje izgleda korisničkog sučelja
- otključavanje uređaja u svrhu korištenja drugih operatora
- zadobivanje potpunog pristupa pohrane podataka
- ekstrakcija skladištenih podataka
- evaluacija sigurnosnog modela sustava i otkrivanje sigurnosnih nedostataka sustava.

Instalacija nepotvrđenih aplikacija izvodi se pomoću trgovina aplikacija trećih strana kao što je Cydia trgovina čije sučelje je prikazano slikom 18. S obzirom na to da aplikacije nisu ograničene Appleovim provjerama sigurnosti, one mogu koristiti inače zabranjena aplikacijska programska sučelja i time dodatno prilagođavati sučelje operativnog sustava. Trgovina Cydia je također mjesto s kojeg je moguće preuzeti aplikacije pomoću kojih se provode različiti forenzički postupci kao što su ekstrakcija skladištenih podataka, [6], [50].



Slika 18. Sučelje Cydia trgovine aplikacija, [51]

Legalnost *jailbreak* postupka nije jasno definirana u svim državama svijeta te u većini slučajeva ovisi o nacionalnom zakonodavstvu države. Općenito se postupak regulira zakonom o digitalnim pravima (eng. *Digital Rights Management* – DRM) koji nije implementiran u sve države svijeta. U slučaju Sjedinjenih Američkih Država, napravljena je revizija Zakona o zaštiti autorskih prava u digitalnom tisućljeću (eng. *Digital Millennium Copyright Act - DMCA*) Jamesa H. Billingtona, prema kojoj je *jailbreak* postupak proglašen u potpunosti legalan postupak kada se koristi u svrhu pokretanja legalno stečenih aplikacija. U slučaju Europske unije, direktiva o računalnim programima također nalaže sličnu iznimku prema kojoj je provođenje postupka opravdano u slučaju pokretanja legalno stečenog softvera, [52].

Unatoč djelomičnoj legalnosti postupaka, Apple svojim sigurnosnim modelom nastoji onemogućiti ili barem otežati izvođenje postupka u svrhu očuvanja sigurnosti uređaja. Rizici provođenja postupka uključuju:

- nenamjerno kršenje zakona u državama gdje legalnost nije jasno definirana
- poništavanje jamstva uređaja
- opterećenje sustava te trajno oštećenje hardvera
- instalacija zlonamjernog softvera
- gubitak podataka
- utjecaj na forenzičku ispravnost podataka stečenih provedenim postupkom, [53].

4.2.1. Vrste *jailbreak* postupaka

Jailbreak postupke moguće je kategorizirati na:

- *tethered jailbreak*
- *untethered jailbreak*
- *semi-tethered jailbreak.*

Tethered jailbreak je vrsta postupka kojom učinjene izmjene nestaju prilikom ponovnog pokretanja iOS uređaja. Postupak se obavlja pokretanjem posebnih aplikacija, posjetom određenim stranicama ili povezivanjem uređaja na stolno računalo putem USB kabela po čemu je također i proizašlo ime postupka. U ovom slučaju iskorištavaju se sigurnosne ranjivosti jezgre sustava, USB upravljačkog programa ili koda instaliranih aplikacija. Nedostatak ovakve vrste postupka je taj što može dovesti do djelomičnog izvršavanja tijeka pokretanja OS-a u slučaju da uređaj nije spojen na stolno računalo prilikom ponovnog pokretanja, [54].

Untethered jailbreak je vrsta postupka kojom se izmjene zadržavaju prilikom ponovnog pokretanja uređaja, pri čemu uređaj ne mora biti povezan na stolno računalo. Međutim, ova metoda je također zahtjevnija jer ju je moguće izvesti isključivo uz pomoć ranjivosti sigurnosnog lanca pokretanja sustava. S obzirom na rijetkost takvih sigurnosnih ranjivosti, većina *untethered* postupaka temeljena je na kombinaciji *tethered* postupka i dodatnih eksploatacija na temelju kojih će se zadržati očuvanost izmjena i nakon ponovnog pokretanja uređaja, [53].

Semi-tethered jailbreak je vrsta postupka koja predstavlja kombinaciju prethodna dva postupka. Ponovnim uključivanjem uređaja, učinjene izmjene se ne zadržavaju na operativnom sustavu kojeg je i dalje moguće u potpunosti pokrenuti bez spajanja na stolno računalo. Svako daljnje modificiranje koda ili pokretanje nepotpisanih aplikacija zahtjeva ponovno obavljanje *jailbreak* postupak nakon ponovnog uključivanja uređaja, [54].

4.2.2. Iskorištavanje sigurnosnih ranjivosti u svrhu *jailbreak* postupka

Jailbreak postupke moguće je provesti isključivo iskorištavanjem sigurnosnih ranjivosti hardverskih i softverskih komponenti iOS uređaja. Postoje tri osnovne razine sigurnosnih ranjivosti:

- Sigurnosne ranjivosti *bootrom* razine – pružaju najveći broj mogućnosti u pogledu *jailbreak* postupka te ih nije moguće softverski ispraviti.

- Sigurnosne ranjivosti *iBoot* razine – pružaju jednak broj mogućnosti kao sigurnosne ranjivosti *bootrom* razine, ali ih je moguće softverski ispraviti.
- Sigurnosne ranjivosti *userland* razina – ograničena mogućnost izvođenja *jailbreak* postupka pri čemu su potrebne najmanje dvije ranjivosti od kojih će jedna biti korištena u svrhu izvršavanja koda, a druga u svrhu eskalacije privilegija. Moguće ih je ispraviti softverski, [54].

Prvi *jailbreak* postupak izveden je 2007. godine kada je George Hotz uklonio hardversku komponentu iPhone uređaja prve generacije s ciljem omogućavanja korištenja mreže drugog telefonskog operatora. Nadalje, iste te godine hakerska grupa *iPhone Dev Team* objavila je prvi službeni alat za *jailbreakanje* uređaja pod nazivom *JailbreakMe* 1.0/*AppSnapp*. Navedeni alat koristio je TIFF sigurnosnu ranjivost koja je omogućila pokretanje koda putem ugrađene aplikacije Internet preglednika. Postupak se izvodio putem JailbreakMe.com web stranice, a omogućio je manja podešavanja korisničkog sučelja kao što je izmjena melodije zvona, [55], [56].

Iphone Dev Team hakerska grupa objavila je 2008. godine novu verziju *jailbreak* alata imena *PwnageTool* za iPhone iOS 2.0. Novim alatom je po prvi puta predstavljena *Cydia* trgovina koja je omogućila preuzimanje i instalaciju aplikacija inače nedostupnih na službenoj App trgovini. Navedeni alat koristio je sigurnosnu ranjivost S5L8900 procesora kojeg su koristili iPhone uređaji prve generacije te iPhone 3G. Kao što je navedeno potpoglavljem 2.3.3., svaki element sigurnosnog tijeka pokretanja mora biti provjeren kako bi uređaj mogao biti uspješno upaljen. Međutim, radi sigurnosne ranjivosti navedenog procesora, iBoot i LLB komponenta nisu ispravno provjerene što je omogućilo zaobilaženje sigurnosnog tijeka pokretanja, [57].

Zbog redovitog i brzog ažuriranja sigurnosnih ranjivosti, provođenje *jailbreak* postupka na gotovo svakoj sljedećoj inačici iOS sustava je zahtijevalo otkrivanje novih sigurnosnih ranjivosti. S obzirom na veliki broj sigurnosnih ranjivosti ali i alata koji ih iskorištavaju, slikom 19. prikazani su samo neki od primjera korištenih proteklih godina. Navedene alate izrađuju zasebne skupine hakera, a unatoč sve manjoj popularnosti *jailbreak* postupka, vidljivo je kako je čak i na najnovijim Apple uređajima moguće provesti *jailbreak* postupak.

Slika 19. Povijest *jailbreak* alata

U povijesti *jailbreak* postupka, jedna od značajnijih metoda bila je limeRa1n. Navedena metoda koristila je sigurnosnu ranjivost A4 procesora korištenog u modelima kao što su iPhone 4 i iPhone 3GS. Ona je omogućila zaobilaženje sigurnosnog tijeka pokretanja uređaja pri čemu Apple sigurnosnu ranjivost nije mogao ispraviti softverskim ažuriranjem već isključivo revizijom hardvera uređaja, [58]. Osim u svrhu *jailbreak* postupka, ona je također korištena i druge svrhu poput učitavanja posebnih inačica OS-a. Novije revizije uređaja otežale su obavljanje *jailbreak* postupka, te je iduća slična sigurnosna ranjivost otkrivena tek deset godina poslije pod nazivom Checkm8.

Checkm8 sigurnosna ranjivost otkrivena je od strane Twitter korisnika pod imenom axi0om. Kao i u slučaju limeRa1na, sigurnosna ranjivost nalazi se unutar *BootRom* komponente. S obzirom na to da je *BootRom* hardverski implementiran dio koda koji si prvi pokreće prilikom uključivanja iOS uređaj, njegove sigurnosne ranjivosti nije moguće ispraviti softverski već isključivo revizijom hardvera.

Checkm8 ranjivost moguće je iskoristiti na svim uređajima koji koriste čipove A5 do A11 uključivši time: iPhone modele od inačice 4S do inačice X što čini gotovo 85% aktivnih Apple pametnih telefona u svijetu, svi iPad uređaji proizvedeni do 2019. godine te Apple pametni satovi serije 1,2 i 3, [59], [60]. Prema [61], osnovne značajke Checkm8 ranjivosti su:

- Zahtjeva fizički pristup pametnom uređaju, odnosno spajanje uređaja na računalo putem USB kabela.
- Omogućuje izvođenje *jailbreak* postupka neovisno o inačici iOS sustava.

- Omogućuje isključivo *tethered jailbreak* metode što ograničava trajnost postupka.
- Ne može biti iskorišten u svrhu zaobilaženja zaštite sigurnosne enklave i Touch ID-a što ograničava pristup skladištenim podacima ako napadač također ne poznaje i certifikat zaključavanja uređaja.
- Omogućuje izvršavanje zlonamjernog koda poput *keylogger*a odnosno softvera koji bilježi i šalje sve unesene podatke na neki drugi uređaj.

Ranjivost uređaja nalazi se u kodu USB upravljačkog programa, a njegovo iskorištavanje zahtjeva pokretanje DFU način rada. Kao što je objašnjeno poglavljem tijeka pokretanja iOS uređaja, DFU način rada služi kako bi se na uređaj prenijela instalacijska datoteka sustava pomoću koje će sustav uređaja biti popravljen ili vraćen na prijašnju inačicu. Kako bi instalacijska datoteka sustava mogla biti prenesena, uređaj prvo provjerava integritet datoteke, a zatim dozvoljava pristup privremenoj pohrani podataka pomoću argumenata pokazivača. Nakon uspješnog prenošenja datoteka na privremenu pohranu uređaja, argumenti pokazivača se resetiraju pri čemu uređaj izlazi iz DFU načina rada i pokreće instalaciju zaprimljene datoteke sustava. U slučaju da datoteka bude neuspješno učitana, uređaj ponovno pokreće DFU način rada.

Upotrebom USB kontrolera, moguće je zaustaviti prijenos datoteke sustava nakon što argumenti pokazivača već budu dodijeljeni. Iskorištavanjem navedene sigurnosne ranjivosti preskače se korak resetiranja argumenata pokazivača, koje je moguće ponovno iskoristiti prilikom sljedećeg pokretanja DFU načina rada u svrhu pokretanja neautoriziranog koda, [62]. Prva, ali i trenutno jedina *jailbreak* metoda koja koristi Checkm8 sigurnosnu ranjivost je checkra1n metoda čija je prva inačica puštena u studenom 2019. godine.

4.2.3. Procedura *jailbreak* postupka

Za potrebe rada, *jailbreak* postupak proveden je na uređaju iPhone 4 prikazanim slikom 20. Uređaj je na tržište izašao u lipnju 2010. godine s predinstaliranom iOS inačicom 4, te je Apple nastavio podržavati uređaj nadogradnjama sve do inačice 7.1.2. Najvažnije karakteristike su prikazane tablicom 3 iz koje je također moguće vidjeti kako postoje tri inačice koje se razlikuju po veličini unutarnje memorije. Model korišten u svrhu rada sadrži 8 GB unutarnje memorije te 512 MB RAM memorije.

Slika 20. Prikaz iPhone 4 uređaja, [63]

Tablica 3. Karakteristike uređaja iPhone 4

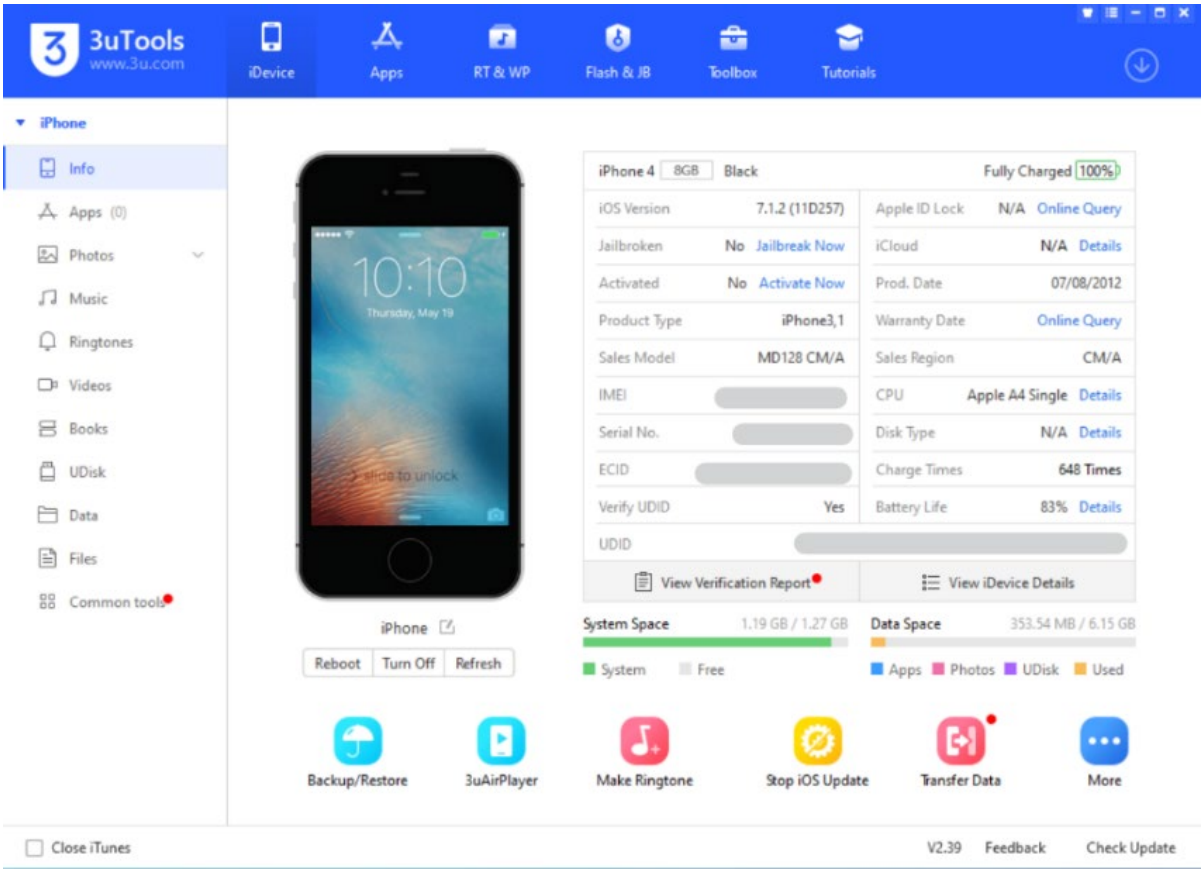
Podržane mreže	GSM, HSPA, WiFi, A-GPS, Bluetooth 2.1
Zaslon	IPS LCD zaslon, 16 milijuna boja, 3.5 incha, 640 x 960 pixela
OS	iOS 4, podržan do inačice iOS 7.1.2.
SoC	Qualcomm SDM660 Snapdragon 660
CPU	1.0 GHz Cortex-A8
GPU	PowerVR SGX535
Pohrana	8 GB / 512 MB RAM, 16 GB / 512 MB RAM, 32 GB / 512 MB RAM, vanjska pohrana nije podržana
Stražnja kamera	5 MP, f/2.8, 1/3.2", 1.75µm, AF
Prednja kamera	VGA
Baterija	1420 mAh

Izvor: [64]

Postupak je proveden pomoću programa 3uTools preuzetog s izvora [65]. Prvim pokretanjem programa dan je niz uputa o pravilnom povezivanju uređaja s računalom. Primjer jedne od važnijih uputa jest postupak odobravanja konekcije putem sučelja telefona prikazanog slikom 21.

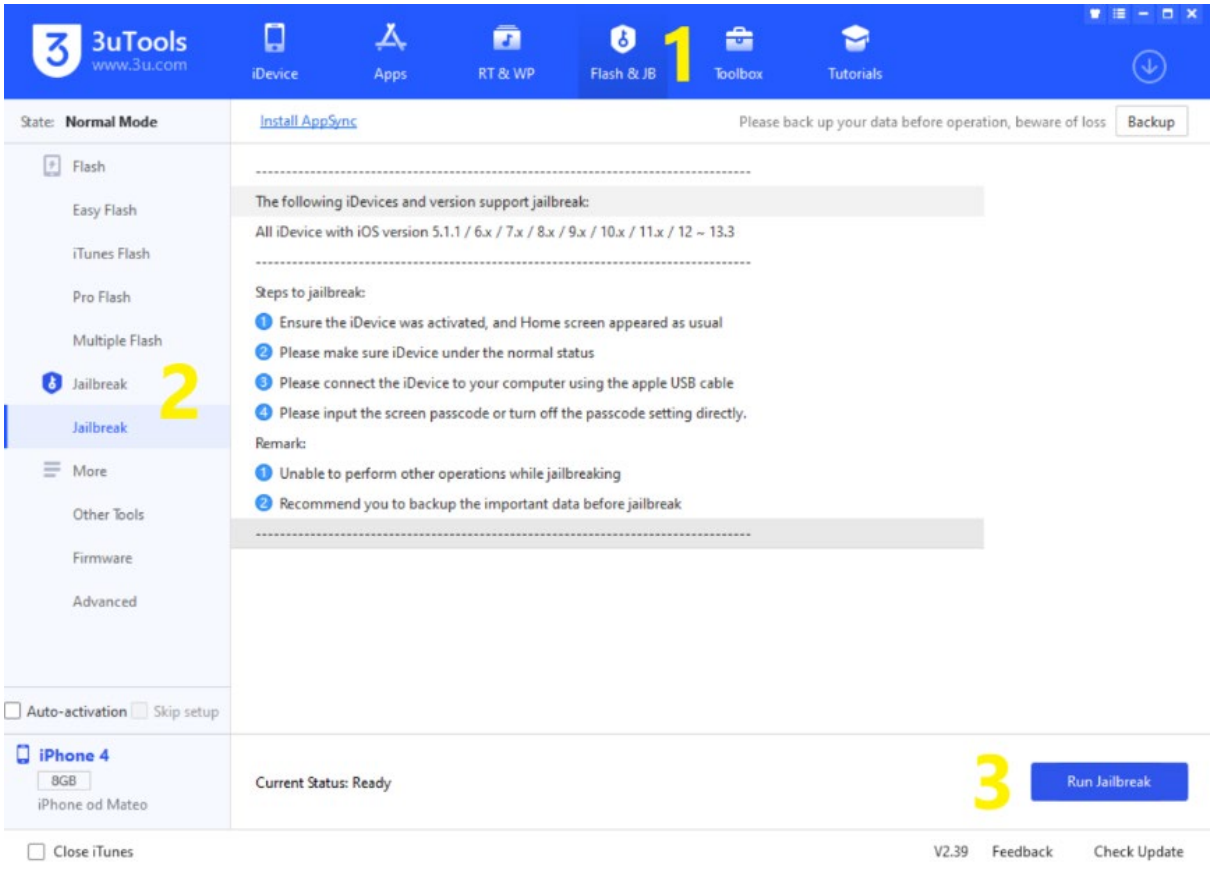
Slika 21. Sučelje za odobravanje konekcije između iPhone uređaja i stolnog računala

Kao što je vidljivo sa slike 22, nakon povezivanja telefona i računala, na korisničkom sučelju 3uTools alata moguće je pregledati osnovne informacije o uređaju kao što je IMEI (eng. *International Mobile Equipment Identity*) broj, ECID (eng. *Exclusive Chip Identification*) broj, informacije o pohrani, datum proizvodnje i ostalo. Nadalje, gornja traka početnog sučelja omogućuje pokretanje mnogih funkcija kao što su: pokretanje *jailbreak* postupka, instalacija aplikacija, izmjena zvuka zvonjave i druge.



Slika 22. Početno sučelje 3uTools programa

Pokretanje *jailbreak* postupka prikazano je slikom 23, pri čemu je redoslijed odabira naznačen brojevima. Na slici je također moguće vidjeti uvjete koje je potrebno ispuniti prije pokretanja *jailbreak* postupka kao što je otključavanje zaslona ekrana ili uklanjanje lozinke zaključavanja.



Slika 23. Sučelje 3uTools alata za pokretanje *jailbreak* postupka

Ovisno o inačici softvera te modelu iPhone uređaja, 3uTool program automatski odabire odgovarajuću *jailbreak* metodu te preuzima datoteke potrebne za obavljanje postupka. U slučaju iPhone 4 modela i inačice sustava iOS 7.1.2. korištena je Pangu7 *jailbreak* metoda koja također zahtjeva konfiguriranje datuma telefona na datum 01.06.2014. Tijekom postupka, na pametni telefon je instalirana aplikacija Pangu koju je moguće pronaći na izborniku aplikacija. Aplikacija je potrebna u svrhu dovršavanja *jailbreak* postupka, a ponovnim pokretanjem pametnog telefona je automatski uklonjena. Završetkom *jailbreak* postupka, na pametni telefon je također automatski instalirana aplikacija Cydia trgovine.

5. Veza forenzičke analize uređaja i *root*/*jailbreak* postupaka

Digitalna forenzika je grana forenzičke znanosti usmjerena na pronalaženje, prikupljanje, analizu i dokumentiranje neobrađenih digitalnih podataka. Pri tome glavni zahtjev digitalne forenzike jest očuvanje integriteta svih prikupljenih podataka kako bi oni mogli biti korišteni u istražne i sudske svrhe. U tom pogledu koristi se pojam forenzičke ispravnosti (eng. *forensically sound*) kojim se kvalificira i opravdava upotreba određene forenzičke tehnologije ili metodologije.

Danas najčešći medij za prijenos i skladištenje digitalnih podataka čine upravo pametni telefoni koji nisu ograničeni lokacijom, a svojim korisnicima pružajući veliki zbir funkcionalnosti poput fotografiranja, navigacije, pregledavanja Internet sadržaja te elektroničkog komuniciranja. Uzevši u obzir različitosti načina korištenja, tehničkih komponenti, implementiranih operativnih sustava ali i sigurnosnih modela, u odnosu na druge elektroničke uređaje, uspostavljena je posebna grana digitalne forenzike imena forenzika mobilnih uređaja.

Forenzika mobilnih uređaja definira odgovarajuću metodologiju i smjernice provođenja ispitivanja mobilnih uređaja, kako bi se osigurao integritet podataka čak i u slučajevima kada nije moguće izbjeći izmjenu konfiguracije uređaja. U slučaju kada nije moguće ispitati ili prikupiti podatke bez promjene konfiguracije uređaja npr. obavljanjem *root* i *jailbreak* postupka, postupke je prethodno potrebno ispitati, potvrditi i dokumentirati kako bi se utvrdilo na koji način navedeni postupak može utjecati na prikupljene podatke.

S obzirom na problematiku provođenja forenzike mobilnih uređaja, sljedećim potpoglavljima bit će definirana sljedeća područja ključna za tematiku diplomskog rada:

- razine ispitivanja mobilnih uređaja
- vrste pohranjenih podataka
- forenzička ispravnost *root* i *jailbreak* postupaka, [6]

5.1. Forenzičke metode i razine ispitivačkih alata

Forenzičko ispitivanje mobilnih uređaja moguće je obaviti zbirom različitih alata koji se razlikuju u ovisnosti o kompleksnosti uporabe, forenzičkoj ispravnosti ali i količini obuhvaćenih podataka. U svrhu kategorizacije svih dostupnih alata koristi se sustav prikazan slikom 24, pri čemu je svaka viša razina tehnički i vremenski zahtjevnija, ali i forenzički ispravnija istovremeno obuhvaćajući veću količinu podataka. Ovisno o željenom ishodu istrage, potrebno je odabrati one alate koji će na najbolji način zadovoljiti postavljene uvjete.

Slika 24. Piramida razina ispitivanja mobilnih uređaja

Ručna ekstrakcije je najmanje zahtjevna razina ispitivanja mobilnih uređaja s obzirom na to da se podaci prikupljaju pomoću korisničkog sučelja operativnog sustava. U svrhu navigacije koristi se tipkovnica uređaja ili zaslon osjetljiv na dodir, a svi podaci ključni za istragu dokumentiraju se pomoću fotoaparata. Unatoč jednostavnosti provođenja ovakvog tipa ispitivanja, glavni nedostatak je mala količina podataka koju je moguće prikupiti. Nepoznavanje sučelja operativnog sustava može utjecati na identifikaciju skladištenih podataka, pri čemu nije uopće moguće oporaviti izbrisane podatke. Iz tog razloga, ručne metode izvode se samo onda kada nije moguće obaviti fizičke ili logičke metode ispitivanja.

Druga razina jest logična ekstrakcija koja zahtjeva povezivanje mobilnog uređaja na radnu stanicu putem USB kabela, RJ-45 kabela, Infrared ili Bluetooth veze. Metode ove razine provode se pomoću aplikacijsko-programskih sučelja putem kojih se šalje niz programskih naredbi. Naredbe zaprima i interpretira procesor uređaja, te na njih odgovara zatraženim

podacima. Ovakve metode oporavka podataka su brze i jednostavne za izvesti radi čega ih danas koristi većina forenzičkih alata. Međutim, kao i u slučaju prethodne razine, moguće je oporaviti samo one podatke koji nisu obrisani s uređaja.

Hex dump je treća razina spomenutog sustava te je ujedno i prva fizička razina forenzičkog ispitivanja. Fizičkim metodama izravno se pristupa podacima zapisanim na *flash* memoriji u svrhu stvaranja bit po bit kopije potpunog datotečnog sustava što uključuje i izbrisane podatke skladištene u neraspoređenom prostoru memorije. Metode ove razine također zahtijevaju povezivanje uređaja na radnu stanicu, a najčešće se izvode slanjem nepotpisanih kodova ili *bootloadera* na uređaj. Skup svih prikupljenih binarnih podataka čini sirovu sliku podataka koja se analizira pomoću tehničkih alata kao što su Autopsy, Forensic Toolkit FTK te Cellebrite Reader.

Metode *chip-off* razine sustava su fizičke metode kojima se podaci prikupljaju izravno s memorijskog čipa uređaja. U tu svrhu čip se fizički uklanja iz uređaja kako bi ga se moglo umetnuti u čitač čipova ili neki drugi mobilni uređaj pomoću kojeg će podaci biti ekstrahirani. Metode temeljene na ovoj razini ispitivanja su skupe i tehnički zahtjevne, pri čemu neispravno obavljen postupak može oštetiti memorijski čip i uništiti sve skladištene podatke. Upravo iz ovih razloga, ove metode se koriste samo u slučajevima kada podatke nije moguće prikupiti ostalim metodama npr. u slučajevima kada je uređaj oštećen.

Posljednja razina ispitivačkih alata je *micro read* razina koja podrazumijeva ručni pregled čipa pomoću elektronskog mikroskopa. Čitav postupak je dugotrajan i skup te zahtjeva veliku količinu tehničkog znanja *flash* memorije i datotečnog sustava uređaja. S obzirom na tehničku zahtjevnost navedenog postupka, alati temeljeni na ovoj razini ispitivanja se koriste isključivo u slučajevima velike važnosti kao što je pitanje nacionalne sigurnosti. Zbog rjeđe učestalosti provođenja postupak, dokumentacija je gotovo nepostojeća. Također, trenutno ne postoji komercijalni alat koji koristi ovu vrstu ispitivanja, [6].

5.2. Vrste memorije i pohranjenih podataka

Pametni telefoni koriste dvije vrste memorije: izbrisiva memorija (eng. *volatile memory*) također poznata kao RAM memorija te neizbrisiva memorija (eng. *nonvolatile memory*) odnosno ROM (eng. *Read Only Memory*) memorija. Podaci izbrisive memorije brišu

se isključivanjem uređaja, ali se zbog brzine pristupa istima koriste u svrhu privremenog skladištenja određenih podataka operativnog sustava te svih aplikacija koje se trenutno izvode na uređaju. RAM memorija može biti vrlo važna u pogledu forenzike uređaja s obzirom na to da može sadržavati važne podatke kao što su korisnička imena, lozinke te ključevi šifriranja.

Podaci neizbrisive memorije zadržani su u pohrani i nakon isključivanja uređaja radi čega se ovakva vrsta memorije koristi u svrhu pohranjivanja operativnog sustava i svih ostalih korisničkih podataka. Vrsta neizbrisive memorije koja se koristi u gotovo svim pametnim telefonima jest *flash* memorija koja je karakterizirana malim fizičkim dimenzijama, brzinom pristupa podacima te nepomičnosti tijekom rada, [66], [67], [68]. Osim unutarnje memorije, podatke je također moguće prikupiti s drugih lokacija uređaja kao što su SIM (eng. *Subscriber Identity Module*) kartica te vanjska memorijska kartica. Neovisno o modelu uređaja, osnovna kategorizacija skladištenih podataka prikazana je tablicom 4.

Tablica 4. Kategorije podataka skladištenih na pametnim telefonima

Kategorija podataka	**Opis podataka**
Podaci komunikacijskih usluga	Povijest upućenih, primljenih i propuštenih poziva; Spremnik primljenih i poslanih SMS te MMS poruka; adresar; podaci lokacije
Podaci instaliranih aplikacija	Podaci nastali uporabom predinstaliranih te naknadno instaliranih aplikacija kao što su: Chrome, Facebook, Google Maps, Gmail itd.
Datoteke	Podaci kreirani od strane korisnika, preuzeti putem Interneta ili zaprimljeni od drugih osoba npr. multimedijski zapisi te dokumenti
Podaci o uređaju	Informacije o uređaju i konfiguraciji uređaja kao što su korisničke zaporke, zapisnici korištenja uređaja, IMEI broj itd.
Neraspoređeni podaci	Podaci izbrisani od strane korisnika, ali i dalje pohranjeni na memoriji uređaja

Izvor: [6]

Današnji operativni sustavi pametnih telefona koriste različite načine upravljanja izbrisanim podacima. U slučaju Android OS-a, svaka datoteka posjeduje metapodatke na temelju kojih je softver može identificira u trenucima kada joj korisnik pokušava pristupiti. Postupkom brisanja datoteke brišu se samo metapodaci, pri čemu sadržaj datoteke ostaje netaknut sve dok ne bude prebrisana novim podacima.

S druge strane, iOS operativni sustav sve informacije o datotekama sprema u odgovarajuće SQLITE baze podataka iz kojih se informacije brišu onda kada korisnik obriše datoteku. Sadržaj podataka, kao i u slučaju Android sustava, ostaje netaknut, međutim ovisno o inačici OS-a, pohranjene datoteke mogu biti kriptirane zasebnim enkripcijskim ključevima. Naredbom brisanja datoteke također se briše i enkripcijski ključ radi čega takve podatke nije moguće pregledati čak ni nakon njihovog oporavljanja, [6].

5.3. Forenzička ispravnost *root* i *jailbreak* postupaka

Pojam forenzičke ispravnosti koristi se u domeni digitalne forenzike kako bi se ocijenila ispravnost prikupljenih podataka. Svaki podatak prikupljen s digitalnog uređaja ne smije biti izmijenjen ni na koji način kako bi se očuvala njegova dokazna vrijednost. Nadalje, pojam forenzičke ispravnosti se također koristi kako bi se opravdala uporaba određene forenzičke metodologije i tehnologije koja mogu utjecati na postupke preslikavanja i obrađivanja podataka. Uvjeti koje pojam forenzičke ispravnosti nalaže su sljedeći:

- Tehnologija korištena u svrhu provođenja forenzičkih postupaka ne smije ni na koji način izmijeniti podatke pohranjene na izvornom disku.
- Dobivena kopija pohranjenih podataka mora biti identična izvornim pohranjenim podatcima.
- Autentičnost preslikanih podataka potrebno je moći dokazati matematičkim otiskom ili generiranim *hash* vrijednostima.
- Svaki korak postupka mora biti jasno dokumentiran u svrhu razumljivosti i ponovljivosti postupka.
- Osoba odgovorna za provođenje forenzičkog postupka mora osigurati da provedeni postupci udovoljavaju navedenim smjernicama ali i drugim državnim zakonima, [69].

Iako podaci uređaja ne bi smjeli biti izmijenjeni, određene elemente pametnog uređaja je ponekad potrebno promijeniti, oštetiti ili čak uništiti kako bi bilo moguće obaviti forenzičke postupke. Jedan takav primjer je upravo *jailbreak* postupak koji čini izmjene u sistemskoj particiji iPhone uređaja. Iako je starije modele iPhone telefona, kao što je u slučaju modela iPhone 4, moguće je forenzički ispitati bez provođenja *jailbreak* postupka, na novijim modelima je postupak često moguć samo nakon obavljanja *jailbreak* postupka. Iz tog razloga obavljanje *jailbreak* postupka je postalo neizbježno u domeni forenzike mobilnih uređaja.

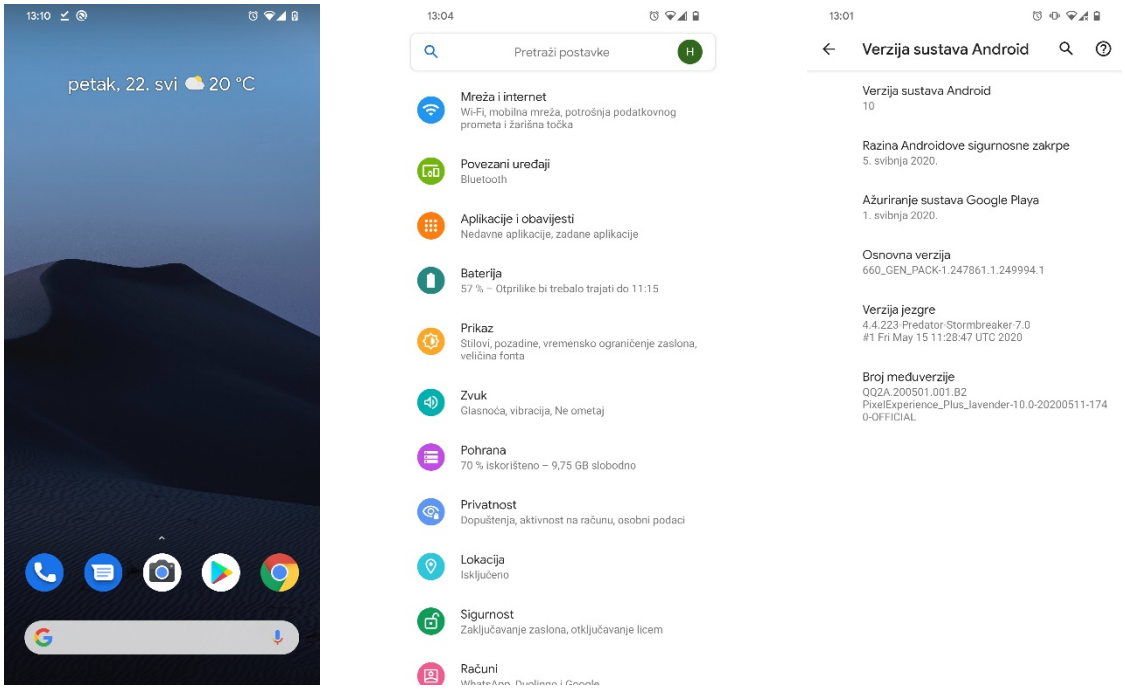
S obzirom na to da su *jailbreak* postupci usko vezani uz sigurnosne ranjivosti koje ih omogućuju, njihova forenzička ispravnost može se znatno razlikovati. Na novijim inačicama iOS operativnog sustava, točnije od inačice 10 pa na dalje, provođenje *jailbreak* postupka je zahtijevalo omogućenu internetsku konekciju. Razlog tome je što su korišteni Apple račun i njegova pridružena lozinka morali biti ovjereni putem Appleovih službenih poslužitelja. Međutim, pojavom checkm8 sigurnosne ranjivosti, stvorena je checkra1n *jailbreak* metoda koja ne zahtjeva internetsku konekciju te ne pravi trajne izmjene u sustavu uređaja. Nedostatak checkm8 sigurnosne ranjivosti je taj što njome nisu obuhvaćeni uređaji noviji od modela iPhone X. Kako bi se održao integritet prikupljenih podataka, svaku metodu potrebno je pojedinačno vrednovati te dokumentirati njihov utjecaj na sustav i skladištene podatke, [53].

U pogledu Android uređaja, forenzička ispravnost *root* postupka također se razlikuje ovisno o korištenoj metodi. Kao što je objašnjeno prethodnim poglavljima, primjenom *one-click-root* aplikacija može doći do neželjenih izmjena u sustavu. Iako takve izmjene nisu nužno zlonamjerne, svejedno mogu narušiti integritet prikupljenih podataka i time ih učini neuporabljivima na sudu. Kao i u slučaju *jailbreak* postupka, utjecaj takvih aplikacija potrebno je jasno definirati kako bi se očuvao integritet podataka. S druge strane, Magisk *root* postupkom ne prave se izmjene u sustavu uređaja, ali je za njegovo izvođenje potrebno ispuniti određene uvjete kao što je otključavanje *bootloadera*. U slučajevima kada proizvođač ne omogućava otključavanje *bootloadera* ili ga otežava, a uređaj ne podržava *one-click-root* aplikacije, *root* postupak može biti proveden isključivo korištenjem sigurnosnih ranjivosti.

6. Postupci ekstrakcije i analize podataka

6.1. Postupci ekstrakcija podataka s Android uređaja

Postupak ekstrakcije podataka obavljen je na dvije inačice Android sustava, pri čemu jedna inačica koristi FDE mehaniku šifriranja podataka, a druga inačica FBE mehaniku šifriranja podataka. Instalacijski paketi OS-a preuzeti su izvora [70] u zip formatu te instalirani pomoću TWRP particije za oporavak. Navedena inačica Android OS-a, čije je sučelje prikazano slikom 25, je modificirana na način da sadrži softverske značajke Google Pixel pametnih telefona što također uključuje aplikacije Google servisa kao i najnovija sigurnosna ažuriranja. Nadalje, prije postupka ekstrakcije, podaci su generirani uobičajenom uporabom pametnog telefona.



Slika 25. Sučelje PixelExperience OS-a

Postupci logičke i fizičke ekstrakcije obavljeni su pomoću Android Debug Bridge (ADB) funkcionalnosti ugrađenih u razvojni alat Android softvera (eng. Software development kit – SDK). Alat je preuzet s izvora [71], a omogućava upravljanje funkcijama pametnog telefona. U tu svrhu, korištena je Unix *shell* klijentsko-poslužiteljska arhitektura sačinjena od tri komponente:

- Klijent, komponenta zadužena za slanje naredbi pametnom telefonu npr. *Command Promt* (CMD) tumač naredbenog retka.
- *Daemon* proces, proces pametnog telefona koji se pokreće odabirom postavke USB otklanjanje pogrešaka. Omogućuje pokretanje naredbi zaprimljenih od klijenta.
- Poslužitelj, komponenta koja upravlja komunikacijom klijenta i poslužitelja, [71].

6.1.1. Procedura logičke ekstrakcije podataka

ADB funkcionalnost koja omogućava logičku ekstrakciju podataka jest naredba `adb backup`. Spomenutom naredbom, izrađuje se sigurnosna kopija skladištenih podataka pri čemu prethodno nije potrebno obaviti *root* postupak. Nadalje, obuhvat skladištenih podataka je moguće daljnje definirati sljedećim parametrima:

- *apk/noapk*, parametar pomoću kojeg se omogućava ili onemogućava sigurnosno kopiranje instalacijskih datoteka aplikacija (eng. *Android Package Kit* – APK)
- *shared*/*noshared*, parametar kojim se omogućava ili onemogućava sigurnosno kopiranje podataka skladištenih na unutarnjoj i vanjskoj pohrani uređaja
- *all*, parametar koji omogućava sigurnosno kopiranje svih dostupnih aplikacija
- *system*/*nosystem*, parametar koji omogućava ili onemogućuje sigurnosno kopiranje sistemskih aplikacija, [72].

Nakon uspješnog povezivanja pametnog telefona s računalom, unutar CMD tumača pokrenuta je naredba:

```
adb backup -apk -shared -all -system
```

Postupak izrade sigurnosne kopije također je bilo potrebno potvrditi putem sučelja pametnog telefona. Kao što je vidljivo na slici 26, ekstrahirani podaci mogu biti dodatno šifrirani proizvoljno definiranom zaporkom, međutim u svrhu diplomskog rada, sigurnosna kopija kreirana je bez unosa zaporke. Nakon uspješne izrade sigurnosne kopije podataka, unutar direktorija SDK alata stvorena je datoteka .ab formata. Zbog toga što program, korišten u svrhu daljnje analize podataka, ne podržava datoteke .ab formata, kreiranu datoteku je također bilo potrebno pretvoriti u .tar format pomoću alata preuzetog s izvora [73].

Slika 26. Prikaz korisničkog sučelja za stvaranje sigurnosne kopije

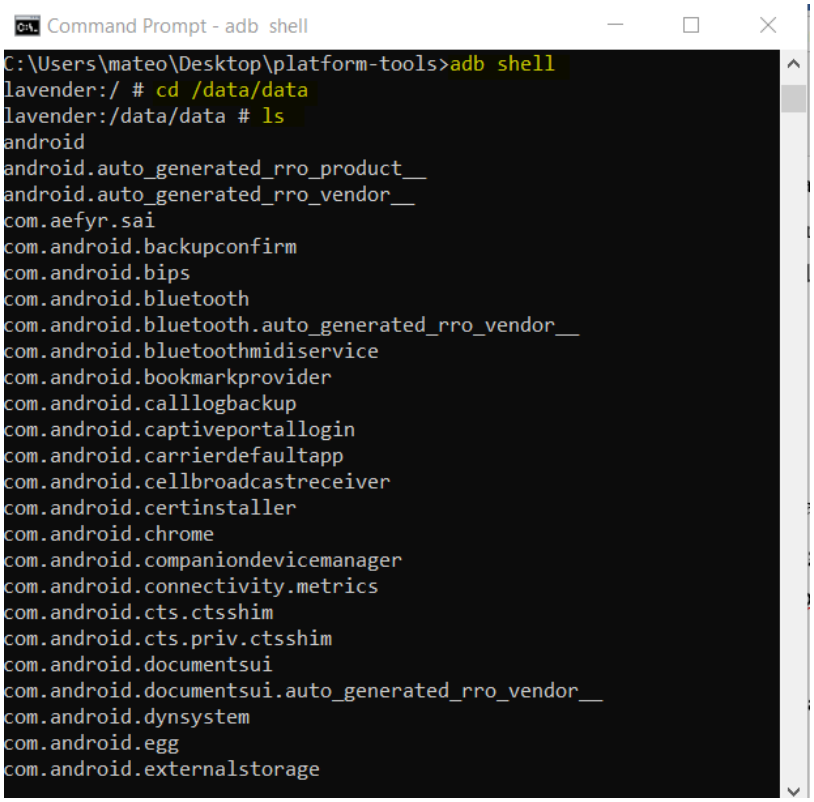
S obzirom na to da se dio korisničkih podataka nalazi u direktoriju */data/data* kojem nije moguće pristupiti bez *root* privilegija, naredbom `adb backup` nije stvorena potpuna kopija svih podataka. Iz tog razloga, također je obavljena ručna logička ekstrakcija podataka nakon što je na uređaju proveden *root* postupak opisan prethodnim poglavljem, uz izuzetak koraka otključavanja *bootloadera* koji je proveden prije instalacije Android OS-a. Potpuna lista datoteka skladištenih unutar direktorija */data/data* dobivena je naredbama:

```
adb shell

cd data/data

ls
```

Prvom naredbom ostvaren je pristup ADB upravljačkom alatu pametnog telefona, dok su druge dvije naredbe korištene u svrhu navigacije na lokaciju željenog direktorija te ispisivanja svih datoteka pohranjenih na navedenoj lokaciji. Kao što je vidljivo na slici 27, unutar prozora CMD tumača naredbenog retka ispisani su nazivi svih skladištenih direktorija unutar direktorija */data/data*.



Slika 27. Prikaz data/data direktorija

Nakon toga, naredbom `adb root` pokrenuta je konekcija s *root* privilegijama. Prema zadanim postavkama Magisk *root* ne podržava pokretanje `adb root` naredbe, pa je na uređaj također prethodno instaliran Magisk modul preuzet s izvora [74]. Ekstrakcija podataka obavljena je za svaki direktorij prikazan prethodno dobivenim popisom, pri čemu je korištena sintaksa:

```
adb pull /data/data/[Naziv direktorija] [Putanja odredišta]
```

Postupkom ručne logičke ekstrakcije podataka svaka je datoteka spremljena u zaseban direktorij, međutim zbog grešaka nastalih u prijenosu, nije bilo moguće preuzeti sve datoteke */data/data* direktorija. Kao što je prikazano slikom 28, greške su uzrokovane nepodržanim nazivima datoteka koje su onemogućile izradu kopije putem CMD tumača. Unatoč tome, uspješno stvorene datoteke će također biti analizirane idućim poglavljem.

```
C:\Users\mateo\Desktop\platform-tools>adb pull /data/data/com.viber.voip C:\Users\mateo\Desktop\platform-tools\Adb_data
adb: error: failed to create directory 'C:\Users\mateo\Desktop\platform-tools\Adb_data\com.viber.voip\files\.Fabric\com.crashly
tics.sdk.android:answers\': Invalid argument
```

Slika 28. Prikaz primjera greške nastalog prilikom prijenosa podataka ručnom logičkom ekstrakcijom

6.1.2. Procedura fizičke ekstrakcije podataka

U svrhu obavljanja fizičke ekstrakcije, na računalo je instaliran ncat alat s izvora [75] koji omogućava uspostavu konekcije putem koje će biti preneseni podaci. S obzirom na to da Android uređaji prema zadanim postavkama ne sadržavaju ncat alat, na pametni telefon je također instaliran Busybox Magisk modul. Busybox je skup različitih Unix softverskih alata te također uključuje ncat, [76].

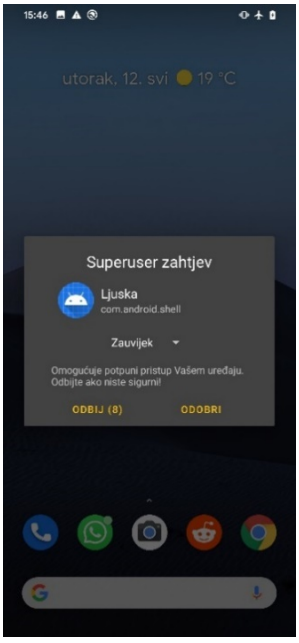
Postupak je izveden pomoću dva prozora CMD tumača naredbi pokrenuta na računalu. Prvi prozor korišten je u svrhu upravljanja funkcijama pametnog telefona, a drugi u svrhu upravljanja konekcijom pametnog telefona i računala. Nakon spajanja pametnog telefona na računalo, u prvom CMD prozoru korištene su sljedeće naredbe:

```
adb -d shell
```

```
su
```

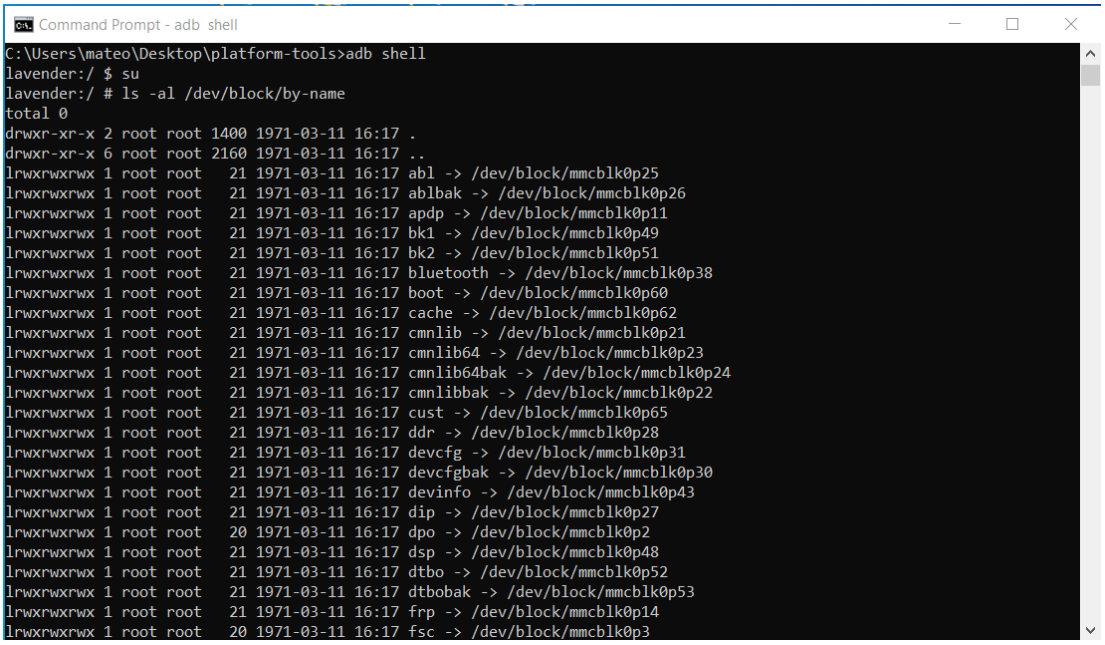
```
ls -al /dev/block/by-name
```

Prvom naredbenom linijom ostvaruje se pristup ADB upravljačkom alatu pametnog telefona pomoću kojeg će biti izvedene naredbe preslikavanja particije korisničkih podataka. Drugom naredbenom linijom, ostvarene su *root* privilegije koje također moraju biti potvrđene unutar korisničkog sučelja pametnog telefona prikazanog slikom 29.



Slika 29. Magisk korisničko sučelje za odobravanje ili odbijanje *root* privilegija

Posljednjom naredbenom linijom dohvaća se popis svih logičkih particija čiji je djelomičan popis prikazan slikom 30. Popis olakšava identifikaciju particije koja sadrži korisničke podatke. Prema popisu, korisnički podaci skladišteni su unutar *mmcblk0p66* logičke particije, međutim daljnjim ispitivanjem utvrđeno je kako je navedena particija šifrirana što, iako ne onemogućava ekstraktiranje podataka, sprječava daljnju analizu podataka. Unatoč tome također je utvrđeno da se, u slučaju FDE mehanike šifriranja, korisnički podaci dešifriraju prilikom prvog otključavanja zaslona te spremaju unutar particije *dm-0*.



Slika 30. Popis svih particija uređaja

Nakon identificiranja ispravne logičke particije, u drugom CMD prozoru ispisuje se naredba pomoću koje se prosljeđuje sva komunikacija povezanog uređaja:

```
adb forward tcp:8888 tcp:8888
```

Kako bi podaci mogli biti preneseni, unutar prvog prozora korištena je Linux `dd` funkcija koja omogućava preslikavanje podataka s jednog mjesta na drugo. Parametri korišteni u sklopu ove naredbe omogućuju identifikaciju lokacije s koje će podaci biti preuzeti, te lokaciju na koju će podaci biti preslikani. Primjer naredbe korištene u svrhu fizičke ekstrakcije podataka FDE inačice sustava glasi:

```
dd if=/dev/block/dm-0 | busybox nc -l -p 8888
```

Posljednjom naredbom, unutar drugog CMD prozora, pokrenuta je ncat konekcija na računalu, te je definiran naziv datoteke unutar koje će biti preneseni svi podaci odabrane particije:

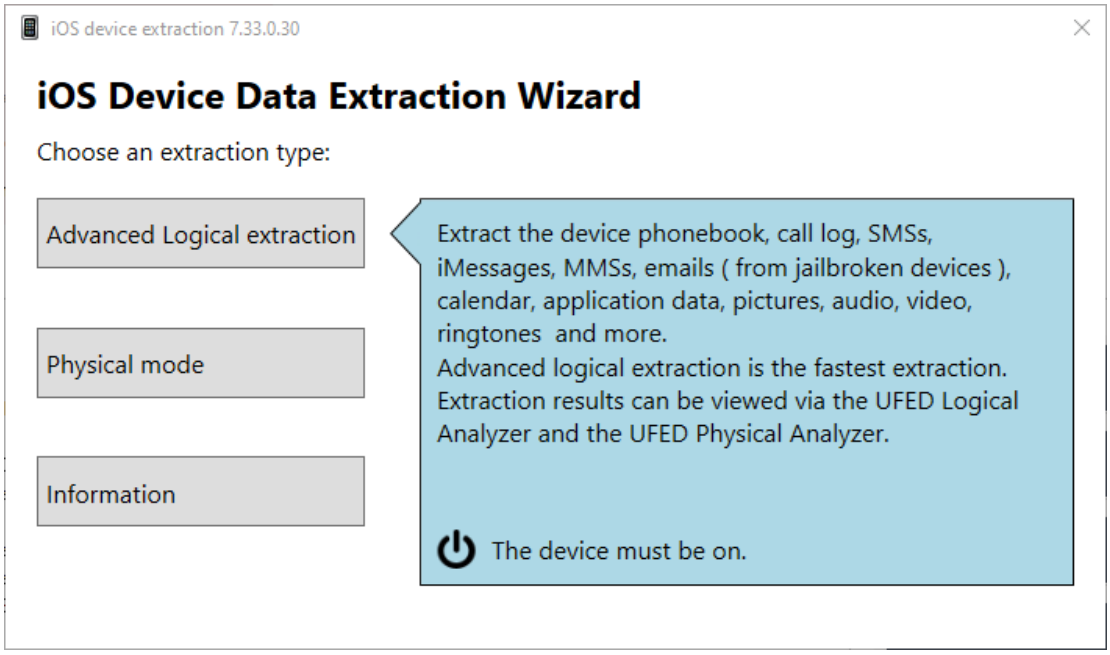
```
ncat.exe 127.0.0.1 8888 > backup.dd
```

Nakon uspješno obavljenog postupka fizičke ekstrakcije, svi podaci sadržani su unutar backup.dd datoteke koja će biti detaljnije analizirana pomoću programa Autopsy.

6.2. Postupci ekstrakcija podataka s iPhone uređaja

Ekstrakcija podataka s prethodno spomenutog iPhone 4 telefona obavljena je pomoću UFED Physical Analyzer programa. Postupak je obavljen u Laboratoriju za sigurnost i forenzičku analizu Fakulteta prometnih znanosti pod nadzorom voditelja dr. sc. Siniše Husnjaka. Zbog zastarjelosti zadnje podržane iOS inačice nije bilo moguće instalirati aplikacije s Apple App trgovine, te su stoga generirani samo osnovni podaci kao što su pozivi, SMS poruke, povijest preglednika i drugi slični podaci.

Kao što je prikazano slikom 31, program omogućava dvije osnovne vrste ekstrakcije podataka tj. fizičku ekstrakciju (eng. *Physical mode*) podataka te naprednu logičku ekstrakciju (eng. *Advanced Logical extraction*) podataka.



Slika 31. Prikaz vrsta ekstrakcije UFED Physical Analyzer programa

Napredna logička ekstrakcija ograničena je količinom podataka koju može oporaviti s uređaja, a u svrhu njezinog izvođenja UFED Physical Analyzer koristi tri metode:

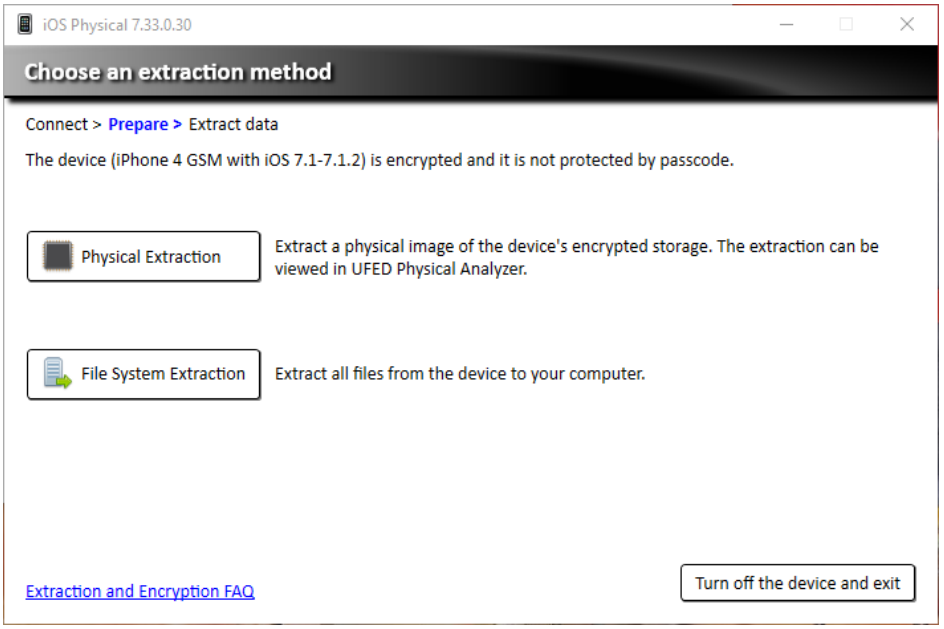
- Metoda 1 - koristi Apple File Connection (AFC) aplikacijsko programskog sučelja u svrhu stvaranja sigurnosne kopije. Navedeno sučelje se uobičajeno koristi prilikom stvaranja iTunes sigurnosnih kopija uređaja.
- Metoda 2 – koristi se u slučaju da je uređaj šifriran te nije poznata zaporka otključavanja. Njome se izdvajaju podaci sigurnosne kopije.
- Metoda 3 – koristi se u slučaju šifriranih i nešifriranih uređaja nad kojima je proveden *jailbreak* postupak, [77].

Iako je naprednu logičku ekstrakciju podataka moguće provesti i u slučaju da nad telefonom nije obavljen *jailbreak* postupak, ona potencijalno može prikupiti manju količinu podataka. Iz tog razloga je za potrebe diplomskog rada, postupak proveden prije i poslije obavljanja *jailbreak* postupka. Prilikom spajanja telefona na računalo, telefon mora biti uključen kako bi ga program mogao prepoznati, a ovisno o modelu uređaja, inačici sustava te *jailbreak* postupku, program automatski odabire neke od prethodno spomenutih metoda. Uspješnim obavljanjem postupka stvorena je datoteka .tar ekstenzije koja je također dodatno procesirana pomoću UFED Physical Analyzer programa.

U odnosu na naprednu logičku ekstrakciju podataka, odabirom fizičke ekstrakcije podataka uređaj je također potrebno isključiti prije povezivanja na računalo te ga pokrenuti u DFU načinu rada kako bi telefon mogao zaprimiti forenzičke programe potrebne za izvršavanje fizičke ekstrakcije. Kao što je prikazano slikom 32, UFED Physical Analyser pruža mogućnost izbora dviju procedura: metodu fizičke ekstrakcije i metodu preslikavanja datotečnog sustava. Metoda preslikavanja datotečnog sustava je poseban oblik logičke ekstrakcije koja koristi različit zbir protokola. Na taj način, osim uobičajenih podataka dobivenih logičkom ekstrakcijom, moguće je također pristupiti i preslikati inače skrivene ili nedostupne podatke.

S druge strane, metodom fizičkom ekstrakcijom stvara se bit po bit preslika potpunog sadržaja unutarnje pohrane telefona. U tu svrhu koristi se modificirani *bootloader* koji se učitava na telefon pomoću jedne od prethodno otkrivenih sigurnosnih ranjivosti. Zbog ovisnosti metode o sigurnosnim ranjivostima, fizičku ekstrakciju podataka moguće izvesti isključivo na starijim iPhone telefonima pri čemu je iPhone 4 posljednji podržani model. Serija

čipova korištena na tom modelu, ali i na starijim modelima, omogućuje izvršavanje nepotpisanog koda na telefonu. U tu svrhu korištena je sigurnosna ranjivost koja također omogućava provođenje *jailbreak* postupka, međutim u ovom slučaju je korištena u svrhu pokretanja zasebnog operativnog sustava pomoću kojeg se ekstrahiraju podaci, [77].



Slika 32. Prikaz načina obavljanja fizičke ekstrakcije podataka

Obavljanjem metode preslikavanja podataka datotečnog sustava također je dobivena datoteka .tar ekstenzije, dok je u slučaju fizičke ekstrakcije stvorena datoteka .img ekstenzije. Sve dobivene datoteke su također procesirane u svrhu daljnje analize i komparacije.

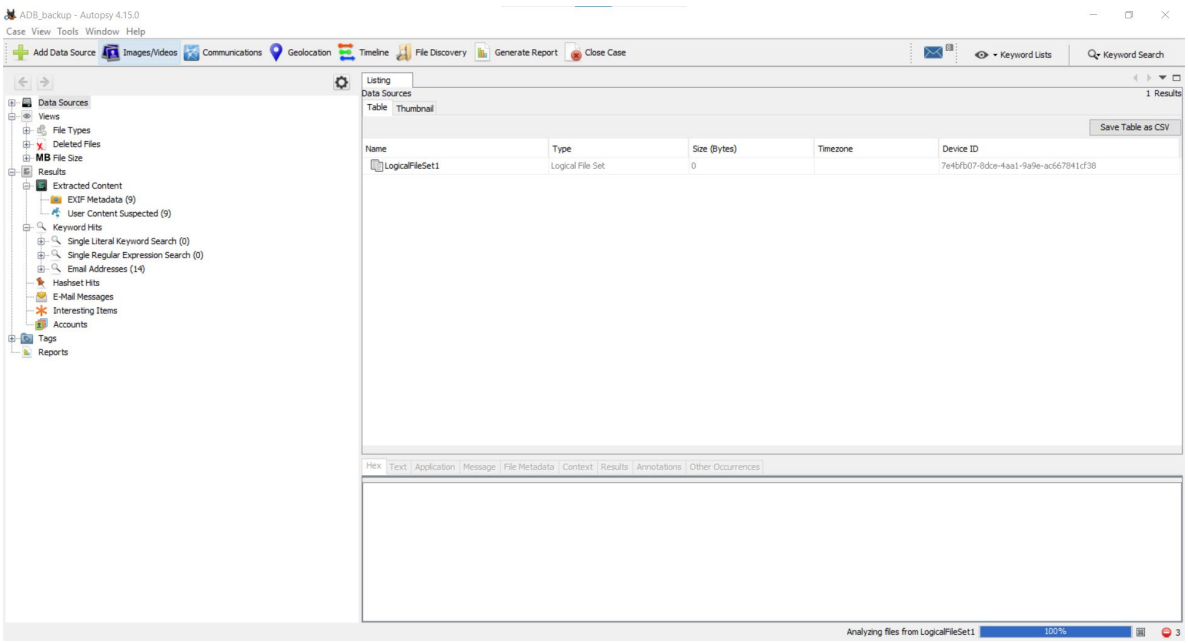
6.3. Analiza i komparacija prikupljenih podataka

6.3.1. Analiza i komparacija podataka Android telefona

Analiza prikupljenih podataka obavljena je pomoću Autopsy alata preuzetog putem web stranice [78]. Nakon instalacije, pokretanjem programa prikazuje se početno sučelje prikazano slikom 33. Putem početnog sučelja moguće je započeti analizu novih podataka ili otvoriti prethodno obavljenu analizu podataka. S obzirom na to da je ekstrakcija podataka Android uređaja obavljena na tri načina, postupak analize obavljen je za sva tri slučaja.

Slika 33. Početno sučelje Autopsy alata

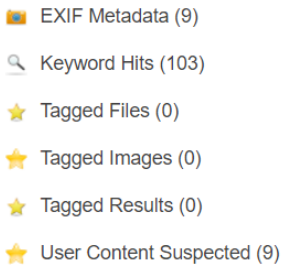
Pritiskom tipke *New Case* otvara se slijed prozora unutar kojih je moguće definirati: naziv slučaja, informacije o ispitivaču, format podataka koji će biti analizirani, putanja direktorija u kojem se nalaze ekstrahirani podataka te putanju direktorija unutar koje će biti spremljeni rezultati analize. Također mogu biti odabrani različiti moduli pomoću kojih se provode različite analize ekstrahiranih podataka. Pokretanjem analize prikazuje se konačno sučelje putem kojeg je moguće pratiti napredak procesa analize te vrstu i količinu podataka koja se analizira. Slikom 32 prikazano je sučelje alata prilikom analiziranja logički ekstrahiranih podataka.



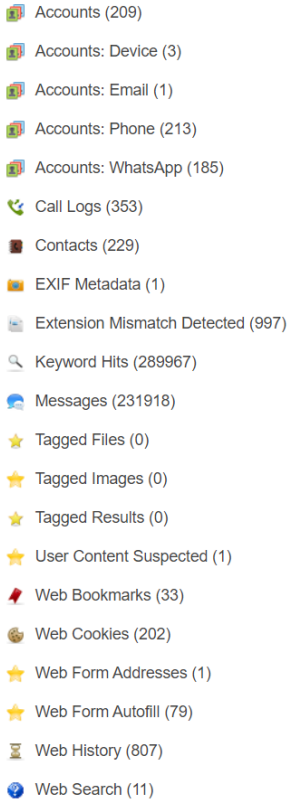
Slika 34. Prikaz glavnog sučelja Autopsy alata

Završetkom postupka analize podataka, Autopsy alat također pruža mogućnost izrade izvješća koja su prikazana slikom 35. Kao što je vidljivo sa slike, analiza podataka dobivenih logičkom ekstrakcijom, bez prethodnog provođenja *root* postupka, dala je najmanju količinu rezultata. Unatoč tome što su postupkom ekstrahirani svi podaci skladišteni na vanjskoj i unutarnjoj pohrani, nije bilo moguće pristupiti korisničkim podacima skladištenim unutar */data/data* direktorija što je utjecalo na manju količinu rezultata. S druge strane, analiza logički prikupljenih podataka, s prethodno obavljenim *root* postupkom, pružila je veću količinu rezultata, međutim nastale greške u prijenosu onemogućile su potpunu ekstrakciju svih podataka. Analizom fizički ekstrahiranih podataka dobivena je najveća količina rezultata pri čemu su također prikupljeni podaci skladišteni unutar neraspoređenog prostora unutarnje pohrane.

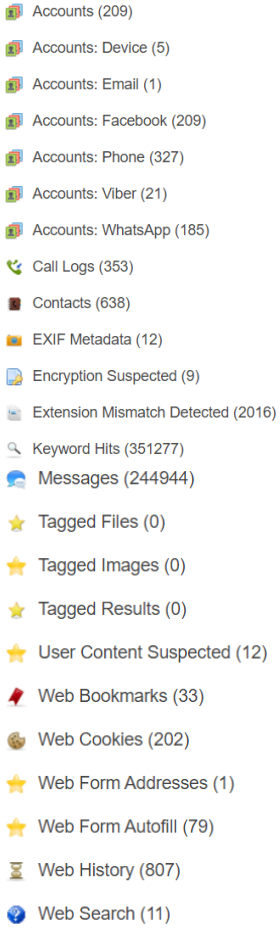
a) Logički ekstrahirani podaci



b) *Root* logički ekstrahirani podaci



c) Fizički ekstrahirani podaci

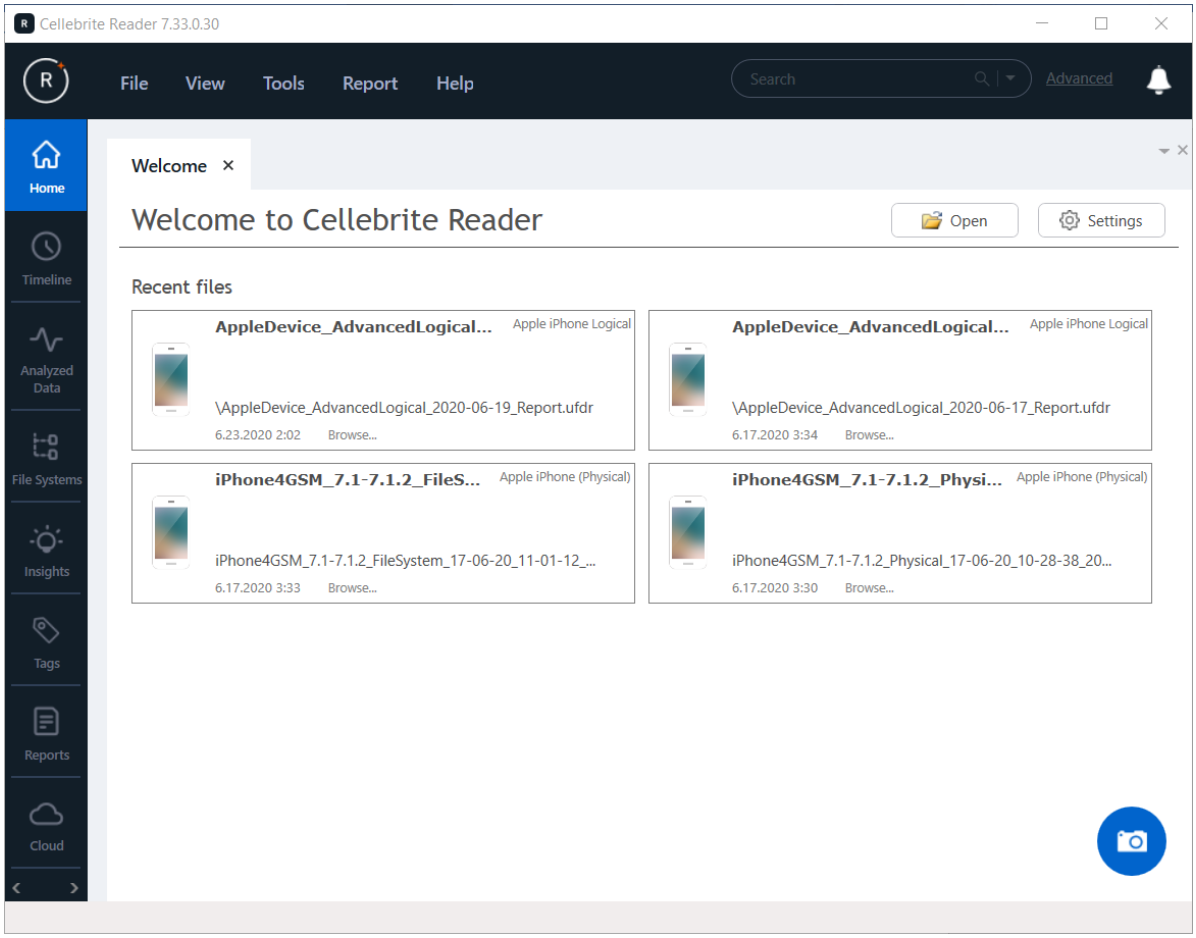


Slika 35. Rezultati analize logičkih ekstrahiranih podataka bez provedenog *root* postupka (slika 35-a), rezultati analize logičkih ekstrahiranih podataka s provedenim *root* postupkom (slika 35-b) i rezultati analize fizički ekstrahiranih podataka s provedenim *root* postupkom (slika 35-c)

Ispitivanje podataka ekstrahiranih s FBE inačice sustava nije moglo biti izvršeno. U slučaju logički ekstrahiranih podataka, podatke nije bilo moguće pretvoriti u ispravan oblik čitljiv Autopsy alatu. Nadalje, podaci dobiveni fizičkom ekstrakcijom bili su u potpunosti šifrirani, te za razliku od FDE inačice, nije postojala particija koja je sadržavala dešifrirane podatke nakon otključavanja zaslona.

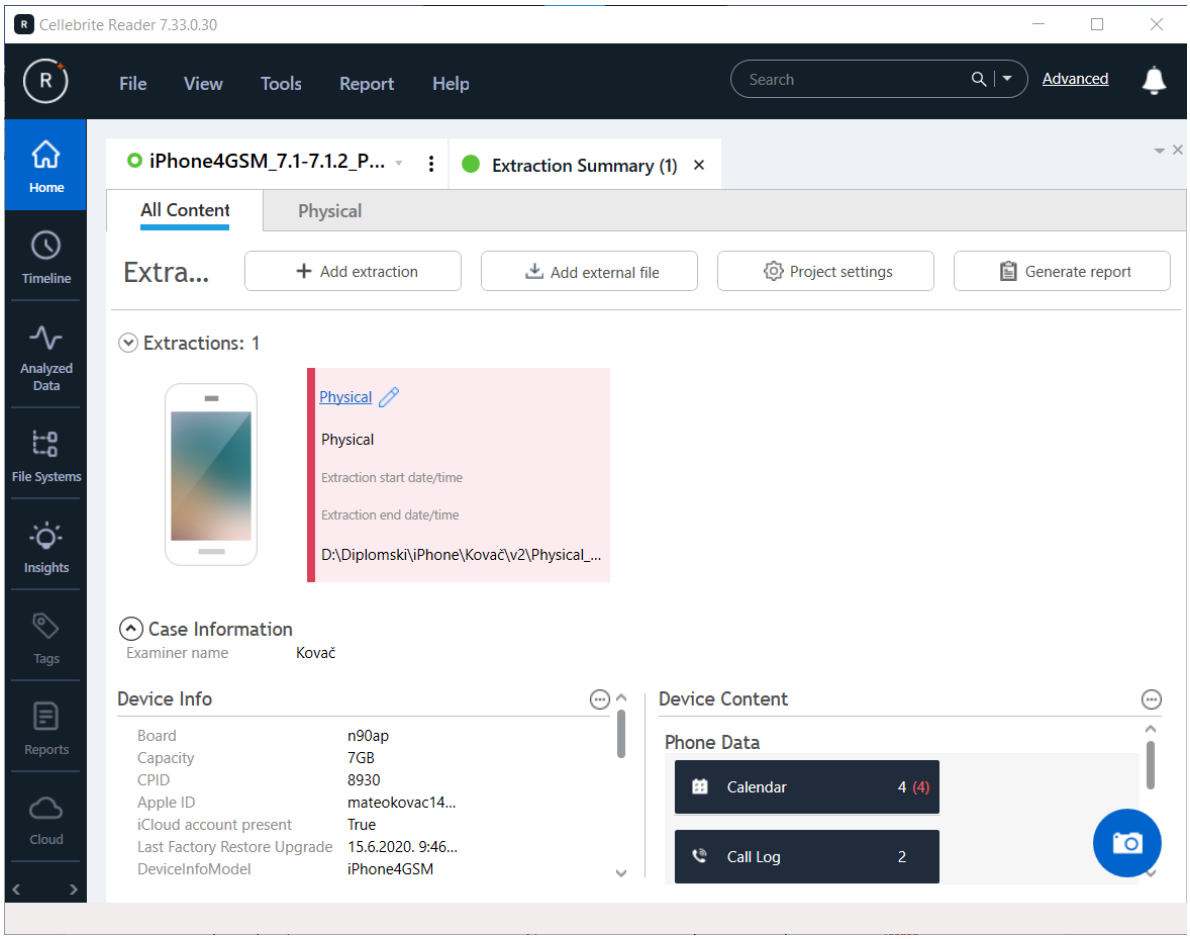
6.3.2. Analiza i komparacija podataka iOS telefona

Analiza podataka prikupljenih s iPhone 4 telefona obavljena je pomoću programa Cellebrite Reader u Laboratoriju za sigurnost i forenzičku analizu Fakulteta prometnih znanosti pod nadzorom voditelja dr. sc. Siniše Husnjaka. Početno sučelje programa, prikazano slikom 36, omogućava očitavanje prethodno analiziranih podataka za potrebe daljnje analize. Pri tome je logička ekstrakcija provedena prije i poslije obavljanja *jailbreak* postupka.



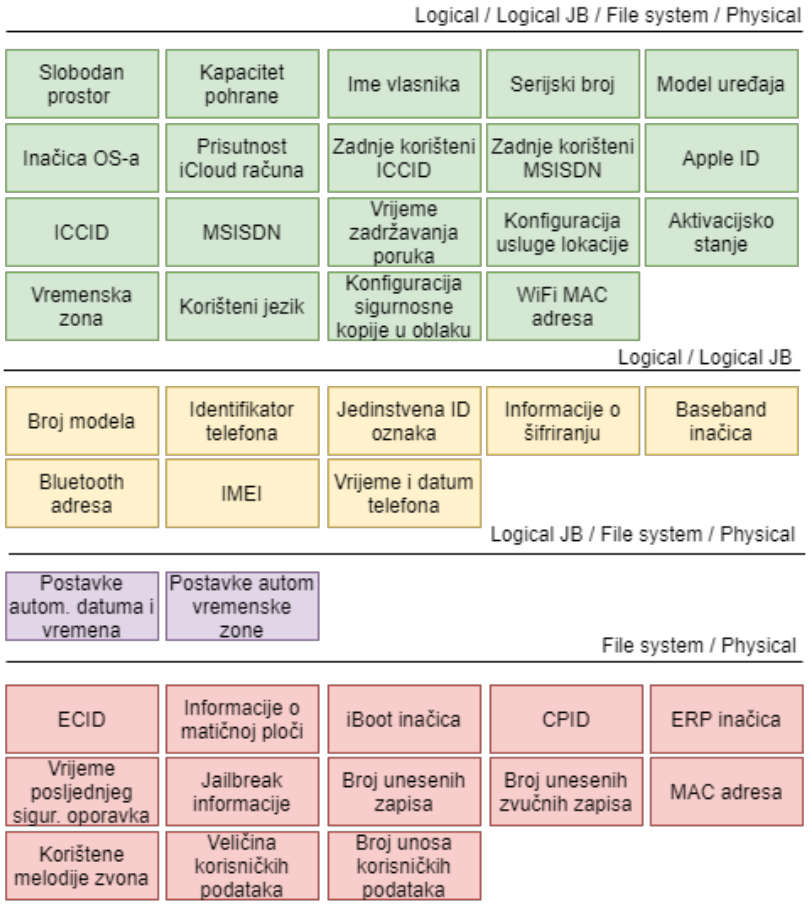
Slika 36. Početno sučelje Cellebrite Reader programa

Kao što je vidljivo na slici 37, očitavanjem ekstrahiranih podataka otvara se sažetak ekstrakcije iz kojeg je moguće očitati osnovne informacije o uređaju, te sadržaj svih pohranjenih podataka.



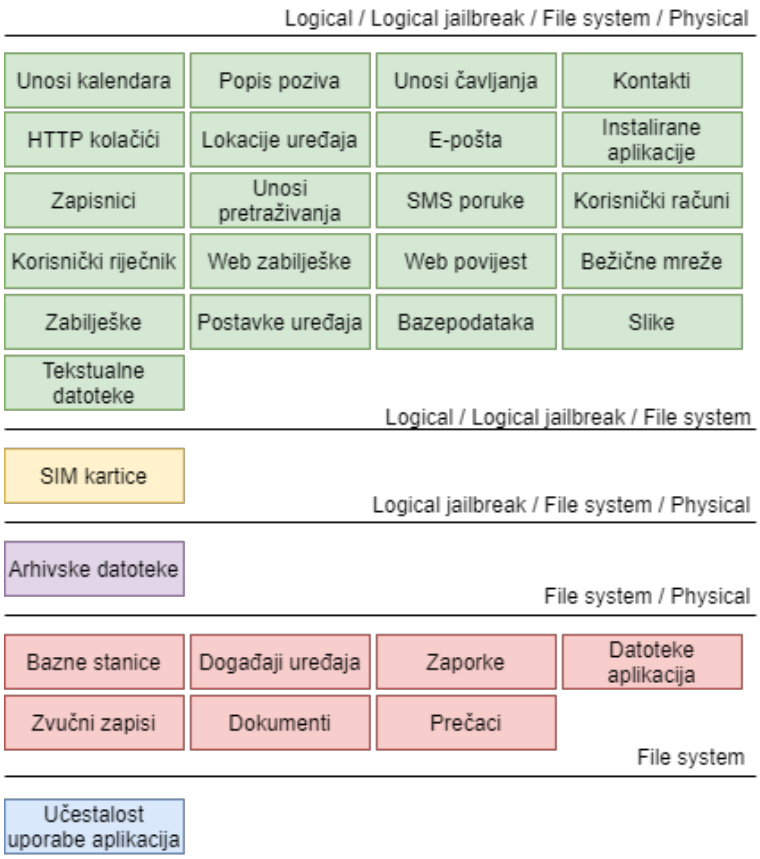
Slika 37. Prikaz sučelja sažetka analiziranih podataka

Slikom 38 prikazane su osnovne prikupljene informacije o uređaju kategorizirane prema korištenim metodama ekstrakcije.



Slika 38. Kategorizirani prikaz osnovnih informacija o uređaju prikupljenih pomoću korištenih metoda ekstrakcije podataka

Vrste prikupljenih podataka prikazane su slikom 39 te su također kategorizirane prema korištenim postupcima. S obzirom na to da je fizičkim postupkom ekstrakcije stvorena bit po bit preslika particije telefona, njome je također obuhvaćen neraspodijeljen prostor na kojem je moguće pronaći izbrisane podatke. Pri tome su oporavljeni samo oni podaci koji su izbrisani uobičajenim putem tj. putem korisničkog sučelja telefona. Podatke izbrisane pokretanjem naredbe vraćanja tvorničkih postavki uređaja nije bilo moguće oporaviti.



Slika 39. Prikaz prikupljenih podataka prema korištenim metodama ekstrakcije podataka

U odnosu na logičke metode ekstrakcije podataka, metoda fizičke ekstrakcije i metoda preslikavanja datotečnog sustava prikupile su više vrsta podataka. Razlog tome su načini provođenja tih postupaka, opisani poglavljem 6.2., koji su omogućili pristupanje inače nedostupnim ili sakrivenim datotekama. Nadalje, logičkom metodom obavljenom nakon provođenja *jailbreak* postupka oporavljena je jedna arhivska datoteka sadržana u direktoriju korisničkog rječnika koju nije bilo moguće oporaviti bez obavljanja *jailbreak* postupka.

7. Zaključak

Zaštita privatnosti i sigurnost podataka danas su jedni od najvažnijih zahtjeva koje proizvođači pametnih telefona moraju ispuniti. Iako ispunjavanje takvih zahtjeva dovodi do bolje zaštite potrošača, također utječe na provođenje legitimnih radnji kao što su forenzička ispitivanja u sklopu kriminalističkih istraga. Digitalna forenzika je posebna grana forenzike koja se bavi prikupljanjem, analizom i dokumentiranjem neobrađenih digitalnih podataka. S obzirom na to da je podatke moguće prikupiti s više različitih uređaja stvorena je posebna podgrana, forenzika pametnih telefona, kojom se nastoji jasnije definirati metodologiju i smjernice provođenja ispitivanja pametnih telefona.

Jedna od često korištenih metoda pristupa osjetljivim podacima u domeni digitalne forenzike je postupak eskalacija privilegija. Riječ je o postupku pomoću kojeg se osigurava pristup administratorskom računu odnosno korisničkom računu koji posjeduje privilegije potrebne za pristupanje svim podacima skladištenim na uređaju. Na tržištu pametnih telefona danas su najkorištenija dva operativna sustava, Android i iOS, a postupci eskalacije privilegija korišteni na tim sustavima nazivaju se *root* i *jailbreak* postupci. Iako je krajnji cilj takvih postupaka isti, njihove procedure se razlikuju u skladu s značajkama sustava.

Zbog otvorenosti Android OS-a, *root* postupke je moguće obaviti bez potrebe iskorištavanja sigurnosnih ranjivosti, odnosno nenamjernih nedostataka hardvera i softvera. Pri tome su na ranijim inačicama korištene *system root* metode kojima su se pravile direktne izmjene u sustavu, dok se na novijim inačicama koristi *systemless root* metode također poznata kao Magisk *root* metoda. Njome se ne prave direktne izmijene u sustavu, već se izmjenjuje inicijalizacijski proces sadržan u particiji *bootloadera*.

Jailbreak postupke je, za razliku od *root* postupaka, moguće izvesti isključivo iskorištavanjem sigurnosnih ranjivosti. Kroz povijest je postojao veći broj *jailbreak* postupaka s obzirom na to da je Apple sigurnosnim ažuriranjem ispravljao svaku nastalu sigurnosnu ranjivost. Najvažnije sigurnosne ranjivosti su one otkrivene na *bootrom* razini zato što je njih nemoguće ispraviti softverski te garantiraju trajnost postupka čak i nakon ažuriranja sustava na noviju inačicu.

Za potrebe diplomskog rada, *root* postupak proveden je na Redmi Note 7 pametnom telefonu pomoću Magisk *root* metode pri čemu su postupci ekstrakcije provedeni prije i poslije *root* postupka. S obzirom na to da fizičku ekstrakciju nije moguće provesti bez administratorskih privilegija, analizom je utvrđeno kako je puno veća količina podataka prikupljena nakon obavljanja *root* postupka pri čemu su također oporavljeni prethodno izbrisani podaci koji su ostali pohranjeni na unutarnjoj pohrani uređaja. Iako je logičku ekstrakciju također bilo moguće provesti uz pomoć *root* privilegija, greške u prijenosu datoteka onemogućile su potpuni obuhvat svih skladištenih podataka.

S druge strane, *jailbreak* postupak proveden je nad iPhone 4 uređaju, te je u svrhu ekstrakcije podataka korišten je UFED Physical Analyzer koji automatiziranim procesom prepoznaje priključeni uređaj. Pomoću navedenog programa bilo je moguće obaviti dvije vrste fizičke ekstrakcije te jedna logička ekstrakcija, pri čemu su fizičke metode zahtijevale upotrebu sigurnosnih ranjivosti, dok je logička metoda koristila Apple programsko sučelje pomoću kojeg se stvaraju iTunes sigurnosne kopije. Rezultati su pokazali kako je najveća količina podataka prikupljena fizičkim metodama, pri čemu nije bilo moguće oporaviti podatke obrisane funkcijom tvorničkog brisanja uređaja.

Sukladno dobivenim rezultatima ali i provedenom teorijskom istraživanju, zaključeno je kako *root* i *jailbreak* postupci imaju veliki značaj u pogledu ekstrakcije podataka, ali su također ograničeni u slučajevima kada uređaj koristi napredniji oblik šifriranja podataka. Provođenjem tih postupaka ili direktnim iskorištavanjem sigurnosnih ranjivosti koje ih omogućuju, moguće je prikupiti veću količinu podataka koja je inače nedostupna prilikom uobičajenog pregleda. Preduvjeti postupka mogu se razlikovati ovisno o odabranoj metodi, naročito u slučaju *jailbreak* postupka, radi čega je važno detaljno dokumentirati sve provedene korake kako bi se sačuvao integritet dobivenih podataka, ali i jasno dokazalo što je u uređaju bilo potrebno izmijeniti kako bi bilo moguće doći do podataka. Forenzička ispravnost navedenih postupaka još uvijek nije jasno utvrđena te se može razlikovati ovisno o odabranoj metodi, međutim u slučajevima kada analizu nije moguće provesti drugačije, *jailbreak* i *root* postupci mogu biti sredstva koja opravdavaju cilj.

Literatura

[1] GCFGlobal. Preuzeto sa: https://edu.gcfglobal.org/en/computerbasics/understanding-operating-systems/1/ [Pristupljeno: svibanj 2020.]

[2] Android Authority. Preuzeto sa: https://www.androidauthority.com/history-android-os-name-789433/ [Pristupljeno: svibanj 2020.]

[3] The Verge. Preuzeto sa: https://www.theverge.com/2019/8/22/20827231/android-10-q-google-name-officially-announced-new-logo-wordmark-desserts [Pristupljeno: svibanj 2020.]

[4] Statcounter. Preuzeto sa: https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202007-202007-bar [Pristupljeno: svibanj 2020.]

[5] Statcounter. Preuzeto sa: https://gs.statcounter.com/android-version-market-share/mobile/worldwide/#monthly-202007-202007-bar [Pristupljeno: svibanj 2020.]

[6] Tamma R, Skulkin O, Mahalik H, Bommisetty S. Practical Mobile Forensics. Birmingham: Packt Publishing Ltd. Third edition; 2018.

[7] Tech Blogon. Preuzeto sa: https://techblogon.com/android-file-system-structure-architecture-layout-details/ [Pristupljeno: svibanj 2020.]

[8] URL: https://a.fsdn.com/con/app/proj/orangefox/screenshots/Screenshot_2018-06-18-17-20-09.png/max/max/1 [Pristupljeno: svibanj 2020.]

[9] DZone. Preuzeto sa: https://dzone.com/articles/android-internals-the-android-os-bootup-process [Pristupljeno: svibanj 2020.]

[10] Finder. Preuzeto sa: https://www.finder.com/ios-operating-system [Pristupljeno: svibanj 2020.]

[11] Statcounter. Preuzeto sa: https://gs.statcounter.com/vendor-market-share/mobile/worldwide/#monthly-202001-202001-bar [Pristupljeno: svibanj 2020.]

[12] iOS Security Guide Document. Preuzeto sa: https://www.apple.com/tr/business/docs/site/iOS_Security_Guide.pdf [Pristupljeno: svibanj 2020.]

[13] Statcounter. Preuzeto sa: https://gs.statcounter.com/ios-version-market-share/mobile/worldwide/#monthly-202001-202001-bar [Pristupljeno: svibanj 2020.]

[14] Elcomsoft blog. Preuzeto sa: https://blog.elcomsoft.com/2018/10/everything-about-ios-dfu-and-recovery-modes/ [Pristupljeno: svibanj 2020.]

[15] Epifani M, Stirparo P. Learning iOS Forensics. Birmingham: Packt Publishing Ltd. First edition; 2015.

[16] Android source. Preuzeto sa: https://source.android.com/security/encryption [Pristupljeno: svibanj 2020.]

[17] Seacord RC, Householder AD. A Structured Approach to Classifying Security Vulnerabilities. Carnegie Mellon Software Engineering Institute. 2005-TN-003.

[18] SearchSecurity. Preuzeto sa: https://searchsecurity.techtarget.com/definition/exploit [Pristupljeno: svibanj 2020.]

[19] CVE Details. Preuzeto sa: https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224 [Pristupljeno: svibanj 2020.]

[20] CVE Details. Preuzeto sa: https://www.cvedetails.com/product/15556/Apple-Iphone-Os.html?vendor_id=49 [Pristupljeno: svibanj 2020.]

[21] Apple Developer. Preuzeto sa: https://developer.apple.com/security-bounty/ [Pristupljeno: svibanj 2020.]

[22] Google Project Zero. Preuzeto sa: https://googleprojectzero.blogspot.com/p/vulnerability-disclosure-faq.html [Pristupljeno: svibanj 2020.]

[23] 9to5Mac. Preuzeto sa: https://9to5mac.com/2019/09/03/ios-exploit-market-report/ [Pristupljeno: svibanj 2020.]

[24] XDA Developers. Preuzeto sa: https://www.xda-developers.com/how-android-security-patch-updates-work/ [Pristupljeno: svibanj 2020.]

[25] Android central. Preuzeto sa: https://www.androidcentral.com/life-android-security-patch-it-goes-google-you [Pristupljeno: svibanj 2020.]

[26] The Verge. Preuzeto sa: https://www.theverge.com/2018/10/24/18019356/android-security-update-mandate-google-contract [Pristupljeno: svibanj 2020.]

[27] Apple Support. Preuzeto sa: https://support.apple.com/en-us/HT201222 [Pristupljeno: svibanj 2020.]

[28] SentinelLabs. Preuzeto sa: https://labs.sentinelone.com/privilege-escalation-macos-malware-path-to-root/ [Pristupljeno: svibanj 2020.]

[29] UpGuard. Preuzeto sa: https://www.upguard.com/blog/privilege-escalation [Pristupljeno: svibanj 2020.]

[30] Tamma R, Tindall D. Learning Android Forensics. Birmingham: Packt Publishing. 2015

[31] Nguyen-Vu L, Chau N, Kang S, Jung S. Android Rooting: An Arms Race beween Evasion and Detection. School of Electronic Engineering. 2017; 4121765.

[32] URL: https://www.fonedog.com/android-toolkit/superuser-management.html [Pristupljeno: svibanj 2020.]

[33] Android Central. Preuzeto sa: https://www.androidcentral.com/root [Pristupljeno: svibanj 2020.]

[34] Electronic Design. Preuzeto sa: Preuzeto sa: https://www.electronicdesign.com/technologies/embedded-revolution/article/21800663/selinux-101-what-you-shoud-know [Pristupljeno: svibanj 2020.]

[35] GitHub. Preuzeto sa: https://topjohnwu.github.io/Magisk/ [Pristupljeno: svibanj 2020.]

[36] URL: https://blogthetech.com/magisk/ [Pristupljeno: svibanj 2020.]

[37] XDA Developers. Preuzeto sa: https://www.xda-developers.com/mediatek-su-rootkit-exploit/ [Pristupljeno: svibanj 2020.]

[38] Gasparis I, Qian Z, Song C, Krishnamurthy SV. Detecting Root Exploits by Learning from Root providers. Proceeding of the 26th USENIX Conference on Security Symposium. 2017;1129-1144.

[39] XDA Developers. Preuzeto sa: https://www.xda-developers.com/best-one-click-root-2018/ [Pristupljeno: svibanj 2020.]

[40] Kaspersky daily. Preuzeto sa: https://www.kaspersky.com/blog/android-root-faq/17135/ [Pristupljeno: svibanj 2020.]

[41] URL: https://i.postimg.cc/QxjK5bfW/xiaomi-redmi-note-7-1.jpg [Pristupljeno: svibanj 2020.]

[42] Mob. Preuzeto sa: https://mob.hr/redmi-note-7-recenzija/ [Pristupljeno: svibanj 2020.]

[43] XDA Developers. Preuzeto sa: https://forum.xda-developers.com/redmi-note-7/how-to/one-redmi-note-7-unlock-bootloader-t3890751 [Pristupljeno: svibanj 2020.]

[44] URL: https://en.miui.com/unlock/ [Pristupljeno: svibanj 2020.]

[45] Android Explained. Preuzeto sa: https://www.androidexplained.com/xiaomi-bootloader-unlock-request/ [Pristupljeno: svibanj 2020.]

[46] URL: https://uploads.tapatalk-cdn.com/20170205/5e20b2ce59dcf00e7f4771c31135ec9d.jpg [Pristupljeno: svibanj 2020.]

[47] URL: https://www.jayceooi.com/wp-content/uploads/2015/07/Redmi-2-Fastboot-Mode.jpg [Pristupljeno: svibanj 2020.]
[48] URL: https://forum.xda-developers.com/redmi-note-7/development/recovery-unofficial-twrp-touch-recovery-t3921637 [Pristupljeno: svibanj 2020.]
[49] URL: https://magiskmanager.com/ [Pristupljeno: svibanj 2020.]
[50] Hackmag. Preuzeto sa: https://hackmag.com/mobile/jailbreaking-for-dummies/ [Pristupljeno: svibanj 2020.]
[51] URL: https://cydia-app.com/download/#Cydia_iOS_13_8211_1331 [Pristupljeno: svibanj 2020.]
[52] MakeUseOf. Preuzeto sa: https://www.makeuseof.com/tag/illegal-root-android-jailbreak-iphone/ [Pristupljeno: svibanj 2020.]
[53] Varenkamp P. iPhone Acquisition Using Jailbreaking Techniques. Faculty od Information Technology and Electrical Engineering. 2019.
[54] Miller C, Blazakis D, Zovi DD, Esser S, Iozzo V, Weinmann RP. iOS Hacker's Handbook. New York City: Wiley. First edition; 2012.
[55] Ars Technica. Preuzeto sa: https://arstechnica.com/gadgets/2007/10/tiff-exploits-for-iphone-safari-mail-released/ [Pristupljeno: svibanj 2020.]
[56] Gizmosscan. Preuzeto sa: https://gizmosscan.com/ios-jailbreaking-history/ [Pristupljeno: svibanj 2020.]
[57] iPhone Dev Team Portal. Preuzeto sa: http://wikee.iphwn.org/s5l8900:pwnage [Pristupljeno: svibanj 2020.]
[58] The iPhone Wiki. Preuzeto sa: https://www.theiphonewiki.com/wiki/Limera1n_Exploit [Pristupljeno: svibanj 2020.]
[59] The iPhone Wiki. Preuzeto sa: https://share.vidyard.com/watch/qE6cyEkxJ5cs7fxwuZ9QUQ? [Pristupljeno: svibanj 2020.]
[60] Cellebrite. Preuzeto sa: https://www.cellebrite.com/en/blog/ios-breakthrough-enables-lawful-access-for-full-file-system-extraction/ [Pristupljeno: svibanj 2020.]
[61] Ars Technica. Preuzeto sa: https://arstechnica.com/information-technology/2019/09/developer-of-checkm8-explains-why-idevice-jailbreak-exploit-is-a-game-changer/ [Pristupljeno: svibanj 2020.]

[62] Habr. Preuzeto sa: https://habr.com/en/company/dsec/blog/472762/ [Pristupljeno: svibanj 2020.]
[63] URL: https://mobilenmore.com/en/apple-iphone-4/ [Pristupljeno: svibanj 2020.]
[64] GSMArena. Preuzeto sa: https://www.gsmarena.com/apple_iphone_4-3275.php [Pristupljeno: svibanj 2020.]
[65] URL: http://www.3u.com/ [Pristupljeno: svibanj 2020.]
[66] Infosecaddicts. Preuzeto sa: https://infosecaddicts.com/physical-acquisition-ios-data/ [Pristupljeno: svibanj 2020.]
[67] The Symbian World. Preuzeto sa: https://sites.google.com/site/i8910wizard/symbian-s60v5-learningcenter/articles/what-types-of-memory-our-mobile-phone-use [Pristupljeno: svibanj 2020.]
[68] How Flash Memory Works. Preuzeto sa: https://computer.howstuffworks.com/flash-memory2.htm [Pristupljeno: svibanj 2020.]
[69] McKemmish R. Advances in Digital Forensics IV.2008;285: 3-15.
[70] URL: https://download.pixelexperience.org/ [Pristupljeno: svibanj 2020.]
[71] Android Developers. Preuzeto sa: https://developer.android.com/studio/command-line/adb [Pristupljeno: svibanj 2020.]
[72] Android Explorations. Preuzeto sa: https://nelenkov.blogspot.com/2012/06/unpacking-android-backups.html [Pristupljeno: svibanj 2020.]
[73] URL: https://forum.xda-developers.com/showthread.php?t=2011811 [Pristupljeno: svibanj 2020.]
[74] URL: https://github.com/evdenis/adb_root [Pristupljeno: svibanj 2020.]
[75] URL: https://nmap.org/ncat/ [Pristupljeno: svibanj 2020.]
[76] Android Acquisition using ADB, root, ncat and DD. YouTube video. Preuzeto sa: https://www.youtube.com/watch?v=KKkvkCgMeMA [Pristupljeno: svibanj 2020.]
[77] Explaining Cellebrite UFED Data Extraction Processes. Preuzeto sa: https://smarterforensics.com/wp-content/uploads/2014/06/Explaining-Cellebrite-UFED-Data-Extraction-Processes-final.pdf [Pristupljeno: svibanj 2020.]
[78] URL: https://www.autopsy.com/ [Pristupljeno: svibanj 2020.]

Popis kratica

OS	Operating system
API	Application programming interface
DVM	Dalvik Virtual Machine
USB	Universal Serial Bus
CPU	Central Proccessing Unit
RAM	Random Access Memory
OTA	Over The Air
ARM	Advanced RISC Machine
GB	Gigabyte
DFU	Device Firmware Upgrade
iBSS	iBoot Single Stage
iBEC	iBoot Epoch Change
LLB	Low Level Bootloader
FDE	Full-disk Encryption
FBE	File-based Encryption
NIST	National Institute of Standards and Technology
SU	Switch user
DAC	Discretionary Access Control
MAC	Mandatory Access Control
TWRP	Team Win Recovery Project
DRM	Digital Rights Management
DMCA	Digital Millennium Copyright Act
IMEI	International Mobile Equipment Identity
ECID	Exclusive Chip Identification
ROM	Read Only Memory
SIM	Subscriber Identity Module
SMS	Short Message Service
MMS	Multimedia Messaging Service

SDK	Software Development Kit
CMD	Command Promt
ADB	Android Debug Bridge
APK	Android Package Kit
AFC	Apple File Connection

Popis slika

Popis tablica

Popis grafikona