Name: Ghulam Abbas
Student ID: Fa22/BSDFCS/030
Section: A
Assignment submitted to: -
Ms. Fatima

Part A: Static Analysis of dd Image of An Android Mobile Using Autopsy

Part 2: Live Log Analysis of Your Own Mobile (Android) Using ADB and Logcat Command line Tools

---

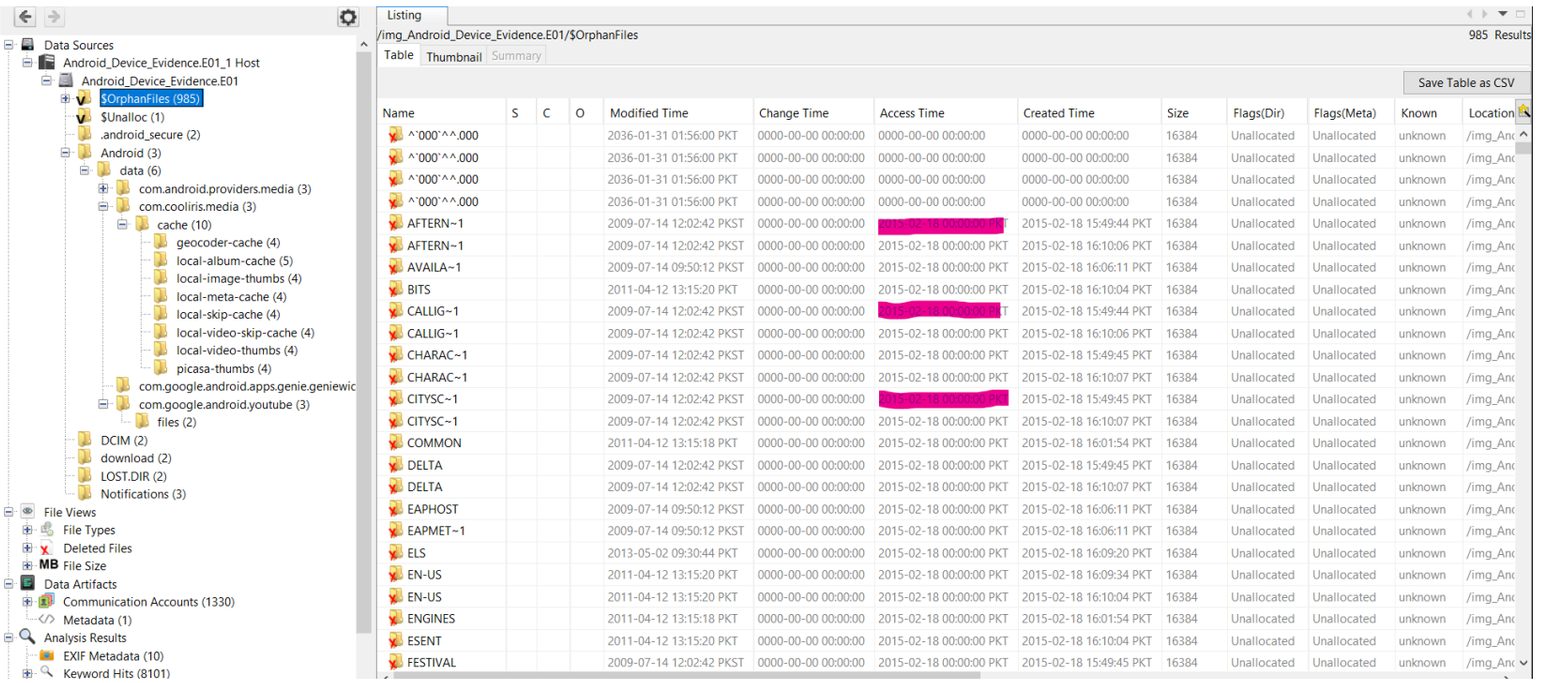
Suspicious Activity Observed in Orphan Files– Case-Based Analysis

During the forensic examination of the Android mobile image—linked to a suspect involved in the unauthorized purchase of shares using the firm's network—suspicious patterns were identified within the $OrphanFiles directory.

Key findings include:

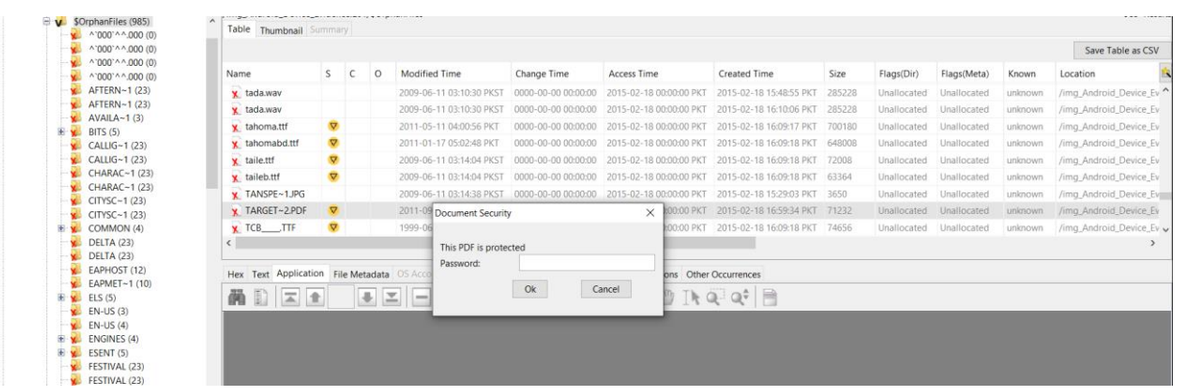
1. **Mass File Deletion Detected**
   A total of 985 orphaned (deleted) files were discovered. All these files were marked as unallocated, meaning they were no longer referenced in the file system and likely deleted. This suggests intentional data removal, possibly to eliminate incriminating evidence related to fraudulent stock transactions.
2. **Uniform File Timestamps**
   A majority of these files share identical or extremely close access and creation times, specifically around 2015-02-18 16:00–16:10 PKT.
   Such a clustered timestamp pattern is unnatural and strongly indicates a bulk operation, possibly a script or wiping tool used to delete multiple files in one go—potentially after the incident to conceal traces.
3. **Suspicious Modification Times**
   Several files show invalid or manipulated Modified timestamps—some with years like 2036, or 0000-00-00, which are either futuristic or null.
   These anomalies suggest timestamp tampering, likely to disrupt forensic timelines and mislead investigators.
4. **Consistent File Sizes (16384 bytes)**
   Numerous files, despite having different names, all have an identical size. This may imply: Use of encrypted containers Application of data padding/wiping tools or files intentionally filled with junk data to obfuscate true content
   Shortened Filenames (8.3 DOS Format). Many filenames follow the DOS-style 8.3 convention (e.g., AFTERN~1), a trait commonly seen in: Recovered fragments from deleted

files. Files from removable FAT-formatted storage (e.g., SD cards) This further confirms that the data was likely deleted and later carved or partially recovered.



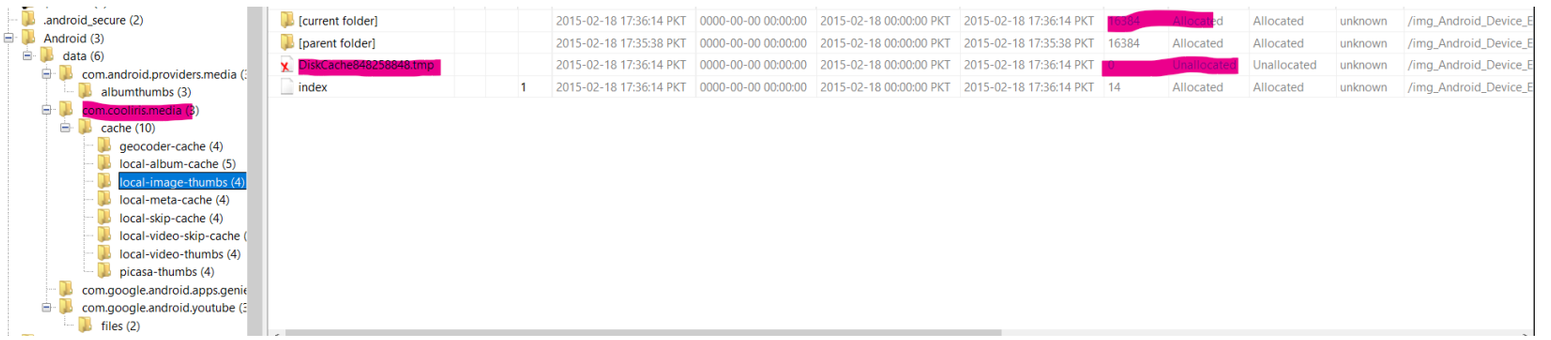
Suspicious Activity Observed in Orphan Files

Found a locked PDF file (TARGET~2.pdf). This PDF file may contain some valuable information for forensics investigation. Take pwd from mobile phone owner or take serach warrant from court for unlock device or folder.



Cache Directory and .tmp Files

Android/data/com.cooliris.media/cache



Deeper inspection, many of the .tmp files inside those folders were marked as *unallocated*—indicating that they had been deleted or no longer referenced by the file system.

When attempting to extract and open these .tmp files through Autopsy, the files appeared to be empty or unreadable, despite the folders showing allocated space.

Suspicious

1. **.tmp files are unallocated**
   These were likely deleted intentionally, possibly to remove evidence such as screenshots, cached media, or transaction records.
2. **Cache folders still exist**
   This suggests a partial wipe, where only content inside folders was deleted—not the folder itself, likely to make the deletion less obvious.
3. **Files show zero or junk data on export**
   Strong indication that data was either overwritten or securely wiped using a tool.
4. **Relevant App (com.cooliris.media)**
   This is a media-handling app, commonly caching screenshots, thumbnails, and media previews—exactly the kind of content that could visually confirm stock transactions.

Suspicious Activity Analysis – Office Document Artifacts

General Psychopathology.ppt, Mood disorders.ppt, Paraphilias.ppt

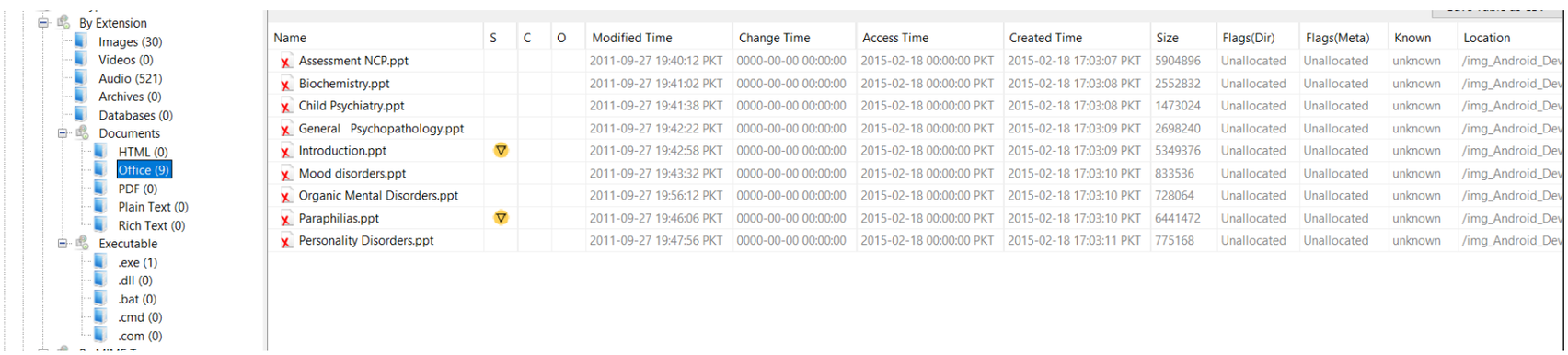
These files may relate to **psychological profiling**, possibly for manipulating targets or creating fake identities.

Child Psychiatry.ppt, Personality Disorders.ppt

Indicates knowledge or use of psychiatric materials, **not expected on a trader's device** unless used maliciously.

Assessment NCP.ppt, Biochemistry.ppt

If the user is not in the medical field, these are out of context. Could indicate **fake credentials or forged medical background**.



Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
Assessment NCP.ppt				2011-09-27 19:40:12 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:07 PKT	5904896	Unallocated	Unallocated	unknown	/img_Android_Dev
Biochemistry.ppt				2011-09-27 19:41:02 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:08 PKT	2552832	Unallocated	Unallocated	unknown	/img_Android_Dev
Child Psychiatry.ppt				2011-09-27 19:41:38 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:08 PKT	1473024	Unallocated	Unallocated	unknown	/img_Android_Dev
General Psychopathology.ppt				2011-09-27 19:42:22 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:09 PKT	2698240	Unallocated	Unallocated	unknown	/img_Android_Dev
Introduction.ppt				2011-09-27 19:42:58 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:09 PKT	5349376	Unallocated	Unallocated	unknown	/img_Android_Dev
Mood disorders.ppt				2011-09-27 19:43:32 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:10 PKT	833536	Unallocated	Unallocated	unknown	/img_Android_Dev
Organic Mental Disorders.ppt				2011-09-27 19:56:12 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:10 PKT	728064	Unallocated	Unallocated	unknown	/img_Android_Dev
Paraphilias.ppt				2011-09-27 19:46:06 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:10 PKT	6441472	Unallocated	Unallocated	unknown	/img_Android_Dev
Personality Disorders.ppt				2011-09-27 19:47:56 PKT	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 17:03:11 PKT	775168	Unallocated	Unallocated	unknown	/img_Android_Dev

Executable File in android phone

This file is found on an Android device image in your case.

Android doesn't run .exe files.

This file belongs on a Windows OS, not on a phone.

That raises a red flag—even if the file itself is legitimate, its presence on a mobile image is abnormal.

Malicious repackaging

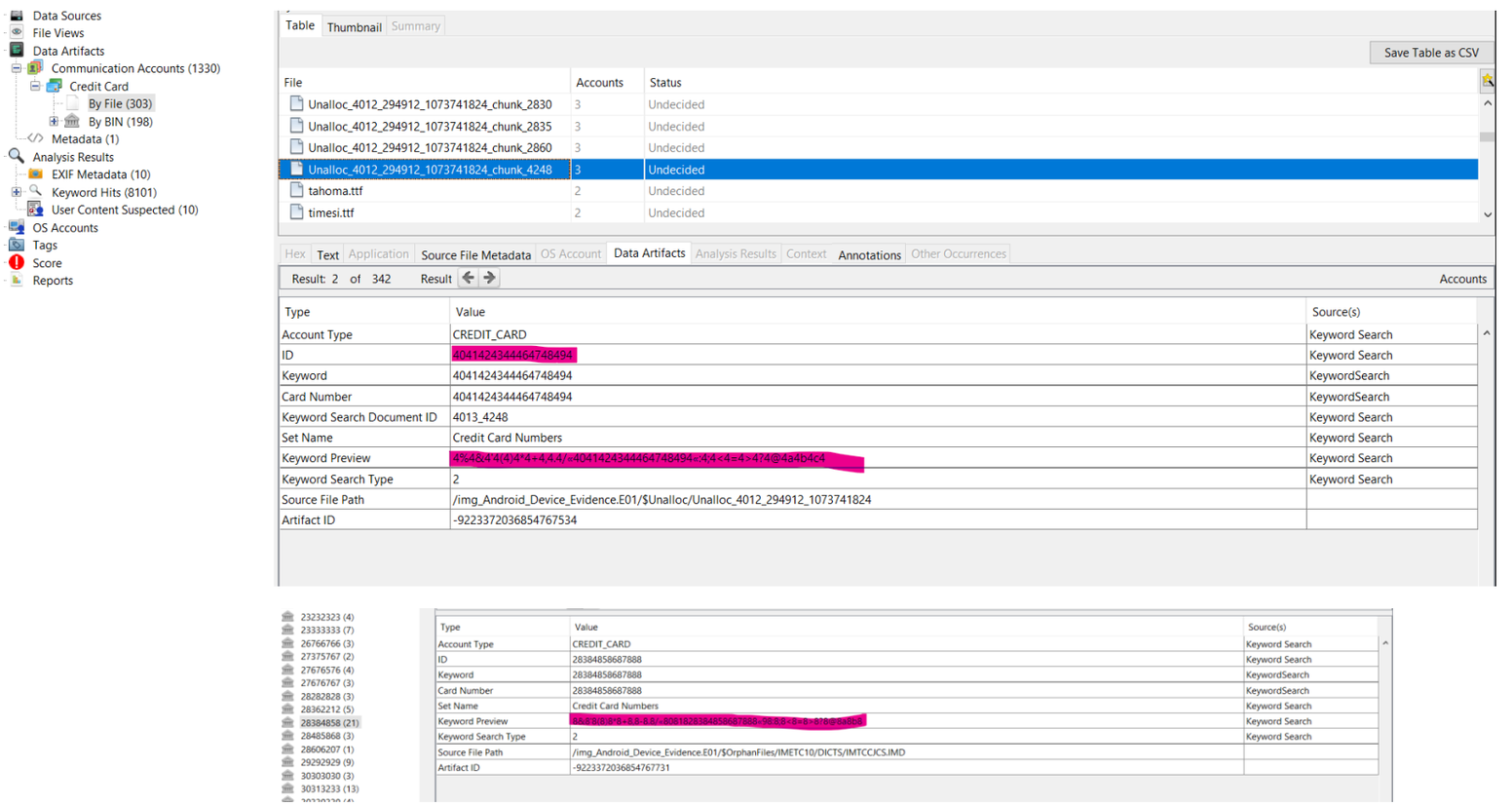
Malware authors often rename legitimate system files to hide payloads.

Highly suspicious unless hash-verified.

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
sapisvr.exe				2009-07-14 08:09:32 PKST	0000-00-00 00:00:00	2015-02-18 00:00:00 PKT	2015-02-18 16:01:54 PKT	44544	Unallocated	Unallocated	unknown	/img_Android_Device_Evidence.l

Suspicious Credit Card Entry Analysis

During the forensic analysis of the Android device image, a suspicious entry was recovered from unallocated space under the credit card keyword search. The value 4041424344464748 49 was identified and, upon decoding, it translated to @ABCDFGHI using ASCII interpretation. This pattern does not correspond to any valid credit card or account number format. Instead, it exhibits characteristics of **synthetically generated data**—commonly used by **carding tools**, **test scripts**, or **malware** to simulate account records. Furthermore, the entry was surrounded by repetitive symbol patterns (4%4&4'4(4)...), suggesting **intentional obfuscation** or **anti-forensics techniques**. Given its location in unallocated space and its artificial structure, this entry is classified as **suspicious**, potentially part of a card fraud scheme or memory-dump artifact from a malicious application.



All transactions showing fake / no valid credit card numbers of entries

In the forensic investigation of Catherine's case, multiple credit card-like entries such as 28384858687888 were found on the suspect's mobile device. Although this number is not valid, its presence indicates that the attacker may have used fake credit card numbers for testing purchase scripts, spoofing transaction records, or probing the financial system for weaknesses. Given that the mobile was used to commit unauthorized trades and contained signs of anti-forensics, these fake numbers may have been a key part of the criminal's technical setup or deception strategy.

This is not valid account type according to my search.

Type	Value	Source(s)
Account Type	CREDIT_CARD	Keyword Search
ID	203050608090	Keyword Search
Keyword	203050608090	KeywordSearch
Card Number	203050608090	KeywordSearch
Keyword Search Document ID	4013_2145	Keyword Search
Set Name	Credit Card Numbers	Keyword Search
Keyword Preview	%+37;agm\ bege !,/,«203050608090« !43468789/:/ !:"$#%	Keyword Search
Keyword Search Type	2	Keyword Search
Source File Path	/img_Android_Device_Evidence.E01/$Unalloc/Unalloc_4012_294912_1073741824	
Artifact ID	-9223372036854767468	

Analysis of IP addresses

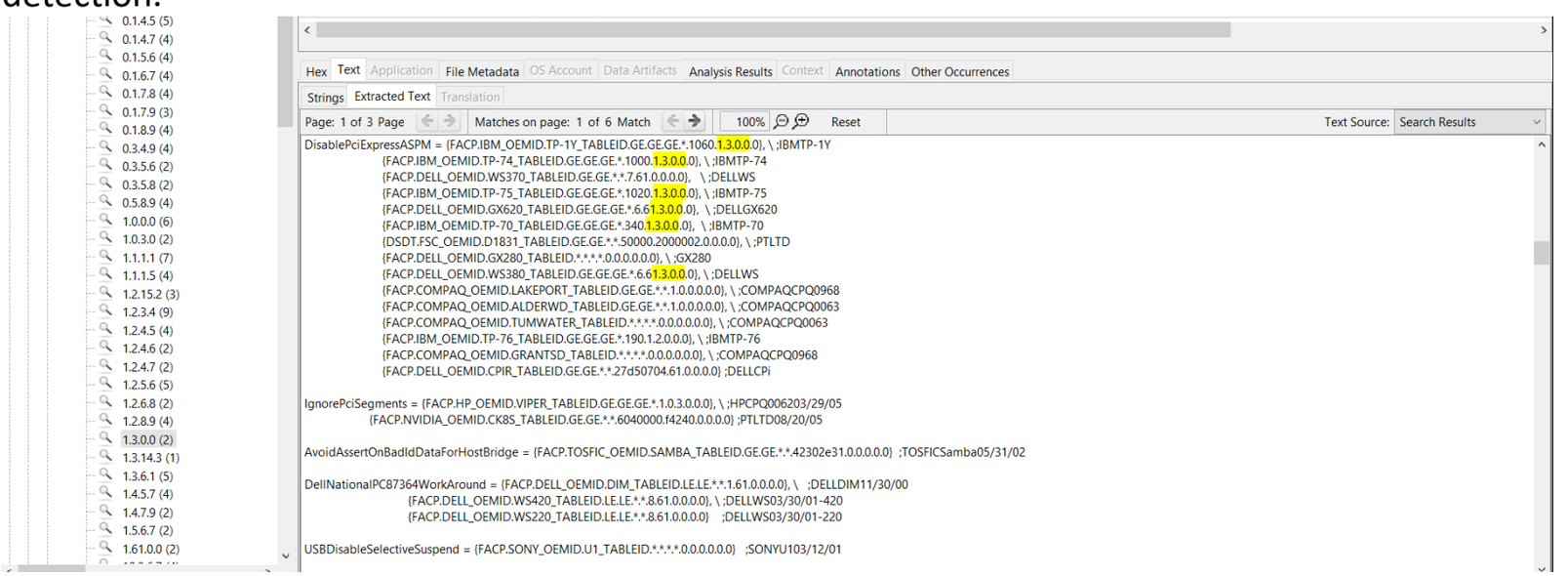
Standard IPv4 addresses only go up to 255.255.255.255.
These entries like 1.3.0.0.0, 6.6.1.3.0.0.0 are not valid IPs.
The format is likely not IP addresses — rather, these are structured hardware/platform version identifiers used in ACPI/firmware configs.

According to my approach

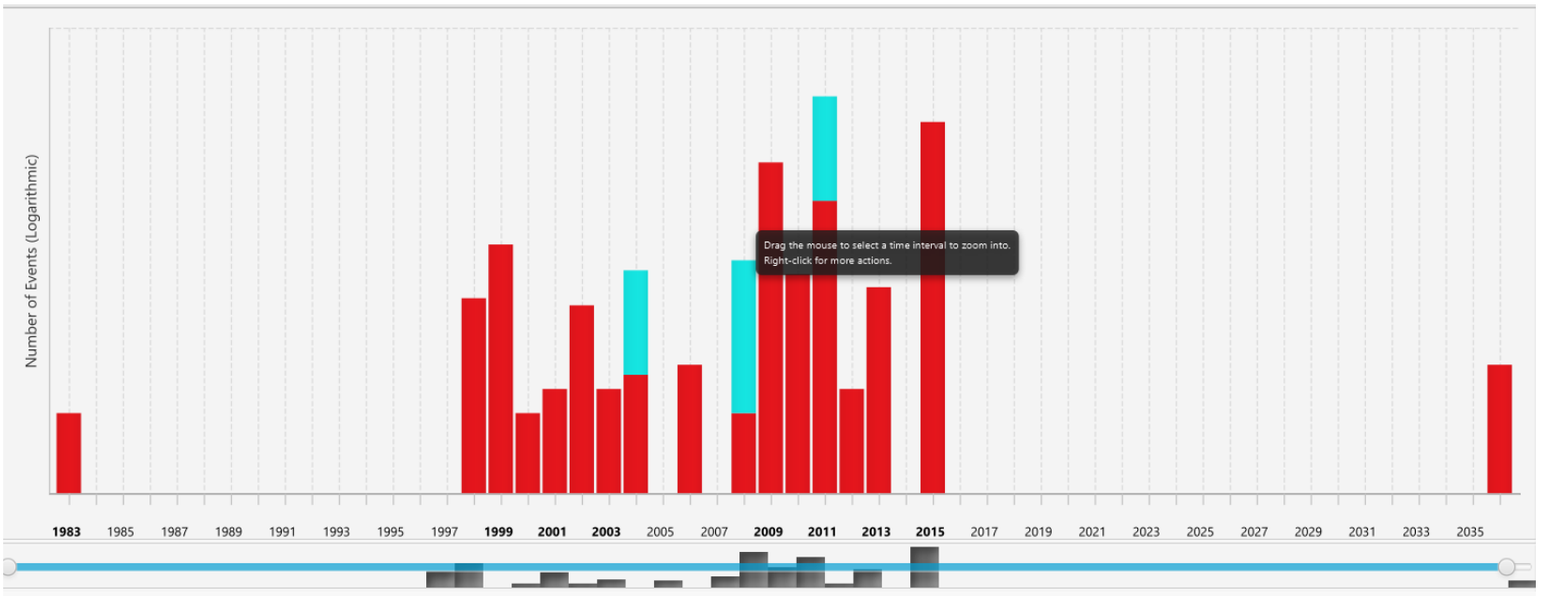
These entries do not contain valid IP addresses, but the structures like 1.3.0.0.0 resemble IP formats. They are ACPI or BIOS identifiers typically seen in Windows-based firmware, not Android. Their presence in a mobile forensic image suggests suspicious activity, possibly involving emulation, spoofing, or malware that includes or fakes PC platform traits. This may be relevant to case involving unauthorized transactions and hidden artifacts.

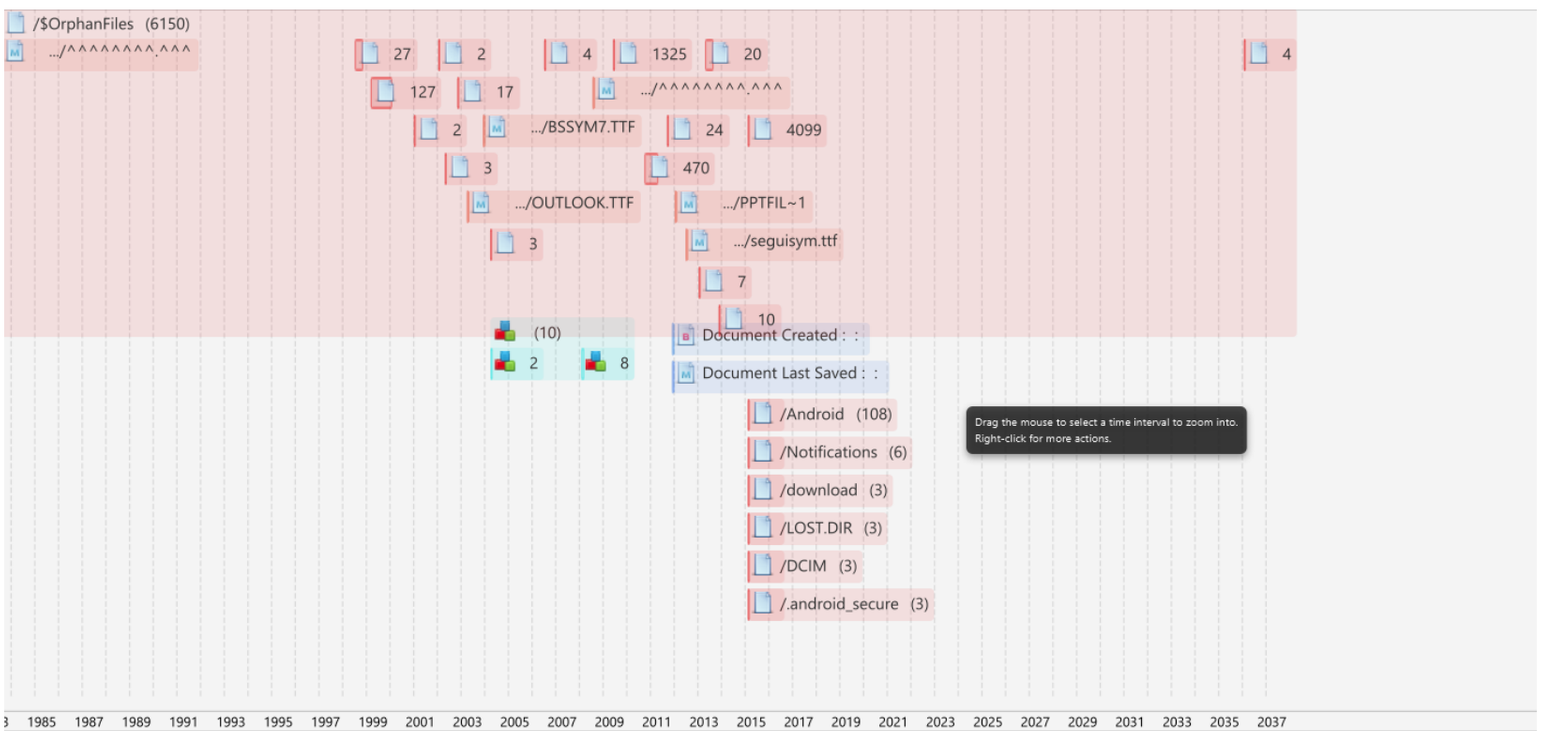
Device Spoofing or Virtual Environments

Some Android malware tools (e.g., test carders, payment spoofers) mimic PC environments.
These entries could be from spoofed BIOS tables, injected to fake system info or bypass detection.

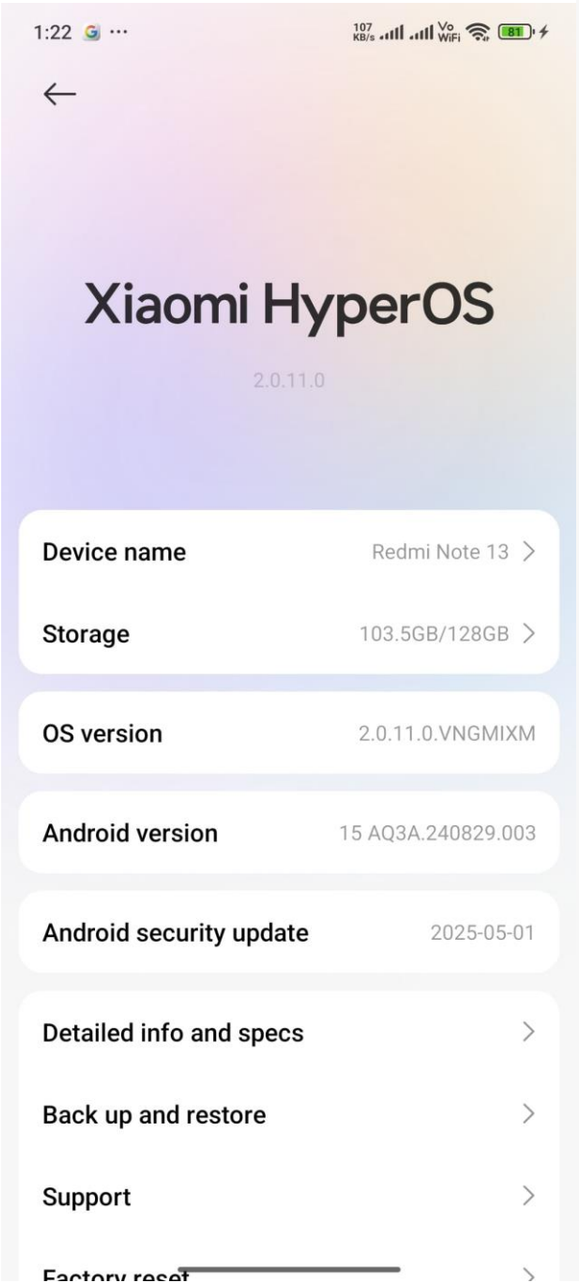


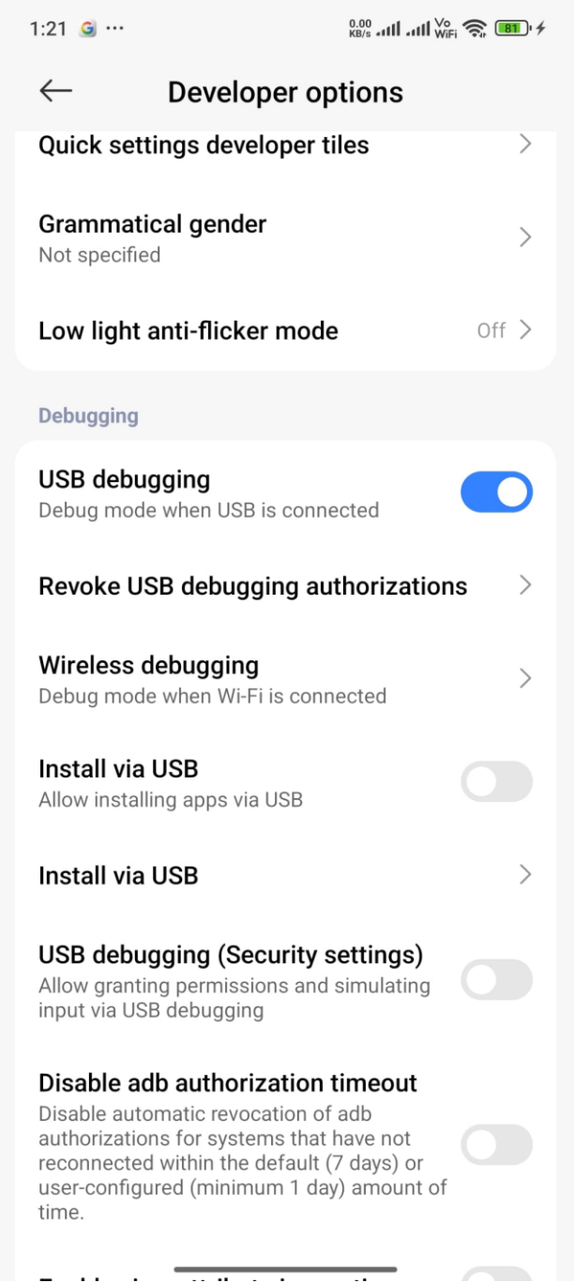
Timeline Analysis / Details of events





1. Enable your mobile ADB mode by following the previous method given in assignment 1
Step 2: Turn on USB Debugging through Developer Options on your Android device. If you don't see Developer Options in the settings, navigate to Settings > About phone, then tap the Build Number (or OS version) seven times to unlock it.

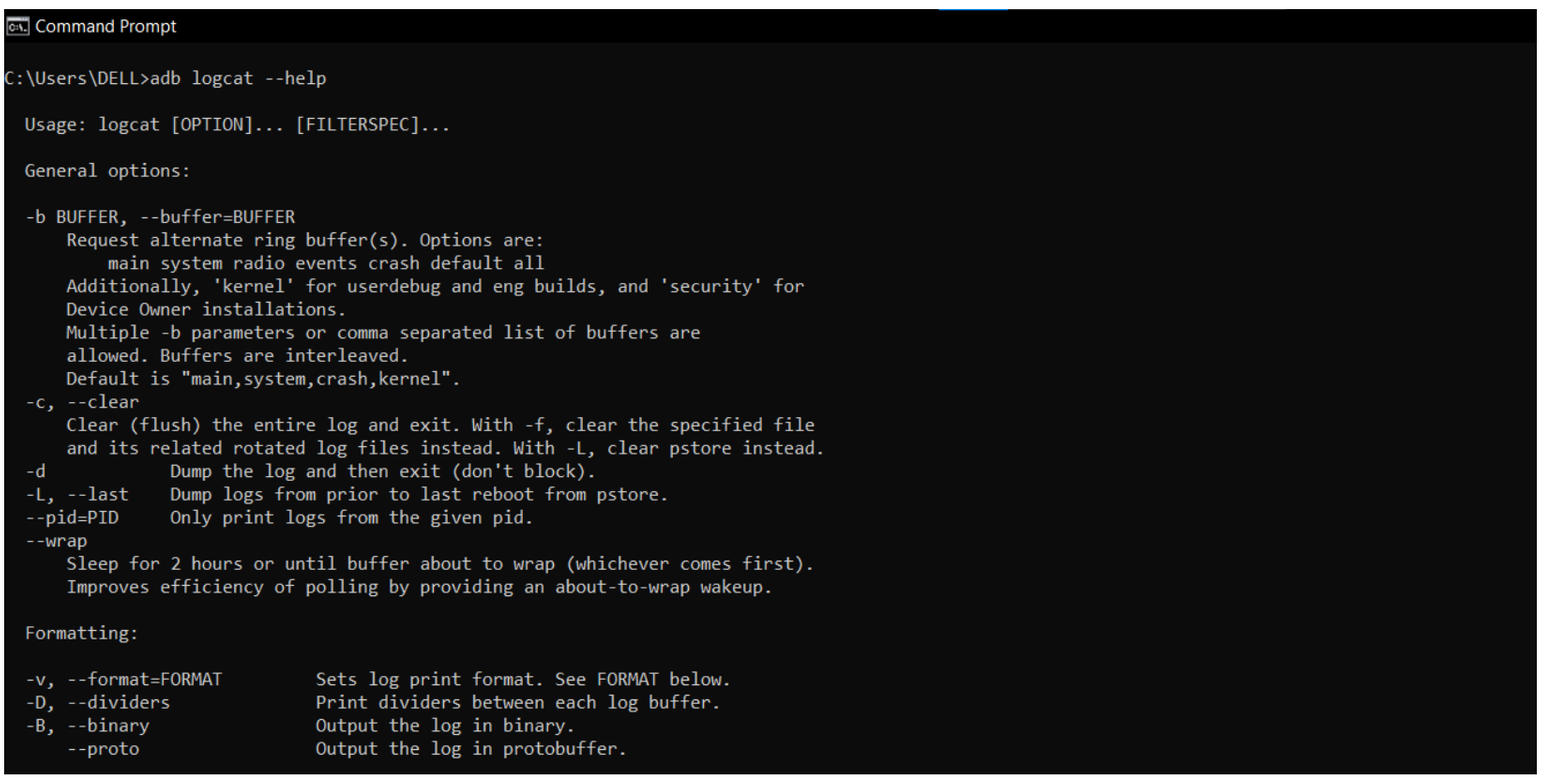




2. Download Logcat command line tool on your workstation

After on the debugging option on android phone connect the android device with workstation through USB. After that run command and ***adb devices / adb logcat*** for live log analysis in workstation command prompt to check device are given response or not.

Connect Use adb shell logcat –version / adb logcat –help to check the existence of logcat within adb tool



```
Command Prompt

C:\Users\DELL>adb logcat --help

  Usage: logcat [OPTION]... [FILTERSPEC]...

  General options:

  -b BUFFER, --buffer=BUFFER
      Request alternate ring buffer(s). Options are:
          main system radio events crash default all
      Additionally, 'kernel' for userdebug and eng builds, and 'security' for
      Device Owner installations.
      Multiple -b parameters or comma separated list of buffers are
      allowed. Buffers are interleaved.
      Default is "main,system,crash,kernel".
  -c, --clear
      Clear (flush) the entire log and exit. With -f, clear the specified file
      and its related rotated log files instead. With -L, clear pstore instead.
  -d            Dump the log and then exit (don't block).
  -L, --last    Dump logs from prior to last reboot from pstore.
  --pid=PID     Only print logs from the given pid.
  --wrap
      Sleep for 2 hours or until buffer about to wrap (whichever comes first).
      Improves efficiency of polling by providing an about-to-wrap wakeup.

  Formatting:

  -v, --format=FORMAT         Sets log print format. See FORMAT below.
  -D, --dividers              Print dividers between each log buffer.
  -B, --binary                Output the log in binary.
      --proto                 Output the log in protobuffer.
```

3. Now I am performing live log analysis by using logcat tool

After on the debugging option on android phone connect the android device with workstation through USB.

Running following logcat commands to analysis different categories logs of android device

I. **adb logcat:** Displays all types of Android logs

```
--------- beginning of system
06-16 15:32:38.693  4043  4250 I MiuiBubbleController: removeBubble: 772 reason=2
06-16 15:32:38.693  4043  4250 I MultiTaskingTaskRepository: onTaskInfoChanged multi list = [5, 4], full = [772]
06-16 15:32:38.695  8412  9133 D ActivityThread: handleBoundsCompatInfoChanged remove: name=com.whatsapp
06-16 15:32:38.696  2709  4303 D WindowManager: Set transition=TransitionRecord{6ebecb7 id=4061 type=CLOSE flags=0x0}, ready=true, SyncId=4061
06-16 15:32:38.697  2709  4303 D BiometricService: canAuthenticate: User=0, Caller=0, Authenticators=[redacted]
06-16 15:32:38.698  2709  4303 D BiometricService/PreAuthInfo: Package: com.whatsapp Sensor ID: 0 Modality: 2 Status: 1
06-16 15:32:38.698  2709  4303 D BiometricService/PreAuthInfo: getCanAuthenticateInternal Modality: 3 AuthenticatorStatus: 1
06-16 15:32:38.698  2709  4303 D AuthService: canAuthenticate, userId: 0, callingUserId: 0, authenticators: [redacted], result: 0
06-16 15:32:38.699  2709  2884 I BLASTSyncEngine: SyncGroup 4061:  Unfinished container:ActivityRecord{2cc6e67 u0 com.whatsapp/.conversationslist.LockedConversationsActivity t772}
06-16 15:32:38.699  2709  2884 I BLASTSyncEngine: isActivitySyncFinished isSyncFinished false ActivityRecord{2cc6e67 u0 com.whatsapp/.conversationslist.LockedConversationsActivity t772}
06-16 15:32:38.701  2709  4303 D BiometricService: canAuthenticate: User=0, Caller=0, Authenticators=33023
06-16 15:32:38.701  2709  4303 D BiometricService/PreAuthInfo: Package: com.whatsapp Sensor ID: 0 Modality: 2 Status: 1
06-16 15:32:38.702  2709  4303 D BiometricService/PreAuthInfo: getCanAuthenticateInternal Modality: 3 AuthenticatorStatus: 1
06-16 15:32:38.702  2709  4303 D AuthService: canAuthenticate, userId: 0, callingUserId: 0, authenticators: [redacted], result: 0
06-16 15:32:38.702  2709  5929 D BiometricService: canAuthenticate: User=0, Caller=0, Authenticators=[redacted]
06-16 15:32:38.702  2709  5929 D BiometricService/PreAuthInfo: Package: com.whatsapp Sensor ID: 0 Modality: 2 Status: 1
06-16 15:32:38.702  2709  5929 D BiometricService/PreAuthInfo: getCanAuthenticateInternal Modality: 3 AuthenticatorStatus: 1
06-16 15:32:38.702  2709  5929 D AuthService: canAuthenticate, userId: 0, callingUserId: 0, authenticators: 33023, result: 0
06-16 15:32:38.717  2709  5929 I BLASTSyncEngine: SyncGroup 4061:  Unfinished container:ActivityRecord[redacted] u0 com.whatsapp/.conversationslist.LockedConversationsActivity t772}
06-16 15:32:38.717  2709  5929 I BLASTSyncEngine: isActivitySyncFinished isSyncFinished false ActivityRecord[redacted] u0 com.whatsapp/.conversationslist.LockedConversationsActivity t772}
06-16 15:32:38.720  2709  5929 D WindowManagerService: onSecureChanged, current win = Window{7ca51ac u0 com.whatsapp/com.whatsapp.conversationslist.LockedConversationsActivity}, hasSecure = true
06-16 15:32:38.720  2709  5929 D WindowManager: wms.Input focus has changed to Window{7ca51ac u0 com.whatsapp/com.whatsapp.conversationslist.LockedConversationsActivity} display=0 updateInputWindows = true
06-16 15:32:38.720  2709  5929 I MultiSenceManagerInternalStub: getInstance
06-16 15:32:38.743  2709  5929 D WindowManager: wms.finishDrawingLocked: mDrawState=COMMIT_DRAW_PENDING Window{7ca51ac u0 com.whatsapp/com.whatsapp.conversationslist.LockedConversationsActivity} in Surface(name=
com.whatsapp/com.whatsapp.conversationslist.LockedConversationsActivity#61359)/@0x11dbc24
06-16 15:32:38.749  2709  2884 D miuiBarFollowAnimation:  BarFollowAnimation core services Flag is effective
06-16 15:32:38.750  2709  2884 D miuiElementAnimation:  ElementAnimation core services Flag is effective
06-16 15:32:38.751  2709  2884 I WindowManager: wms.showSurfaceRobustly mWin:Window{7ca51ac u0 com.whatsapp/com.whatsapp.conversationslist.LockedConversationsActivity}
06-16 15:32:38.752  2709  2884 D WindowManager: Final targets: [ActivityRecord[redacted] u0 com.whatsapp/.Conversation t772 f}}, ActivityRecord{2cc6e67 u0 com.whatsapp/.conversationslist.LockedConversationsActivit
y t772}]
06-16 15:32:38.755  2709  2884 D WindowManager: Calling onTransitionReady info={id=4061 t=CLOSE f=0x0 trk=0 opt={t=FROM_STYLE} r=[0@Point(0, 0)] c=[{null m=CLOSE f=IN_TASK_WITH_EMBEDDED_ACTIVITY|FILLS_TASK leash
=Surface(name=ActivityRecord{651455e u0 com.whatsapp/.Conversation t772}#[redacted] sb=Rect(0, 0 - 1080, 2400) eb=Rect(0, 0 - 1080, 2400) d=0 bc=fffafafa component=com.whatsapp/.Conversation},{null m=TO_FR
ONT f=IN_TASK_WITH_EMBEDDED_ACTIVITY|FILLS_TASK leash=Surface(name=ActivityRecord{2cc6e67 u0 com.whatsapp/.conversationslist.LockedConversationsActivity t772}#61336)/@0xf4d54bd sb=Rect(0, 0 - 1080, 2400) eb=Rect
(0, 0 - 1080, 2400) d=0 bc=fffafafa component=com.whatsapp/.conversationslist.LockedConversationsActivity}] mk=[false] noAni=[false] df=[false] rsa=[false] oa=[true]}, mToken=Token{[redacted]
b7 id=4061 type=CLOSE flags=0x0}}
06-16 15:32:38.760  4043  4250 D MiuiFreeformModeAnimation: startAnimation change: {null m=CLOSE f=IN_TASK_WITH_EMBEDDED_ACTIVITY|FILLS_TASK leash=Surface(name=ActivityRecord{651455e u0 com.whatsapp/.Conversatio
n t772}#61345)/@0x1a1d5bb sb=Rect(0, 0 - 1080, 2400) eb=Rect(0, 0 - 1080, 2400) d=0 bc=fffafafa component=com.whatsapp/.Conversation} info.getType(): 2
06-16 15:32:38.760  4043  4250 D MiuiFreeformModeAnimation: startAnimation change: {null m=TO_FRONT f=IN_TASK_WITH_EMBEDDED_ACTIVITY|FILLS_TASK leash=Surface(name=ActivityRecord{2cc6e67 u0 com.whatsapp/.conversa
tionslist.LockedConversationsActivity t772}#61336)/@0xaa92ad8 sb=Rect(0, 0 - 1080, 2400) eb=Rect(0, 0 - 1080, 2400) d=0 bc=fffafafa component=com.whatsapp/.conversationslist.LockedConversationsActivity} info.get
Type(): 2
06-16 15:32:38.760  4043  4250 D SoScStageCoordinator: inCloseTransition:false pairsStartedWithCloseTransition:false isRequestCloseTriggerTask:false inSingleSoScState:false inSoScUnactivie:true closeStage:null
06-16 15:32:38.761  4043  4250 D TransitionAnimationHelper:  overrideType == ANIM_FROM_STYLE
```

The screenshot shows **Android logcat logs** with system and app activity. Key events include:

- **WhatsApp Process Activity**: A misspelled package (com.whitsap) suggests WhatsApp-related operations, possibly background tasks or transitions.
- **Task Management**: MultiBodiesController logs indicate a task removal (reason code **772**), while MultiskleinflashRepository tracks task changes ([5, 4] → [772]).

- **Activity Transition**: Unimodemnger logs a transition with type **LOSE** (likely an app being backgrounded or closed).
- **UI/Usage Check**: BidUserInterface checks permissions for "Usage, Callers, AnthemInterests."
- **Corrupted Data**: A long string of zeros appears to be malformed binary data (not meaningful).

II. **adb logcat -v threadtime:** Displays all the Android logs with date and time

```
06-16 17:02:50.510  2709  2762 I Telecom : TelecomServiceImpl$1: isInEmergencyCall: false: TSI.iIEC@2XE
06-16 17:02:50.513  2709  2762 I Telecom : TelecomServiceImpl$1: isInEmergencyCall: false: TSI.iIEC@2XI
06-16 17:02:50.531  2709  5923 D CompatChangeReporter: Compat change id reported: 311208629; UID 10267; state: ENABLED
06-16 17:02:50.534  2709  2790 D DisplayManagerService: Drop pending events for gone uid 99644
06-16 17:02:50.542  2709  2881 D Aurogon :  uid = 10140 switch to BG
06-16 17:02:50.543  2709  3604 D Aurogon :  onForegroundActivitiesChanged packageName = com.miui.home
06-16 17:02:50.546  2709  4344 W ActivityManager: Unable to start service Intent { pkg=com.miui.securitycenter cmp=com.miui.securitycenter/com.miui.apppredict.service.AiService } U=0: not found
06-16 17:02:50.547  2709  2881 D Aurogon :  uid = 10140 switch to FG
06-16 17:02:50.553  2709  2709 D HyperNotificationManagerService: BroadcastReceiver ACTION_SCREEN_OFF
06-16 17:02:50.576  2709  3649 I MR2ServiceImpl: Updating composite discovery preference | preference: RouteDiscoveryRequest{ preferredFeatures={}, activeScan=false }, active routers: []
06-16 17:02:50.580  2709  5926 D Aurogon :  packageName = com.google.android.gms isAllowWakeUpList
06-16 17:02:50.588  2709  5926 D Aurogon :  packageName = com.google.android.gms isAllowWakeUpList
06-16 17:02:50.625  2709  4303 I Telecom : TelecomServiceImpl$1: isInEmergencyCall: false: TSI.iIEC@2XM
06-16 17:02:50.650  2709  2899 I ActivityManager: Start proc 7217:com.miui.msa.global:ui/u0a180 for service {com.miui.msa.global/com.xiaomi.ad.internal.splash.uiprocess.SplashUIService} caller=com.miui.msa.globa
l
06-16 17:02:50.659  2709  2898 I SmartPower: com.snapchat.android/10336(29918): idle->background(4450ms) R(broadcast start Intent { act=android.intent.action.SCREEN_OFF flg=0x50200010 }) adj=250.
06-16 17:02:50.676  2709  3604 D GreezeManager: THAW uid = 10336 pid = [ 29918 ]  reason : broadcast caller : 1000
06-16 17:02:50.679  2709  3604 I PowerManagerServiceImpl: Partial wakeLock:[PARTIAL_WAKE_LOCK                '*job*/com.snapchat.android/androidx.work.impl.background.systemjob.SystemJobService' DISABLED (uid=1
000 pid=2709 pkg=android ws=WorkSource{10336 com.snapchat.android})], disabled: false, procState: 0, reason: greeze
06-16 17:02:50.704 12550 12709 D MilletUtils: setUidState, uid = 10336 allow = true
06-16 17:02:50.708  2709  3906 I SmartPower: com.miui.msa.global:ui/10180(7217): died->background(33019ms) R(process start ) adj=-10000.
06-16 17:02:50.715  2709  5926 D Aurogon :  packageName = com.google.android.gms isAllowWakeUpList
06-16 17:02:50.722  7217  7235 D MiuiDownscaleImpl: set override inverted scale : 1.0
06-16 17:02:50.737  7217  7217 I ActivityThread: SetSystemFontMap done
06-16 17:02:50.743  2709  5923 D Aurogon :  packageName = com.google.android.gms isAllowWakeUpList
06-16 17:02:50.744  2709  2898 I SmartPower: com.google.android.as/10126(21334): idle->background(4364ms) R(broadcast start Intent { act=android.intent.action.DREAMING_STARTED flg=0x50000010 }) adj=100.
06-16 17:02:50.748  7217  7247 D ActivityThread: the event that target app starts is : AppLaunchEvent{mType='441', mProcessName='com.miui.msa.global:ui', mPackageName='com.miui.msa.global', mIsMainProcess='false
', mVersionCode='2025021700', mLaunchTime='2025-06-16'}
06-16 17:02:50.761  2709  2899 I ActivityManager: Start proc 7258:com.google.android.apps.wellbeing/u0a147 for broadcast {com.google.android.apps.wellbeing/com.google.android.apps.wellbeing.alarm.NextAlarmClockC
hangedReceiver_Receiver} caller=android
06-16 17:02:50.809  7217  7217 I ActivityThread: TrafficStats init done
06-16 17:02:50.810  2709  4344 I Telecom : TelecomServiceImpl$1: isInEmergencyCall: false: TSI.iIEC@2XQ
06-16 17:02:50.812  2709  4344 I SmartPower: com.google.android.apps.wellbeing/10147(7258): died->background(786ms) R(process start ) adj=-10000.
06-16 17:02:50.814  2709  5926 D CompatChangeReporter: Compat change id reported: 319212206; UID 10148; state: ENABLED
06-16 17:02:50.819  2709  3604 D isUidValid: uid = 10147
```

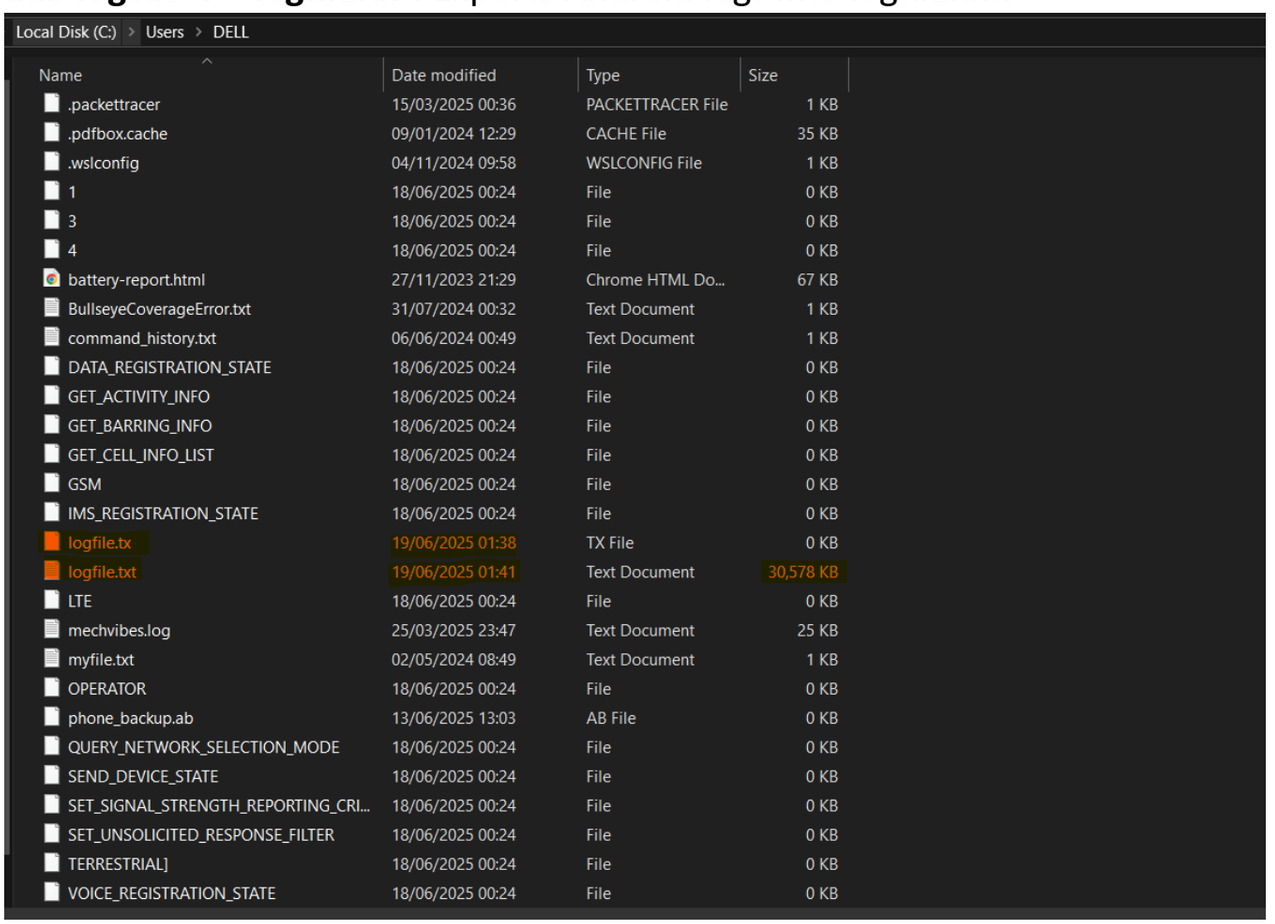
This logcat snapshot shows real-time system events on an Android device. Key highlights:

- TelecomServiceImpl repeatedly reports that the device is **not in an emergency call**.
- The system handles **foreground/background app transitions**, especially for apps like com.google.android.gms, com.miui.msa.global, and com.snapchat.android.
- Services like **MR2ServiceImpl** and **DownscaleImpl** are invoked, likely to manage power or display.

- A **wake lock** is acquired due to background processing by Snapchat's JobScheduler service.
- Several apps (including Snapchat and Mi apps) start/stop processes in response to **broadcast intents** like ACTION_SCREEN_OFF or DREAMING_STARTED.
- The log confirms that some services failed to start due to component errors (not found) or configuration (e.g., disabled).

Overall, this is a typical Android system reacting to screen state changes and background service execution involving system and third-party apps

III. **adb logcat -v > logfile.txt:** Exports Android logs into logfile.txt.

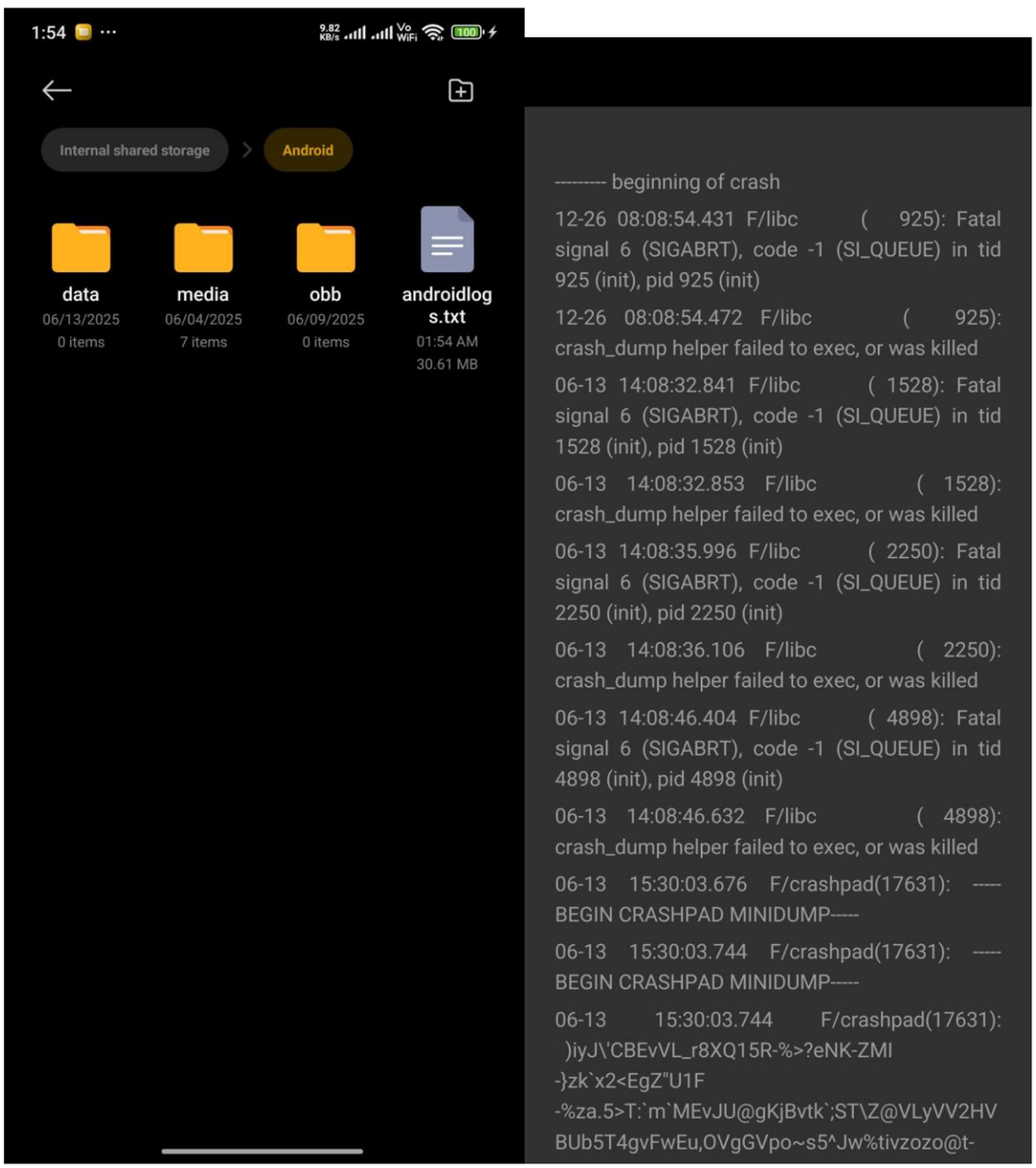


Local Disk (C:) > Users > DELL

Name	Date modified	Type	Size
.packettracer	15/03/2025 00:36	PACKETTRACER File	1 KB
.pdfbox.cache	09/01/2024 12:29	CACHE File	35 KB
.wslconfig	04/11/2024 09:58	WSLCONFIG File	1 KB
1	18/06/2025 00:24	File	0 KB
3	18/06/2025 00:24	File	0 KB
4	18/06/2025 00:24	File	0 KB
battery-report.html	27/11/2023 21:29	Chrome HTML Do...	67 KB
BullseyeCoverageError.txt	31/07/2024 00:32	Text Document	1 KB
command_history.txt	06/06/2024 00:49	Text Document	1 KB
DATA_REGISTRATION_STATE	18/06/2025 00:24	File	0 KB
GET_ACTIVITY_INFO	18/06/2025 00:24	File	0 KB
GET_BARRING_INFO	18/06/2025 00:24	File	0 KB
GET_CELL_INFO_LIST	18/06/2025 00:24	File	0 KB
GSM	18/06/2025 00:24	File	0 KB
IMS_REGISTRATION_STATE	18/06/2025 00:24	File	0 KB
logfile.tx	19/06/2025 01:38	TX File	0 KB
logfile.txt	19/06/2025 01:41	Text Document	30,578 KB
LTE	18/06/2025 00:24	File	0 KB
mechvibes.log	25/03/2025 23:47	Text Document	25 KB
myfile.txt	02/05/2024 08:49	Text Document	1 KB
OPERATOR	18/06/2025 00:24	File	0 KB
phone_backup.ab	13/06/2025 13:03	AB File	0 KB
QUERY_NETWORK_SELECTION_MODE	18/06/2025 00:24	File	0 KB
SEND_DEVICE_STATE	18/06/2025 00:24	File	0 KB
SET_SIGNAL_STRENGTH_REPORTING_CRI...	18/06/2025 00:24	File	0 KB
SET_UNSOLICITED_RESPONSE_FILTER	18/06/2025 00:24	File	0 KB
TERRESTRIAL]	18/06/2025 00:24	File	0 KB
VOICE_REGISTRATION_STATE	18/06/2025 00:24	File	0 KB

IV. **adb logcat -f <file_name> :** Saves all the Android logs into the android path's specified file.

```
C:\Users\DELL>adb logcat -v time -f /sdcard/Android/androidlogs.txt
^C
C:\Users\DELL>
C:\Users\DELL>
```



Purpose of command

- Captures all Android system logs (logcat)
- Adds timestamps (-v time)
- Saves logs in my Android device at:
- **Path: /sdcard/Android/androidlogs.txt**
- A text file androidlogs.txt is continuously written while the command runs.

V. **adb logcat "*:E":** Displays only error and fatal log messages.

```
--------- beginning of system
06-16 17:14:58.040  2709  3611 E ProcessManager: error write exception log
06-16 17:14:58.246 17109 17109 E MQSEventManagerDelegate: failed to get MQSService.
06-16 17:14:58.269 17109 17127 E MQSEventManagerDelegate: failed to get MQSService.
06-16 17:14:58.375 17109 17109 E ForceDarkHelperStubImpl: getForceDarkOriginState transact failed , mService: null
06-16 17:15:00.892  2709  3145 E HandwritingModeController: Cannot get requestId: Handwriting was not initialized.
06-16 17:15:02.108  2709  3650 E FingerprintProvider/defaultHIDL: initAuthStatistics
06-16 17:15:02.844  2709  3611 E ProcessManager: error write exception log
06-16 17:15:03.239  2709  2791 E FingerprintUnlockRateData: calculateUnlockCnt: appAuthMap doesn't contains:com.whatsapp
06-16 17:15:03.248  2709  5924 E VibratorManagerServiceImpl: vibrationThreadSetParameters
06-16 17:15:03.249  2709  5924 E VibratorManagerServiceImpl: skipjava.lang.NullPointerException: Attempt to invoke interface method 'void vendor.hardware.vibratorfeature.IVibratorExt.setUsageExt(int)' on a null
object reference
06-16 17:15:03.417  2709  3650 E AidlResponseHandler: Client monitor is not an instance of com.android.server.biometrics.sensors.AcquisitionClient
06-16 17:15:04.548  2709  6898 E TaskPersister: File error accessing recents directory (directory doesn't exist?).
06-16 17:15:05.788  2709  3650 E FingerprintProvider/defaultHIDL: initAuthStatistics
06-16 17:15:06.590  4064  4186 E ActivityThread: Failed to find provider info for com.miui.personalassistant.servicedeliver.system.provider
06-16 17:15:06.693  2709  5921 E VibratorManagerServiceImpl: vibrationThreadSetParameters
06-16 17:15:06.704  2709  2791 E FingerprintUnlockRateData: calculateUnlockCnt: appAuthMap doesn't contains:com.whatsapp
06-16 17:15:07.913  2709  6898 E TaskPersister: File error accessing recents directory (directory doesn't exist?).
06-16 17:15:08.746  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:08.746  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:08.746  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:08.746  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:08.746  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:11.343  2709  6898 E TaskPersister: File error accessing recents directory (directory doesn't exist?).
06-16 17:15:12.138  2709  3611 E ProcessManager: error write exception log
06-16 17:15:13.111  2709  3649 E MR2SystemProvider: Could not map this selected device attribute type to an available route: 1
06-16 17:15:13.155  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:13.155  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:13.155  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:13.155  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:13.155  2709  3086 E JobScheduler.Background: Couldn't determine stopped state for unknown package: com.google.android.apps.turbo
06-16 17:15:13.340  2709  3649 E MR2SystemProvider: Could not map this selected device attribute type to an available route: 1
06-16 17:15:13.872  2709  3830 E JobScheduler: ResourceBudgetManagerInternal is null
06-16 17:15:14.495  2709  3437 E NotificationVibratorHelper: Error creating vibration waveform with pattern: [0]
06-16 17:15:14.500  2709  3437 E NotificationVibratorHelper: Error creating vibration waveform with pattern: [0]
06-16 17:15:14.501  2709  3437 E NotificationVibratorHelper: Error creating vibration waveform with pattern: [0]
06-16 17:15:14.501  2709  3437 E NotificationVibratorHelper: Error creating vibration waveform with pattern: [0]
06-16 17:15:16.055  2709  3611 E ProcessManager: error write exception log
06-16 17:15:16.158  2709  3611 E ProcessManager: error write exception log
06-16 17:15:16.206  2709  3611 E ProcessManager: error write exception log
06-16 17:15:19.068  2709  3611 E ProcessManager: error write exception log
06-16 17:15:21.461  2709  3437 E NotificationVibratorHelper: Error creating vibration waveform with pattern: [0]
06-16 17:15:21.467  2709  3437 E NotificationVibratorHelper: Error creating vibration waveform with pattern: [0]
06-16 17:15:21.467  2709  3437 E NotificationVibratorHelper: Error creating vibration waveform with pattern: [0]
```

The command adb logcat "*:E" filters and displays only **error-level logs** from all system and app components. These logs include **crashes, exceptions, and critical failures**, helping developers or analysts troubleshoot issues quickly. It excludes all lower-level logs like warnings, info, and debug, making it useful for focusing on serious problems in the system or specific apps. **Use this when you're troubleshooting:** (App crashes, System failures, Security violations, Android app development issues)

VI. **adb logcat “application_or_tag_name:*” “*:S”:** Displays the Android logs of a given application

```
06-19 01:18:16.233 31314 31396 W chromium: [WARNING:net/cert/ev_root_ca_metadata.cc:119] Not implemented
06-19 01:18:17.496 31314 31314 W chromium: [WARNING:chrome/browser/android/compositor/compositor_view.cc:376] Child process died (type=6) pid=17257)
06-19 01:18:17.496 31314 31314 W chromium: [WARNING:chrome/browser/android/compositor/compositor_view.cc:376] Child process died (type=6) pid=17257)
06-19 01:18:20.184 31314 31314 I MIUIInput: [MotionEvent] ViewRootImpl windowName 'com.android.chrome/org.chromium.chrome.browser.customtabs.CustomTabActivity', { action=ACTION_DOWN, id[0]=0, pointerCount=1, eve
ntTime=256455476, downTime=256455476, phoneEventTime=01:18:20.179 } moveCount:0
06-19 01:18:20.448 31314 31314 I MIUIInput: [MotionEvent] ViewRootImpl windowName 'com.android.chrome/org.chromium.chrome.browser.customtabs.CustomTabActivity', { action=ACTION_UP, id[0]=0, pointerCount=1, event
Time=256455744, downTime=256455476, phoneEventTime=01:18:20.447 } moveCount:60
06-19 01:18:20.939 31314 31314 W cr_BotControlsStacker: Using mTotalHeight before initialization
06-19 01:18:20.946 31314 31396 W chromium: [WARNING:net/cert/ev_root_ca_metadata.cc:119] Not implemented
06-19 01:18:20.959 31314 31327 E JavaBinder: *** Uncaught remote exception! Exceptions are not yet supported across processes. Client PID 5669 UID 10156.
06-19 01:18:20.959 31314 31327 E JavaBinder: java.lang.RuntimeException: java.lang.ClassCastException: android.view.ViewStub cannot be cast to android.widget.ImageButton
06-19 01:18:20.959 31314 31327 E JavaBinder:    at org.chromium.base.task.PostTask.e(chromium-TrichromeChromeGoogle6432.aab-stable-715108933:21)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at org.chromium.chrome.browser.customtabs.CustomTabsConnection.M(chromium-TrichromeChromeGoogle6432.aab-stable-715108933:233)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at ZU0.q(chromium-TrichromeChromeGoogle6432.aab-stable-715108933:11)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at hV0.onTransact(chromium-TrichromeChromeGoogle6432.aab-stable-715108933:958)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at android.os.Binder.execTransactInternal(Binder.java:1507)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at android.os.Binder.execTransact(Binder.java:1451)
06-19 01:18:20.959 31314 31327 E JavaBinder: Caused by: java.lang.ClassCastException: android.view.ViewStub cannot be cast to android.widget.ImageButton
06-19 01:18:20.959 31314 31327 E JavaBinder:    at org.chromium.chrome.browser.customtabs.features.toolbar.CustomTabToolbar.a0(chromium-TrichromeChromeGoogle6432.aab-stable-715108933:14)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at TU0.call(chromium-TrichromeChromeGoogle6432.aab-stable-715108933:316)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at java.util.concurrent.FutureTask.run(FutureTask.java:264)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at org.chromium.base.task.TaskRunnerImpl.runTask(chromium-TrichromeChromeGoogle6432.aab-stable-715108933:31)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at android.os.MessageQueue.nativePollOnce(Native Method)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at android.os.MessageQueue.next(MessageQueue.java:355)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at android.os.Looper.loopOnce(Looper.java:203)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at android.os.Looper.loop(Looper.java:337)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at android.app.ActivityThread.main(ActivityThread.java:9593)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at java.lang.reflect.Method.invoke(Native Method)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:593)
06-19 01:18:20.959 31314 31327 E JavaBinder:    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:936)
06-19 01:18:21.329 31314 31314 D UserSceneDetector: invoke error.
06-19 01:18:21.481 31314 31314 I cr_E2E_DCController: setSafeAreaConstraint: false
06-19 01:18:21.561 31314 31408 W chromium: [WARNING:net/spdy/spdy_session.cc:3009] Received WINDOW_UPDATE for invalid stream 1
06-19 01:18:21.853 31314 31408 W chromium: [WARNING:net/spdy/spdy_session.cc:3009] Received WINDOW_UPDATE for invalid stream 1
06-19 01:18:21.870 31314 31408 W chromium: [WARNING:net/spdy/spdy_session.cc:3009] Received WINDOW_UPDATE for invalid stream 1
06-19 01:18:22.864 31314 31397 W chromium: [WARNING:net/cert/ev_root_ca_metadata.cc:119] Not implemented
06-19 01:18:23.694 31314 17281 W adservices: To enable debug api, include ACCESS_ADSERVICES_AD_ID permission and enable advertising ID under device settings
```

adb logcat --pid=31314

running: adb logcat --pid=31314

This command shows only logs **from the Chrome process** (PID 31314). The screenshot captures real-time output of that command.

What's Happening in the Screenshot

1. Warning & Info Logs from Chromium

Example:

W chromium: [WARNING:net/cert/ev_root_ca_metadata.cc:119] Not implemented

- These are **Chrome internal network/security logs**
- This one is a **warning about missing certificate metadata** support
- Not a problem for normal users — just developer diagnostics

2. User Input Detected (Touch Events)

Example:

I MIUIInput: [MotionEvent] ViewRootImpl windowName='com.android.chrome...CustomTabActivity'

- Chrome received **touch input** (ACTION_DOWN / ACTION_UP)
- Indicates user **tapped on a custom tab or link**

---

3. Runtime Crash — Java Exception

E JavaBinder: java.lang.ClassCastException: android.view.ViewStub cannot be cast to android.widget.ImageButton

- This is a **runtime error** caused by bad code in Chrome or a custom tab extension
- It's trying to cast a ViewStub (placeholder UI element) as an ImageButton — which fails
- It shows the exact class and method:
- at org.chromium.chrome.browser.customtabs.features.toolbar.CustomTabToolbar

---

4. Thread Execution / Stack Trace

- All the lines after the error are **stack traces** showing the internal Chromium/Android classes where the crash propagated
- This is **very useful for debugging or reverse engineering**

---

5. Performance Monitor Warnings

W MessageMonitor: PerfMonitor: Slow Operation...

- Chrome's **UI thread** took too long to pause/start activity
- Indicates **performance lag**, possibly due to the crash

Types of Data Shown (from adb logcat --pid=31314)

Type	Description
W (Warning)	Non-fatal issues like unimplemented features or slow performance
E (Error)	Fatal exceptions or crashes (e.g. ClassCastException)
I (Info)	Informational logs like touch input and Chrome state
D (Debug)	Optional diagnostics (not visible in this part but included in logcat)

VII. **adb logcat -b events “gsm_service_state_change” “*:S”**

```
C:\Users\DELL>adb logcat -b events "gsm_service_state_change" "*:S"
* daemon not running; starting now at tcp:5037
* daemon started successfully
--------- beginning of events
06-18 21:45:41.760  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 21:45:41.868  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 21:55:30.793  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 21:55:30.881  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 22:36:04.575  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 22:36:04.669  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 22:55:37.777  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 22:55:38.003  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 22:58:00.833  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 22:58:01.054  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 23:39:48.553  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 23:39:48.687  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 23:57:13.722  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 23:57:13.929  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 23:57:18.392  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-18 23:57:18.601  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:04:26.298  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:04:27.061  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:04:30.054  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:04:30.091  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:06:45.268  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:06:45.425  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:06:49.214  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:06:49.357  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:07:10.728  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:07:11.853  4028  4028 I gsm_service_state_change: [0,0,0,0]
06-19 00:09:17.831  4028  4028 I gsm_service_state_change: [0,0,0,1]
06-19 00:09:39.386  4028  4028 I gsm_service_state_change: [0,1,0,0]
06-19 00:10:01.297  4028  4028 I gsm_service_state_change: [0,0,0,1]
06-19 00:10:12.064  4028  4028 I gsm_service_state_change: [0,1,0,0]
^C
```

The gsm_service_state_change entries in your logcat output represent changes in the **GSM network service state** on your Android device. These logs are generated by the telephony subsystem and indicate different states of cellular connectivity.

Reason why and how the signals show a variation?

Before, During & After Call Signal Changes

Before Call (Idle State)

- **Signal:** [0,0,0,0]

- **Meaning:**
  - **Voice:** In service (0)
  - **Data:** Disconnected (0)
  - **No Roaming** (0,0)

During Call (Active Call)

- **Signal:** [0,0,0,1]
- **Meaning:**
  - **Voice still active (**0**)**
  - **Data roaming activated (**1**)**
- **Possible Reasons:**
  - Your phone **switched to a roaming network** temporarily for the call (common in 2G/3G fallback).
  - Carrier might have **network handover issues**.

After Call (Terminated Call)

- **Signal:** [0,1,0,0]
- **Meaning:**
  - **Voice normal (**0**)**
  - **Data reconnected (**1**)**
  - **Back to home network (**0,0**)**

Why Did [0,0,0,1] **Happen?**

Non-VoLTE Call (2G/3G Fallback):

- If your phone **doesn't support VoLTE**, it must switch to older networks (2G/3G) for calls.
- Some carriers **roam temporarily** during this switch.

Network Handover Glitch:

- A weak signal may force your phone to **roam briefly** before stabilizing.

Carrier-Specific Behavior:

- Some carriers **share towers**, causing false roaming flags.

VIII. **Adb logcat -b radio:** Display all radio events

This log shows network-related events:

- Your device is connected to Telenor LTE with good signal.
- It’s checking VoLTE feature support (and supports Dual VoLTE).
- Time zone and SIM details are being parsed.
- Google Services are requesting cell tower information.
- There are no errors, and logs indicate normal modem and radio activity.

```
C:\Users\DELL>adb logcat -b radio
--------- beginning of radio
06-20 12:16:34.067  1612  2581 D RequestManager: [RequestManager.cpp: 358] [Loc_hal_worker(1612,2581)] listenForResponses: Got a response parcel
06-20 12:16:34.145  4028  4028 I BatteryStateManager: BatteryStateManager onReceive action android.intent.action.BATTERY_CHANGED
06-20 12:16:44.126  4028  4028 D MiuiNetworkControllerImpl[0]: request cell info list without optimization: cts
06-20 12:16:44.126  4028  4028 D MiuiNetworkControllerImpl[0]: processCacheCellInfoList is empty function,return false
06-20 12:16:44.128  4028  4028 D RILJ    : [0070]> GET_CELL_INFO_LIST [PHONE0]
06-20 12:16:44.148  4028  4028 D SST     : [0] handleMessage: received event 43
06-20 12:16:44.149  4028  4325 D RILJ    : [0070]< GET_CELL_INFO_LIST [CellInfoLte:{mRegistered=YES mTimeStamp=598219467116383ns mCellConnectionStatus=0 CellIdentityLte:{ mBandwidth=2147483647 mMcc=410 mMnc=
AlphaLong=Telenor mAlphaShort=Telenor mAdditionalPlmns={} mCsgInfo=null} CellSignalStrengthLte: rssi=-51 rsrp=-78 rsrq=-14 rssnr=2147483647 cqiTableIndex=2147483647 cqi=2147483647 ta=2147483647 miuiLevel=0 
mizedLevel=0 level=4 parametersUseForLevel=1 android.telephony.CellConfigLte :{ isEndcAvailable = false }}, CellInfoLte:{mRegistered=NO mTimeStamp=598219467116383ns mCellConnectionStatus=0 CellIdentityLte:{
dwidth=2147483647 mMcc=null mMnc=null mAlphaLong= mAlphaShort= mAdditionalPlmns={} mCsgInfo=null} CellSignalStrengthLte: rssi=-55 rsrp=-85 rsrq=-18 rssnr=2147483647 cqiTableIndex=2147483647 cqi=2147483647 t
7483647 miuiLevel=0 mOptimizedLevel=0 level=4 parametersUseForLevel=1 android.telephony.CellConfigLte :{ isEndcAvailable = false }}] [PHONE0]
06-20 12:16:44.149  4028  4028 D LocaleTracker-0: processCellInfo: cell info=[CellInfoLte:{mRegistered=YES mTimeStamp=598219467116383ns mCellConnectionStatus=0 CellIdentityLte:{ mBandwidth=2147483647 mMcc=4
nc=06 mAlphaLong=Telenor mAlphaShort=Telenor mAdditionalPlmns={} mCsgInfo=null} CellSignalStrengthLte: rssi=-51 rsrp=-78 rsrq=-14 rssnr=2147483647 cqiTableIndex=2147483647 cqi=2147483647 ta=2147483647 miuiL
0 mOptimizedLevel=0 level=4 parametersUseForLevel=1 android.telephony.CellConfigLte :{ isEndcAvailable = false }}, CellInfoLte:{mRegistered=NO mTimeStamp=598219467116383ns mCellConnectionStatus=0 CellIdenti
:{ mBandwidth=2147483647 mMcc=null mMnc=null mAlphaLong= mAlphaShort= mAdditionalPlmns={} mCsgInfo=null} CellSignalStrengthLte: rssi=-55 rsrp=-85 rsrq=-18 rssnr=2147483647 cqiTableIndex=2147483647 cqi=21474
 ta=2147483647 miuiLevel=0 mOptimizedLevel=0 level=4 parametersUseForLevel=1 android.telephony.CellConfigLte :{ isEndcAvailable = false }}]
06-20 12:16:44.150  4028  4028 D LocaleTracker-0: updateLocale: countryIso = pk, countryIsoDebugInfo = OperatorNumeric(41006): MccTable.geoCountryCodeForMccMnc("MccMnc{mcc='410', mnc='06'}")
06-20 12:16:44.150  4028  4028 D LocaleTracker-0: updateLocale: timeZoneCountryIso = pk, timeZoneCountryIsoDebugInfo = OperatorNumeric(41006): MccTable.geoCountryCodeForMccMnc("MccMnc{mcc='410', mnc='06'}")
06-20 12:16:44.150  4028  4028 D NitzStateMachineImpl: handleCountryDetected: countryIsoCode=pk, mLatestNitzSignal=null
06-20 12:16:44.152  4028  4028 D MiuiPhoneApp: recordResourceAccessStatistics callingPackage=com.google.android.gms event=event_request_cell_info_list state=SUCCESS
06-20 12:16:46.130  4028  4028 D SST     : [0] handleMessage: received event 43
06-20 12:16:50.612  4028  4028 I BatteryStateManager: BatteryStateManager onReceive action android.intent.action.BATTERY_CHANGED
06-20 12:16:53.480  4028  4028 D QtiDSMGR-1: handleMessage: 5
06-20 12:16:53.481  4028  4028 D QtiDSMGR-0: handleMessage: 5
06-20 12:17:01.472  4028  4028 D QtiDSMGR-1: handleMessage: 5
06-20 12:17:01.473  4028  4028 D QtiDSMGR-0: handleMessage: 5
06-20 12:17:04.807  4028  4325 D FeatureConfiguration: Checking feature: FEATURE_DUAL_VOLTE
06-20 12:17:04.807  4028  4325 D FeatureConfiguration: Feature FEATURE_DUAL_VOLTE build is supported, featureSupported=true
06-20 12:17:04.811  4028  4325 D ImsManagerIM [0]: getLocalImsConfigKeInt() for key:68, result: 0
06-20 12:17:04.817  4028  4325 D FeatureConfiguration: Checking feature: FEATURE_DUAL_VOLTE
06-20 12:17:04.817  4028  4325 D FeatureConfiguration: Feature FEATURE_DUAL_VOLTE build is supported, featureSupported=true
06-20 12:17:04.817  4028  4325 D ImsManagerIM [0]: getLocalImsConfigKeInt() for key:68, result: 0
06-20 12:17:05.289  4028  4325 D FeatureConfiguration: Checking feature: FEATURE_DUAL_VOLTE
06-20 12:17:05.289  4028  4325 D FeatureConfiguration: Feature FEATURE_DUAL_VOLTE build is supported, featureSupported=true
06-20 12:17:05.289  4028  4325 D ImsManagerIM [0]: getLocalImsConfigKeInt() for key:68, result: 0
06-20 12:17:05.297  4028  4325 D FeatureConfiguration: Checking feature: FEATURE_DUAL_VOLTE
06-20 12:17:05.297  4028  4325 D FeatureConfiguration: Feature FEATURE_DUAL_VOLTE build is supported, featureSupported=true
06-20 12:17:05.297  4028  4325 D ImsManagerIM [0]: getLocalImsConfigKeInt() for key:68, result: 0
06-20 12:17:11.495  4028  4028 D QtiDSMGR-1: handleMessage: 5
06-20 12:17:11.496  4028  4028 D QtiDSMGR-0: handleMessage: 5
06-20 12:17:25.524  4028  4028 D QtiDSMGR-1: handleMessage: 5
06-20 12:17:25.524  4028  4028 D QtiDSMGR-0: handleMessage: 5
06-20 12:17:29.355  2360  2654 E RILQ    : =RILM= dsiGetVerboseCallEndReason type= 6 ,code= 120
06-20 12:17:48.727  4028  4028 D QtiDSMGR-1: handleMessage: 5
06-20 12:17:48.727  4028  4028 D QtiDSMGR-0: handleMessage: 5
```