NFSU JOURNAL OF CYBER SECURITY & DIGITAL FORENSICS

NFSU – Journal of Cyber Security and Digital Forensics
Volume – 2, Issue – 1, June 2023
E-ISSN - 2583-7559

ANDROID MOBILE CYBERCRIMES AND ITS FORENSIC

Bandu B. Meshram
Professor, & Chairman, Jeman Educational Society,
Navi, Mumbai, Maharashtra,(India)
bbmeshram.jes@gmail.com

Manish Kumar Singh,
Head Of Law Department ,
NIMS, School of Law.
NIMS University Rajasthan, Jaipur(India)
manishksinghlaw@gmail.com

***Abstract*—Mobile Systems are developed each day with enhanced features such as cameras, personal digital assistants and the global positioning systems. Although these features have been tailored to meet user's needs, they have simplified criminals' activities of committing crimes without physically being available at the crime scene. The statement of the problem about mobile forensic is defined and research methodology used in this experimentation is discussed. The mobile attack surface area is identified with attacks on mobile communication systems. The attacks on the smartphones are identified during the research and its categorization is made into mobile attacks using operating systems, mobile applications, internet browser based attacks, network based attacks, hardware based attacks. The researchers also investigate the types of attacks based on mobile surface namely network based attack, device based attack, data center based attack, operating systems based attack and browser based attacks and the OWASP top 10 mobile vulnerabilities and attacks. The list of attacking tool, mobile forensic tools which are used at each steps of mobile forensic are identified and evidence paths are also listed by experimentation. The deployment environment for android mobile forensic is set to carry out the research experimentation and diagnosis of attack to recover, preserve, analyses its data and related materials so that the investigation agency presents the evidence in a court of law. The practical experimentation shows the mobile forensic steps for initial response, data acquisition and duplication - recovering deleted files to extract the evidence. Secondly the mobile forensic software tool consisting of initialization, acquisition and investigation is proposed. The experimentation and proposed tool does the forensics of information like SMS, MMS, Contacts, Call Logs, Gmail Account, Hardware Details, network information related to Wi-Fi, Bluetooth and third party applications and social media.**

Keywords—mobile forensic attacks, process model , data extraction, recovery, initialization, acquisition, investigation, evidence.

I. BACKGROUND AND MOTIVATION

Mobile forensic is monitoring and analysis of mobile traffic for the purposes of information gathering, evidence collection, preservation and extracting legal evidence, Evidence analysis for intrusion detection or attacker detection, preparing evidence report to produce it into the court of law ..

.

Article 39A of the Indian Constitution [1] provides for equal justice and free legal aid: "The State shall secure that the operation of the legal system promotes justice, on a basis of equal opportunity, and shall, in particular, provide free legal aid, by suitable legislation or schemes or in any other way, to ensure that opportunities for securing justice are not denied to any citizen by reason of economic or other disabilities. There is a need to spread the awareness of mobile cyber-crimes among the people of India, The ignorant citizens about cyber-crime can be easily cheated by the hackers to steal data and money. . In global mobile market more than 60% smart phones are used by android in market . Smart phones have been an integral part of our daily life. Every day, Android phones are used to make calls, send texts, check emails, snap photos, and surf the web. Between 2017 and 2022, the Android operating system was used by more over 50% of all smartphones.. However, because

of their widespread use, Android devices have become the most appealing targets for cyber thieves.

The paper is organized as below.

Section II explore the research problem.

Section III identify the mobile attack surface area and explore attacks on mobile device. Section IV presents the experimentation and diagnosis research for mobile forensic. Section V presents the proposed Mobile forensic tool .Lastly section VI Concludes the results.

II . STATEMENT OF THE RESEARCH PROBLEM

The digital evidence under sec 45, 45A, 46 of Evidence Act [2] can then be produced in the court of law by expert's attendance in court[3] (Sec 293 Cr.P.C) for punishing the Mobile cybercriminal under IPC[4] OR IT Act 2000[5],Hence the research problem is titled as

"Android Mobile Cybercrimes and Its Forensic **"**

A. Aim of Research

Aim : " To investigate android mobile cybercrimes and digital forensic evidence for the prosecution of mobile systems attacker/hacker in the court of law"

B. Research Objectives

The core objectives are.

- To explore cybercrime surface area and attacks in the Android mobile System.
- To do the research experimentation and diagnosis for mobile forensic
- To explore Information and paths of evidence that resides on android mobile device.
- To propose and implement the mobile forensic framework

C. Research Methodology Used

This section describes research methodology with respect to mobile forensic research design, population and sampling, data collection and measurement of the results.

1)Research Design :As the research is critical, the research design will be based on the exploratory research studies, diagnostic studies and experimental studies to do the qualitative research for in-depth understanding of the phenomenon of android mobile forensic

Research design[8] is made with respect to

a)Sample Design : collection of data about entities connected to mobile using email, social media, messaging and communication by mobile; text, voice communication or video chatb b)Observational Design: observation of various log files and the data into the path files.c))Statistical Design**:** communication frequency, abuse language used or vulgar video etc. d) Operational Design**:** forensic identification based on all above parameters.

2) The research design adopted to do this research is descriptive, diagnostic, exploratory and experimental research.

a) Exploratory Study to formulate the problem for more precise investigation using review literature and experience survey to acquire the knowledge of experience people in the domain.

b) Descriptive Research studies aims to describe the facts and the situations they are concerned with what? is done using attack types. c) Diagnostic Study is adopted using mobile forensic process and mobile tools to know about the attacker and forensic investigation.

3).Literature Survey: The extensive literature survey is done by using database search engines, various journal papers and papers of national and international conferences, Google digging and sci-hub is also used for the literature survey.

4).Population and Sampling**:** In order to carry out the research, convenience sampling is be used.

5).Data Collection Procedure**:** The observation method is used to collect the primary data set. The physical live data collection, data extraction, forensic duplication and data analysis is done.

II. MOBILE ATTACKS

A. Mobile Systems Attack Surface

There are five interlinked attack surfaces areas as shown in Fig. 1

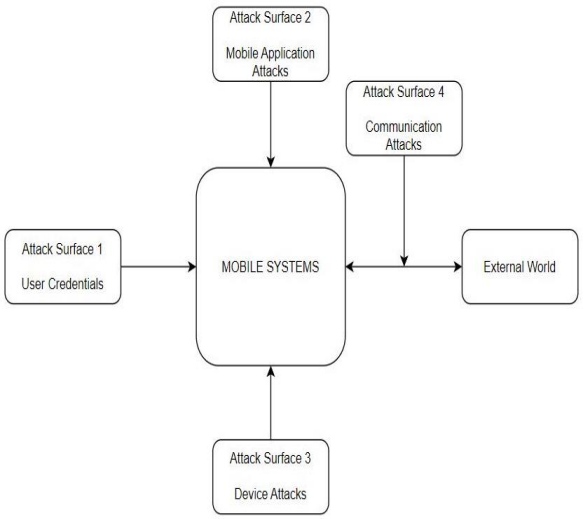
1: Stealing User Credentials:

2 Attack on Application Integrity.

3. Attack on Device Integrity and android operating system

4. Attack on API Channel Integrity: via public Wi-Fi connections.

5.Attack on API and Service application weakness or loopholes . that cybercriminals target.



.

Fig.1 : Attack surface area

B. Types of Attacks Based on Surface

The types of attacks based on surface namely **(i)**Network based attack, (ii) Device based attack, (iii) Data center based attack,(iv) Operating Systems based attack (v)browser based attacks are enumerated in Table 1.

NFSU JOURNAL OF CYBER SECURITY & DIGITAL FORENSICS

NFSU – Journal of Cyber Security and Digital Forensics
Volume – 2, Issue – 1, June 2023
E-ISSN - 2583-7559

TABLE 1. THE TYPES OF ATTACKS BASED ON MOBILE SURFACE AREA

Network[12][14]	***Device[13] [16]***
Wi-Fi • Rogue Access Point • Man in the Middle • DNS Poisoning • Denial of Service ((WI-FI) • Man-In-The-Middle (MITM) • Denial of Service Attack(DOS) • Eavesdropping • Wi-Fi Sniffing • Man-in-the-Middle Attack • Rogue Access and Hot Spot • Denial of Service(WI-FI) • Evil Twin Access Point • Attacks Through SMS	Browser Based • Phishing • Framing • Clickjacking Data Caching Phone Based • SmiShing • Baseband Attac
Bluetooth • BlueSnarfing • Bluejacking • Bluebugging	Hardware Based Attacks • Cold Boot Attacks • Hard Drive Side Channel Attacks • SIM Card fraud • Camera Based Attacks
Internet Browser Based Attacks • Rooting the device • Single Sign-On(SSO) • Client Side Injection • Improper Session Handling • Browser Exploits • Phishing	System Based • Phone Rooting • Carrier Loaded Software • JailBreaking • Operating systems bases attacks
Operating systems bases attacks[15][17] • Exploiting custom permission vulnerability • Exploiting URI permission Flaw • Permission re-delegation attack • Local root Exploit	
Data Centre Based Attack	
Application based[18][19] Mobile OWASP 10 Attack • Improper Platform Usage, • Insecure Data Storage, • Insecure Extraneous Functionality	Database • Data Dumping • SQL, BOF

C. The OWASP Top 10 Mobile Application Attacks

TABLE 2. THE OWASP TOP 10 MOBILE VULNERABILITIES AND ATTACKS [19]

Vulnerabilities (OWASP TOP 10)	***Attack***
Authentication Insecure ,Authorization Insecure	SQL injection, xss[20]
Platform Usage Improper Data Storage Insecure , Cryptography Insufficient, Code Quality of Client , Extraneous Functionality	Database leak, Data leak[21], Code smell [22], Malware[23],
Tampering of Code Reverse Engineering	Repackaging [24] Malware [23][25], data leak[21], Phishing[26]
Communication Insecure	MITM[27] ·Data leak[21]

D. Android Malware Categories

The Summary of Android malware categories[28], [29][30][31] is shown in Table 4.

TABLE 3. ANDROID MALWARE CATEGORIES

Malware Category	***Purpose***	***Common android Malware Families***
Adware	Provides users with intrusive pop-up advertisements.	appadgexin, batmobi, ewind, shedun, pandaad,
Backdoor	Hides in the background to exploit the device covertly.	mobby, kapuser, hiddad, dendroid, levida
File Infector	The files are infected, particularly Mobile application executable (APK) files.	tachi, leech, gudex, commplat,

PUA	Interrupts the device's normal operation.	scamapp, utchi, cauly,umpay, apptrack, secapk, wiyun, youmi
Ransomware	The file is encrypted and a ransom is demanded for the user to access the data.	lockscreen, slocker,congur, masnu, fusob, jisut, koler,
Riskware	Possibly exposes the smartphone to vulnerabilities.	badpac, mobilepay, wificrack, triada, skymobi,
Scareware	Uses fear to entice the user to download malicious apps by creating fear in their minds.	avpass, mobwin, and fakeapp
Spyware	Spies on the device and sends information to a remotely controlled server in order to steal information.	spynote, qqspy, spydealer, smsthief, spyagent,
Trojan	trojan-dropper,Trojan-banker, trojan-spy, and trojan-sms are all characterized by acting like impersonators that steal information from the device.	guerrilla, gugi, hqwar, obtes, and hypay

E.Attack experimentation fo Forensic Path Information

In this section, we have made the attack experimentation by using the software tools and listed the paths in which investigator can find the evidence of attack.

1) Information Gathering Attack

TABLE 6. INFORMATION GATHERING

Software Tool	Purpose	Attack Path
Red_Hawk	All in one tool for Information Gathering and Vulnerability Scanning	Database Path C:/Users/Kali/rootfs/red_hawk/db/report.txt
Phoneinfoga	to scan phone numbers	TOOL PATH C:\Users\USER\AppData\Local\Packages\KaliLinux\LocalState\rootfs\phoneinfoga

F. All vulnerability and Phishing Attack Paths

TABLE 5. PHISHING AND VULNERABILITY ATTACK

Software Tool	*Purpose*	*Path*
Software Engineering Toolkit	penetration testing framework for making ethical attacks	Credential Database Path C:/Users/Kali/Rootfs/Set/Db/Database.Txt Attack Path /Data/Data/Com.Facebook.Katana DDOS ATTACK C:/Users/Kali/Rootfs/Set/Db/Database.Txt
Zphiser	phishing Tool in wide area network.	Get IP Address - C:\Users\User\Appdata\Local\Packages\Kalilinux.54290c8133fee_Ey8k8hqnwqnmg\Localstate\Rootfs\Zphiser\Ip.Txt **Get Credentials - Username & Password** C:\Users\User\Appdata\Local\Packages\Kalilinux.54290c8133fee_Ey8k8hqnwqnmg\Localstate\Rootfs\Zphiser\Usernames.Dat **Attack Path** /Data/Data/Com.Instagram.Android/Cache /Data/Data/Com.Facebook.Katana /Data/Data/Com.Snapchat.Android

G. Forensic Hardware, Software Tools

The mobile forensic tools which are used at each steps are listed in Table 7.

TABLE 7. STANDARDIZED HARDWARE & SOFTWARE TOOLS

Process Name	***Hardware Used***	***Software Used***
Seizure	Mobile phone evidence box, Phone jammer, Faraday bag	-
Acquisition & Deleted Data	Fernico ZRT, EDEC	XRY Logical[35],

Recovery	Eclipse	Oxygen Forensic Suite[36], Lantern, FTK Imager Lite[38], Final Mobile Forensic Tool[37].
Identification & Examination	Fernico ZRT, EDEC Eclipse	XRY Logical, Oxygen Forensic Suite, Lantern
Evaluation & Analysis	-	Open Source Android Forensic &Final Mobile Forensic Tool.
Admission as Evidence	evidence box, Phone jammer, Faraday bag	Cellebrite Touch, Encase Forensic[39].

III. ANDROID FORENSIC: EXPERIMENTATION & DIAGONISTIC RESEARCH

This section presents descriptive, diagnostic exploratory and experimental research for mobile forensic.

A. Access Points

Before heading towards attacks, it's important to have a look on various access points of a mobile device as shown in Table 8.

TABLE 8 : ACCESS POINT

Access Points	*Paths In Android*
BLUETOOTH	/FileManager/InternalStorage/MyBluetooth
WIFI	/FileManager/InternalStorage/data/misc/wifi/wpa_supplicant.conf
Hotspot	/FileManager/InternalStorage/data/misc/wifi/softap.conf
USB port	/FileManager/ExternalStorage/mnt
Headphone jack	/FileManager/ExternalStorage/system/etc/
Microphones	/FileManager/InternalStorage/res/raw/
WhatsApp	/FileManager/InternalStorage/WhatsApp
Recycle bin	/FileManager/InternalStorage/Photos /Recent Deleted
Hard Disk (IDE)	PCIROOT(0)#PCI(0100)#ATA(C01T03L00)
IMEI Number	Go to setting → about phone → you will get phone number, mobile model name, model number, serial number, IMEI Number. Go to status information under about phone you will get device mac address of mobile.

B. Setting Deployment Environment

This section provides the technicality about the mobile forensic process models [42] [43].

Fig. 2 shows the deployment environment for experimentation

1)Create Android Forensic Workstation

a) Install Java SE Development Kit (JDK) with SDK On Forensic Workstation.

b)Android Virtual Device **(**an emulator/**AVD) :**To create a new AVD (on the Windows workstation), perform the following command.

C:\android\BBM>android avd

c)Connecting an Android device to a forensic workstation..

Fig 2: Experimentation Environment for Mobile Forensic

d) Installing the device drivers on workstation e) Accessing the connected Android device by workstation**:** using the USB cable. g)Install Android Debug Bridge (adb) on forensic **workstation**. f)Set Android Device in USB debugging Mode e)Accessing the device using adb j) Loading Android Forensic Tools On Forensic Workstations.

C .Mobile Seizure

1)Procedure for Seizing Mobile devices Forensic

- Isolate the attacked mobile using (1) Faraday box, Airplane Mode + Disabling Wi-Fi and Hotspots, or 2) Cloning the device SIM card. Or use Faraday box
- Documentation of evidences with tags and labels of devices and tool used..
- Transporting Mobile Device Evidence to Forensic laboratory.

- Prepare the chain of custody for mobile forensic investigation procedure.

D. Identification (Backups of Mobile data)

Before backing up the mobile data or replica imaging of SIM Card , , the mobile forensic investigator must recover the deleted data and then take the whole backup of data..*[41].* Table 8 shows :mobile parts &information recovered from them.

TABLE 8 :MOBILE PARTS &INFORMATION RECOVERED

Sr.No.	Data is stored into three different locations.	What type of information do we expect to recover from mobile parts ?
1.	Subscriber Identity Module (SIM) Card-smart card consists of Storage and processor.	ICCID(Card Identifier), IMSI(Subscriber Identity), Text based User Data(SMS, Contacts and calls).. Phone numbers of calls made/received , Contacts , SMS details (time/date, recipient, etc.) , SMS text Service Provider Data
2.	Memory Car	Media files –pictures, audio, video &documents. Smart phone applications &databases , Any kind of files
3	Mobile device	Calls, contacts, SMS, soci Media Data, Multimedia data browser,History,location Information.

1).Manual data extraction:. The following steps are used to obtain a forensic image of a rooted Android device. .

- Enter following commands to gain root access of mobile.
  i .c:\user\BBM >adb reboot
  ii. c:\user\BBM > adb kill-server
  iii. c:\user\BBM > adb start-server
  iv. c:\user\BBM > adb root
- Install the Android Terminal Emulator application[49].
- Executing the `mount` command::

shell@Android:/ $ mount

Once the mounting is done, execute the DD command to bit-by-bit image of the device into new SD card:

C:\android\BBM> $ *dd if=/dev/block/Block number of=/sdcard/tmp.image*

The preceding command will make a bit-by-bit image of the Blocknumber (data partition) and copy the image file to an SD card.

2).Logical data extraction **:** Logical data extraction[44] techniques extract the data present on the device by accessing the file system[44].

a) You can use MSAB XRY tool for logical data extraction or adb pull command. You can extract the data from . various locations of data as given below:

- Shared preferences file location application /data directory/ shared_pref.
- Internal storage location /data/data
- External storage location /sdcard directory.
- SQLite database location /data/data/PackageName/ database.db
  from which we can pull the data.

C:\android\BBM>adb.exe pull /data C:\BBM \ pulleddatafolder
pull: /data/data/ /www.jes.com - > C:\ pulleddatafolder /data/ / www.jes.com
pull: /data/data/com.mymobile.android/lib/libpng2.so -> C:\ pulleddatafolder /data/ com.mymobiler.android/lib/libpng2.so
pull:/data/system.notfirstrun->C:\pulleddatafolder /system.notfirstrun
`output:`
700 files pulled. 0 files skipped**.**
---

3) Obtaining HASH of memory image file

The output of the the sha256sum utility is redirected to the hash .txt file in the user's home directory.

C:\android\BBM >$ find ~ -type f –exec sha256 sum { } \ ; > ~/ hash .txt.*txt*

4)Subscriber Identity Module Analysis(SIM) **:** SIMXtractor tool developed by CDAC extract the information from the SIM cards. It is also used for SIM card Reader , SIM Imager & Analyzer.

C. Android Data Recovery Techniques

Recovering deleted data from an Android device involves following scenarios:

1)Recovering deleted data from an SD card[46] using Remo Recovery Tool and Android.Tools such as LiME[51] can be used to do complete the memory capture

2)Recovering data deleted from internal memory: Recovering files which are deleted from Android's internal memory (such as SMS, contacts, app data, and more) are supported by cellebrite UFED or oxygen forensic.

3) Recovering deleted files by parsing SQLite database deleted data can be recovered from unallocated blocks and free blocks using forensic tools such as Oxygen Forensics SQLite Viewer **or** You can reuse the existing scripts[52].

4)The steps to recover deleted SMS/MMS using the adb command-line tool from the mmssms.db file:

C:\android\BBM>adb.exe
pull
/data/data/com.android.providers.telephony/databases
C:\ android\BBM\ pulleddatafolder
pull: building file list...
pull:
/data/data/com.android.providers.telephony/databases/mmssms.db -journal -> C:
pulleddatafolder /mmssms.db-journal
pull:
/data/data/com.android.providers.telephony/databases/telephony .db-journal -> C:
pulleddatafolder /telephony.db-journal
pull:
/data/data/com.android.providers.telephony/databases/mmssms.db -> C:\ android\BBM
pulleddatafolder /mmssms.db
pull:
/data/data/com.android.providers.telephony/databases/telephony .db -> C:\ :\ pulleddatafolder /telephony.db
10 files pulled. 0 files skipped.

You can view the mmssms.db file using Oxygen Forensics SQLite Viewer tool.

b) you can also recovering files using file-carving carving tools such as Scalpel from unallocated space. The DiskDigger software can recover different types of files from both the internal memory and SD cards. You can also restore the contacts on the device through the Google account configured with the device.

E. Examination & forensic Analysis

AccessData, Sleuthkit, and Final Mobile forensic are some popular forensic software products that have analytic capabilities with forensic analysis functionality[46]. The exact path of various files may vary slightly depending on the device and the version of Android, but it is commonly found in the following paths which may contain various subfolders.

a)Extracting physically damaged device information **:** by viewing the build.prop file present in the /system folder, as follows:

C:\android\BBM>$ cat build.prop
cat build.prop

b)Extracting call logs **:** The information about call logs is stored in the contacts2. db file located at /data/data/com.android.providers.contacts/databases/.

c)Extract the data from mobile device to forensic workstation using adb.exe commond

C:\android\BBM> adb.exe
pull
/data/data/com.android.providers.contacts
C:\android\BBM> pulleddatafolder
pull: building file list...
.. ..
pull:

data/data/com.android.providers.contacts/databases/contacts2.db -> C:\android\BBM> pulleddatafolder /databases/contacts2.db
pull:
/data/data/com.android.providers.contacts/databases/profile.db -> C:\ pulleddatafolder /databases/profile.db
pull:
/data/data/com.android.providers.contacts/databases/profile.db-journal -> C:\android\BBM> pulleddatafolder /databases/profile.db-journal

d)Analyse the extracted data at workstation using SQLite Browser :. The calls table present in the contacts2.db file provides information about the call history.

e) Extracting SMS/MMS **:**The mmssms.db file which is present under the /data/data/com.android. providers. Telephony/databases location contains the necessary message logs. using similar adb.exe pull procedure used in Section E (c)

f)Extracting browser history: The the browser history is located at /data/data/com.Android.Browser/ browser2.db.

g)Analysis of Social networking sites: The Social Media and IM chat applications data is stored in for example /data/data/com.facebook.katana OR /sdcard/WhatsApp/ folder and navigate to the databases folder.

h) Android File systems

1) /boot: partition contains the kernel and RAM disk.

2)/system:. Folder contains system-related files. can be viewed by using the command:

C: \android\BBM > $ cd /system
cd /system
C:\android\BBM >/system $ ls

3)/recovery: This allows the device to boot into the recovery mode to take the backup..

4)/data: The contents of the data folder can be viewed using the command:

C:\android\BBM > adb.exe shell
C: \android\BBM > $ cd /data
cd /data
C:\android\BBM >$data # ls

5) /cache: The cache partition is used to store frequently accessed data and some of the logs for faster retrieval..

6) /misc:, The folder contains information about hardware settings, USB settings, and the On/Off state of the device.

7) The contents of the file systems in the proc folder can be viewed by using the following command:

C:\android\BBM > $ cat /proc/filesystems

8) To find various folders under the sys directory in an Android device:

C:\android\BBM >$ cd /sys

cd /sys

C:\android\BBM > $ ls

The following command shows all information about the CPU of the device:

C:\android\BBM >$ $ cat /proc/cpuinfo

9) To show adb shell is connected to the device give the commond:

C:\android\BBM >$ $ ls -l /dev/pts

3)Mobile phone's memory **:** . the memory content is "cloned using FTK imager, or Final Mobile forensic

10)Viewing Android file systems on an Android device[45]**.** . The "tree" command is used to see hierarchy of folders for the specified directory.

C:\android\BBM OMFT > $ tree -L 1

-L 1 option specifies the depth of the directory tree to display. In this case, it is set to 1, which means only the immediate subdirectories and files in the current directory will be shown.

IV PROPOSED MOBILE FORENSIC FRAMWORK

Based on the mobile forensic life cycle, the optimized mobile forensic tool is proposed having three steps[48][47] namely Mobile Seizure, Acquisition and Analysis of evidence as depicted in fig .3.

Steps for Experiment:

1. Connect theSmartphone to the PC in debugging mode
2. In terminal, run the following command to list all the connected devices
   adb devices
3. Then you can run following command to see all contents ofthe log file created by system.
   Adblogcat

Based on the functioning of components, the proposed tool is divided into 3 modules .namely Initialization, Acquisition and forensic Analysis

Initialization

- The seized mobile device is rooted.
- ADB (Android Debug Bridge), Java is installed on the computer device.
- The mobile phone is then connected to the computer for data acquisition.

1)Mobile Seizure: The seized mobile device is rooted and ADB (Android Debug Bridge), Java is installed on the computer device.

2)Acquisition: The mobile phone is then connected to the computer for data acquisition. The potential evidence can be obtained from SMS, Contacts and Call Logs,Wi-Fi, Bluetooth Logs, Browser History and Gmail,3rd party application data. The data is extracted from the mobile device by executing DataExtractor.java. This class uses ADB (Android Debug Bridge), Linux Terminal as Super User and AFLogical tool to extract data from the device. The extracted data contains log files, CSV files and SQLite database files and are stored on computer storage permanently. This data shall then be used for analysis.

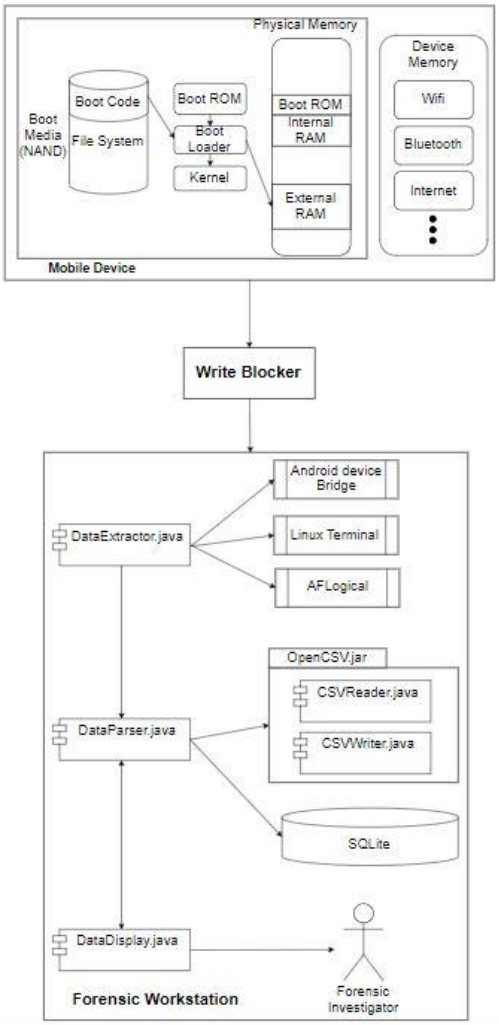


Fig 3. Proposed optimized mobile forensic tool(OMFT)

Following are some potential data locations in the android deviceas shown in Table 10

TABLE 10: TYPE OF EVIDENCE AND LOCATIONS

Type of Evidence	Potential Data Target Location
Contacts	/sdcard/forensics/*date.time*/contacts_phones.csv
Call History	/sdcard/forensics/*date.time*/calllog_calls.csv
SMS	/sdcard/forensics/*date.time*/sms.csv

MMS	/sdcard/forensics/*date.time*/mms.csv
WhatsApp	/data/data/com.WhatsApp/databases/msgstore.db
Browser	/data/data/com.android.browser/databases/browser.db
Wi-Fi	/data/misc/wifi/wpa_supplicant.conf
Bluetooth	From cmd output, via following command- adb shell dumpsystembluetooth
Apps installed	/data/app
Account	From cmd output, via following command- adb shell dumpsys account
Logcat	From cmd output, via following command- adblogcat -d -v time
proc file system	/proc/ filesystems

B. Proposed Data Structures Design for Forensic Tool

The data structure design for the potential data locations in the android device file system is given below table 8 to table

Table 11: SMS.csv

Attribute	Explanation
Address	Contact name or number (if the number is not saved)
Type	Value '1' = Received SMS Value '2' = Sent SMS
Body	Actual content of the SMS
Date	Date of the SMS in encrypted format

Table 12: CallLog.csv

Attribute	Explanation
number	Contact number of the other party in call
date	Date of the call in encrypted format
duration	Total duration of the call in seconds
name	Name of the other party in call

Table 13: Contacts Phones.csv

Attribute	Explanation
number	Contact number of a person
name	Saved name of the contact

Table 14: msgstore.db (WhatsApp)

Attribute	Explanation
key_remote _jid	A unique id given by WhatsApp server to identify a person or group
subject	Name of the group or "NULL" in case of a single person
data	Actual content of the message

Table 15: bluetooth.txt

Attribute/Flag	Explanation
Bluetooth	Gives current Bluetooth status "ON" or "OFF"
Local Address	Gives the MAC address of the device
Local Name	Gives the Bluetooth visible name of the device
isDiscovering()	Gives the discovering other device status "true"= Searching is ON "false"= Searching is OFF
Known Devices	Lists all the available devices along with the pairing information

Table 16: wpa_supplicant.conf (Wi-Fi)

Attribute	Explanation
ssid	Name of the Wi-Fi hotspot or access point
Psk	Pre-Shared Key i.e. the password to join to the hotspot Stored in an encrypted format
key_mgmt	Indicates which security protocol is in use
Priority	Indicates the priority of different hotspots; device prioritizes them; it connects to the hotspot with lower value when more than one is available

Table 17: history.txt (Web Browser)

Attribute	Explanation
Name	Title name of the webpage
url	URL of the webpage

Table 18: logcat.txt

Attribute	Explanation
Date	Calendar date of event (day-month)
timestamp	Time of the event (hours-minutes-seconds.milliseconds)
Name	Name of the event/activity
Pid	Gives the process id
description	Describes the event

Table 19: apps.txt

Attribute	Explanation
Type	Indicates what kind of app it is "system" = System App "data" = User App
Name	Name of the app

C. Proposed Algorithm Design for Forensic Tool

The following are the algorithms for various activities performed by the proposed tool.

Algorithm 1: Searching Connected Device

Pre-condition: Phone is connected and debugging mode isON in the phone

Input: None

Output: Found or Not Found

Procedure:

1. *Run the process "adb devices" on terminal*
2. *Parse output of above process line by line*
3. *WHILE(Not End of Output)*
4. *IF (line contains "List of devices attached") THEN*
5. *IF (next line contains "device") THEN*
6. *Print "Device Found"*
7. *Exit*
8. *End of IF*
9. *End of IF*
10. *Go to next line*
11. *End of WHILE*

Algorithm 2: Extraction of potential data

Pre-condition: Phone is connected and debugging mode isON phone

Input: None

Output: Data on computer

Procedure:

1. Run following processes on terminal
2. Start the AFLogical App (Open Source Tool) to convert and extract the potential database files into *.csv* files

   P1= "*adb shell am start -a android.intent.action.MAIN –n com.viaforensics.android.aflogical_ose/ com.viaforensics.android.ForensicsActivity*"
3. Extract Contacts, Calls, MMS, SMS, Device information to /sdcard/forensics/

   P2= "*adb shell am start -a android.intent.action.MAIN –n com.viaforensics.android.aflogical_ose/ com.viaforensics.android.ExtractAllData*"
4. Copy the data from device to computer folderP3= "*adb pull /sdcard/forensics/ /home/user/Mobile_Forensic*"
5. Copy Wi-Fi data from mobile to computer folderP5= "*adb pull /data/misc/wifi/wpa_supplicant.conf /home/User/MobileData/wifi/*"
6. Copy Browser data from mobile device to computerP6= "*adb pull /data/data/com.android.browser/ databases/ browser2.db /home/user/MobileData/browser/*"
7. Store Logcat output into a logcat.txt fileP7= "*adblogcat -d -v time > /home/user/MobileData/logcat/logcat.txt*"
8. List all the extracted files

   P8 = "*ls /home/user/Mobile_Forensic*"
9. Store the output file_name to string
10. Return string

Algorithm 3: Parsing and Displaying Contacts

Pre-condition: files should be extracted from phone

Input: Contacts Phones.csv

Output: Display Contact List

Procedure:

1. *FR=FileReader("/home/user/Mobile_Forensic/" + folder + "/Contacts Phones.csv")*
2. *WHILE(Not End of File)*
3. *Line=FR.ReadLine()*
4. *String set[]=Separate words from Line*
5. *Print set[10],set[7]*
6. *Go to NextLine*
7. *End of WHILE*

Algorithm 4: Parsing and Displaying Call Logs

Pre-condition: files should be extracted from phone

Input: Extracted files

Output: Display Call History

Procedure:

1. *FR=FileReader("/home/user/Mobile_Forensic/"+ folder + "/CallLog Calls.csv")*
2. *WHILE(Not End of File)*
3. *Line=FR.ReadLine()*
4. *String set[]=Separate words from Line*
5. *Print set[6],set[1],set[3],set[2]*
6. *Go to NextLine*
7. *End of WHILE*

Algorithm 5: Parsing and Displaying SMS

Pre-condition: files should be extracted from phone

Input: Extracted files

Output: Display Messages

Procedure:

1. *CSVReader csv=new CSVreader("/home/user /Mobile_Forensic/" + folder + "/SMS.csv")*
2. *Make various list for storing sms data. Addresslist, Namelist, Bodylist ,.*

ArrayList<String>addresslist = new Arraylist<String>();

3. *WHILE(Not End of File)*
4. *WHILE(Not End of Row)*
5. *Adresslist = csv.getColumn[0];*
6. *Bodylist = csv.getColumn[1];*
7. *End of WHILE*
8. *End of WHILE*
9. *Make a Jlist of addresslist elements*

JList jlist1= new JList.setModel(new DefaultListModel(addresslist));

10. *Map Bodylist to TextPane according to addresslist*
11. *Display this Jlist and TextPane in GUI*

Algorithm 6: Parsing and Displaying Wifi Information

Pre-condition: files should be extracted from phone

Input: Extracted files

Output:Display Wi-Fi information

Procedure:

1. *FR=FileReader("/home/user/MobileData/wifi/wpa_ supplicant.conf")*
2. *WHILE(Not End of File)*
3. *Line=FR.ReadLine()*
4. *IF(Line contains "network") THEN*
5. *Print Line=FR.ReadLine()*
6. *Print Line=FR.ReadLine()*
7. *Print Line=FR.ReadLine()*
8. *Print Line=FR.ReadLine()*
9. *Go to NextLine*
10. *End of IF*
11. *End of WHILE*

Algorithm7: Parsing and Displaying Browser History

Pre-condition: files should be extracted from phone

Input: Extracted files

Output: Display browser history

Procedure:

1. FR=FileReader("/home/user/MobileData/browser/history.txt")
2. WHILE(Not End of File)
3. Line=FR.ReadLine()
4. String set[]=Separate words from Line by "\\"

5. Print set[1],set[2]
6. Go to NextLine
7. End of WHILE

Algorithm 8:Parsing and Displaying WhatsApp Messages

Pre-condition: Files should be extracted from phone

Input: Extracted files

Output: Display WhatsApp messages

Procedure:

1. *Read the database file using sqlite3 in linux terminal*
   "Sqlite3 "/home/user
   /MobileData/WhatsApp/Databases/msgstore.db"
2. *Get the total number of conversations in the database*
   "sqlite3 /home/user/mdata/msgstore.db
   select count(distinct key_remote_jid)
   from messages;""
3. *Get all the conversation IDs*
   "sqlite3 /home/user/mdata/msgstore.db "
   select distinct key_remote_jid from messages;" "
4. *Fetch messages from database for each conversation*
   according to its ID"sqlite3
   /home/user/mdata/msgstore.db
   "selectkey_remote_jid,data from messages where
   data is not null and key_remote_jid=
   '919986934603@s.WhatsApp.net';" "
5. *Display all the fetched messages using the Forensic Tool GUI*

Algorithms 9 Print Forensic Report report Generation

Input : ADB setup
Output : Mobile's overall information is segregated and represented in a tabular report using Web Browser
Variables: ADB serial No., Shell permission, Manufacturer, Model, Android version, Build name, WI-Fi MAC address, Local time, Android time, Accounts, Contacts, Call logs, SMS messages, Whatsapp messages attack paths.
Algorithm:

Step 1: Create report.html file
Step 2: Append the following data
ADB serial No.,Shell permission, Manufacturer ,Model ,Android version ,Build name ,WI-Fi MAC address Local time ,Android time ,Accounts ,Contacts ,Call logs ,SMS messages ,Whatsapp messages ,attack log contents
Step 3 : Save the html report in current working directory

.Analysis by Forensic Investigator: The data stored in CSV files and SQLite database files is parsed in order to be presented in human understandable format using OpenCSV libraries and SQLite queries. The parsed data is then assigned to Java Swing components likeJLists, JTextPane and JTables etc. to have a forensic report. Apart from this we have used Final Mobile forensic Tool to know the exact location of the attacker.

V RESULTS AND CONCLUSION

The Standard of Criminal Justice administration depends much on the exposure of Judges, Lawyers, Prosecutors and investigators to cyber-crimes jurisprudence. Section 1 the statement of the research problem explores the aims of research, research objectives and scope of the problem and the research methodology used.

Section II deals with attacks on mobile device systems. Section III. Explores the android forensic: experimentation & diagnostic research which present the deployment environment with respect to hardware and software forensic tools for mobile forensic. Section IV presents proposed mobile forensic software architecture, Data structure and algorithms design**.** Based on the mobile forensic life cycle, the optimized mobile forensic tool is proposed having three steps namely Initialization, Acquisition and Analysis of evidence with admissible forensic report into court of law.

ACKNOWLEDGMENT

We thank Mr. Dhanraj Wanjari former ACP, digital forensic expert and Dr. Ankush Dhanvijay IPS, crime expert, Dr. Prashant Lokhande, digital forensic expert about their inputs about the cybercrimes. Their valuable comments and suggestions have been very useful in enhancing the work and the ways it can be achieved. Thanks to Jeman educational society, Navi Mumbai for providing computing infrastructure for the conduction of the experiment.

REFERENCES

[1]M.P. Jain, Indian Constitutional 1776 (18 Ed. 2018), Lexisnexis (1962)
[2] The Indian Evidence Act, 1872
[3] The Code Of Criminal Procedure, 1973
[4] The Indian Penal Code 1860
[5] The Information Technology Act, 2000

[6]Bill Nelson et.al. Guide to Computer Forensics and investigations, Fourth Edition, Cenage Learning India Private Limited, Delhi 110092.

[7] Konstantia Barmpat salou, Tiago Cruz, et.al. Current and Future Trends in Mobile Device Forensics: A Survey, ACM Computing Surveys, Vol. xx, No. xx, Article 7. Publication date: January 2018

[8] C. R. Kothari, Gaurav Garg ,Research Methodology Methods And Techniques, New Age International Publisher, 2020

[9] Mobile ecosystems https://pages.nist.gov/mobile-threat-catalogue/background/mobile-attack-surface/mobile-ecosystem.html

[10] Platform Architecture, https://developer.android.com/guide/platform, 2021

[11] ANDROID SYSTEMS ARCHITECTURE (2/12/ 2022) HTTPS://WWW.GEEKSFORGEEKS.ORG/ANDROID-SYSTEM-ARCHITECTURE/

[12] Dean Wasil, Omar Nakhila, "Exposing Vulnerabilities in Mobile Networks: A Mobile Data Consumption Attack", IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems, 2017.

[13]Kevin Curran, Vivian Maynes and Declan Harkin, "Mobile device security", Inderscience Int. J. Information and Computer Security, Vol. 7, No. 1, 201.

[14].Milad Taleby Ahvanooey, Prof. Qianmu Li, Mahdi Rabbani, Ahmed Raza Rajput, "A Survey on Smartphones Security: Software Vulnerabilities, Malware, and Attacks", (IJACSA) International Journal of Advanced Computer Science and Applications Vol. 8, No. 10,30-45, 2017.

[15]Bahman Rashidi_and Carol Fung, "A Survey of Android Security Threats and Defenses", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 6, number: 3, pp. 3-35, 2015.

[18]. AASHIRVAD KUMAR, TOP 10 ATTACKS AND VULNERABILITIES OF OWASP MOBILE90(2022): https://detoxtechnologies.com/top-10-attacks-and-vulnerabilities-of-owasp-mobile/

[19]Aashirvad Kumar, Top 10 Attacks and Vulnerabilities of OWASP Mobile 2022 : https://detoxtechnologies.com/top-10-attacks-and-vulnerabilities-of-owasp-mobile/

[20]. Bao, Wenying, Wenbin Yao, Ming Zong, and Dongbin Wang. Cross-site Scripting Attacks on Android Hybrid Applications. In Proceedings of the 2017 International Conference on Cryptography, Security and Privacy, pp. 56-61. ACM, 2017

[21]H. Zhang, Z. Li, H. Shahriar, D. Lo, F. Wu and Y. Qian, Protecting Data in Android External Data Storage, 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019, pp. 924-925, DOI: 10.1109/COMPSAC.2019.00143. and Applications Conference (COMPSAC), 2019, pp. 924-925, DOI: 10.1109/COMPSAC.2019.00143.

[22].Palomba, F., Di Nucci, D., Panichella, A., Zaidman, A., and De Lucia, A. (2019). On the impact of code smells on the energy consumption of mobile applications. Information and Software Technology, 105:43–55.

[23].Rajan Thangaveloo and Wong Wang Jing DATDroid: Dynamic Analysis Technique in Android Malware Detection (March 2020) International Journal on Advanced Science Engineering and Information Technology

[24]Haoyu Ma, Shijia Li, Debin Gao, Daoyuan Wu, Qiaowen Jia, Chunfu Jia. Active Warden Attack: On the (In)Effectiveness of Android App Repackage-Proofing [IEEE 2021]

[25]Mobile Phones are Under Malware Attack:http://anti-virus-software-review.toptenreviews.com/mobile-phones-are-under-malware-attack.html

[26]Chen, L. Fan, C. Chen, M. Xue, Y. Liu and L. Xu, GUI-Squatting Attack: Automated Generation of Android Phishing Apps, in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 6, pp. 2551-2568, 1 Nov.-Dec. 2021, doi: 10.1109/TDSC.2019.2956035.

[27].S. Kaka, V. N. Sastry and R. R. Maiti, "On the MitM vulnerability in mobile banking applications for android devices," 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2016, pp. 1-6, doi: 10.1109/ANTS.2016.7947811.

[28]Rajan Thangaveloo and Wong Wang Jing DATDroid: Dynamic Analysis Technique in Android Malware Detection International Journal on Advanced Science Engineering and Information Technology, March 2020

[29]Mahindru, A., & Sangal, A. L. (2020). MLDroidâ€ framework for Android malware detection using machine learning techniques. Neural Computing and Applications, 33(10), 5183â€ 5240. doi:10.1007/s00521-020-05309-4

[30]Agrawal, P., & Trivedi, B. (2020). Machine Learning Classifiers for Android Malware Detection. Advances in Intelligent Systems and Computing, doi:10.1007/978-981-15-5616-6_22

[31]M. Koli, J. D. (2018). RanDroid: Android malware detection using random machine learning classifiers. In: International Conference on Technologies for Smart City Energy Security and Power (ICSESP) IEEE, Mar 2018.

[32] Zareen, A., & Baig, S. (2017). Notice of violation of IEEE publication principles: Mobile phone forensics: Challenges, analysis and tools classification. 2017 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering. https://doi.org/10.1109/sadfe.2010.24

[33].R. Umar, I. Riadi, and M. Zamroni, "Mobile Forensic Tools Evaluation for Digital Crime Investigation," Int. J. Adv. Sci. Eng. Technol., vol. 8, June, 2018. pp. 949–955,

[34] Imam Riadi, Ahmad Dahlan etal, A Study of Mobile Forensic Tools Evaluation on Android-Based LINE

Messenger, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 10, 2018 www.ijacsa.thesai.org

[35]XRY Logical Tool: https://www.msab.com/product/xry-extract/xry-logical/

[36]Oxygen Forensic Detective, Advanced Software To Extract Data From Multiple Sources: https://www.oxygen-forensic.com/en/

[37]Final Mobile ForensicsTool: http://www.mobileforensicscentral.com/, https://finaldata.com/

[38] Raj Chandel,Comprehensive Guide on FTK Imager, (November 6, 2020): https://www.hackingarticles.in/comprehensive-guide-on-ftk-imager/

[39]EnCase Forensic Tool : https://e-forensic.ca/products/encase-forensic-suite/

[40]Physical acquisition of a locked Android device https://www.digitalforensicscorp.com/blog/physical-acquisition-of-a-locked-android-device/

[41] SIM forensics: Extraction and preparation of digital evidence using Sim Xtractor. (2019). International Journal of Innovative Technology and Exploring Engineering, 9(2S2),868-870. https://doi.org/10.35940/ijitee.b1135.1292s219

[42].Practical mobile forensic approaches:https://subscription.packtpub.com/book/networking-and-servers/9781786464200/1/ch01lvl1sec12/practical-mobile-forensic-approaches

[43] Dimitar Kostadinov, The mobile forensics process: steps and types: (July 6, 2019): https://resources.infosecinstitute.com/topic/mobile-forensics-process-steps-types/

[44]Bommisetty, S., Tamma, R., & Mahalik, H. Practical mobile forensics. Packt Publishing. 2014.

[45]Hoog, *Android forensics: Investigation, analysis, and mobile security for Google Android*. Elsevier, . 2011.

[46]Skulkin, O., Tindall, D., & Tamma, R*Learning Android forensics: Analyze Android devices with the latest forensic tools and techniques* (2nd ed.). Packt Publishing. .2018.

[47]The Forensic Process Analysis of Mobile Device (2015) Dasari Manendra Sai et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, 4847-4850www.ijcsit.com4847

[48]. Zareen, A., & Baig, S. (2017). Notice of violation of IEEE publication principles: Mobile phone forensics: Challenges, analysis and tools classification. 2017 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering. https://doi.org/10.1109/sadfe.2010.24

[49]Android Terminal Emulator : https://github.com/jackpal/Android- Terminal-Emulator/wiki

[50]Remo Recovery tool for Android software from http://www.remosoftware.com/remo-recover-for-android

[51]. LiME : https://code.google.com/p/lime-forensics/.

[52] Open source Python scripts to parse the SQLite files for deleted records:http:// az4n6.blogspot.in/2013/11/python-parser-to-recover-deleted-sqlite. html