UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET

Primena ADB alata u analizi Android uređaja

Seminarski rad
Studijski program: Računarstvo i informatika
Modul: Softversko inženjerstvo

Mentor:
Prof. dr Bratislav Predić

Student:
Željko Vasić, br.ind. 1808

Sadržaj

1. Android Debug Bridge (ADB)

Android Debug Bridge je alat komandne linije koji omogućava komunikaciju između korisnika alata i povezanog Android uređaja ili emulatora. ADB je sastavni deo Android SDK-a (*Software Development Kit*). ADB je jedan od najvažnijih alata za upravljanje Android uređajima. Upravljanje Android uređajima obuhvata upravljanje njihovim operativnim sistemima, razvijanje aplikacija, digitalnu forenziku i administraciju uređaja. ADB funkcije se obično izvršavaju komandama u komandnoj liniji (*CMD*) na računaru.

Sastoji se od tri osnovne komponente:

1. **Klijent** - pokreće se na računaru i aktivira se izdavanjem adb komandi u komandnoj liniji.
2. **Server** - radi kao pozadinski proces na računaru. Server upravlja komunikacijom između klijenta i uređaja ili emulatora.
3. **ADB demon(adbd)** - pokreće se na samom Android uređaju. Služi kao posrednik između računara i operativnog sistema uređaja. ADB demon prima komande sa računara, interpretira ih i prosleđuje ih Android uređaju. Prima rezultat komande sa uređaja i vraća ga računaru. Njegovo pokretanje se kontroliše pomoću podešavanja *USB Debugging* koje se otključava u modu za programere.

ADB podržava komuniciranje preko Wi-Fi konekcije i preko USB konekcije. USB konekcija je češće korišćen i bezbedniji način komunikacije, pa će fokus ovog rada biti na ovoj vrsti povezivanja.
Da bi ADB mogao da funkcioniše, korisnik najpre mora aktivirati opcije za programere (*Developer options*) na uređaju i uključi USB debagiranje (*USB Debugging)*. Aktiviranje opcije za programere se radi tako što se 7 puta uzastopno klikne na broj verzije uređaja. Time se otključavaju opcije koje su dostupne samo u programerskom režimu rada. Nakon toga se može omogućiti USB debagiranje gde se korisnicima dodeljuju posebne privilegije.

USB debagiranje omogućava korisnicima:

- Pristup unutrašnjem stanju aplikacija
- Instalira i uklanja aplikacije bez provera bezbednosti
- Pristup logovima sistema
- Kopiranje fajlova na uređaj
- Kopiranje fajlova sa uređaja
- Analiziranje performansi uređaja
- Izvršavanje kritičnih komandi

Jedan od problema ADB-a je taj što dobija veoma visok nivo privilegija i funkcioniše čak i kada je ekran zaključan. To znači da ako zlonamerni napadač dobije pristup ADB-u može instalirati malicione softvere sa svim privilegijama na povezani uređaj. Aplikacija sa svim privilegijama može da ima potpunu kontrolu nad sistemom. Pored toga, komande *push* i *pull* omogućavaju prenos datoteka između računara i uređaja. Pomoću tih komandi se mogu slati i preuzimati fajlovi iz bilo kog direktorijuma na uređaju, kako iz interne memorije tako i sa SD kartice. Ako ADB koristi osoba koja nije vlasnik uređaja može doći do curenja ličnih informacija sa uređaja.
Zbog svega ovoga ADB je česta meta hakerskih napada.

Primeri ADB komandi:

adb install <putanja_do_aplikacije> instalira aplikaciju sa eksternih uređaja
adb push <local> <remote> slanje fajla sa računara na uređaj
adb pull <remote> <local> preuzimanje fajla sa uređaja na računar
adb backup/restore <local> kreira backup fajlova sa uređaja

2. Sigurno USB debagiranje

Sa širenjem funkcionalnosti ADB alata, rasla je i potreba za dodatnim bezbednosnim poboljšanjima zbog sve češćih napada. Jedno od najznačajnijih poboljšanja je bilo uvođenje sigurnog USB debagiranja (*Secure USB Debugging*). To je mehanizam koji omogućava da samo autorizovani računari mogu uspostaviti ADB konekciju sa uređajem. Ako se uređaj pomoću USB-a poveže sa neautorizovanim računarom, taj računar ne može videti fajlove niti uspostaviti ADB vezu sve dok se ne ručno sa uređaja ne dozvoli pristup tom računaru. Kada se uređaj prvi put povezuje sa novim računarom, pojavljuje se dijalog koji traži potvrdu korisnika. Ako korisnik omogući pristup i čekira opciju "*Always allow from this computer*", ubuduće se neće tražiti potvrda za taj računar.

Naravno, da bi dodatna bezbednosna poboljšanja imala smisla neophodno je najpre da mobilni uređaj bude zaštićen jakom lozinkom, ukoliko uređaj nije zaključan lozinkom svako može pristupiti informacijama na njemu.

Za implementaciju sigurne autentifikacije procesa u USB debagiranju se koristi RSA autentifikacija sa 2048-bitnim ključem. Uređaj najpre šalje nasumičnu 20-bajtnu poruku računaru. Računar potpisuje tu poruku svojim privatnim ključem i vraća potpisanu poruku nazad do uređaja. Uređaj prima odgovor i proverava validnost potpisa koristeći odgovarajući javni ključ. Ako je potpis validan, uređaj potvrđuje da poruka potiče od autorizovanog računara koji poseduje odgovarajući privatni ključ i

dozvoljava USB debagiranje. U suprotnom se ne dozvoljava konekcija sa tim računarom. Bez ovog mehanizma bi svako mogao da ima pristup informacijama uređaja. Ovim mehanizmom se sprečava neovlašćen pristup podacima i funkcijama uređaja.

3. Ograničenja ADB alata

ADB alat je alat koji se najčešće koristi u okviru logičke forenzike uređaja. Njegovo osnovno ograničenje je to što ne pristupa fizičkom sloju memorije uređaja, već radi isključivo sa podacima koji su dostupni kroz operativni sistem i sistemske API-je.

Ograničenja ADB:

1. Nedostupnost pristupa nedodeljenom prostoru memorije

Kada korisnik obriše fajl, operativni sistem ne briše njegov sadržaj odmah, već samo uklanja referencu na taj fajl iz tabele fajl sistema (npr. MFT ili inode). Prostor koji je fajl zauzimao postaje nedodeljen (*unallocated*), ali podaci i dalje fizički postoje na disku dok ne budu prepisani novim sadržajem. S obzirom da ADB radi na logičkom nivou, on ne može da pristupi ovim podacima.

2. Obrisane fajlove je nemoguće povratiti

Obrisani fajlovi ne mogu da se povrate korišćenjem ADB-a i ne mogu se izvući informacije sa fizičkog sloja memorije, što predstavlja značajno ograničenje u forenzičkim analizama.

3. Ograničen pristup zaštićenim particijama

Bez root pristupa ADB ne može da izvuče informacije sa zaštićenih particija uređaja (kao što je folder */data/data* u kome se nalaze podaci aplikacija). Za potpunu analizu zaštićenih particija uređaja je neophodan root pristup ili korišćenje naprednijih forenzičkih alata.

4. Root pristup i njegove posledice u digitalnoj forenzici

Root pristup predstavlja najviši mogući nivo privilegija u Android operativnom sistemu. Korisnik sa root pristupom ima potpunu kontrolu nad sistemom uključujući sve datotekama, servise, procese i podešavanja uređaja. Ima pristup čak i kritičnim sistemskim fajlovima. U standardnom režimu korišćenja korisnik i aplikacije rade u kontrolisanom sandbox okruženju koje sprečava pristup kritičnim sekcijama uređaja. Root pristup uklanja ta ograničenja i omogućava pristup particijama */system* i */data/data*.

Korisnik ili proces sa root pristupom sistemu može da:

- Menja sistemske particije
- Pokreće dodatne servise koji menjaju logove uređaja
- Menja vreme pristupa fajlovima
- Prikrije tragove malvera i da mu još veće privilegije
-

Zbog ovoga se podaci sa rootovanog uređaja smatraju manje pouzdanim i kompromitovanim, jer je moguće da su menjani čime gube svoj integritet.

4.1. Zaključani i otključani bootloader

Bootloader je program koji inicijalno pokreće Android sistem. Proizvođači Android uređaja zaključavaju bootloader po defaultu da bi se sprečilo modifikovanje sistema i instaliranje neproverenih softvera.
Zaključani bootloader sprečava modifikaciju sistemskih particija čime se garantuje integritet sistema i podataka u njemu. Predstavlja bezbedonosnu meru protiv neovlašćenih promena u sistemu.
Otključani bootloader uklanja ova ograničenja dozvoljava sve što je zabranjeno u zaključanom bootloaderu. Omogućava instaliranje root alata i menja proces verifikovanog podizanja sistema(*Verified Boot*) čime se narušava integritet podataka u sistemu. Omogućava pristup folderima */system, /vendor, /persist* i */data/data.* U tim folderima se čuvaju konfiguracioni fajlovi, sesije i kolačići i zaštićeni sistemski logovi. U digitalno forenzici je cilj prikupiti nemodifikovane podatke i podatke koji su forenzički validni.

5. Primena ADB-a u digitalnoj forenzici

ADB pruža širok spektar mogućnosti za logičku analizu uređaja i prikupljanje fajlova koji se nalaze u korisničkom prostoru i na sistemskom nivou. U forenzičkom procesu je ADB posebno vredan kada uređaj nije root-ovan i kada je potrebno pristupiti podacima bez narušavanja integriteta sistema. ADB se često koristi kao prvi korak u inicijalnoj analizi uređaja, pre primene kompleksnijih metoda za dubinsku analizu. Koristi se pri analiziranju kompromitovanih uređaja. ADB omogućava dokumentovanje trenutnog stanja uređaja korišćenjem screenshot funkcija, smanjujući rizik od kompromitovanja dokaza.
Korišćenjem ADB-a na uređaju se može otkriti:

- Instaliran maliciozni softver
- Neautorizovan fizički pristup
- Curenje podataka
- Pokušaj rootvanja

Osnovna ADB komanda je komanda *adb devices k*oja prikazuje listu svih trenutno povezanih uređaja na računar.

```
C:\platform-tools>adb devices
List of devices attached
DUJNW20604001937        device
```

5.1. Prikupljanje podataka

Podaci koji su dostupni kroz standardne Android API-je se mogu prikupljati bez pristupa fizičkom sloju memorije, tkzv. **logička ekstrakcija podataka**.

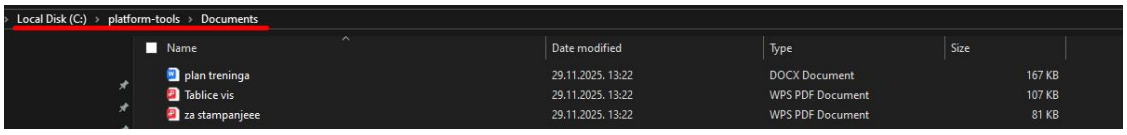
5.1.1. Pristup korisničkim folderima

adb pull /*PutanjaDoFoldera*/

Ovom komandom se pristupa sadržaju koji se nalazi na navedenoj putanji, to može biti fajl ili ceo folder.

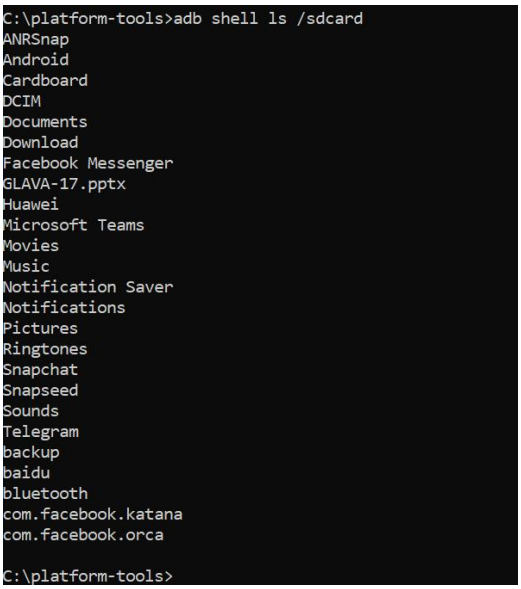
```
C:\platform-tools>adb pull /sdcard/Documents
/sdcard/Documents/: 3 files pulled, 0 skipped. 8.3 MB/s (361263 bytes in 0.041s)
```

Izvršavanjem ove komande se celokupan sadržaj foldera koji se nalazi na putanji /sdcard/Documents kopirao na računar, u folder iz koga je pozvana komanda pull.



5.1.2. Pregled sadržaja foldera

Komandom *adb shell ls /putanjaDoFoldera* se izlistava celokupni sadržaj tog foldera:



5.1.3. Pristup internim folderima aplikacije

Komanda adb shell run-as omogućava da se komanda izvrši kao da je pokrenuta iz konteksta specifične aplikacije.

adb shell run-as com.ime.aplikacije cp fajl_u_folderu /sdcard/

Ova komanda radi samo ako je aplikacija instalirana sa debuggable manifest zastavicom (što je retko kod komercijalnih aplikacija, ali uobičajeno u fazi razvoja ili testiranja). Ako je aplikacija debuggable, analitičar može koristiti ovu tehniku da prekopira fajlove iz internog zaštićenog foldera (/data/data/...) na dostupnu lokaciju (/sdcard/), odakle se zatim mogu preuzeti komandom adb pull. Ovo je način da se obavi logička ekstrakcija iz zaštićenih particija bez root pristupa.

5.1.4. Kreiranje backup-a uređaja

adb backup [BackupOpcija] -f <putanja_do_fajla.ab> [paket_ili_aplikacije]

Ova funkcija kreira backup aplikacija i podataka sa telefona u jedan fajl koji se čuva na računaru. Fajl je formata .ab i može sadržati podatke aplikacija ali ne i fizičke fajlove kao što su fotografije ili video zapisi.

[*BackupOpcija*] može biti neka od opcija:
-apk - uključuje i APK fajlove instaliranih aplikacija
-noapk - ne uključuje APK fajlove, nego samo podatke aplikacije
-shared - uključuje deljenu memoriju (SD karticu / interne foldere)
-all - backup svih aplikacija i podataka

Primer za kreiranje backupa:

```
C:\platform-tools>adb backup -noapk -f maps_backup.ab com.google.android.apps.maps
WARNING: adb backup is deprecated and may be removed in a future release
Now unlock your device and confirm the backup operation...
```

Ova komanda će u fajl *maps_backup.ab* sačuvati podešavanja aplikacije Google Maps, sačuvane lokacije i istoriju pretrage lokacija.

5.1.5. Pristup logovima

adb logcat čita sve sistemske i aplikacione logove koje Android generiše u realnom vremenu. Logovi mogu sadržati informacije o greškama aplikacija, sistemske događaje, obaveštenja, aktivnosti mreže...

```
C:\platform-tools>adb logcat
--------- beginning of system
07-19 14:16:32.244   837   852 I storaged: storaged_t::event: storage_info refreshed.
07-19 14:16:34.080  1402  1560 I chatty  : uid=1000(system) monitor thread expire 10 lines
07-19 14:16:43.941  1402  1643 I chatty  : uid=1000(system) UEventObserver expire 14 lines
07-19 14:17:32.246   837   852 I storaged: storaged_t::event: storage_info refreshed.
07-19 14:17:34.094  1402  1560 I chatty  : uid=1000(system) monitor thread expire 10 lines
07-19 14:17:45.379  1402  1643 I chatty  : uid=1000(system) UEventObserver expire 14 lines
07-19 14:17:55.194 15278 15278 D ActivityThread: Attach thread to application
07-19 14:17:55.406 15278 15299 W DE J ImageProcessorAlgoImpl: [effect] initAlgoXmlPath() error! can't find xml
07-19 14:18:16.103  1402  1643 I chatty  : uid=1000(system) UEventObserver expire 14 lines
07-19 14:18:32.249   837   852 I storaged: storaged_t::event: storage_info refreshed.
07-19 14:18:34.110  1402  1560 I chatty  : uid=1000(system) monitor thread expire 10 lines
07-19 14:19:17.545  1402  1643 I chatty  : uid=1000(system) UEventObserver expire 14 lines
07-19 14:19:32.251   837   852 I storaged: storaged_t::event: storage_info refreshed.
07-19 14:19:34.126  1402  1560 I chatty  : uid=1000(system) monitor thread expire 10 lines
07-19 14:20:13.359  1402  1758 I chatty  : uid=1000(system) Thread-7 expire 11 lines
07-19 14:20:13.359  1402  1565 I chatty  : uid=1000(system) HwActivityTaskM expire 11 lines
07-19 14:20:13.374  1402  6773 I chatty  : uid=1000(system) Binder:1402_15 expire 17 lines
07-19 14:20:15.482  2141 14752 W ContextImpl: Calling a method in the system process without a qualified user: android.app.C
extWrapper.bindService:705 com.huawei.nb.client.RemoteServiceConnection.open:132 com.huawei.nb.client.Proxy.connect:172 com.
aServiceProxyManager.connectDataService:86
07-19 14:20:18.980  1402  1643 I chatty  : uid=1000(system) UEventObserver expire 21 lines
07-19 14:20:32.132  1402  1668 I chatty  : uid=1000(system) AlarmManager expire 21 lines
07-19 14:20:32.253   837   852 I storaged: storaged_t::event: storage_info refreshed.
07-19 14:20:34.138  1402  1560 I chatty  : uid=1000(system) monitor thread expire 10 lines
07-19 14:20:42.198  2141  4525 W ContextImpl: Calling a method in the system process without a qualified user: android.app.C
extWrapper.bindService:705 com.huawei.nb.searchmanager.service.SearchManager.connectRemoteSearchService:192 com.huawei.nb.se
k:759 com.huawei.nb.searchmanager.service.SearchManager.intentIdle:665
07-19 14:20:42.232 15081 15100 W ContextImpl: Calling a method in the system process without a qualified user: android.app.C
extWrapper.bindService:705 com.huawei.nb.client.RemoteServiceConnection.open:132 com.huawei.nb.client.Proxy.connect:172 com.
aServiceProxyManager.connectDataService:86
07-19 14:21:32.255   837   852 I storaged: storaged_t::event: storage_info refreshed.
07-19 14:21:34.155  1402  1560 I chatty  : uid=1000(system) monitor thread expire 10 lines
07-19 14:21:51.142  1402  1643 I chatty  : uid=1000(system) UEventObserver expire 14 lines
07-19 14:22:32.258   837   852 I storaged: storaged_t::event: storage_info refreshed.
07-19 14:22:34.173  1402  1560 I chatty  : uid=1000(system) monitor thread expire 10 lines
```

5.1.6. Kreiranje screenshotova

adb shell screencap /putanjaDoFajla

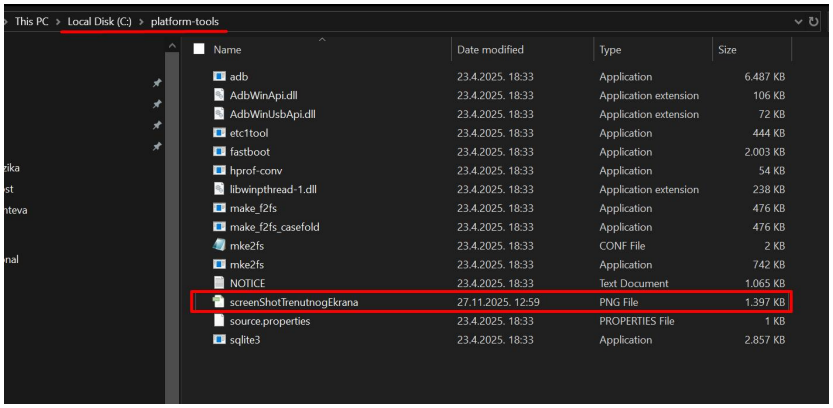
```
C:\platform-tools>adb shell screencap -p /sdcard/screenShotTrenutnogEkrana.png
```

Ovom komandom se snimak kreira i čuva na putanji koja je navedena u komandi.

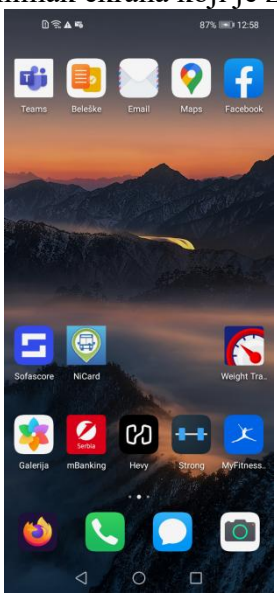
adb pull */putanjaDoFajla*

```
C:\platform-tools>adb pull /sdcard/screenShotTrenutnogEkrana.png
/sdcard/screenShotTrenutnogEkrana.png: 1 file pulled, 0 skipped. 32.4 MB/s (1429156 bytes in 0.042s)
```

Nakon izvršavanja pull komande, snimak ekrana uređaja koji je snimljen prethodnom komandom će biti sačuvan u folderu platform-tools gde je pokrenuta komanda.



Ako otvorimo tu sliku, vidimo snimak ekrana koji je zabeležen:



5.2. Pregled i analiza aplikacija

ADB omogućava uvid u internu strukturu fajlova Android sistema. Iako bez root pristupa nije moguće čitati zaštićene direktorijume, ipak se mogu dobiti informacije koje su korisne za forenzičku analizu.

5.2.1. Pregled instaliranih aplikacija

adb shell pm list packages -f

Rezultat ove komande je lista svih instaliranih aplikacija uključujući i skrivene i sistemske aplikacije. Na ovaj način se može pronaći maliciozni softver za praćenje lokacije, snimanje poziva, pristup porukama...

```
C:\platform-tools>adb shell pm list packages -f
package:/data/app/com.sofascore.results-EWUt4-MOpSFOKrR_RkE8nw==/base.apk=com.sofascore.results
package:/system/app/HiFolder/HiFolder.apk=com.huawei.hifolder
package:/data/app/io.strongapp.strong-MOlYhDOY40LUrwYPFI3EUw==/base.apk=io.strongapp.strong
package:/system/priv-app/CtsShimPrivPrebuilt/CtsShimPrivPrebuilt.apk=com.android.cts.priv.ctsshim
package:/hw_product/app/HwCamera2/base-master.apk=com.huawei.camera
package:/system/emui/oversea/overlay/HwPermissionControllerOverlay.apk=com.huawei.permissioncontroller.overlay
package:/data/app/com.huawei.android.tips-prefrX738tU7k6GGRWtDEQ==/base.apk=com.huawei.android.tips
package:/system/priv-app/HwCameraKit/HwCameraKit.apk=com.huawei.camerakit.impl
package:/data/app/monitoryourweight.bustan.net-fqlAuzaKoolNn1n8TLI1bw==/base.apk=monitoryourweight.bustan.net
package:/system/app/HwSynergy/HwSynergy.apk=com.huawei.synergy
package:/system/app/HwLauncher6/HwLauncher6.apk=com.huawei.android.launcher
package:/data/app/com.android.mediacenter-1ZLr3RpIoFuT1cGCwxZxtw==/base.apk=com.android.mediacenter
package:/system/priv-app/TelephonyProvider/TelephonyProvider.apk=com.android.providers.telephony
package:/system/app/HwJetPack/HwJetPack.apk=com.huawei.androidx
package:/system/priv-app/CalendarProvider/CalendarProvider.apk=com.android.providers.calendar
package:/data/app/com.alibaba.aliexpresshd-1zZ1s0bpSR-8D80pfx1nzA==/base.apk=com.alibaba.aliexpresshd
package:/system/app/FeatureFramework/FeatureFramework.apk=com.huawei.featurelayer.featureframework
package:/data/app/com.huawei.health-8m8nfjJROScbzcQ-YOK3MQ==/base.apk=com.huawei.health
package:/hw_product/app/HiCard/HiCard.apk=com.huawei.hicard
package:/data/app/com.huawei.hidisk-BfpEbygd5ZfVpfPn-vz5bw==/base.apk=com.huawei.hidisk
package:/system/app/HiView/HiView.apk=com.huawei.hiview
package:/system/app/HwIAware/HwIAware.apk=com.huawei.iaware
package:/system/app/HwBluetoothImport/HwBluetoothImport.apk=com.huawei.bluetooth
package:/system/priv-app/MediaProvider/MediaProvider.apk=com.android.providers.media
package:/data/app/com.touchtype.swiftkey-jQIX1hs9Mwa2EyMwDl1Zow==/base.apk=com.touchtype.swiftkey
```

5.2.2. Analiza .apk fajlova

Na primer, hoćemo da analiziramo .apk fajl aplikacije Google maps.
Najpre komandom

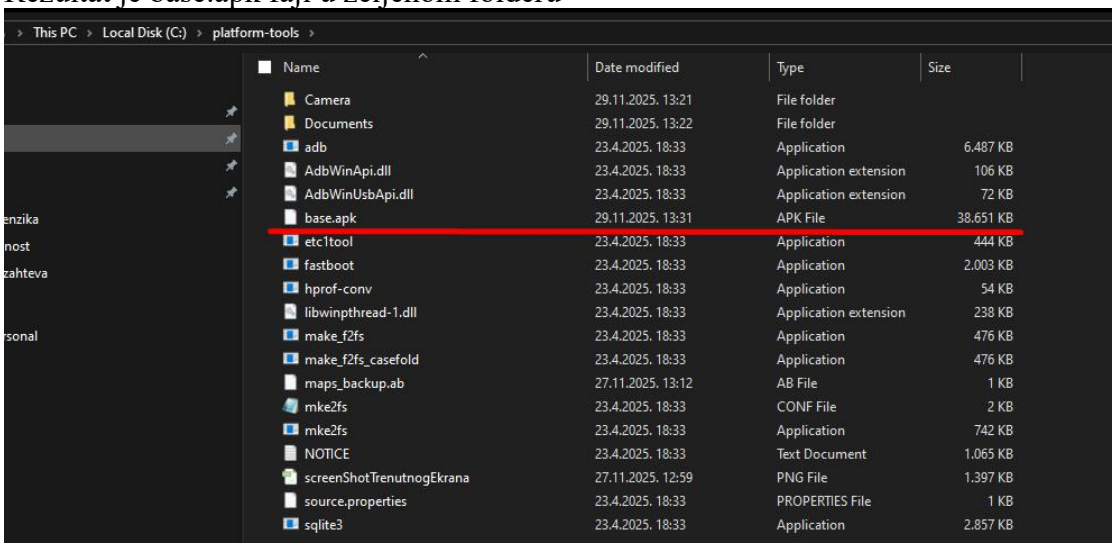
```
C:\platform-tools>adb shell pm path com.google.android.apps.maps
package:/data/app/com.google.android.apps.maps-2lRyDWjnzkRFR2xFTHsFVA==/base.apk
```

Pronalazimo putanju u sistemu do željene aplikacije, u ovom slučaju Google maps.

Nakon toga se izvršava komanda pull nad dobijenom putanjom:

```
C:\platform-tools>adb pull /data/app/com.google.android.apps.maps-2lRyDWjnzkRFR2xFTHsFVA==/base.apk
/data/app/com.google.android.apps.maps-2lRyDWjnzkRFR2xFTHs...le pulled, 0 skipped. 36.5 MB/s (39577840 bytes in 1.033s)
```

Rezultat je base.apk fajl u željenom folderu



Ovom komandom se omogućava preuzimanje instaliranih APK paketa koji se mogu dalje analizirati alatima kao što su JADX ili APKTool.

5.3. Prikupljanje sistemskih informacija i metapodataka

5.3.1. Informacije o hardveru i softveru

1. adb shell getprop

Prikazuje sistemska svojstva Android uređaja. To su interne promenljive koje Android koristi da opiše tehničke parametre kao što su verzija sistema, model uređaja, sigurnosna podešavanja, mrežne informacije...
Omogućava analitičaru uvid u kritične informacije bez root pristupa.

Funkcija getprop čita sve podatke iz Android sistemskog servisa:

```
C:\platform-tools>adb shell getprop
[aaudio.mmap_exclusive_policy]: [1]
[aaudio.mmap_policy]: [2]
[bastet.service.enable]: [true]
[bg_fsck.pgid]: [420]
[bt.dpbap.enable]: [1]
[bt.max.hfpclient.connections]: [2]
[camera.ignore.chat.facebeauty]: [1]
[config.disable_consumerir]: [true]
[config.hw.power.saving.always.netconnect]: [false]
[config.hw.power.saving.autoquit]: [false]
[dalvik.vm.appimageformat]: [lz4]
[dalvik.vm.boot-dex2oat-threads]: [4]
[dalvik.vm.checkjni]: [false]
[dalvik.vm.dex2oat-Xms]: [64m]
[dalvik.vm.dex2oat-Xmx]: [512m]
[dalvik.vm.dex2oat-max-image-block-size]: [1048576]
[dalvik.vm.dex2oat-minidebuginfo]: [true]
[dalvik.vm.dex2oat-resolve-startup-strings]: [true]
[dalvik.vm.dex2oat-threads]: [4]
[dalvik.vm.dexopt.secondary]: [true]
[dalvik.vm.heapgrowthlimit]: [384m]
[dalvik.vm.heapmaxfree]: [8m]
[dalvik.vm.heapminfree]: [2m]
[dalvik.vm.heapsize]: [512m]
[dalvik.vm.heapstartsize]: [8m]
[dalvik.vm.heaptargetutilization]: [0.75]
[dalvik.vm.image-dex2oat-Xms]: [64m]
[dalvik.vm.image-dex2oat-Xmx]: [64m]
[dalvik.vm.image-dex2oat-threads]: [4]
[dalvik.vm.isa.arm.features]: [default]
[dalvik.vm.isa.arm.variant]: [cortex-a15]
[dalvik.vm.isa.arm64.features]: [default]
```

Moguće je čitati i pojedinačne podatke korišćenjem specifičnih getprop funkcija:

Android verzija:

```
C:\platform-tools>adb shell getprop ro.build.version.release
10
```

Sigurnosna patch verzija:

```
C:\platform-tools>adb shell getprop ro.build.version.security_patch
2020-10-01
```

Model uređaja:

```
C:\platform-tools>adb shell getprop ro.product.model
ART-L29
```

Da li je uređaj spreman za debagiranje:

```
C:\platform-tools>adb shell getprop ro.debuggable
0
```

Indikatori da li je uređaj rootovan:

```
C:\platform-tools>adb shell getprop ro.secure
1

C:\platform-tools>adb shell getprop ro.boot.verifiedbootstate
green
```

Ako je *ro.debuggable=1* i *ro.secure=0*, to znači da je uređaj najverovatnije root-ovan i kompromitovan. U ovom primeru vidimo da to nije slučaj i da je uređaj siguran. Dodatno vidimo da je *verifiedbootstate = gree*n što znači da je bootloader zaključan, da su sve particije u originalnom stanju, nijedan deo sistema nije modifikovan, uređaj nije rootovan i da je uređaj potpuno bezbedan.

2. adb shell dumpsys

Adb shell dumpsys je sistemska dijagnostička komanda koja prikazuje stanje svih ključnih Android servisa koji trenutno rade u operativnom sistemu uređaja.
Omogućava uvid u trenutno stanje uređaja bez direktnog pristupa fajl sistemu.
Ovo je jedan od najmoćnijih ADB alata jer omogućava uvid u aktivne aplikacije, sistemske servise, stanje senzora, notifikacije, procese u sistemu, Wi-Fi, memoriju...

Osnovna komanda *adb shell dumpsys* izlistava sve servise na uređaju i rezultat može biti ogroman. Zbog toga se najčešće cilja određeni servis koji je potrebno ispitati.

Activity service:
Koristi se za prikaz trenutno aktivnih aplikacija i identifikaciju onoga šta je korisnik poslednje radio na uređaju.
Komanda za activity service je a*db shell dumpsys activity*

```
C:\platform-tools>adb shell dumpsys activity
ACTIVITY MANAGER SETTINGS (dumpsys activity settings) activity_manager_constants:
  max_cached_processes=32
  background_settle_time=60000
  fgservice_min_shown_time=2000
  fgservice_min_report_time=3000
  fgservice_screen_on_before_time=1000
  fgservice_screen_on_after_time=5000
  content_provider_retain_time=20000
  gc_timeout=5000
  gc_min_interval=60000
  full_pss_min_interval=1200000
  full_pss_lowered_interval=300000
  power_check_interval=300000
  power_check_max_cpu_1=25
  power_check_max_cpu_2=25
  power_check_max_cpu_3=10
  power_check_max_cpu_4=2
  service_usage_interaction_time=1800000
  usage_stats_interaction_interval=7200000
  service_restart_duration=1000
  service_reset_run_duration=60000
  service_restart_duration_factor=4
  service_min_restart_time_between=10000
  service_max_inactivity=1800000
  service_bg_start_timeout=15000
  service_bg_activity_start_timeout=10000
```

Korišćenjem filtriranja, forenzičar može brzo da utvrdi koja je aplikacija bila u fokusu kada je uređaj zaplenjen. Ovaj izlaz direktno ukazuje na aktivnost aplikacije koja je trenutno na ekranu, što je ključno za rekonstrukciju ponašanja korisnika:

adb shell dumpsys activity | grep 'mFocusedStack'

Package service:
Koristi se za prikaz svih instaliranih aplikacija, njihovih dozvola i podešavanja, kao i vreme instalacije i ažuriranja aplikacije.
Komanda za package service je a*db shell dumpsys package*

```
Legacy install sessions:
  {}

Active APEX packages:

  com.android.apex.cts.shim
    Version: 1
    Path: /system/apex/com.android.apex.cts.shim.apex
    IsActive: true
    IsFactory: true
  com.android.conscrypt
    Version: 291601510
    Path: /system/apex/com.android.conscrypt.apex
    IsActive: true
    IsFactory: true
  com.android.media
    Version: 291601612
    Path: /system/apex/com.android.media.apex
    IsActive: true
    IsFactory: true
  com.android.media.swcodec
    Version: 291601611
    Path: /system/apex/com.android.media.swcodec.apex
    IsActive: true
    IsFactory: true
```

Battery service:
Pokazuje trenutno stanje baterije, kada i koliko je uređaj bio punjen. Može da se koristi za rekonstruisanje vremena aktivnosti uređaja.

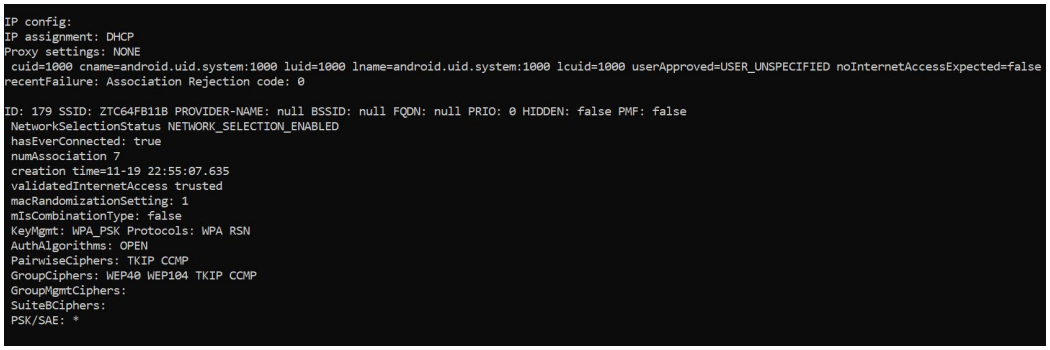
Komanda *adb shell dumpsys battery* se koristi za prikaz trenutnog stanja baterija:

```
C:\platform-tools>adb shell dumpsys battery
Current Battery Service state:
  AC powered: false
  USB powered: true
  Wireless powered: false
  Max charging current: 0
  Max charging voltage: 0
  Charge counter: 391000
  status: 2
  health: 2
  present: true
  level: 83
  scale: 100
  voltage: 4262
  temperature: 250
  technology: Li-poly
```

Komanda *adb shell dumpsys batterystats* se koristi za prikaz statistike baterije

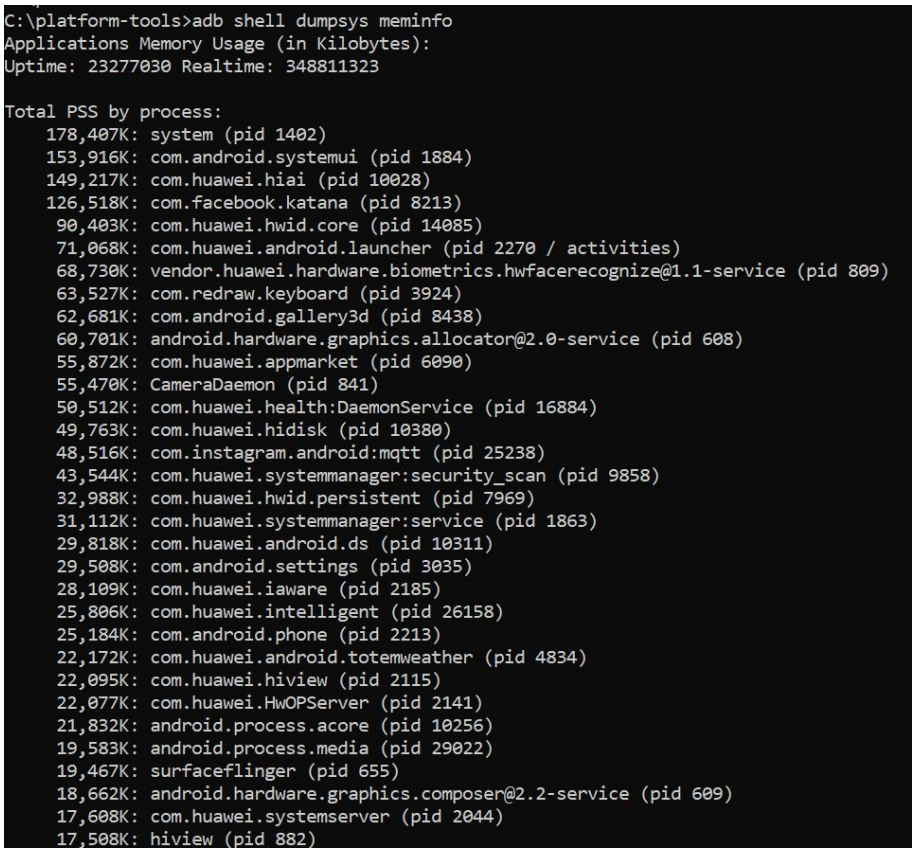
```
  Daily from 2022-07-20-04-33-15 to 2022-07-21-08-36-27:
    Discharge step durations:
      #0: +4h47m58s2ms to 82 (screen-off, power-save-off, device-idle-on)
      #1: +5h19m24s0ms to 83 (screen-off, power-save-off, device-idle-on)
      #2: +3h24m58s978ms to 84 (power-save-off)
      #3: +3h20m0s991ms to 85 (power-save-off, device-idle-on)
      #4: +4h28m47s9ms to 86 (power-save-off, device-idle-on)
        Discharge total time: 17d 19h 2m 59s 600ms  (from 5 steps)
        Discharge screen off time: 21d 2h 8m 20s 100ms  (from 2 steps)
        Discharge screen off device idle time: 21d 2h 8m 20s 100ms  (from 2 steps)
  Daily from 2022-07-19-12-22-21 to 2022-07-20-04-33-15:
    Discharge step durations:
      #0: +3h23m19s971ms to 88 (power-save-off, device-idle-on)
        Discharge total time: 14d 2h 53m 17s 100ms  (from 1 steps)
    Charge step durations:
      #0: +4m56s959ms to 90 (screen-off, power-save-off, device-idle-off)
      #1: +4m56s958ms to 89 (screen-off, power-save-off, device-idle-off)
      #2: +4m56s958ms to 88 (screen-off, power-save-off, device-idle-off)
      #3: +4m56s959ms to 87 (screen-off, power-save-off, device-idle-off)
      #4: +6m49s611ms to 86 (power-save-off, device-idle-off)
      #5: +5m58s398ms to 85 (screen-on, power-save-off, device-idle-off)
      #6: +2m23s361ms to 84 (screen-on, power-save-off, device-idle-off)
      #7: +20s480ms to 83 (screen-on, power-save-off, device-idle-off)
      #8: +10s238ms to 82 (screen-on, power-save-off, device-idle-off)
      #9: +20s480ms to 81 (screen-on, power-save-off, device-idle-off)
      #10: +20s482ms to 80 (screen-on, power-save-off, device-idle-off)
      #11: +10s238ms to 79 (screen-on, power-save-off, device-idle-off)
      #12: +30s722ms to 78 (screen-on, power-save-off, device-idle-off)
      #13: +30s717ms to 77 (screen-on, power-save-off, device-idle-off)
      #14: +20s481ms to 76 (screen-on, power-save-off, device-idle-off)
      #15: +20s480ms to 75 (screen-on, power-save-off, device-idle-off)
      #16: +20s474ms to 74 (screen-on, power-save-off, device-idle-off)
      #17: +30s727ms to 73 (screen-on, power-save-off, device-idle-off)
        Charge total time: 3h 36m 10s 600ms  (from 18 steps)
        Charge screen off time: 8h 14m 55s 800ms  (from 4 steps)
        Charge screen on time: 1h 34m 31s 300ms  (from 13 steps)
  Daily from 2024-07-25-04-26-20 to 1970-01-01-03-31-33:
    Discharge step durations:
      #0: +1h58m20s997ms to 2 (screen-off, power-save-off, device-idle-on)
      #1: +2h4m58s979ms to 3 (screen-off, power-save-off, device-idle-on)
```

Wi-Fi service:
Prikazuje informacije o trenutnoj Wi-Fi konekciji i istoriji konektovanja na Wi-Fi.



Process service:

Komanda *adb shell dumpsys meminfo* prikazuje informacije o memoriji koju trenutno koriste aplikacije.



Komanda *adb shell dumpsys cpuinfo* prikazuje informacije o tome koje aplikacije i koliko troše CPU. Ovo su dinamičke informacije koje se stalno menjaju.

```
C:\platform-tools>adb shell dumpsys cpuinfo
Load: 43.11 / 43.21 / 43.34
CPU usage from 9338ms to 1346ms ago (2025-11-29 14:20:27.266 to 2025-11-29 14:20:35.258):
  1.6% 1402/system_server: 0.8% user + 0.7% kernel / faults: 80 minor
  1.2% 8213/com.facebook.katana: 0.9% user + 0.2% kernel / faults: 39 minor
  0.4% 2185/com.huawei.iaware: 0.2% user + 0.2% kernel / faults: 52 minor
  0.1% 2044/com.huawei.systemserver: 0% user + 0% kernel
  0.2% 9460/kworker/u16:4: 0% user + 0.2% kernel
  0% 1/init: 0% user + 0% kernel / faults: 36 minor
  0% 8/ksoftirqd/0: 0% user + 0% kernel
  0.1% 9/rcu_preempt: 0% user + 0.1% kernel
  0% 18/ksoftirqd/1: 0% user + 0% kernel
  0% 78/mailbox-16: 0% user + 0% kernel
  0.1% 442/logd: 0.1% user + 0% kernel
  0% 589/netd: 0% user + 0% kernel / faults: 23 minor
  0% 657/powerlogd: 0% user + 0% kernel
  0% 758/irqbalance: 0% user + 0% kernel
  0% 837/storaged: 0% user + 0% kernel / faults: 1 minor
  0.1% 896/chargemonitor: 0% user + 0.1% kernel
  0.1% 1041/wlan_bus_rx/sdi: 0% user + 0.1% kernel
  0.1% 1079/hisi_hcc: 0% user + 0.1% kernel
  0% 1253/oam_hisi: 0% user + 0% kernel
  0% 6489/chargelogcat-c: 0% user + 0% kernel
  0% 10256/android.process.acore: 0% user + 0% kernel
  0.1% 11268/kworker/u16:0: 0% user + 0.1% kernel
  0% 12474/kworker/2:2: 0% user + 0% kernel
  0% 12915/kworker/5:0: 0% user + 0% kernel
 +0% 13117/kworker/3:0: 0% user + 0% kernel
0.9% TOTAL: 0.3% user + 0.3% kernel + 0% iowait + 0.1% irq + 0.1% softirq
```

3. adb shell cat /proc/cpuinfo

Ova komanda prikazuje informacije o procesoru kao što su model procesora, broj jezgara tog procesora, arhitekturu procesora i serijski broj procesa. Ne govori ništa o aktivnim procesima i opterećenju.

```
C:\platform-tools>adb shell cat /proc/cpuinfo
Processor       : AArch64 Processor rev 2 (aarch64)
processor       : 0
BogoMIPS        : 3.84
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 1
BogoMIPS        : 3.84
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 2
BogoMIPS        : 3.84
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4

processor       : 3
BogoMIPS        : 3.84
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision    : 4
```

4. adb shell cat /proc/meminfo

Ova komanda prikazuje detaljne informacije o RAM memoriji Android uređaja. Prikazuje informacije o ukupnoj i slobodnoj RAM memoriji, memoriji koju sistem koristi za keširanje (buffers, cached), memoriji koju trenutno koriste procesi (active, inactive), memoriju koju koristi kernel (slab)...

```
C:\platform-tools>adb shell cat /proc/meminfo
MemTotal:        3768328 kB
MemFree:           62948 kB
MemAvailable:    1844688 kB
Buffers:            4216 kB
Cached:          1714296 kB
SwapCached:       161000 kB
Active:          1293620 kB
Inactive:        1514356 kB
Active(anon):     679544 kB
Inactive(anon):   418664 kB
Active(file):     614076 kB
Inactive(file):  1095692 kB
Unevictable:         592 kB
Mlocked:             592 kB
SwapTotal:       2293756 kB
SwapFree:        1421792 kB
Dirty:               124 kB
Writeback:             0 kB
AnonPages:       1073264 kB
Mapped:           545848 kB
Shmem:              8676 kB
Slab:             313968 kB
SReclaimable:     146608 kB
SUnreclaim:       167360 kB
KernelStack:       54560 kB
PageTables:        74028 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     4177920 kB
Committed_AS:   84418288 kB
VmallocTotal:   263061440 kB
VmallocUsed:           0 kB
VmallocChunk:          0 kB
CmaTotal:         458752 kB
CmaFree:               0 kB
IonTotalCache:       640 kB
IonTotalUsed:      60980 kB
RsvTotalUsed:     276484 kB
```

5.3.2. Analiza mrežne aktivnosti

adb shell netstat

Ova komanda prikazuje mrežne aktivnosti uređaja. Otkriva otvorene portove, aktivne mrežne konekcije i komunikaciju sa drugim serverima. Tu spadaju aktivne TCP i UDP konekcije, portovi koji čekaju dolazne konekcije, IP adrese povezanih uređaja i stanja konekcija. Otkriva se koje aplikacije imaju aktivne konekcije, da li neka aplikacija šalje podatke u pozadini(bez znanja korisnika) i da li komunicira sa nekim serverima sa kojima ne bi trebalo da komunicira.

```
C:\platform-tools>adb shell netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.0.24:45556      edge-star-shv-02-:https ESTABLISHED
tcp        0      0 192.168.0.24:45562      edge-star-shv-02-:https ESTABLISHED
tcp6       0     32 ::ffff:192.168.0.:50312 ecs-80-158-38-48.:https LAST_ACK
tcp6       0      0 ::ffff:192.168.0.:49894 edge-mqtt-mini-sh:https ESTABLISHED
tcp6       0      0 ::ffff:192.168.0.:50008 ecs-160-44-193-95.:5223 ESTABLISHED
udp     4608      0 192.168.0.24:bootpc     192.168.0.1:bootps      ESTABLISHED
udp        0      0 192.168.0.24:46118      anycast-dns.sbb.:domain ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State           I-Node Path
unix  162    [ ]         DGRAM                       15677 /dev/socket/logdw
unix  2      [ ]         DGRAM                      270873 /dev/socket/wpa_wlan0
unix  2      [ ]         DGRAM                      272583 /data/vendor/wifi/wpa/sockets/wlan0
unix  9      [ ]         DGRAM                       21930 /dev/socket/statsdw
unix  53     [ ]         DGRAM                       16560 /dev/socket/powerlogdw
unix  2      [ ]         DGRAM                      267154 /data/vendor/wifi/wpa/sockets/p2p0
unix  3      [ ]         STREAM     CONNECTED        29052
unix  3      [ ]         SEQPACKET  CONNECTED       710434
unix  3      [ ]         STREAM     CONNECTED        24533
unix  3      [ ]         SEQPACKET  CONNECTED        35248
unix  3      [ ]         SEQPACKET  CONNECTED        31340
unix  2      [ ]         DGRAM                       20591
unix  2      [ ]         DGRAM                       57408
unix  2      [ ]         DGRAM                       25322
unix  2      [ ]         DGRAM                         284
unix  2      [ ]         DGRAM                       18499
unix  2      [ ]         DGRAM                      719898
unix  2      [ ]         DGRAM                      716346
unix  2      [ ]         DGRAM                      338086
unix  3      [ ]         SEQPACKET  CONNECTED        46356
unix  2      [ ]         DGRAM                       32094
unix  3      [ ]         SEQPACKET  CONNECTED        29186
unix  2      [ ]         DGRAM                      656549
unix  3      [ ]         SEQPACKET  CONNECTED       464427
unix  3      [ ]         SEQPACKET  CONNECTED        29990
unix  2      [ ]         DGRAM                       16553
unix  2      [ ]         DGRAM                      319406
```

Ako u rezulatu ove komande postoji *tcp 0 0 0.0.0.0:5555 0.0.0.0:* LISTEN*
To znači da bilo ko u mreži može da se poveže na taj uređaj i preuzme kontrolu što je izuzetno opasno i čini uređaj podložnim za napade.

6. Zaključak

Android Debug Bridge(ADB) je jedan od najvažnijih alata koji se koriste u digitalnoj forenzici Android uređaja. Njegova najveća prednost je ta što omogućava programeru uvid u trenutno stanje operativnog sistema uređaja, pristup korisničkim fajlovima, pregled instaliranih aplikacija i analizu sistemskih servisa, bez potrebe za root pristupom. Time se značajno smanjuje rizik od narušavanja integriteta podataka što je od ključne važnosti u forenzičkim istragama. Korišćenjem ADB se mogu dobiti informacije o hardveru, softveru, sistemskim servisima, aktivnim konekcijama i mrežnim procesima uređaja koji se analizira. Zbog visokih nivoa dozvola koje ima ADB neophodno je uvesti mere bezbednosti poput Secure USB Debugging-a i RSA autorizacije i ograničenja pristupa uređaju dok korisnik ručno ne dozvoli pristup. Ovim merama se uređaj štiti od neovlašćenog pristupa. Iako pristup omogućava potpuni pristup svim sistemskim particijama i servisima on može narušiti integritet dokaza jer rootvan uređaj dozvoljava menjanje sistemskih logova, vremena pristupa fajlovima i prikrivanje aktivnosti malicioznog softvera. Zbog toga se root pristup izbegava i koristi se samo ako je apsolutno neophodan.
ADB je prvi, najbezbedniji i najmanje invanzivan korak u analiziranju stanja uređaja jer ne zahteva root pristup nad uređajem. Na taj način omogućava logičku ekstrakciju podataka i njihovu analizu bez menjanja sadržaja uređaja i integriteta podataka. ADB omogućava forenzičarima da analiziraju uređaje, pronaladju maliciozne softvere ako postoje i dokumentuju stanje uređaja bez rizika od narušavanja integriteta dokaza. Zbog toga je pravilna upotreba ADB alata uz poštovanje forenzičkih principa od ključnog značaja za validnost digitalnih dokaza.