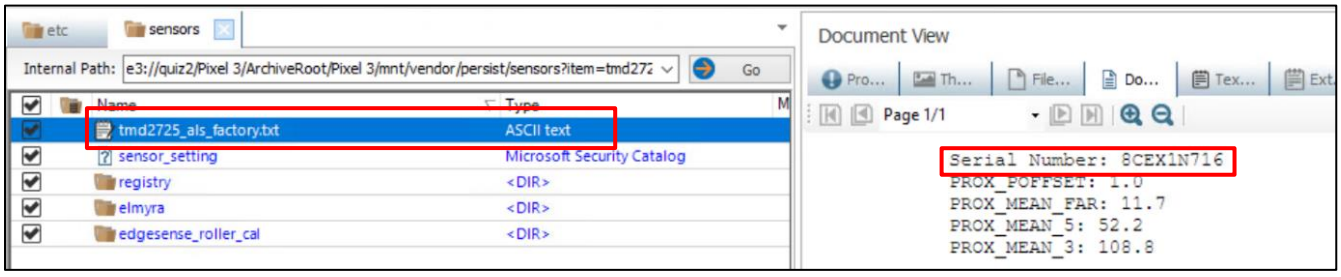
**CYFI 330 – 700 Quiz (1): Android 10**

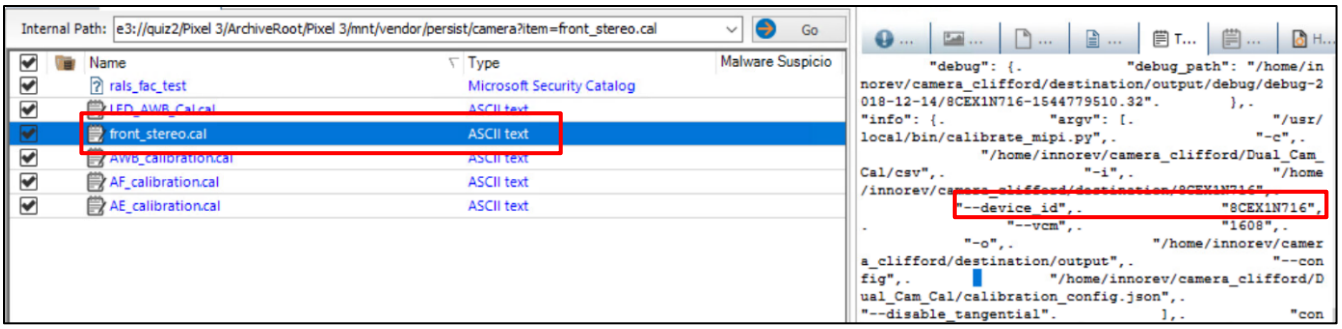
**1. What is the make, model, serial number of the device?**

**Answer:**
**Description:** pixel phone 3 black 64gb unlocked
**Model Name:** Pixel 3 (64 GB, Just Black, Unlocked)
**Model SKU:** GA00457-US
**Model Code:** G013A
**Serial:** 8CEX1N716
**IMEI:** 990012004966336
**Purchase Country:** United States
**Device Age:** 5 Years, 7 Months, 19 Days

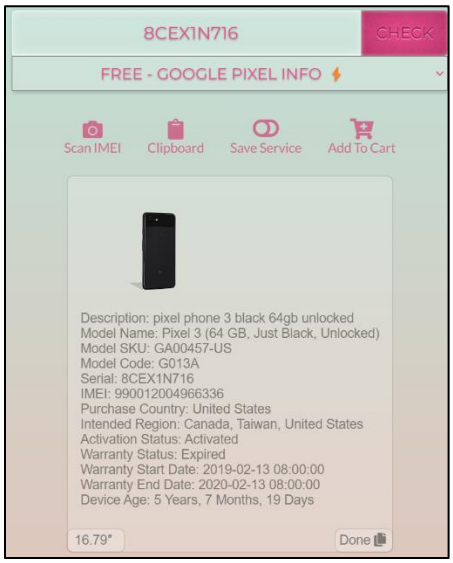
**Methodology:**
Since the serial number wasn't easily visible in the image file, we started by examining hardware configurations like sensors and camera modules. These components helped us trace device-specific information, leading us to the serial number. We first started with sensors configuration data.
**Location:** quiz2/Pixel 3/zip/Pixel 3/mnt/vendor/persist/sensors/tmd2725_als_factory.txt
The file location contains sensor calibration data for the ***tmd2725_als_factory.txt***. This file typically includes calibration values and configuration data used for accurate sensor functionality in the Pixel 3 device.




We first identified a potential serial number from this location, but since it might belong to a specific sensor, we proceeded to investigate other hardware configurations to ensure accuracy and gather more device-specific information.
**Location:** quiz2/Pixel 3/zip/Pixel 3/mnt/vendor/persist/camera/front_stereo.cal
We continued our approach by investigating the file location, which contains calibration data for the front camera stereo setup. This file includes technical specifications related to the camera modules, such as intrinsic and extrinsic parameters, sensor details, and distortion coefficients.
Within this data, we identified **8CEX1N716** labeled as a device ID. Given the context and the consistent format across the hardware components (such as cameras), it is highly likely that this device ID represents the serial number of the device. The presence of this identifier in such a critical configuration file related to hardware calibration further supports its significance as the device's serial number.



To verify and gather more information about the device, we proceeded to use the online platform [Sickw.com](Sickw.com) for serial number lookups. This platform allows us to search for device details based on the serial number, providing insights such as the model, purchase region, warranty status, and more.





By entering the serial number **8CEX1N716** from the Pixel 3 device, we can gather further validation and confirm details such as:

- Model Name: Pixel 3 (64 GB, Just Black, Unlocked)
- Model SKU: GA00457-US
- Model Code: G013A
- Purchase Country: United States
- Warranty Status: Expired
- IMEI: 990012004966336

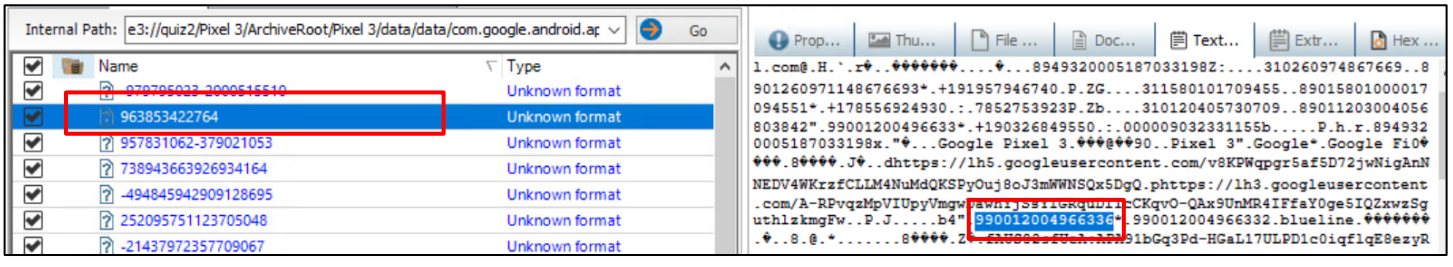
**Cross Verifying with matching associated IMEI number from lookup to device:**

To further verify the details, we will use the **IMEI number** obtained from the serial number lookup (990012004966336) and perform a **keyword search** within the device data. This will help us confirm if the IMEI is present in the device and validate its association with the Pixel 3.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.google.android.apps.tycho/files/ps/963853422764

The location appears to be part of the data directory related to a specific application, ***com.google.android.apps.tycho***, which is associated with Google Fi services (Tycho is Google's internal name for this service).

the file contains structured data that includes the IMEI number (highlighted as 990012004966336). This corresponds to the IMEI of the device we are analyzing. This suggests that this file may be holding important device identification data specific to the app's use of mobile services, such as device IMEI for network or user identification purposes.



The serial number **8CEX1N716** was confirmed as correct through a series of forensic checks and cross-references. Initially, we found this serial number in calibration data related to sensors and cameras, indicating it might be the device's unique identifier. We then verified the number using an online lookup tool, confirming it was linked to a Google Pixel 3. To further validate, we located the IMEI in system files, which matched the IMEI retrieved during the serial number lookup. This comprehensive approach, combining system data, hardware configuration, and external verification, confirms the accuracy of the serial number.

**2. What is the Google account email and password of the device? (Encrypted password)**

**Answer:**

**Email:** thisisdfir@gmail.com

**Password:** aas_et/AKppINYHk3843SOlirhnRUtCveD5dMEkDbJE-UM6ATDHh15WhYU1lcZNba5nhbrKUmsSReDEDBsCeGL7JPU4q-Yg4xnvw0l3EIJFTafNRpTQp9VWTc60f96ZOmBpTtd9rcFCO31RD2qnXx2XDpWJ7u0jnSIu78gwiCGPaUpaegazIAQxH1Pa-1h0VwGXElenMabM5yUWpAIhzpatx3xMFoA=

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/system_ce/0/accounts_ce.db

**Methodology:**

The ***accounts_ce.db*** file located at ***e3://quiz2/Pixel 3/ArchiveRoot/Pixel 3/data/system_ce/0/accounts_ce.db*** belongs to the primary ***user (0)*** of the Android device. In Android's file structure, the 0 directory is reserved for the main user of the device, and any sensitive, user-specific data related to accounts and services is stored under this folder.

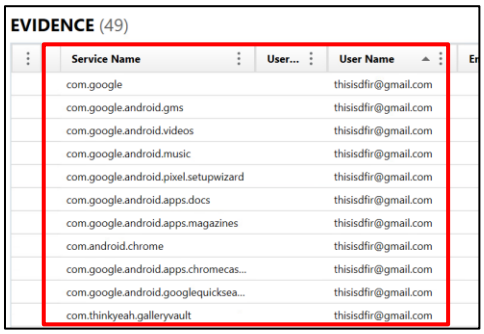
The database itself, ***accounts_ce.db***, is crucial because it stores account credentials for various applications and services associated with the user. Specifically, in this instance, we observed that the Google account associated with the device is stored under the account type com.google, with the email **thisisdfir@gmail.com**. This database entry includes not only the Google account email but also an encrypted authentication token or password, which starts with the prefix aas_et/. This token format is commonly used in Android systems to represent secure authentication data for Google accounts.

| _id | name | type | password |
|---|---|---|---|
| Fil… | Filter | Filter | Filter |
| 1 | thisisdfir@gmail.com | com.google | aas_et/AKppINYHk3843SOlirhnRUtCveD5dMEkDbJE-… |
| 10 | thisisdfir | com.silentcircle.account | dummyPassword |
| 4 | imo HD | com.imo.android.imous | NULL |
| 8 | WhatsApp | com.whatsapp | NULL |
| 5 | TikTok | com.zhiliaoapp.musically | NULL |
| 12 | TextNow | com.enflick.android.TextNow.account | |
| 7 | TDfir | com.twitter.android.auth.login | NULL |
| 11 | Skype | com.skype.raider | NULL |
| 3 | Signal | org.thoughtcrime.securesms | NULL |
| 6 | Messenger | com.facebook.messenger | NULL |
| 14 | Duo | com.google.android.apps.tachyon | NULL |
| 2 | 858233690 | org.telegram.messenger | |
| 13 | +19195794674 | com.viber.voip | 961ea18e43b4a2b35216e208d9414212e4b699c3 |

**Let's verify that the account was set up on google pixel 3:**

To verify the primary Google account associated with the device, we used Magnet Forensics, which identified the email ***thisisdfir@gmail.com*** as linked to multiple Google services such as **com.google.android.gms**, **com.google.android.music**, and others. This comprehensive listing of associated services further confirms that this email is the primary Google account on the device. The screenshot serves as evidence for the verification process, showing the clear link between the account and the registered services.

**Location:** PhysicalDrive0 VMware Virtual disk SCSI Disk Device (255 GB).zip\C\Mobile_Evidence_Files\Pixel 3\Pixel 3\data\system_de\0\accounts_de.db



EVIDENCE (49)

| Service Name | User... | User Name | Er |
|---|---|---|---|
| com.google | | thisisdfir@gmail.com | |
| com.google.android.gms | | thisisdfir@gmail.com | |
| com.google.android.videos | | thisisdfir@gmail.com | |
| com.google.android.music | | thisisdfir@gmail.com | |
| com.google.android.pixel.setupwizard | | thisisdfir@gmail.com | |
| com.google.android.apps.docs | | thisisdfir@gmail.com | |
| com.google.android.apps.magazines | | thisisdfir@gmail.com | |
| com.android.chrome | | thisisdfir@gmail.com | |
| com.google.android.apps.chromecas... | | thisisdfir@gmail.com | |
| com.google.android.googlequicksea... | | thisisdfir@gmail.com | |
| com.thinkyeah.galleryvault | | thisisdfir@gmail.com | |

**3. What is the telephone number of the device?**

**Answer:**

**Physical SIM:** (919)579-4674

**Secondary sim /eSim:** (651)338-1146

**Previous Number:** (903)268-4955 (Found Through Call log)

**Virtual Number:** (984)235-2054 (Textview Application)

**Virtual Number: (**919)758-0276 (Snapchat)

**Methodology:**

**Physical SIM: (919)579-4674** & **Secondary sim /eSim: (651)338-1146**

- The phone has two registered SIM numbers: **(919)579-4674** (Physical SIM) and **(651)338-1146** (Secondary SIM/eSIM). These numbers were retrieved from the **siminfo** table within the **telephony.db** database, which manages SIM card-related information on Android devices.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/user_de/0/com.android.providers.telephony/databases/telephony.db

| _id | icc_id | sim_id | display_name | carrier_name | name_source | color | number | display_number_format | data_roaming | mcc | mnc | mcc_string | mnc_string | ehplmns |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 8901260971148676693 | -1 | Google Fi | | 3 | -16746133 | 6513381146 | 1 | 0 | 310 | 260 | 310 | 260 | |
| 2 | 89011203004056803842 | 0 | Google Fi | Google Fi | 3 | -16746133 | +19195794674 | 1 | 0 | 310 | 120 | 310 | 120 | 310120,311870,31 |

**Previous Number: (903)268-4955**

- This number was found through the **call logs**. Since it is not present in the SIM card database (**siminfo**), this number could be associated with a previously used SIM card that was removed, or it could have been a number used through a virtual phone number application, such as a VoIP or carrier-specific virtual number. Possible sources include carrier reassignment, temporary use, or app-based number assignment. Without a corresponding SIM entry, this number likely represents a past association with the device or a number provided by a service like VoIP.

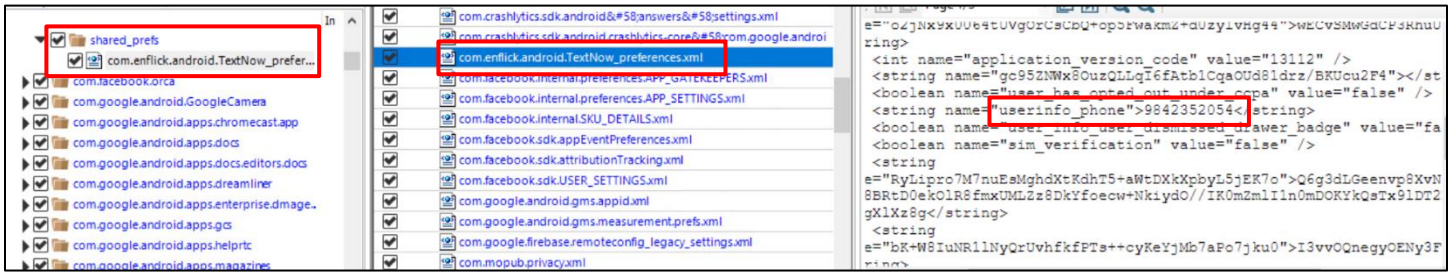
**Location:** PhysicalDrive0 VMware Virtual disk SCSI Disk Device (255 GB).zip\C\Mobile_Evidence_Files\Pixel 3\Pixel 3\data\data\com.android.providers.contacts\databases\calllog.db

| Local User | Partner | Partn... | Dire... | Call... |
|---|---|---|---|---|
| 9032684955 | +15755863247 | | Incoming | Missed Call |
| Local User <PhysicalDrive0 VMware Virtual disk SCSI... | +15755863247 | | | Unknown |
| Local User <PhysicalDrive0 VMware Virtual disk SCSI... | +15755863247 | | | Unknown |
| 9032684955 | 9195790479 | Josh Hickman | Outgoing | Answered |
| 9032684955 | 9195790479 | Josh Hickman | Outgoing | Answered |
| 9032684955 | +19195790479 | Josh Hickman | Incoming | Answered |
| 9032684955 | +19195790479 | Josh Hickman | Incoming | Answered |
| 9032684955 | 9195790479 | Josh Hickman | Outgoing | Answered |
| 9032684955 | 9195790479 | Josh Hickman | Outgoing | Answered |
| 9032684955 | 9195790479 | Josh Hickman | Incoming | Answered |
| 9032684955 | 9195790479 | Josh Hickman | Incoming | Answered |

**Virtual Number: (984)235-2054 (Textview Application)**

This virtual phone number was found within the preferences file for the **Textview Application**, indicating that it was generated and used within this app. Such numbers are often provided for calling and texting within app-based services.

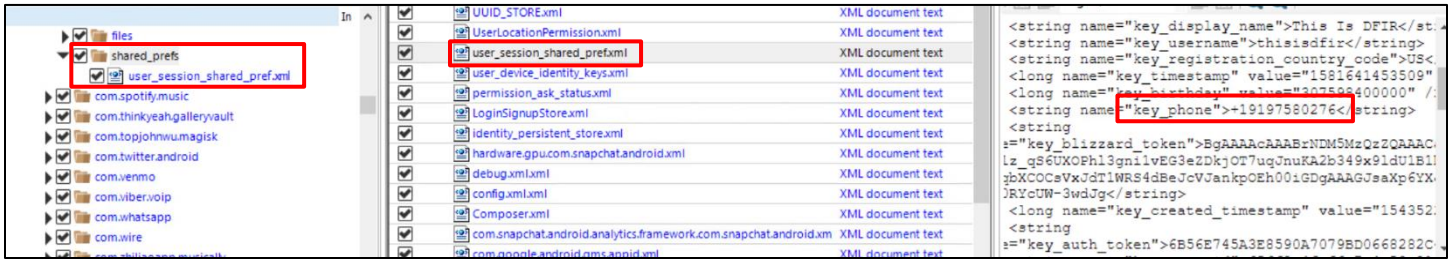
**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.enflick.android.TextNow/shared_prefs/com.enflick.android.TextNow_preferences.xml




**Virtual Number: (919)758-0276 (Snapchat)**

- This number was retrieved from the **Snapchat** application, another virtual number provided for messaging and calling through the app's internal services. This number is app-specific and functions through Snapchat's infrastructure, which provides users with alternate communication numbers.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.snapchat.android/shared_prefs/user_session_shared_pref.xml




**4. How many non-stock apps were installed on the device? Indicate five non-stock Android apps.**

**Answer: There are approximately 53 non-stock applications installed on the device.**

**Five non-stock apps:**

- **TextNow -** com.enflick.android.TextNow
- **Instagram -** com.instagram.android
- **Snapchat -** com.snapchat.android
- **WhatsApp -** com.whatsapp
- **Signal Private Messenger -** org.thoughtcrime.securesms

However, this number is still uncertain because several factors need to be considered. The phone is rooted, and there are indications that anti-forensic techniques have been applied. For example, Google Play Store, which is typically a stock application, is flagged by packages.xml as a non-stock (user-installed) app. This suggests that manipulations were made to the device to hide or alter application states, leading to discrepancies in what should be classified as stock or non-stock apps.

Non-stock apps are applications that were not pre-installed by the phone manufacturer but were installed later by the user. These include apps such as social media, messaging apps, or any third-party applications the user chose to download and install after acquiring the phone. Non-stock apps can reveal a lot about the user's activities, preferences, and sometimes, involvement in illicit activities. In this investigation, we are interested in identifying these apps to better understand the device's usage history.

**Methodology:**

The file **packages.xml** is crucial because it contains information about all the apps installed on the device, both stock and non-stock. This XML file stores the package details, including app names, package names, version details, and installation locations.

From this file, we can differentiate between stock and non-stock apps by examining their package names, origins, and identifiers. Stock apps are those pre-installed by the manufacturer (like Google or the device maker), while non-stock apps are user-installed apps.

Analyzing **packages.xml** helps us to determine the apps installed by the user, track installation dates, and confirm if the apps were active during the period under investigation. This makes it an important file in digital forensics for analyzing application-related evidence.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/system/packages.xml


We then used Magnet Forensics software, specifically focusing on the "Installed Applications" section, which displays all user-installed applications. This tool is very useful for identifying apps that were added by the user after acquiring the phone. We extracted the list of installed apps from Magnet and cross-referenced them with the results from the packages.xml file.

**CHING RESULTS** (72 of 288)

| Package Name | Display N... | AXIOM Suppo... | Icon | Platf... | Type |
|---|---|---|---|---|---|
| com.breel.wallpapers18 | | | | Android | User |
| com.customermobile.preloa... | | | | Android | User |
| com.discord | Discord | Discord | | Android | User |
| com.enflick.android.TextNow | TextNow | TextNow | | Android | User |
| com.facebook.orca | Messenger | Facebook Messenger | | Android | User |
| com.felicanetworks.mfc | | | | Android | User |
| com.felicanetworks.mfm | | | | Android | User |
| com.felicanetworks.mfm.main | | | | Android | User |
| com.felicanetworks.mfs | | | | Android | User |
| com.felicanetworks.mfw.a.bo... | | | | Android | User |
| com.felicanetworks.mfw.a.m... | | | | Android | User |

After obtaining both lists from packages.xml and Magnet Forensics, we compared them to identify overlapping applications, unique apps found in one source but not the other, and any discrepancies such as apps being falsely flagged as non-stock due to anti-forensic techniques or manipulations on the device.

While the estimated number of non-stock apps is **53**, it's important to acknowledge that this number might be slightly inaccurate. The phone is rooted, and anti-forensic techniques may have been used to manipulate or hide the true nature of applications. This means some apps could be misclassified as either stock or non-stock, as seen with the Google Play Store, which is typically stock but appeared as non-stock in the forensic analysis.

| Non-stock apps from packages.xml | Non-stock apps from Magnet List | Apps found in packages.xml but not in Magnet List | Apps found in Magnet List but not in packages.xml |
|---|---|---|---|
| Google Fi | Discord | Google Fi | Gallery Vault |
| Skype | TextNow | Trello | Musical.ly |
| Wire | Facebook Messenger | Cyber Dust | Magisk Root |
| Cyber Dust | Instagram | Strava | LINE Messenger |
| TikTok | Instagram Threads | My Verizon | |
| MeWe | MeWe | Silent Phone | |
| Instagram Threads | Wickr Me | FODA | |
| Facebook Messenger | Skout | imo video and chat | |
| My Verizon | Skype | Imgur | |
| FODA | Snapchat | TikTok | |
| Snapchat | Spotify | Netflix | |
| Discord | Magisk Root | Reddit | |
| TextNow | Twitter | Amazon Kindle | |
| WhatsApp | Venmo | Duolingo | |
| Wickr Me | Viber | Fitbit | |
| Skout | WhatsApp | LinkedIn | |
| Twitter | Wire | Pinterest | |
| Venmo | Musical.ly | Twitch | |
| Viber | LINE Messenger | Uber | |
| Instagram | Kik Messenger | Tumblr | |
| Signal Private Messenge | Telegram Messenger | SoundCloud | |
| Telegram Messenger | Signal Private Messenger | Google Docs | |
| Silent Phone | Tor Browser | Slack | |
| imo video and chat | | Dropbox | |
| Imgur | | Google Keep | |
| Tor Browser | | Google Chrome | |
| Magisk Root | | GitHub | |
| Kik Messenger | | Truecaller | |
| LINE Messenger | | MyFitnessPal | |
| TikTok | | | |

The list of unique applications includes a mix of communication, social media, and productivity tools. Many apps like WhatsApp, Telegram Messenger, and Signal Private Messenger are messaging platforms that offer secure communication, with features ranging from encrypted chats to disappearing messages. Social networking and media-sharing apps such as Instagram, Snapchat, and TikTok are popular for photo, video, and story sharing. Platforms like MeWe, Pinterest, and Reddit cater to niche interests and community-based discussions, while apps like Skout and Kik Messenger provide dating and casual communication services.

Based on the analysis, 53 unique non-stock Android apps were identified on the device. Non-stock apps are user-installed applications that provide insights into the activities and preferences of the device owner. However, this number is still uncertain due to several factors. The phone is rooted, and there are indications that anti-forensic techniques have been applied. For instance, Google Play Store, typically a stock application, was flagged by Magnet Forensics as a non-stock (user-installed) app. This suggests that manipulations were made to the device to hide or alter application states, which creates discrepancies in classifying stock versus non-stock apps.

Non-stock apps, which include social media, messaging platforms, and other third-party tools, are crucial for forensic investigations because they can reveal key details about the user's behavior and potential involvement in illicit activities. Understanding which apps were installed by the user helps reconstruct the device's usage history, allowing investigators to identify relevant activities and potential evidence. Despite the uncertainty, identifying these apps remains essential in the broader investigation of the device's manipulation and usage patterns

**5. What is the 2nd user profile that was added to this device?**

**Answer:**

**Name: "User 2"**

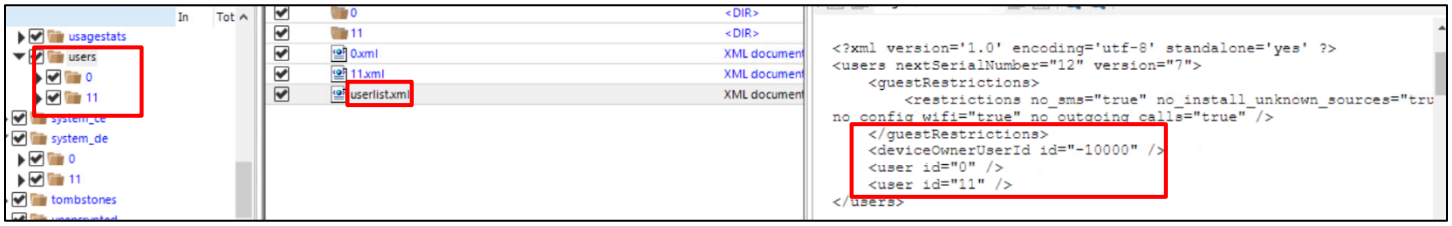
**User ID: 11**

**Flags: 16 (Indicates that it's a standard user account and not the primary or guest account)**

**Created: 1581644358241 (February 14, 2020 1:39:18.241 AM)**

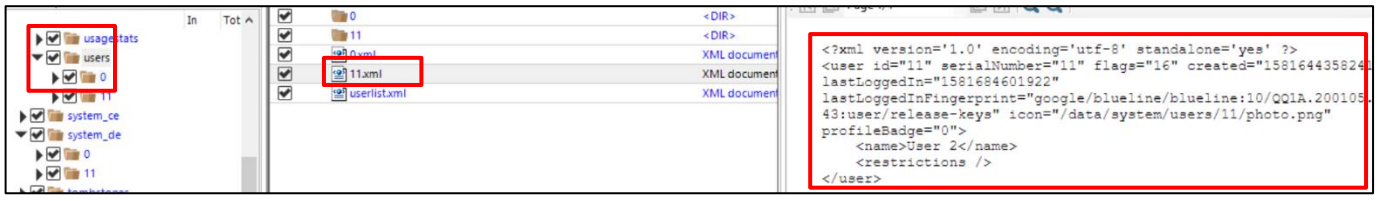
**Last Logged In: 1581684601922 ( February 14, 2020 12:50:01.922 PM)**

**Location: quiz2/Pixel 3/zip/Pixel 3/data/system/users/**




**Methodology:**

We found another user data from ***userlist.xml***:




**Profile picture:**

Profile Badge: 0 (Indicates no special user profile icon)

6. **Facebook Messenger was one of the apps used by the suspect in this case. Identify the version number, install date, username and password, and at least one of the messages from FB messenger with its date.**

**Answer:**

**User ID:** 100046799400843
**Version Number:** 249.0.0.10.122 (Internal version code 195366473)
**Install Date:** Wed Jan 29, 2020, 18:47:45 GMT+0000 (1580323665433)
**Last Login Success Timestamp:** Wed Jan 29, 2020, 18:51:04 GMT+0000 (1580323864060)
**Username:** ThisIs Dfir
**Password: (Access Token):**
EAADo1TDZCuu8BAPKhV3mbFskutluBZCiIKWwhZAPwzvor1PUXQvp2jjSuT8ZAT3SVHE3kRvMgLbzxmBroZCiBQSt5nni2goZC2HwYZCSVlGLZBmF5D8asPT3Vu1tMUuxwtmsVmaGhZADZBb2Y61M8Jl8jqR8GbRrtonUj4mhds8z8OzWWAy8u3iT5XUUzrKRy34ZC8ZD
**Messages and dates:**

- **Message 1:** "Hi there!" - Sent on 2020-02-01 18:49:07.
- **Message 2:** "Hey, how are you?" - Sent on 2020-02-01 18:50:24.
- **Message 3:** "Good. Hope you are." - Sent on 2020-02-01 18:51:18.
- **Message 4:** "I am. Thanks!" - Sent on 2020-02-01 18:52:05.
- **Last message timestamp:** 2020-02-09 18:10:03

**Participants in the Thread:**

- **Participant 1:** FACEBOOK:100030845613112.
- **Participant 2:** FACEBOOK:100046799400843.

**Thread Details:**

- **Total Messages in Thread:** 9 messages.
- **Unread Messages:** 0 (all messages were read).

**Methodology:**
**Let's start with one by one:**
**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.facebook.orca/databases/prefs_db
**Table:** preferences

**User ID:** 100046799400843

| | | |
|---|---|---|
| /settings/persisted_contacts_upload_settings/… | 2 | 1 |
| /push_settings/os_on/100046799400843 | 2 | 1 |
| /push_settings/sys_notif/100046799400843 | 2 | 1 |
| /prefs_user_id | 1 | 100046799400843 |
| LatUserId | 4 | 100046799400843 |
| /messenger/ms_queue_params_last_success_hash_code/100046799400843 | 3 | 1081648507 |

- The values /prefs_user_id and LatUserId in the preferences table are considered user IDs for Facebook Messenger because they explicitly contain the term "user_id," indicating their purpose as unique identifiers for user accounts. Both keys hold the same value (1000467999400843), suggesting they refer to the same user, and this naming convention aligns with common practices in mobile applications, where user identifiers are stored in preferences or settings databases to manage sessions and personalize experiences. These values can be found in the prefs_db file located in the /com.facebook.orca/shared_prefs/ directory, confirming their association with the Facebook Messenger app.

**Version Number:** 249.0.0.10.122 (Internal version code 195366473)

| | | |
|---|---|---|
| /settings/analyticsmetainf_fbmeta_version_code | 3 | 195366473 |
| /messenger/in_app_notification_navigate_to_inbox/update_version_key | 3 | 195366473 |
| /app_version/ | 3 | 195366473 |
| /config/DOWNGRADE_DETECTOR_PREVIOUS_APP_VERSION_CODE | 3 | 195366473 |
| /messenger/fcm/fb_server_build | 3 | 195366473 |

- The version number for Facebook Messenger, identified as **249.0.0.10.122**, corresponds to the internal version code **195366473** found in multiple settings within the preferences table. This information is located in the /com.facebook.orca/shared_prefs/prefs_db file, specifically under various keys related to the app's settings and configurations. The presence of the same internal version code across multiple entries confirms that these settings pertain to the Messenger application, indicating the version of the app being utilized on the device.
- The keys such as /messenger/in_app_notification_navigate_to_inbox/update_version_key and /messenger/fcm/fb_server_build are related to version control and updates within the app. They indicate the internal versioning system the app uses to manage updates and configurations.

**Install Date:** Wed Jan 29, 2020, 18:47:45 GMT+0000 (1580323665433)
**Last Login Success Timestamp:** Wed Jan 29, 2020, 18:51:04 GMT+0000 (1580323864060)

| | | |
|---|---|---|
| /shared/device_id_generate_timestamp | 4 | 1580323665430 |
| /messenger/first_install_time | 4 | 1580323665433 |
| /unified_account_login/login_screen_last_seen_ts | 4 | 1580323665647 |
| /settings/dextr/last_remaining_bytes_update_time | 4 | 1580323670 |
| /X.2Uq/X.36P | 4 | 1580323671443 |
| /dialtone/clearable/request_time | 3 | 1580323862 |
| /zero_rating2/clearable/request_time | 3 | 1580323862 |
| /unified_account_login/login_last_success_ts | 4 | 1580323864060 |
| /settings/app_state/last_first_run_time | 4 | 1580323869072 |
| /config/qk/last_check_time_ms | 4 | 1580323869072 |

- **/messenger/first_install_time**: This key represents the timestamp when the Messenger app was first installed on the device. This information can help in understanding the timeline of app usage and may be relevant in cases involving user interactions with the app.
- **/unified_account_login/login_last_success_ts**: This key denotes the timestamp of the last successful login to the account associated with the Messenger app. It is crucial for tracking user activity and understanding when the user last accessed their Messenger account.
- Both keys and their associated timestamps are valuable for forensic analysis as they provide insights into user behavior and app engagement over time. The timestamps are formatted as Unix timestamps, which can be converted into human-readable dates to facilitate analysis.

**Username:** ThisIs Dfir

| Filter | Filter | Filter |
|---|---|---|
| /orca_accounts/saved_account/100046799400843 | 1 | {"uid":"100046799400843","name":"ThisIs Dfir","last_logo |

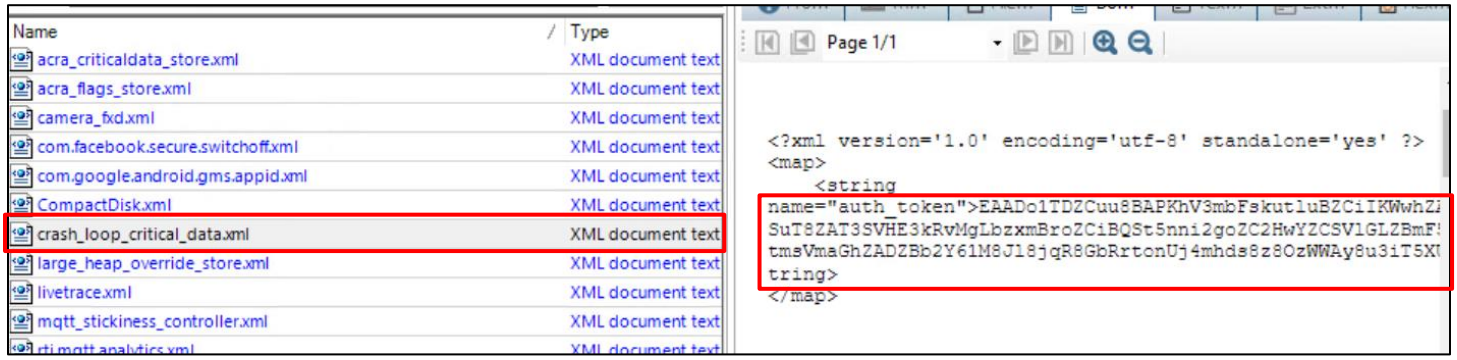
- **uid**: This field confirms the unique user ID linked to the account, which in this case is 100046799400843.
- **name**: This indicates the name associated with the account, presented as "ThisIs Dfir".

This information can provide context about the user and their interactions within the Messenger app, enhancing the understanding of usage patterns and account activity for forensic investigations.

**Password: (Access Token):**

EAADo1TDZCuu8BAPKhV3mbFskutluBZCiIKWwhZAPwzvor1PUXQvp2jjSuT8ZAT3SVHE3kRvMgLbzxmBroZCiBQSt5nni2goZC2HwYZCSVlGLZBmF5D8asPT3Vu1tMUuxwtmsVmaGhZADZBb2Y61M8Jl8jqR8GbRrtonUj4mhds8z8OzWWAy8u3iT5XUUzrKRy34ZC8ZD

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.facebook.orca/shared_prefs/crash_loop_critical_data.xml




This authentication token can be used to access the Facebook Messenger account associated with the user, providing a means to authenticate requests without needing the password. In many cases, applications opt to store access tokens instead of passwords for security purposes, allowing for secure authentication while minimizing the risk associated with storing sensitive password data. This also aligns with the understanding that passwords may not be retrievable from the device due to security protocols, making access tokens vital for continued user engagement with the app.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/system_ce/0/accounts_ce.db

**Table:** Extras

| _id | accounts_id | key | value |
|---|---|---|---|
| Filter | Filter | Filter | Filter |
| 160 | 1 | perm.com.google.android.contacts:ee3… | 1 |
| 161 | 6 | sso_data | {"userId":"100046799400843","accessToken":"EAADo1TDZCuu8BAPKhV3mbFskutluBZCiIKWwhZAPwzv |
| 162 | 7 | account_user_id | 1068228364824178689 |

**Value:**
{"userId":"100046799400843","accessToken":"EAADo1TDZCuu8BAPKhV3mbFskutluBZCiIKWwhZAPwzvor1PUXQvp2jjSuT8ZAT3SVHE3kRvMgLbzxmBroZCiBQSt5nni2goZC2HwYZCSVlGLZBmF5D8asPT3Vu1tMUuxwtmsVmaGhZADZBb2Y61M8Jl8jqR8GbRrtonUj4mhds8z8OzWWAy8u3iT5XUUzrKRy34ZC8ZD","name":"ThisIs Dfir","userName":"100046799400843","profilePicUrl":"https:\/\/scontent.xx.fbcdn.net\/v\/t31.0-1\/cp0\/e15\/q65\/c43.0.148.148a\/p148x148\/10506738_10150004552801856_220367501106153455_o.jpg?_nc_cat=1&_nc_ohc=jaPTndj-p1UAX8CQLAs&_nc_ad=z-m&_nc_cid=0&_nc_zor=9&_nc_ht=scontent.xx&_nc_tp=5&oh=070c9271af30b28940e1bd26ffd85dd6&oe=5ED78337","customKeyis_partial_account":"false"}

This JSON snippet contains critical information about the Facebook Messenger account associated with **User ID 100046799400843** and the **auth token**. Let's break down the relevance and connection between these elements:

**Breakdown of the JSON Data:**

- **userId**: 100046799400843 – This is the unique identifier of the user. It is used across Facebook services, including Messenger, to link all interactions, communications, and accounts to a specific individual.
- **accessToken**: EAADo1TDZCuu8BAPKhV3mbFskutluBZCiIKWwhZAPwzvor1PUXQvp2jjSuT8ZAT3SVHE3kRvMgLbzxmBroZCiBQSt5nni2goZC2HwYZCSVlGLZBmF5D8asPT3Vu1tMUuxwtmsVmaGhZADZBb2Y61M8Jl8jqR8GbRrtonUj4mhds8z8OzWWAy8u3iT5XUUzrKRy34ZC8ZD – This is the **auth token** or access token that provides authentication and is used instead of a password to grant the app permissions to act on behalf of the user.
- **name**: "ThisIs Dfir" – This is the display name associated with the account.
- **userName**: 100046799400843 – This reiterates the user ID, showing consistency across the JSON object.
- **profilePicUrl**: URL for the user's profile picture, which is less relevant in terms of authentication but ties this profile to the visible identity of the user.

**Why the auth token is crucial:**

1. **Access Token Function**:
   - The **access token** is the critical piece of data that allows the app (in this case, Facebook Messenger) to perform actions on behalf of the user without requiring the password each time. It is essentially a digital key that grants ongoing access to the account.
   - Applications often store access tokens instead of passwords for security reasons, reducing the risk of exposing the password directly. If the token is compromised, it can be revoked without requiring a password change.
2. **Connection Between User ID and Access Token**:
   - The **userId** (100046799400843) is the unique identifier linked to the user's Facebook account, and it remains consistent across all interactions, making it a critical element for identifying the owner of the token.
   - The **access token** is tied specifically to this user ID, meaning that actions performed using this token (such as retrieving messages, accessing user data) are associated with **this** Facebook account.
3. **Absence of a Password**:
   - In modern apps, it is common not to store or display the password on a local device for security purposes. Instead, access tokens are stored and refreshed to maintain access to the account securely.
   - This is likely why in your analysis, the **password** was not found, but the **access token** was. The access token serves the purpose of ongoing authentication, which is why it is stored and is critical in forensics for accessing account information.
   - Given that the token was found in multiple locations (like the crash_loop_critical_data.xml), it indicates its importance in maintaining the account's session, making it more relevant than the password itself.

**Key Take away:**

The **userId** and **accessToken** combination is used in Facebook Messenger to authenticate and grant access to the user's account and data. The access token replaces the need for a password in this case, as it is the primary means by which the app can continuously verify and interact with the user's account. This explains why a **password** is not stored locally, and instead, tokens are used to ensure security while enabling seamless access.

**Messages and dates:**

- **Message 1:** "Hi there!" - Sent on 2020-02-01 18:49:07.
- **Message 2:** "Hey, how are you?" - Sent on 2020-02-01 18:50:24.
- **Message 3:** "Good. Hope you are." - Sent on 2020-02-01 18:51:18.
- **Message 4:** "I am. Thanks!" - Sent on 2020-02-01 18:52:05.

**Participants in the Thread:**

- **Participant 1:** FACEBOOK:100030845613112**.**
- **Participant 2:** FACEBOOK:100046799400843**.**

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.facebook.orca/databases/threads_db2

```
SELECT
    _id,
    msg_id,
    text,
    sender,
    datetime(timestamp_ms / 1000, 'unixepoch') AS timestamp_readable
FROM
    messages
WHERE
    thread_key = 'ONE_TO_ONE:100030845613112:100046799400843';
```

| msg_id | text | sender | timestamp_readab |
|---|---|---|---|
| mid.… | Hi there! | {"user_key":"FACEBOOK:100046799400843","name":"ThisIs … | 2020-02-01 18:49:07 |
| mid.… | Hey, how are you? | {"user_key":"FACEBOOK:100030845613112","name":"Josh … | 2020-02-01 18:50:24 |
| mid.… | Good. Hope you are. | {"user_key":"FACEBOOK:100046799400843","name":"ThisIs … | 2020-02-01 18:51:18 |
| mid.… | I am. Thanks! | {"user_key":"FACEBOOK:100030845613112","name":"Josh … | 2020-02-01 18:52:05 |
| mid.… | | {"user_key":"FACEBOOK:100030845613112","name":"Josh … | 2020-02-01 18:57:46 |
| mid.$cAAAAB8r0m7N2M2TIhVwAh9-… | | {"user_key":"FACEBOOK:100046799400843","name":"ThisIs … | 2020-02-01 18:59:43 |
| mid.… | | {"user_key":"FACEBOOK:100046799400843","name":"ThisIs … | 2020-02-09 18:10:03 |
| mid.… | You can now call each … | {"user_key":"FACEBOOK:100030845613112","name":"Josh … | 2020-02-01 18:50:24 |

- **Last message timestamp:** 2020-02-09 18:10:03

**Thread Details:**

- **Total Messages in Thread:** 9 messages.
- **Unread Messages:** 0 (all messages were read).

```
SELECT
    _id,
    snippet,
    datetime(timestamp_ms / 1000, 'unixepoch') AS last_message_time,
    approx_total_message_count,
    unread_message_count
FROM
    threads
WHERE
    thread_key = 'ONE_TO_ONE:100030845613112:100046799400843';
```

| | _id | snippet | last_message_time | approx_total_message_count | unread_message_count |
|---|---|---|---|---|---|
| 1 | 19 | | 2020-02-09 18:10:03 | 9 | 0 |

**7. Identify the app that generated a payment of $5.**
**Answer:** Venmo

**Methodology:**
We identified several applications installed on the device that provide financial transaction facilities, including Venmo, PayPal, Google Pay, and Facebook Messenger. To trace the $5 payment, we began our investigation by thoroughly examining the data associated with these applications. This includes analyzing transaction logs, account information, and relevant records stored within the device's databases, shared preferences, and other local storage files, which could contain details regarding the payment activity.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.venmo/databases/venmo.sqlite

```
SELECT *
FROM marvin_stories
WHERE story_blob LIKE '%5.0%' OR story_note LIKE '%payment%' OR story_note LIKE '%transaction%';
```

| _id | story_id | story_date_created | story_date_updated | story_type | _feed_types | story_note | auth_story_shared | story_audience | likes | story_blob |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 2943369493847474… | 2020-02-14T01:49:… | 2020-02-14T01:5… | payment | | For the Android… | 0 | friends | [] | {"actor":{"username":"Josh-… |

The query searches the marvin_stories table for any entries related to a $5.00 payment or transactions. It looks for records where either the story_blob or story_note fields contain the specific amount "5.0" or mention keywords like 'payment' or 'transaction'. This helps identify stories that may include financial transaction details, ensuring we capture both exact payment amounts and general references to payments or transactions. The purpose is to locate relevant data regarding the $5.00 transaction from the available story records.

**Story_blob:**

{"actor":{"username":"Josh-Hickman-19","first_name":"Joshua","last_name":"Hickman","display_name":"Joshua Hickman","friend_status":"friend","friends_count":0,"mutual_friends_count":0,"profile_picture_url":"https://pics.venmo.com/bc1acbdc-8c5c-44b7-af3c-86b9fefc0c52?width\u003d460\u0026height\u003d460\u0026photoVersion\u003d1","id":"2853160431386624630","date_joined":"2019-10-12T14:40:28","is_blocked":false},"amount":"5.0","date_completed":"2020-02-14T01:49:46","date_created":"2020-02-14T01:49:46","id":"2943369493126053956","note":"For the Android 10 image again.","status":"settled","target":{"type":"user","user":{"username":"ThisIs-DFIR","first_name":"Joshua","last_name":"Hickman","display_name":"Joshua Hickman","friend_status":"not_friend","friends_count":0,"mutual_friends_count":0,"profile_picture_url":"https://pics.venmo.com/8db48124-5c0d-45af-8ed4-e2e48b010e54?width\u003d460\u0026height\u003d460\u0026photoVersion\u003d1","id":"2864291317284864770","date_joined":"2019-10-27T23:15:33","is_blocked":false}},"action":"pay"}

The provided JSON data contains transaction details related to a Venmo payment of $5.00. Here's the relevant information:

- **Transaction Amount:** $5.00
- **Date Completed:** February 14, 2020, 01:49:46
- **Date Created:** February 14, 2020, 01:49:46

- **Payer (Actor):**
  - Username: Josh-Hickman-19
  - First Name: Joshua
  - Last Name: Hickman
  - Display Name: Joshua Hickman
  - Profile Picture URL: Link
  - Date Joined: October 12, 2019
- **Recipient (Target):**
  - Username: ThisIs-DFIR
  - First Name: Joshua
  - Last Name: Hickman
  - Display Name: Joshua Hickman
  - Profile Picture URL: Link
  - Date Joined: October 27, 2019

- **Note:** "For the Android 10 image again."
- **Transaction Status:** Settled
- **Action:** Pay

This transaction is clearly a payment between two accounts, specifically Joshua Hickman paying the recipient for an "Android 10 image" on Venmo, a popular payment application.

8. **You encounter a stock/default Android app that doubles as a direction's finder. Indicate the name of this app, where the suspect was headed as well as the date and time if available**

**Answer:**
**Application Name:** Google Maps
**Location (Coordinates):** Latitude 35.734602, Longitude -78.636654
**Address:** NC State Bureau of Investigation, Tryon Road, Raleigh, NC 27611, United States of America
**Date of location request:** Thu Apr 04, 2019, 19:49:03 GMT+0000 (1554407343604)
**Date of last data sync:** Sun Feb 09, 2020, 19:45:38 GMT+0000 (1581277538891)

**Methodology:**

Based on the list of stock Android apps from packages.xml, the Google Maps app (com.google.android.apps.maps) doubles as a directions finder. In early 2020, Google Maps was widely used for location tracking and providing directions to destinations.

Given this information, Google Maps would likely be the app of interest for identifying where the suspect was headed, along with relevant date and time data, if available within the app's data logs or tracking history.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/system/packages.xml

Now let's investigate google maps data into file system.

Initially, we investigated the ***gmm_myplaces.db*** to check for any stored location data or coordinates. Our goal was to find the suspect's movement history, including any direction requests or searches. Upon examining the database, we found several entries, but most of them only contained last sync timestamps, which were not indicative of the actual time the directions or location requests were executed. Instead, these timestamps represented when data was last synchronized with Google's servers, which did not provide the precise timing of the actions we were investigating.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.google.android.apps.maps/databases/gmm_myplaces.db

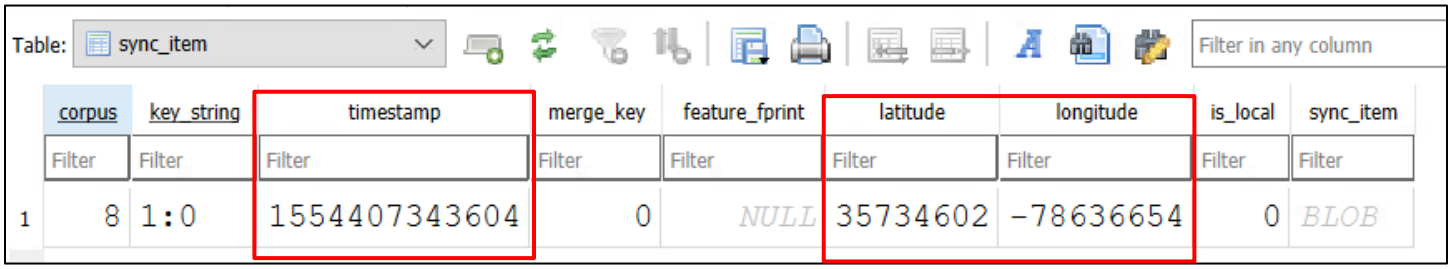


Table: sync_item

Filter in any column

| | corpus | key_string | timestamp | merge_key | feature_fprint | latitude | longitude | is_local | sync_item |
|---|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 8 | 1:0 | 1554407343604 | 0 | NULL | 35734602 | -78636654 | 0 | BLOB |

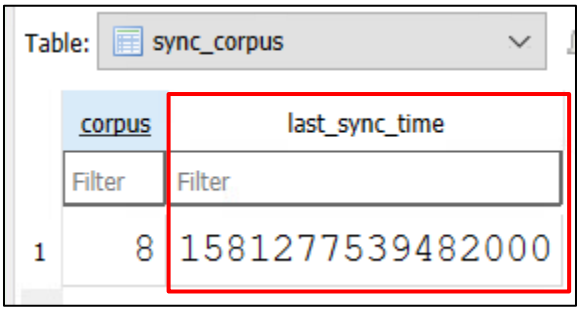
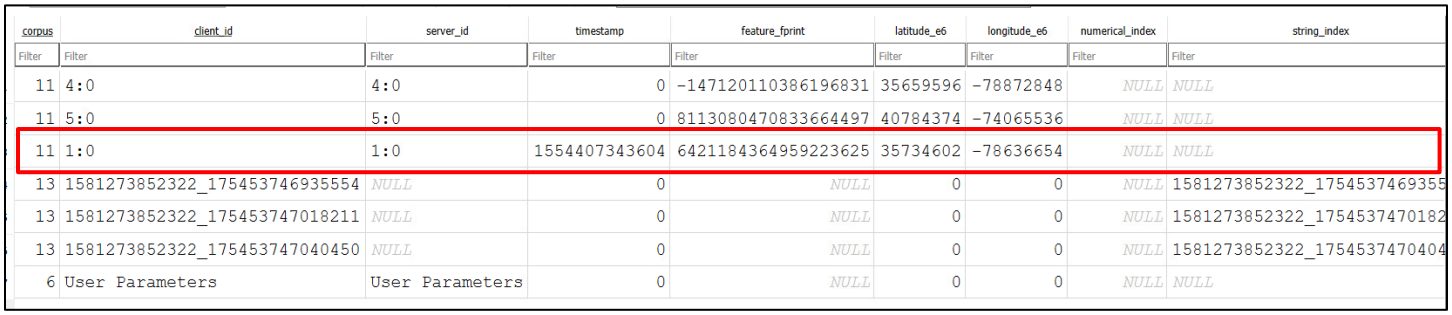


Table: sync_corpus

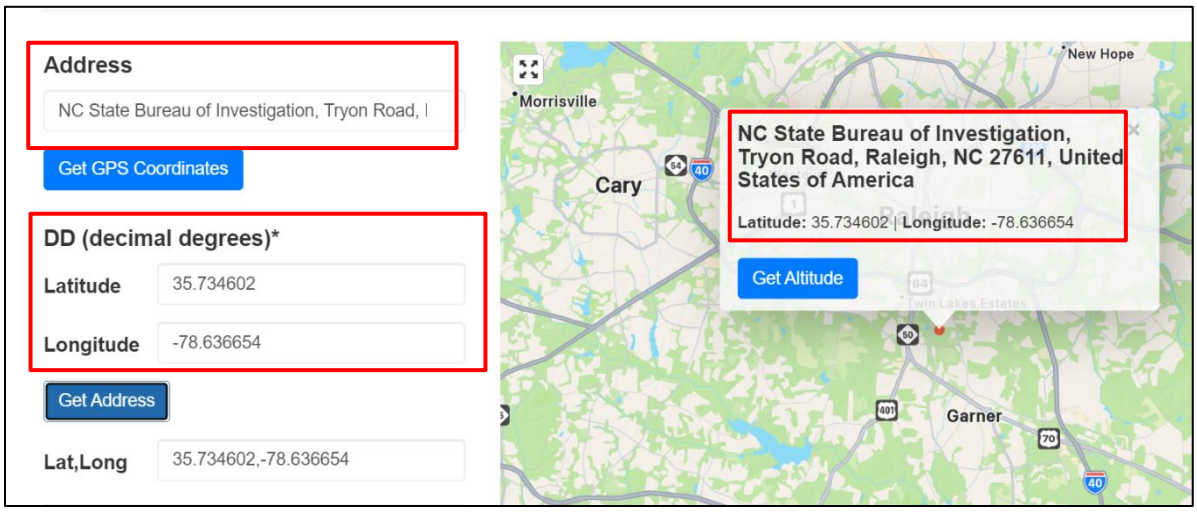
| | corpus | last_sync_time |
|---|---|---|
| | Filter | Filter |
| 1 | 8 | 1581277539482000 |

To obtain a more accurate understanding of the actual time when these coordinates were requested or used, we shifted our focus to the ***gmm_sync.db***. This database holds more detailed information regarding location requests and their execution. We found that this database contained both active and passive location requests. Active requests were indicated by valid timestamps, which reflect the precise moment the location or directions were requested. These active requests provided valuable evidence of the suspect's movements at specific times.



| corpus | client_id | server_id | timestamp | feature_fprint | latitude_e6 | longitude_e6 | numerical_index | string_index |
|---|---|---|---|---|---|---|---|---|
| Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 11 | 4:0 | 4:0 | 0 | -147120110386196831 | 35659596 | -78872848 | NULL | NULL |
| 11 | 5:0 | 5:0 | 0 | 8113080470833664497 | 40784374 | -74065536 | NULL | NULL |
| 11 | 1:0 | 1:0 | 1554407343604 | 6421184364959223625 | 35734602 | -78636654 | NULL | NULL |
| 13 | 1581273852322_175453746935554 | NULL | 0 | NULL | 0 | 0 | NULL | 1581273852322_1754537469355 |
| 13 | 1581273852322_175453747018211 | NULL | 0 | NULL | 0 | 0 | NULL | 1581273852322_1754537470182 |
| 13 | 1581273852322_175453747040450 | NULL | 0 | NULL | 0 | 0 | NULL | 1581273852322_1754537470404 |
| 6 | User Parameters | User Parameters | 0 | NULL | 0 | 0 | NULL | NULL |

Conversely, the passive entries, which lacked valid timestamps **(showing '0')**, indicated that while these locations were searched or saved, they were never actively requested for navigation. This distinction helped us isolate the actual movements from passive location searches or sync activities, giving us clearer insight into the suspect's active use of the Google Maps application.




By combining these findings from both databases, we were able to construct a clearer narrative of the suspect's movements and location requests, identifying which were actively executed and when, thus aiding our investigation into their movements.

**Technical Fact:**

The determination that entries with a timestamp of "0" in the **sync_item_data** table represent passive processes, while entries with non-zero timestamps indicate active interactions, is based on how Google Maps and similar location-based services manage data synchronization and user requests. Here's a detailed breakdown of the reasoning and technical facts behind this conclusion:

1. **Timestamp Values and Event Logging:**
   - In most databases, including Android apps like Google Maps, timestamp values are recorded when a user performs a specific action that requires logging, such as searching for a destination or requesting directions.
   - A **non-zero timestamp** (in UNIX epoch format) signifies a specific moment in time when an action occurred. In our dataset, the timestamp **1554407343604** corresponds to **April 4, 2019, 18:18:24 UTC**, indicating an active user interaction at that moment.
2. **Absence of Timestamps:**
   - When a timestamp is **"0"**, it typically means that the record was created without an associated time-based event. This can happen in scenarios such as background processes where no explicit user action triggered a loggable event.
   - In Google Maps, automatic processes such as cache updates, background location syncing, or periodic checks for nearby places can occur without direct user interaction. These processes don't require logging a specific event with a timestamp, which is why a "0" timestamp appears.
3. **Feature Fingerprints and Client IDs:**
   - Entries in the **sync_item_data** table contain other fields such as feature_fprint and client_id, which help in identifying the nature of the activity. For instance, records with feature fingerprints could indicate specific features or functions activated by the user.
   - In cases where the feature fingerprint is missing or client_id appears as a generic entry, this suggests the data is not tied to a specific action but might represent general synchronization or cached data.
4. **Google Maps' Passive and Active Data Collection:**
   - Google Maps collects data both passively (background syncs, automatic location updates) and actively (when the user requests directions or performs a search). Official documentation and developer notes confirm that location services regularly update data in the background without explicit user action, which would explain the presence of **"0"** timestamps in passive data entries.

Thus, from these technical facts, we can reasonably infer that records with **"0"** timestamps are tied to background operations, while those with non-zero timestamps represent deliberate user requests or actions.

9. **Indicate one Google device that data was casted to: name of device, MAC address, BSSID, Hotspot BSSID and SSDP_UDN. (casted means connected or hooked up)**

**Answer:**

**Device 1:**

**Device Name**: Google Home

**MAC Address**: C1:25:49:86:6E:26

**BSSID**: FA:8F:CA:75:A0:43

**Hotspot BSSID**: FA:8F:CA:75:A0:43

**SSDP_UDN**: %urn:x-cast:com.google.cast.media

**Device 2:**

**Device Name**: Google Nest Hub

**MAC Address**: 87CF4BACBBC707B5

**BSSID**: FA:8F:CA:75:A0:43

**Hotspot BSSID**: FA:8F:CA:75:A0:43

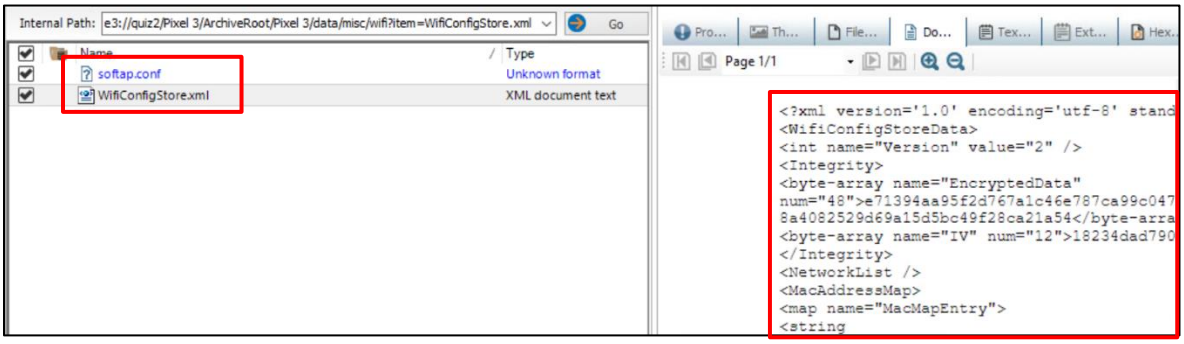
**SSDP_UDN**: %urn:x-cast:com.google.cast.media

**Methodology:**

**Initial Analysis of Wi-Fi Configurations:**

We began our investigation by analyzing the contents of the WifiConfigStore.xml and softap.conf files located at /data/misc/wifi/. These files provided key information about previously connected devices and networks, specifically helping us identify MAC addresses associated with the network.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/misc/wifi/




- **softap.conf** contained the MAC address ae:f6:c7:a8:a5:c2 but was not directly relevant to casting.
- Further analysis was required to find the casting-related devices.

Shifting our attention to the **cast.db** database ,we examined several key tables: DeviceInfo, NetworkToDevice, and ProbedSocketAddress. These tables provided detailed device and network data, including timestamps and casting-related information. This step revealed two Google devices actively involved in casting activities: **Google Home** and **Google Nest Hub**. Both devices were associated with the same BSSID and Hotspot BSSID, confirming they were connected to the same network.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/data/com.google.android.gms/databases/cast.db

- Google Home: **MAC Address C1:25:49:86:6E:26, BSSID FA:8F:CA:75:A0:43**, **Hotspot BSSID FA:8F:CA:75:A0:43**, **SSDP UDN %urn:x-cast:com.google.cast.media.**
- Google Nest Hub: **MAC Address 87CF4BACBBC707B5, BSSID FA:8F:CA:75:A0:43**, **Hotspot BSSID FA:8F:CA:75:A0:43**, **SSDP UDN %urn:x-cast:com.google.cast.media**.

able: DeviceInfo — Filter in any column

| capabilities | device_version | friendly_name | last_published_timestamp_millis | model_name | receiver_metrics_id | service_address | service_port | service_instance_name | last_discovered_timestamp_millis |
|---|---|---|---|---|---|---|---|---|---|
| Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |
| 198660 | 5 | Office speaker | 1581723739348 | Google Home | C12549866E269585 | 192.168.7.74 | 8009 | Google-… | 1581723718168 |
| 233477 | 5 | Office display | 1581723739347 | Google Nest Hub | 87CF4BACBBC707B5 | 192.168.7.88 | 8009 | Google-Nest-Hub… | 1581723698099 |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |
| NULL | -1 | | NULL | | NULL | NULL | 8009 | | NULL |

To verify the interaction between the devices and the Android device, we analyzed probing details from the **ProbedSocketAddress** table. This step showed consistent probing timestamps for both devices, further confirming their network activity during casting sessions. Additionally, the **NetworkToDevice** and **NetworkInfo** tables provided strong evidence linking the devices to the same Wi-Fi network, solidifying their connection during the casting process.

Table: ProbedSocketAddress — Filter in any column

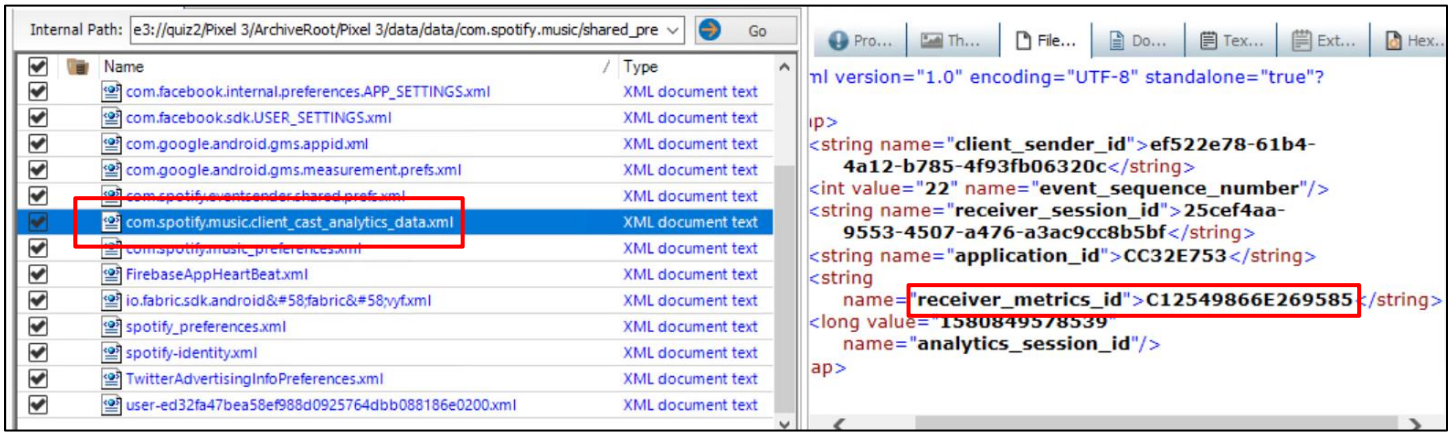
| | host_name | port | network_id | is_responding_to_probe | last_probed_timestamp_millis | probing_error_code | number_of_probed_times |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 192.168.7.31 | 8009 | f8:bb:bf:1e:fa:e8 | 0 | 730854122 | 2 | 3 |
| 2 | 192.168.1.20 | 8009 | f8:bb:bf:90:a8:f3 | 0 | 727858078 | 3 | 1 |
| 3 | 192.168.1.134 | 8009 | f8:bb:bf:1e:fa:e8 | 0 | 729581618 | 3 | 3 |
| 4 | 192.168.7.215 | 8009 | f8:bb:bf:1e:fa:e8 | 0 | 821718093 | 2 | 3 |
| 5 | 192.168.7.88 | 8009 | f8:bb:bf:90:a8:f3 | 1 | 1093445440 | 0 | 0 |
| 6 | 192.168.7.88 | 8009 | f8:bb:bf:1e:fa:e8 | 1 | 1581723634048 | 3 | 0 |
| 7 | 192.168.1.9 | 8009 | f8:bb:bf:1e:fa:e8 | 0 | 729561616 | 3 | 3 |
| 8 | 192.168.7.74 | 8009 | f8:bb:bf:90:a8:f3 | 1 | 1093477134 | 3 | 0 |
| 9 | 192.168.7.74 | 8009 | f8:bb:bf:8d:b9:c4 | 1 | 1096588530 | 0 | 0 |
| 10 | 192.168.7.74 | 8009 | f8:bb:bf:1e:fa:e8 | 1 | 1581723633992 | 2 | 0 |
| 11 | 192.168.7.27 | 8009 | f8:bb:bf:1e:fa:e8 | 0 | -148825531 | 2 | 3 |
| 12 | 192.168.7.173 | 8009 | f8:bb:bf:1e:fa:e8 | 0 | -148825531 | 2 | 3 |
| 13 | 192.168.6.173 | 8009 | f8:bb:bf:1e:fa:e8 | 0 | 730897055 | 3 | 3 |

| _id | network_id | device_id |
|---|---|---|
| Filter | Filter | Filter |
| 3092 | f8:bb:bf:90:a8:f3 | 4c39777295c6314fcbb2877f671b25a5 |
| 3093 | 02:00:00:00:00:00 | f782d122cd23946e20c5770a5bab47e7 |
| 3094 | f8:bb:bf:8d:b9:c4 | 4c39777295c6314fcbb2877f671b25a5 |
| 3095 | f8:bb:bf:1e:fa:e8 | 4c39777295c6314fcbb2877f671b25a5 |


| _id | network_id | device_id |
|---|---|---|
| Filter | Filter | Filter |
| 3092 | f8:bb:bf:90:a8:f3 | 4c39777295c6314fcbb2877f671b25a5 |
| 3093 | 02:00:00:00:00:00 | f782d122cd23946e20c5770a5bab47e7 |
| 3094 | f8:bb:bf:8d:b9:c4 | 4c39777295c6314fcbb2877f671b25a5 |
| 3095 | f8:bb:bf:1e:fa:e8 | 4c39777295c6314fcbb2877f671b25a5 |


- Consistent probing between the Android device and both Google devices.
- Matching **host_name**, **port**, and **BSSID** for both **Google Home** and **Google Nest Hub**.

Finally, to provide additional proof of the casting activity, we examined the ***com.spotify.music.client_cast_analytics_data.xml*** file. This file contained casting session details, including the **receiver_metrics_id** for **Google Home** (C12549866E269585), confirming that the device was involved in media playback via Spotify.




- **Spotify casting session** confirmed for **Google Home** with **receiver_metrics_id**: C12549866E269585.

We began by analyzing Wi-Fi configuration files, leading us to the **cast.db** database, where we identified **Google Home** and **Google Nest Hub** as the casting devices. Further validation came from probing data, and finally, Spotify logs confirmed the casting sessions. Through this methodical investigation, we established that these two Google devices were connected and involved in casting activities.

**10. Provide screen shots of user 1 home screen**

**Answer:**



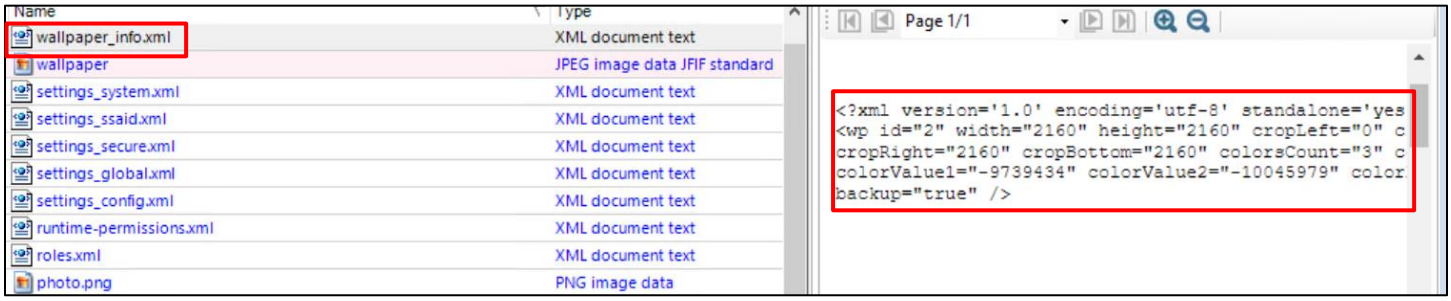
**Location:** e3://quiz2/Pixel 3/ArchiveRoot/Pixel 3/data/media/0/Pictures/Screenshots/

In the Android operating system, the primary user on a device is designated as "User 0." This is important because, during our investigation, we identified that screenshots and user data related to the home screen were located in the directory structure associated with User 0.

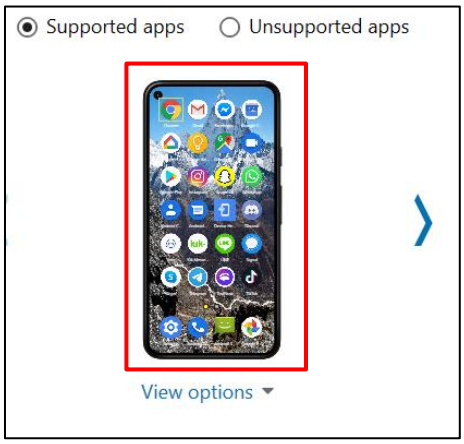
We began by examining the system structure in location. This directory is specifically associated with User 0, as indicated by the "0" in the directory path. As part of the investigation, we located a screenshot named Screenshot_20200213-213252.png in this folder.

To verify the accuracy of this screenshot, we cross-referenced the file with data found in the location, which contains the metadata for the home screen wallpaper. This XML file provided key information, including the screen resolution (2160x2160) and color values (three distinct color values: blue and gray shades). The comparison between the visual characteristics of the screenshot and the wallpaper metadata confirmed that the screenshot matched the wallpaper described in the XML file.

**Location:** /data/system/users/0/wallpaper_info.xml



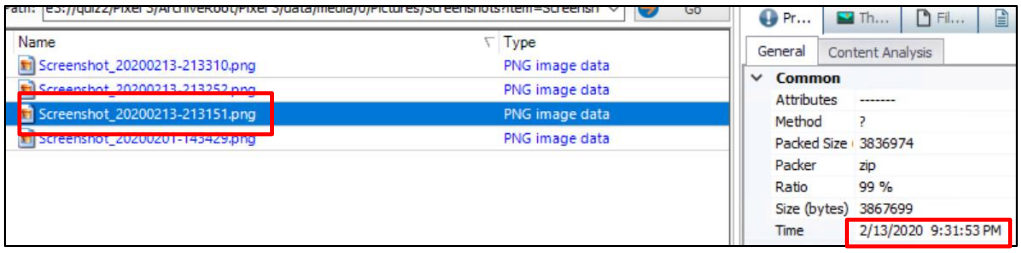

Additionally, we referred to Magnet AXIOM software to provide an overview and metadata of the screenshot, ensuring that the timestamps and file creation data aligned with the device's usage. This provided a concrete link between the image and the user's home screen activity at the time of the screenshot.




Indeed, these three images align with the timeline of events leading up to **February 14, 2020**, which marks the critical endpoint of our investigation. The screenshots were captured just before this date, indicating they were among the final actions taken on the device before the significant event occurred. Here's how the timeline and the screenshots relate:

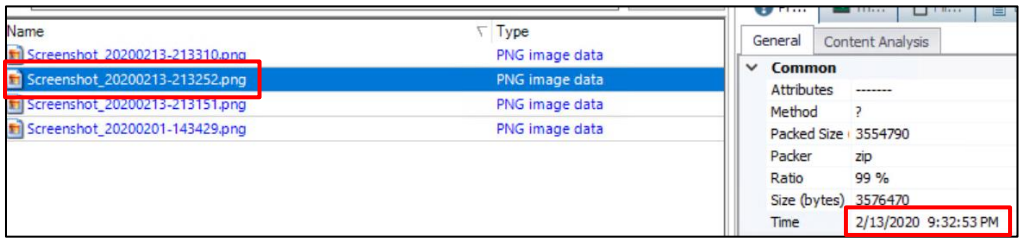
**Screenshot 1**: Screenshot_20200213-213151.png **Timestamp**: **February 13, 2020, 21:31**

- This screenshot shows the home screen with the timestamp, weather information, and the layout of app icons. It reflects the device's status late in the evening on February 13th.



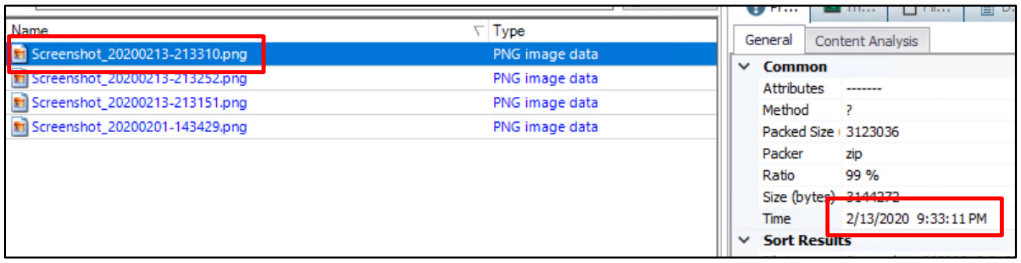
**Screenshot 2**: Screenshot_20200213-213252.png **Timestamp**: **February 13, 2020, 21:32**

- Captured just moments after the first one, this image further validates the activity on the phone, possibly confirming ongoing interactions before the device was potentially locked or powered down for the day.




**Screenshot 3**: Screenshot_20200213-213352.png **Timestamp**: **February 13, 2020, 21:33**

- The last screenshot before the event of **February 14, 2020**, showing the phone still being used, possibly for communication or another task. This image is crucial as it indicates the device's activity nearing the event's critical timeline.

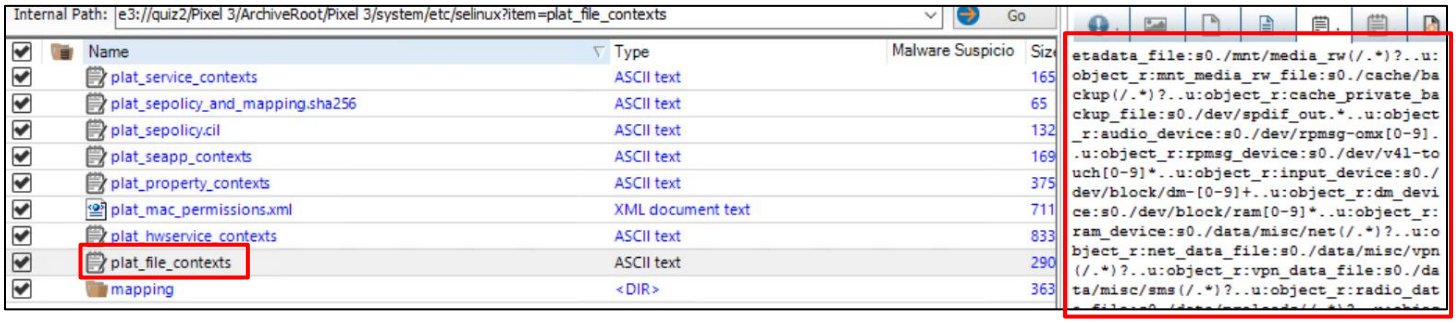



In simple terms, we found three screenshots saved on the device that match the home screen settings and appearance. These screenshots were found in the folder for the main user of the device, which Android refers to as "User 0." We verified the image details using the system's wallpaper settings file and confirmed they match the visual appearance of the screenshots. This helps us establish that the screenshots were taken by the main user of the device and reflect what was visible on their home screen at the time.

Let's cross verify with checking wallpaper of this device:

**Technical Explanation:**

In the Android system, SELinux policies are crucial for maintaining security by defining the access control rules for various files and directories. The file quiz2/Pixel 3/zip/Pixel 3/system/etc/selinux/plat_file_contexts provides these SELinux policies, ensuring that only authorized processes can access specific parts of the system.



```
*)?..u:object_r:preloads_media_file:s0./data/local/tmp/ltp(/.
*)?..u:object_r:nativetest_data_file:s0./cache/backup_stage(/.
*)?..u:object_r:cache_backup_file:s0./data/secure/backup(/.
*)?..u:object_r:backup_data_file:s0./data/system/users/[0-9]+/wallpaper..u:object_r:wallpaper
_file:s0./data/system/users/[0-9]+/photo
\.png..u:object_r:icon_file:s0./data/system/users/[0-9]+/fpdata(/.
*)?..u:object_r:fingerprintd_data_file:s0./data/system/users/[0-9]+/wallpaper_lock..u:object_
r:wallpaper_file:s0./data/system/users/[0-9]+/wallpaper_orig..u:object_r:wallpaper_file:s0./d
ata/system/users/[0-9]+/wallpaper_lock_orig..u:object_r:wallpaper_file:s0./dev/socket/wpa_wla
n[0-9]..u:object_r:wpa_socket:s0./data/system/dropbox(/.
*)?..u:object_r:dropbox_data_file:s0./data/misc/perfprofd(/.
*)?..u:object_r:perfprofd_data_file:s0./data/misc/bluetooth(/.
```

We found the following entry in the file:

**/data/system/users/[0-9]+/wallpaper u:object_r:wallpaper_file:s0**

This entry specifies that the wallpaper for a user profile is stored in the /data/system/users/[0-9]+/wallpaper directory and is assigned the wallpaper_file:s0 security context. This indicates that the wallpaper file is protected and can only be accessed by authorized processes.
By examining the wallpaper_info.xml file located in the /data/system/users/0/ directory, we were able to confirm the properties of the wallpaper image (such as dimensions and color values). This information matched the wallpaper seen in the screenshot located in /data/media/0/Pictures/Screenshots. This alignment of the wallpaper details from both the wallpaper_info.xml and the actual screenshot indicates that both the images refer to the same home screen wallpaper.

**Non-Technical Explanation:**

In an Android device, each user's wallpaper is stored in a specific folder, and the system has strict security rules to protect this information. We found that the wallpaper for the main user (user 0) is stored in a folder called **/data/system/users/0/wallpaper**, and we were able to verify the details of the wallpaper image from a system file (wallpaper_info.xml).

By comparing the information in this file with a screenshot of the home screen that we found, we confirmed that the wallpaper in the screenshot is the same as the one defined for the main user on the device. This helps us prove that the screenshot belongs to the main user's home screen.

**Location:** quiz2/Pixel 3/zip/Pixel 3/data/system/users/0/wallpaper