RECOVERING DELETED DATA FROM ANDROID DEVICE USING ADB COMMAND

[1]Malakar Khushboo Shivkinkar, [2]Dr. Priyanka Sharma
[1]Student, Raksha Shakti University, Ahmedabad,
[2]Professor, Department of IT & Telecommunication, Raksha Shakti University, Ahmedabad
[1]Cyber Security,
[1]Raksha Shakti University, Ahmedabad, India

Abstract: The smart phone market is growing very rapidly. Now a days technologies are updating everyday which are becoming direct targets of criminals and hackers. Mobile contains a lot of important, sensible and sensitive data which is of an investigator's attention. There are following internal and external data available on Android devices like SMS, MMS, emails, call logs, contacts, gallery photos, calendars, notes, browser history, GPS locations, passwords, and data stored on SD cards, and internal memory data etc. Thus, it is necessary that with the help of open and commercial tools Investigators are aware of all the techniques and method used for extracting data from Android Device. ADB (USB Debugging) method which utilizes a built-in protocol within the operating system.

This paper proposed the recovering of deleted data extraction of android based smartphones iball Andi4.5M enigma version 4.4.2 specified with adb command tools, and paper highlights various commands available in terms of physical acquisition.

***Index Terms*: Mobile Phone Forensics, adb command, dd, Android forensic, Cyber Forensic.**

I. INTRODUCTION

Many of the portable devices like PDAs, Smartphone, phones, tables and many other electronic handheld devices are running using software program called operating system to manage the android or windows applications and hardware. Many mobile operating systems are available in the cyber world such as Hewlett-Packard's webOS, Google Android, Apple iOS, BlackBerry operating system, Nokia's Symbian and Microsoft's Windows Phone operating system. The mobile operating system is the software platform on top of which other programs, called application programs, can run on mobile devices. As usability of mobile devices increasing day to day life and become more widespread so it is important for forensic investigator to handle the device found in crime scenes.

Now a days it is easy to handle it with the help of different Forensic tool kit like Faraday Bag's, write blocker's, UFED 4PC etc. The primary aim of mobile or android forensic is to retrieval, gathering or recovery of sensitive data present in device and to take proper evidence from mobile devices as it is a part of digital forensics.

The main purpose of a forensic tool is to obtain data present in Mobile Device without modifying the data and to maintain the integrity of the data. The tool should provide critical updates in time to keep pace of the rapid changes of Mobile Device hardware and software. Basically there are two types of forensic tool it can be either forensic or non-forensic, which each of them providing different challenges and tasks as well as permitting for different solutions.

Forensic tools are tools that are designed primarily for uncovering data from different versions of Mobile Devices, while non-forensic tools are not designed for uncovering data but can be manipulated for that purpose. Before getting into actual forensics it is important to understand Android operating system, Android architecture, file systems, directory structures, and how and where the data is stored on the devices.

Forensic investigation matters while collecting, preserving, Acquainting, Documenting digital evidence to present it in court. Preserving mobile device without altering data to maintain the integrity it is nearly impossible because mobile device constantly transmits data using the network, Wi-Fi, or Bluetooth and other mobile apps like Xender. That's it is necessary to document each and every small detail about steps taken starting from seizure, preservation, acquisition with the presentation in court.[5]

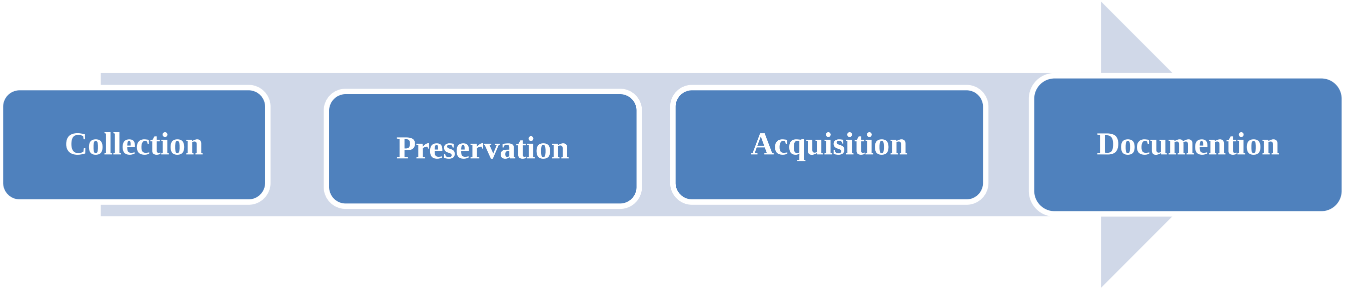


Figure 1: MOBILE FORENSIC PROCESS

II.PHYSICAL ACQUISITION

Physical extraction implies a bit-by-bit copy of the entire flash memory of a mobile device. This extraction method not only enables the acquisition of intact data, but also data that is hidden or has been deleted.

Deleted data can be recovered from many levels:

The first is the file system level. During the file system reconstruction process, it's possible, in many cases and occasions, to recover deleted files or hidden files.

The second level is retrieving deleted information from database files. In some database files that can be found on smartphones, it's possible to recover deleted records, such as SMS, MMS, call log entries, contacts, messages etc. Supported data types obtained using physical extraction include intact and deleted passwords, installed applications, geographical tags, location information, media files such as photos and videos taken by the user, GPS fixes, emails, chats and many more.[2]

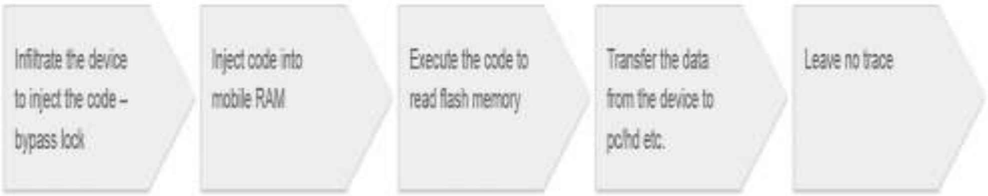

Infiltrate the device to inject the code – bypass lock
Inject code into mobile RAM
Execute the code to read flash memory
Transfer the data from the device to pc/hd etc.
Leave no trace

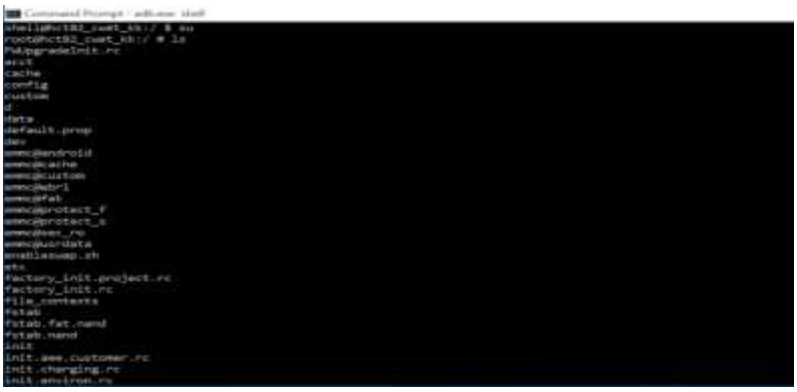
Figure 2: PHYSICAL ACQUISITION PROCESS

*This process can vary for different devices

This type of extraction enables the maximum amount of deleted and hidden data to be recovered. Usually it is difficult to achieve physical data from the mobile device as the manufacturers of mobile devices secure against arbitrary analysis of the device's internal memory.

III. ANDROID FILE HIERARCHY

Android organizes its data in files and folders that helps a forensic to analyst narrow down their research to specific locations of the device. Based on the device manufacturer and the underlying Linux version, the structure of this hierarchy may have a few insignificant changes. To see the complete file hierarchy in windows, we need to have root access the device with the help different android application like KingoRoot.apk and BusyBox.apk in android devices.



```
Command Prompt - adb.exe shell
shell@hct82_cwet_kk:/ $ su
root@hct82_cwet_kk:/ # ls
FWUpgradeInit.rc
acct
cache
config
custom
d
data
default.prop
dev
emmc@android
emmc@cache
emmc@custom
emmc@ebr1
emmc@fat
emmc@protect_f
emmc@protect_s
emmc@sec_ro
emmc@usrdata
enableswap.sh
etc
factory_init.project.rc
factory_init.rc
file_contexts
fstab
fstab.fat.nand
fstab.nand
init
init.aee.customer.rc
init.charging.rc
init.environ.rc
```

Figure 3: Android File Hierarchy

IV.RESEARCH METHODOLOGY

Memory Storage of android devices

In android smart phone there are four types of memories are available.

i) Phone memory: In this the actual operating systems partitions and OS related files are located.

ii) Internal memory: In this memory all installed applications, its data, gallery and others important things are there.

iii) External memory is an extension to internal memory and it is used to store app related information, backup data, videos, photos and other information on user interest.

iv) RAM: This memory area contains details about currently running processes, data structures and information related to communication between various running processes. This is a volatile memory and very crucial for forensic investigation. The forensic analysis of this volatile memory is also called live forensic analysis. Currently, this paper discusses about major types of acquisition methods namely physical acquisition by using Command Line called adb command. [3]

ANDROID DEBUG BRIDGE (ADB)

Android Debug Bridge (adb) is a flexible command-line tool that runs on different OS like Linux, windows, Ubuntu and that lets you communicate with a connecting device. It provides access to a Unix shell that you can use to run a different commands on a device and enables a variety of device actions, such as installing and debugging apps like KingoRoot, BuyBox. It is a client-server program that includes three components:

i) **A client**, which sends commands. The client runs on your development machine.

ii) **A daemon (adb)**, which runs commands on a device. The daemon runs as a background process on each device.

iii) **A server**, which manages communication between the client and the daemon. The server runs as a background process on your development machine. [8]

To get root permission first we have configure USB Debugging of the device so, on most Android devices, do the following: go to “Menu” -> “Settings” -> “Applications” -> “Development” and then click “USB debugging” to enable ADB [1].

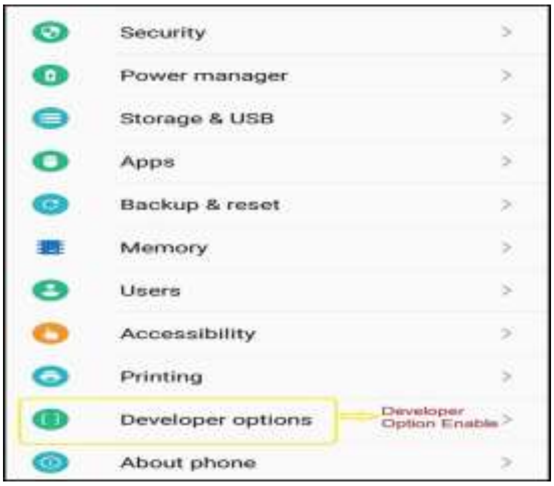
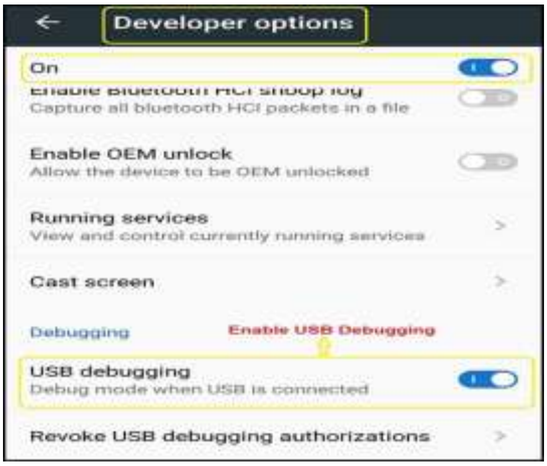


Figure 4: Selecting Developer Options



Developer options

On

Enable Bluetooth HCI snoop log
Capture all bluetooth HCI packets in a file

Enable OEM unlock
Allow the device to be OEM unlocked

Running services
View and control currently running services

Cast screen

Debugging

Enable USB Debugging

USB debugging
Debug mode when USB is connected

Revoke USB debugging authorizations

Figure 5: Enabling USB Debugging

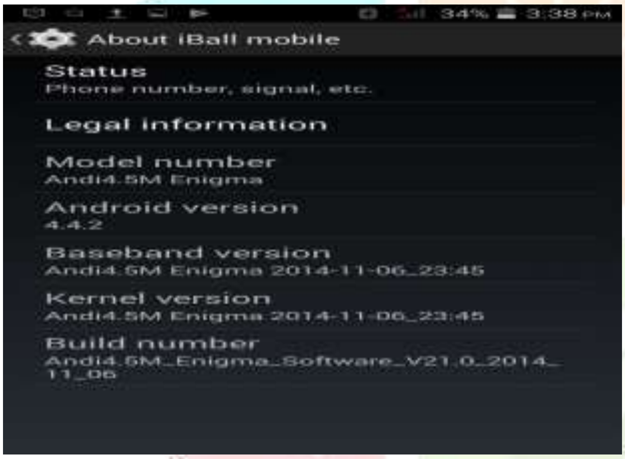
WHY TO ROOT THE ANDROID PHONE

Rooting Android Phone – Rooting is the process of authorizing the users of Android phones to gain the highest privilege i.e. the user have privilege on an Android Phone. Android is based on Linux as discussed. Thus, gaining root access is same as gaining root user access or administrative access on Linux OS.

Why Root an Android Phone – In Android each application is assigned a UID and is run as a separate process, and each application is separated so that one application does not access the data of another application. UID's assigned to each application that are stored in packages (XML file in **/data/system** folder). UIDs, stores the Android permissions of each program as well. The private data of each application is stored in the **/data/data** location and is accessible only to that application. The data present at this location cannot be accessed if the phone is not rooted since a normal user cannot access the application data. However, rooting a phone will allow us to access the data present in any location of the devices. Thus, it is necessary to root the Android Phone. [6]

ANDROID FORENSIC LAB SETUP

For Lab setup I have used Windows 10 Operating System and Android Mobile iball Andi4.5M enigma version 4.4.2 for physically extracting data of the device [1].



34% 3:38 PM

About iBall mobile

Status
Phone number, signal, etc.

Legal information

Model number
Andi4.5M Enigma

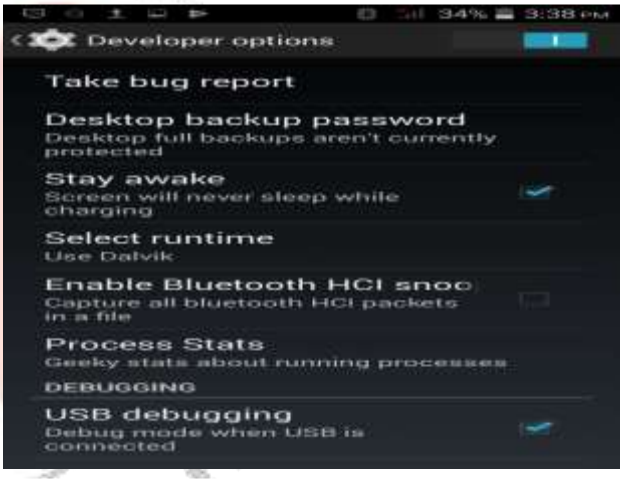
Android version
4.4.2

Baseband version
Andi4.5M Enigma 2014-11-06_23:45

Kernel version
Andi4.5M Enigma 2014-11-06_23:45

Build number
Andi4.5M_Enigma_Software_V21.0_2014_11_06

Figure 6: Details of Device iball Andi4.5m Enigma



34% 3:38 PM

Developer options

Take bug report

Desktop backup password
Desktop full backups aren't currently protected

Stay awake ✓
Screen will never sleep while charging

Select runtime
Use Dalvik

Enable Bluetooth HCI snoop ☐
Capture all bluetooth HCI packets in a file

Process Stats
Geeky stats about running processes

DEBUGGING

USB debugging ✓
Debug mode when USB is connected

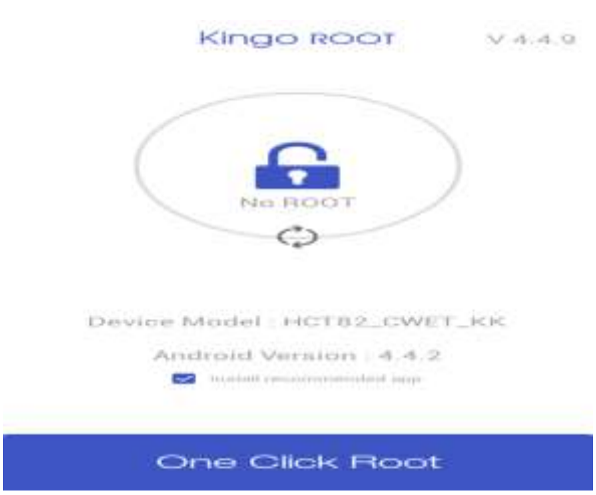
Figure 7: Enabling USB Debugging of iball Andi4.5M enigma version 4.4.2

HOW TO ROOT

In this section, it shows how to Root an Android Phone using different android application. In my case, I have rooted iball Andi4.5M enigma mobile version 4.4.2. Rooting steps for Android mobile will vary as per the mobile manufacturer so the steps may vary from your mobile manufacturer. There are many tools and application available for rooting any device, and it varies from a different manufacturer.

These are following steps to root the android device:

- Download KingoRoot.apk file on android device.
- Allow installation of apps from unknown sources on your Android device.
  Settings > Security > Unknown Sources
- Install and launch KingoRoot[9]
- Press "One Click Root" on the main interface.
- Wait a few seconds until the result appear.

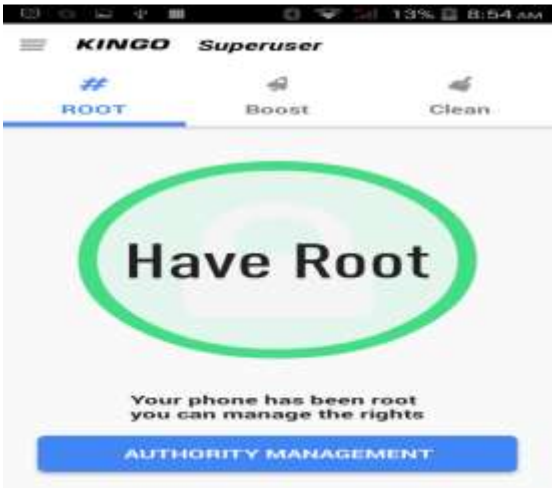


Device Model : HCT82_CWET_KK

Android Version : 4.4.2

☑ Install recommended app

Figure 8: Rooting in Progress with KingoRoot app

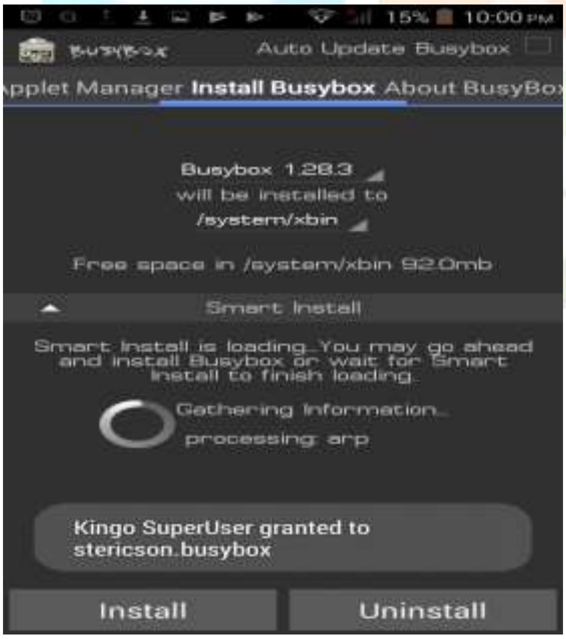


Your phone has been root
you can manage the rights

Figure 9: Device Connected Notification

These are following steps for super user access of rooted android device:

- Head to the Google Play Store from your rooted Android device. [10]
- Install **BusyBox**, this app was created by developer Stephen Ericson, and
- It is constantly updated with the latest BusyBox commands.
- It will ask for Superuser access, so tap "Grant" on this popup.



Busybox 1.28.3
will be installed to
/system/xbin

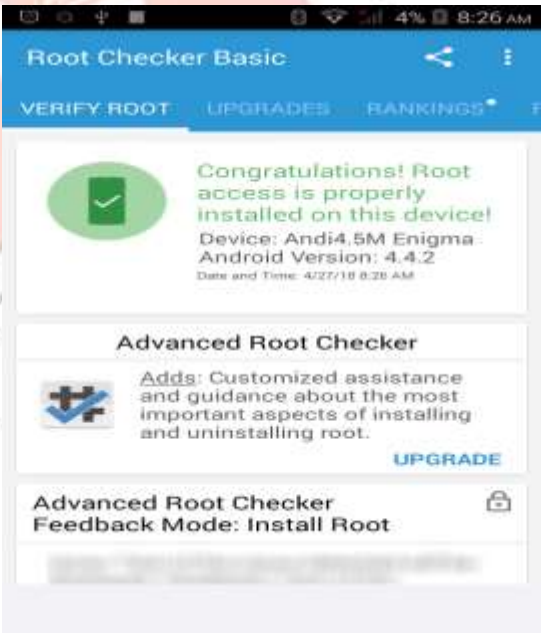
Free space in /system/xbin 920mb

Smart Install is loading...You may go ahead and install Busybox or wait for Smart Install to finish loading

Gathering Information...
processing: arp

Kingo SuperUser granted to stericson.busybox

Figure 10: SuperSU Installed successfully using BusyBox app



Congratulations! Root access is properly installed on this device!

Device: Andi4.5M Enigma

Android Version: 4.4.2

Date and Time: 4/27/18 8:26 AM

Advanced Root Checker

Adds: Customized assistance and guidance about the most important aspects of installing and uninstalling root.

Advanced Root Checker
Feedback Mode: Install Root

Figure 11: To check Root Access on device using Root Checker app

GETTING STARTED WITH ADB COMMAND

Android Debug Bridge (ADB) – In Android forensics, ADB plays an important role. It is present in <sdk_path>/platform-tools folder of the device.

Example: This is my path of <sdk_path> C:\Users\Khushboo\Desktop\adb\platform-tools. There you will find an executable called adb.exe as shown and for tcp connection ncat.exe.

Example: In this practical my windows 10 OS it is located **C:\Users\Khushboo\Desktop\adb\platform-tools**. There you will find an executable called adb.exe and for tcp connection ncat.exe.

Android Debug Bridge as the name suggests it acts as a channel between computer and the mobile phone. It usually runs with a non-privilege case account. Thus, it will not provide access to internal application data. But on a rooted phone, ADB will run with root shell account and deliver access to internal application data, OS files and folders.

Using ADB to access the device – Connect the device to the computer. After connecting the device to the computer and before issuing adb commands, it is helpful to know whether the mobile phone is connected to the adb server. This can be done using the “adb.exe” devices command. This command lists out all the devices that are connected to the computer, as shown in the following command.

For Help command:

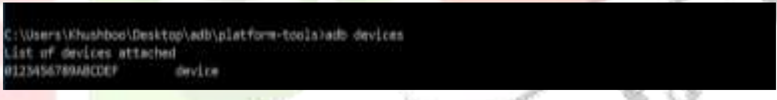
```
C:\Users\Khushboo\Desktop\adb\platform-tools >adb.exe –h

C:\Users\Khushboo\Desktop\adb\platform-tools >ncat.exe –h
```

To show the directory of <sdk_path>/platform-tools folder

```
C:\Users\Khushboo\Desktop\adb\platform-tools >dir

C:\Users\Khushboo\Desktop\adb\platform-tools >adb.exe devices
```

List of devices attached

0123456789ABCDEF device



```
C:\Users\Khushboo\Desktop\adb\platform-tools>adb devices
List of devices attached
0123456789ABCDEF        device
```

Figure 12: List of devices attached

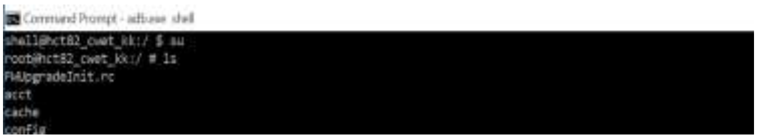
Above command shows the List of devices that I attached on it. And for issuing Shell Commands to the Mobile Phone – As stated above Android runs on Windows 10 shell prompt in VMware and provides a way to access the shell. Using ADB, we can access or gain a shell on Android Phone. Once we access or gain shell, we can run most of the Linux commands. We can gain shell access on mobile using adb.exe command as shown below –[3]

```
C:\Users\Khushboo\Desktop\adb\platform-tools >adb.exe shell

1|shell@hct82_cwet_kk:/ $ ls
```

For super user access you will have to allow USB debugging on your device. By accessing super user **1|shell@** change into **root@.**

```
1|shell@hct82_cwet_kk:/ $ su

root@hct82_cwet_kk:/ # ls /data
```

For instance, as shown in the below command line, ls command can be used to view all the files within a directory.



```
shell@hct82_cwet_kk:/ $ su
root@hct82_cwet_kk:/ # ls
FWUpgradeInit.rc
acct
cache
config
```

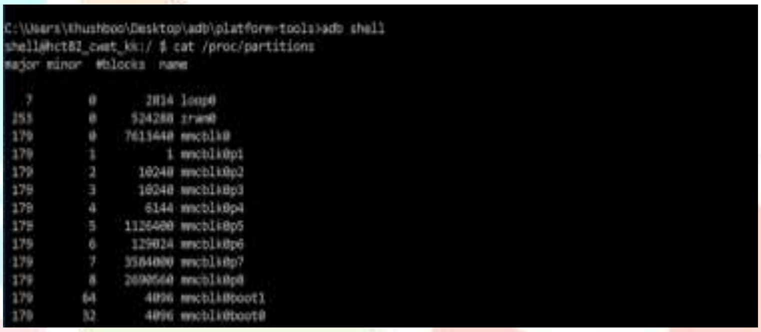
Figure 13: Superuser(su) access Shell Prompt

DETERMINING WHAT TO IMAGE

When imaging a computer, an examiner must first find what the drive is mounted or not as; */dev/sda*, for example. The same is true when imaging an Android device. The first step is to launch the ADB shell and view the */proc/partitions* file using the following command:

root@hct82_cwet_kk:/ # cat /proc/partitions

The output will show all partitions on the device:



```
C:\Users\Khushboo\Desktop\adb\platform-tools>adb shell
shell@hct82_cwet_kk:/ $ cat /proc/partitions
major minor  #blocks  name

   7        0       2014 loop0
 253        0     524288 zram0
 179        0    7613440 mmcblk0
 179        1          1 mmcblk0p1
 179        2      10240 mmcblk0p2
 179        3      10240 mmcblk0p3
 179        4       6144 mmcblk0p4
 179        5    1126400 mmcblk0p5
 179        6     129024 mmcblk0p6
 179        7    3584000 mmcblk0p7
 179        8    2690560 mmcblk0p8
 179       64       4096 mmcblk0boot1
 179       32       4096 mmcblk0boot0
```

Figure 14: All partitions on the device

In the output shown in the preceding screenshot, *mmcblk0* is the entirety of the flash memory on the device. This is because the private data of all the applications are stored in this folder. Thus, the security is enforced by Android. Only the root user has access to this location.

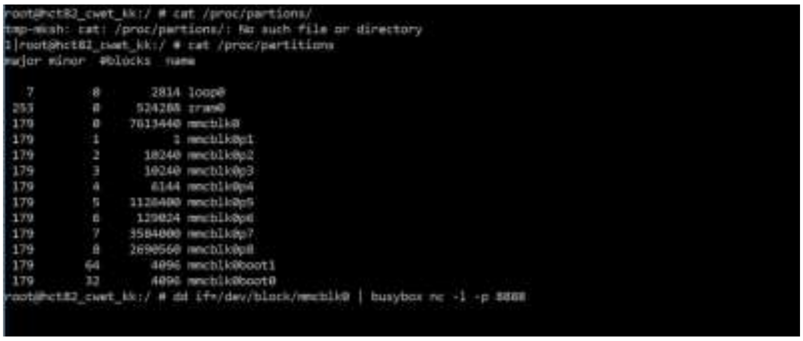
As shown in the above command, through rooting the device private data of all the applications can now be seen easily by navigating to the respective files and folders. Hence, the ADB tool on a rooted device allows us to access all the data of applications installed on the android device.

GET ACCESS TO NETCAT

Netcat provides a resourceful means of investigating a network from the back-end side of the servers and further establish any new connection inside the networks using the aforementioned protocols. It has the capability to be run on its own or through scripts, command line or other programs. It is apparent how the data partition has the following entry to it:

Command:

root@hct82_cwet_kk:/#ddif=/dev/block/mmcblk0 | busyboxnc –l –p 8888



```
root@hct82_cwet_kk:/ # cat /proc/partions/
tmp-mksh: cat: /proc/partions/: No such file or directory
1|root@hct82_cwet_kk:/ # cat /proc/partitions
major minor  #blocks  name

   7        0       2814 loop0
 253        0     524288 zram0
 179        0    7613440 mmcblk0
 179        1          1 mmcblk0p1
 179        2      10240 mmcblk0p2
 179        3      10240 mmcblk0p3
 179        4       6144 mmcblk0p4
 179        5    1126400 mmcblk0p5
 179        6     129024 mmcblk0p6
 179        7    3584000 mmcblk0p7
 179        8    2690560 mmcblk0p8
 179       64       4096 mmcblk0boot1
 179       32       4096 mmcblk0boot0
root@hct82_cwet_kk:/ # dd if=/dev/block/mmcblk0 | busybox nc -l -p 8888
```

Figure 15: Port 8888 Establish Connection

Local port 8888 has to get forwarded to remote port 8888 using the following command. “adb forward tcp: 8888 tcp: 8888”as shown in the following command.

C:\Users\Khushboo\Desktop\adb\platform-tools >adb forward tcp: 8888 tcp: 8888

In my system, Local port 8888 has to get forwarded to remote port 8888 using the following command. “adb forward tcp: 8888 tcp: 8888”as shown in the above image. The actual meaning of such command is that when a connection gets established on port 8888 on the local machine, port 8888 on the Android device will receive the same connection redirected from the local machine’s port.

C:\Users\Khushboo\Desktop\adb\platform-tools >ncat.exe 127.0.0.1 8888 > android.dd

Extracting the data partition could be performed using the following command which relies on the “dd” tool along with the netcat tool.

V. RESULT

Recovering deleted data from device using autopsy

Recovering deleted data is analyzed by open source tool called Autopsy. When a user deletes any data from the Android device, the data is not actually erased or deleted [4]. What gets deleted is the pointer to this particular data. All file systems contain metadata that maintains information about the hierarchy of files, file names, file structure or so on. Hence, it is possible to partially recover the deleted data from open source tool and command prompt of windows. [7]

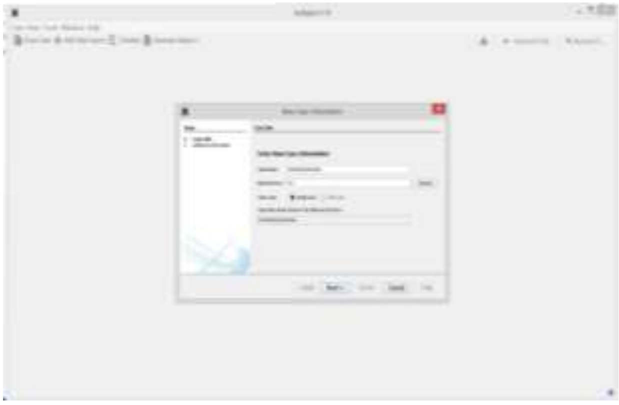


Figure 16: Create a New Case in Autopsy 4.1.0

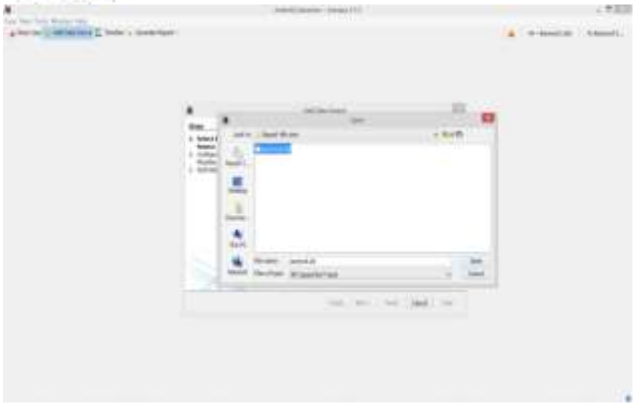


Figure 17: Select .dd file and open this in tool which was created by adb command

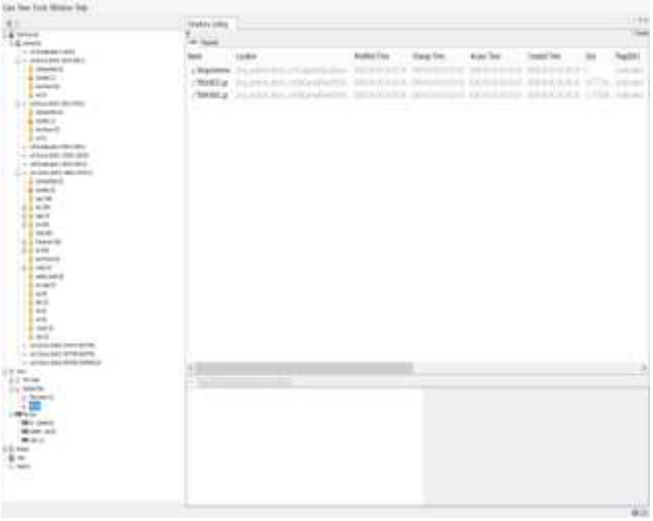


Figure 17: Shows Extracted file which was deleted

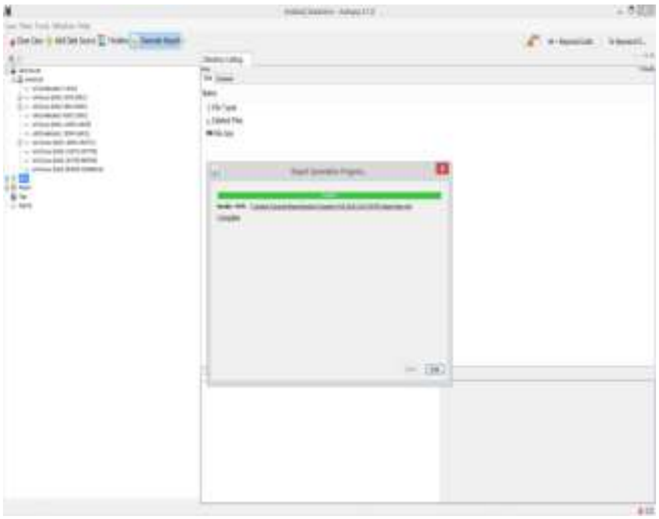


Figure 18: Report generation in different format like html,excel,text file

VI. CONCLUSION

Using open source tool like adb command and Autopsy I conclude that it is only extracting recently deleted file from the android devices .The forensics analysis of Android phone and Android application involves different tools and technique than traditional forensics, as the version or security of devices upgrades new methods are to be studied for Mobile forensics. Apart from other challenges like extracting and recovering deleted with open source, live data, bypassing screen lock and password, maintaining the integrity of mobile data and application data is the prime challenge faced in any Android Forensics. Though lots of open and commercial tools are available for Mobile Forensics, there are breaches to be filled, and a lot needs to be done in this direction.

REFERENCE

[1] SudipHazra and PrabhakarMateti,"Challenges in Android Forensics" Communications in Computer and Information Science, November 2017

[2]Source:http://ec2-107-23-31-70.compute1.amazonaws.com/mobileforensics/capabilities/operations/physical-extraction

[3] Jeff Lessard and Gary Kessler, "Android Forensics: Simplifying Cell Phone Examinations", Small Scale Digital Device Forensics Journal Vol. 4, No.1, ISSN# 1941-6164, September 2010.

[4] Imam Riadi and Sunardi,"Forensic Investigation Technique on Android's Blackberry Messenger using NIST Framework",October 2017.

[5] Venkateswara Rao V. and A. S. N. Chakravarthy,"Survey on Android Forensic Tools and Methodologies"International Journal of Computer Applications (0975 – 8887), Volume 154 – No.8, November 2016.

[6] Platform Architecture, https://source.android.com/images/android_framework.

[7] D. Manson, A. Carlin, S. Ramos, A. Gyger, M. Kaufman, and J. Treichelt, "Is the open way a better way? Digital forensics using open source tools," in System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on. IEEE, 2007, p. 266b.

[8] Android Debug Bridge (adb), Source:https://developer.android.com/studio/command-line/adb.

[9] KingoRoot.apk: KingoRoot – Android Rooting Forensic Application. Available at https://kingroot.net/thank-you-for-downloading-kingroot-for android/

[10]BusyBox Application: Available at https://play.google.com/store/apps/details?id=stericson.busybox&hl=en_IN