DFIR Review

How Android Bluetooth Connections Can Determine if a Driver had Their Hands on the Wheel During an Accident

Heather Mahalik

Published on: Nov 05, 2020

License: [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Synopsis

<p>Forensic question: How can Android Bluetooth artifacts be used in distracted driving cases?</p>
<p>OS: Android versions 6-10 were used in this research.</p>
<p>Tools: Magnet Acquire</p> <p>Public Roots for access</p> <p>Cellebrite UFED</p> <p>Cellebrite Physical Analyzer</p> <p>DCode</p> <p>DB Browser for SQLite</p>	

Here's how to unlock critical evidence from Android devices using Bluetooth connections to vehicles.

By Heather Mahalik, Senior Director of Digital Intelligence

Author's note: I'd like to extend my special thanks to Josh Hickman for allowing me to use his public images, for taking my calls to validate my sanity during my research, and for peer-reviewing this blog.

When it comes to scanning and tracking paired Bluetooth devices, Android devices are completely different from iOS phones. Historically, more files were accessible that tracked connectivity in older Android versions. We used to be able to track connectivity to Bluetooth, NFC, USB, and more within /data/com.android.connectivity.metrics/databases/events.db. This file is no longer present on the

Android devices we tested. In addition, connections to a car via Bluetooth exist all over the Android device. If you are trying to determine if a user was driving hands-free, you have your work cut out for you.

This blog will focus on determining if Bluetooth devices were paired to an Android device. We'll then look at how to determine actual connection times. For the examples in this blog, I used Josh Hickman's public Android image¹ so that you can validate my findings. This research includes validation from Josh's image and the Android images from myself and my colleagues.

Diving Into The Data

The easiest way to approach Bluetooth on Android is to start with what is parsed inside of your tool. Below we can see what [Physical Analyzer](#) detected for Bluetooth and Bluetooth devices.

Physical Analyzer Parsed Bluetooth

The Bluetooth listing under “Device Connectivity” shows paired devices. This device is the Bluetooth connection to Josh Hickman's Nissan Rogue. (Note: it is stored as “rouge” in the config file and in Josh's car. He confirmed that for me as the typo is strange). Notice the lack of a timestamp? We will circle back to that shortly.

>> Device Connectivity
 Translate
Go to ▾

Device Name: Rouge
 Device type: Unknown
 Timestamp:
 Connectivity method: Bluetooth
 Connectivity nature: Paired
 Artifact Family:
 Source Repository Path:
 Source: Bluetooth Devices
 Source file: [Google_G013A_Pixel_3.zip/data/misc/bluedroid/bt_config.conf : 0x1F9 \(Size: 919 bytes\)](#)

Device Identifiers

Source

Recognized Device
 Translate
Go to ▾

Device Name: Rouge
 Device Type: Unknown
 Device Model:
 Serial number:
 Account:

Device Identifiers

MAC Address b4:ec:02:73:ff:93

AdditionalInfo

Physical Analyzer Parsed Bluetooth - Paired

The next connection for Bluetooth is listed as “Devices > Bluetooth Devices.” This includes “detected” devices. For this test image, the Nissan Rogue (Rouge) was detected.

×	Name	Device Type	Device Identifiers	Source	Source file information
	Rouge	Unknown	MAC Address b4:ec:02:73:ff:93	Bluetooth Dev...	bt_config.conf_0x1f9

Physical Analyzer Parsed Bluetooth - Detected

Sadly, this is where the easy part ends. If you just needed to prove pairing, consider yourself lucky. But what if you needed timestamps for this pairing? What if you needed to whether the user was in a hands-free state when something occurred?

Anytime a connection is made on an Android device (USB, Bluetooth, etc.) there are traces left all over the device. To start, we can manually to examine the bt_config.conf file to determine “Paired” and

“Detected” devices. The file system view of parsed Bluetooth data is shown below. You can easily get to the location by clicking on the **“Source file”** in Physical Analyzer.

Physical Analyzer File System View

Some may open the configuration file and start poking around. We now need to make sense of this data.

Physical Analyzer Hex View of bt_config.conf

Upon closer examination, you may notice the timestamp at the top of the bt_config.conf file. This is *not* when the Bluetooth connection was established, but when the device was first set up for use. Make sure your tool is not parsing this as a connection timestamp.

Below, we can see that Josh started using his device on January 29, 2020. Don't believe me? Read his documentation file for this extraction. The Bluetooth wasn't first connected until the next day. Stay tuned for details on how to piece that together.

```
[Info].FileSource = Empty.TimeCreated = 2020-01-29 15:38:12..[Metrics].Salt256Bit = 8c68fa3461653b701e23f6bac4ae6fa71e56f7610758288c0fc27eb7228379ba..[Adapter].Address = 7c:d9:5c:ac:a2:ce.LE_LOCAL_KEY_IRK = 195d5c0827a16e7e95b8780cc4243049.LE_LOCAL_KEY_IR = 6aec03cd0258857111ffe969e845ec72.LE_LOCAL_KEY_DHK = e506ba23332577b905649764bcd9e248.LE_LOCAL_KEY_ER = f6fdc1e629635e0d40e479d02694a0ab.ScanMode = 0.DiscoveryTimeout = 120..[b4:ec:02:73:ff:93].Manufacturer = 72.LmpVer = 8.LmpSubVer = 30120.Name = Rouge.DevClass = 7936.DevType = 1.LinkKeyType = 5.PinLength = 0.LinkKey = acf8f79b5ecf1b517c84fa2c0fc57d15.AddrType = 0.Service = 0000111e-0000-1000-8000-00805f9b34fb 0000110b-0000-1000-8000-00805f9b34fb 0000110e-0000-1000-8000-00805f9b34fb 00001133-0000-1000-8000-00805f9b34fb 00000000-0000-1000-8000-00805f9b34fb 00000000-0000-1000-8000-00805f9b34fb .HfpVersion = 0701.HfpSdpFeatures = 2f00.AvdtpVersion = 0301..
```

Physical Analyzer Hex View of bt_config.conf

Depending on the Bluetooth device, you may retrieve timestamps. This was not the case when I was examining Josh's device because his paired Bluetooth connection was with a vehicle. [Android Auto](#) is a beast of its own, and timestamps and correlations are everywhere. Josh and I both conducted research on Android Auto a few years ago and we both continue learning as we dig through this data.

For this blog, I am going to provide you with a glimpse into how to put a person in a car, connected to Android Auto, when an activity occurs. To do this, we need to examine many files to establish connection start times, run times, and end times. From here, you can then correlate (timeline) the activity of interest.

There are many ways to approach this. One could simply search for "Bluetooth" across the parsed data or you can dive right in like I did. I copied the Mac address of Josh's car and did a Hex search in Physical Analyzer.

To properly conduct a Hex search across the entire device extraction, you must first go to the File Systems model and select the **“FileDump”** by double clicking.

Next, select the magnifying glass and type in your keyword of interest. (Refer to [Hex Diving Blog](#) for more details on this type of keyword search).

Physical Analyzer Hex Search

This search produced 106 results. I recommend sorting by **“Source”** and examining the files of interest.

#	Offset	Length	Value	Source	More
1	0x3A38A5F	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
2	0x3A38A79	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
3	0x3A38A7A	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
4	0x3A38A7B	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
5	0x3A38A7C	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
6	0x3A38A7D	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
7	0x3A38A7E	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
8	0x3A38A7F	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
9	0x3A38A80	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
10	0x3A38A81	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
11	0x3A38A82	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
12	0x3A38A83	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
13	0x3A38A84	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
14	0x3A38A85	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
15	0x3A38A86	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
16	0x3A38A87	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
17	0x3A38A88	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
18	0x3A38A89	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
19	0x3A38A8A	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
20	0x3A38A8B	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
21	0x3A38A8C	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
22	0x3A38A8D	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
23	0x3A38A8E	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
24	0x3A38A8F	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
25	0x3A38A90	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
26	0x3A38A91	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
27	0x3A38A92	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
28	0x3A38A93	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
29	0x3A38A94	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
30	0x3A38A95	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
31	0x3A38A96	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
32	0x3A38A97	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
33	0x3A38A98	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
34	0x3A38A99	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
35	0x3A38A9A	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
36	0x3A38A9B	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
37	0x3A38A9C	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
38	0x3A38A9D	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
39	0x3A38A9E	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
40	0x3A38A9F	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
41	0x3A38AA0	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
42	0x3A38AA1	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
43	0x3A38AA2	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
44	0x3A38AA3	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
45	0x3A38AA4	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
46	0x3A38AA5	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
47	0x3A38AA6	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
48	0x3A38AA7	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
49	0x3A38AA8	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
50	0x3A38AA9	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
51	0x3A38AAA	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
52	0x3A38AAB	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
53	0x3A38AAC	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
54	0x3A38AAD	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
55	0x3A38AAE	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
56	0x3A38AAF	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
57	0x3A38AAG	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
58	0x3A38AAH	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
59	0x3A38AAI	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
60	0x3A38AAJ	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
61	0x3A38AAK	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
62	0x3A38AAL	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
63	0x3A38AAM	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
64	0x3A38AAN	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
65	0x3A38AAO	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
66	0x3A38AAP	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
67	0x3A38AAQ	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
68	0x3A38AAR	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
69	0x3A38AAS	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
70	0x3A38AAT	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
71	0x3A38AAU	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
72	0x3A38AAV	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
73	0x3A38AAW	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
74	0x3A38AAX	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
75	0x3A38AAY	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
76	0x3A38AAZ	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
77	0x3A38A8F	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
78	0x3A38A8E	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
79	0x3A38A8D	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
80	0x3A38A8C	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
81	0x3A38A8B	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
82	0x3A38A8A	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
83	0x3A38A89	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
84	0x3A38A88	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
85	0x3A38A87	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
86	0x3A38A86	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
87	0x3A38A85	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
88	0x3A38A84	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
89	0x3A38A83	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
90	0x3A38A82	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
91	0x3A38A81	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
92	0x3A38A80	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
93	0x3A38A7F	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
94	0x3A38A7E	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
95	0x3A38A7D	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
96	0x3A38A7C	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
97	0x3A38A7B	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
98	0x3A38A7A	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
99	0x3A38A79	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
100	0x3A38A78	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
101	0x3A38A77	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
102	0x3A38A76	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
103	0x3A38A75	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
104	0x3A38A74	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
105	0x3A38A73	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	
106	0x3A38A72	0x11	B4EC:02:73FF:93	/data/data/com.google.android.gms/databases/canonicaldata.db	

Physical Analyzer Search Results

You can now double-click on the source and it will take you to that file in the “File System View.” Determining usage is the hard part, but we now have some files to lead us down that trail. Also note, I include all of these files for reference at the end of this blog.

Databases of interest can be examined using the SQLite Wizard inside PA where you can leverage the query builder to check columns of interest, decode timestamps, rename columns, and add it as a parser into Physical Analyzer.

SQLite Wizard in Physical Analyzer

Or, you can export the database out of Physical Analyzer and parse in your preferred tool. This query may help you determine relevance of this file to your investigation. The results will show that the “modllyear” is 18 MY, but his vehicle is actually a 2019. It appears that Nissan may associate all years to specific model parts (Nissa_Bosch_SUV). I am not a vehicle expert but wanted to mention it as the query provide this output.

SELECT

id,

manufacturer,

model,

datetime(connectiontime/1000,'unixepoch','localtime') AS "connection time",

modelyear,

vehicleid,

CASE

when bluetoothConnectionAllowed = 1 then "Bluetooth Allowed"

else "Bluetooth Not Allowed"

end AS "Permissions"

from allowedcars

DB Browser for SQLite showing my query results

At this point, we see a connection time of February 2, 2020 at 14:09 (localtime). This is where having test data is extremely helpful and we discuss testing like this in our Cellebrite podcast “Carved From Unallocated”. Josh provides documentation with his public images so that you can validate the data! I looked at what my query provided and found that he last connected his Android device to his car on this exact date and time.

Date	Time	Action
01/30/2020	08:56	Connect to car & setup app
	08:58	Started “NPR Up First” podcast
	09:00	Invoked Google Assistant and asked for directions: “I need directions to 7629 Purfoy Road in Fuquay-Varina, North Carolina”
	09:01	Started navigation in Maps to location
	09:16	Disconnected from car
02/09/2020	13:44	Connect to car
	13:46	“Give me directions to Sir Walter Coffee in Holly Springs, North Carolina.”
	13:47	Started “NPR Up First” podcast
	13:49	Got directions to Sir Walter Coffee in Holly Springs, NC
	13:51	Started “Digital Forensics Survival Podcast.”
	13:59	Disconnect from car
	14:09	Connect to car
		Continued “Digital Forensics Survival Podcast.”
	14:16	Disconnect from car

A snippet from Josh’s public documentation (Bluetooth Activity)

We are now getting somewhere. Now we know the last time Josh connected his Android device via Bluetooth to his car. Let’s keep digging! Another file that caught my eye was coffee_preferences.xml. This file helped determine when the device was first paired via Bluetooth using Android Auto. The decoded timestamp below (yes, there are others, but this is the one that matters) shows the timestamp of January 30, 2020 at 08:56 AM (localtime).

Physical Analyzer File Format Viewer and DCode for Validation

When we look at Josh's testing documentation, we can see we have established our primary date of interest correctly!

Date	Time	Action
01/30/2020	08:56	Connect to car & setup app
	08:58	Started "NPR Up First" podcast
	09:00	Invoked Google Assistant and asked for directions:
		"I need directions to 7629 Purfoy Road in Fuquay-Varina, North Carolina"
	09:01	Started navigation in Maps to location
02/09/2020	09:16	Disconnected from car
	13:44	Connect to car
	13:46	"Give me directions to Sir Walter Coffee in Holly Springs, North Carolina."
	13:47	Started "NPR Up First" podcast
	13:49	Got directions to Sir Walter Coffee in Holly Springs, NC
	13:51	Started "Digital Forensics Survival Podcast."
	13:59	Disconnect from car
	14:09	Connect to car
		Continued "Digital Forensics Survival Podcast."
	14:16	Disconnect from car

A snippet from Josh's public documentation (Bluetooth Activity)

What we have done so far is not an easy task, which is why I hope this blog helps you along your way. I am 100% positive that different devices will store the data in different files. However, you now have methods to guide you along the way.

Please refer to the [presentation](#) that Sarah Edwards and I provided at the DFIR Summit in 2019² as it provides a look at the Samsung side of Android Auto and Apple CarPlay as well. You will see that this research provided even more information than my presentation at the DFIR Summit showed as the device was made by a different manufacturer and the Android OS was older. This is the perfect example of why we all must continue researching, learning, and sharing our findings!

The following files were tested for **Android Auto** connections only (you may find more if you search like I did in this blog):

/data/misc/bluedroid/bt_config.conf

/data/data/com.google.android.gms/shared_prefs/coffee_preferences.xml

/data/data/com.google.android.projection.gearhead/shared_prefs/app_state_shared_preferences.xml
- tracks last runtime for Bluetooth

/data/data/com.google.android.projection.gearhead/shared_prefs/com.google.android.gms.analytics.prefs.xml - tracks first run time and monitoring start time.

/data/data/com.google.android.projection.gearhead/shared_prefs/common_user_settings.xml - Can be used to verify mac address for connected automobile.

/data/data/com.google.android.gms/shared_prefs/bluetooth_addresses_prefs.xml - Can be used to verify mac address for connected automobile.

/data/user_de/0/com.android.bluetooth/shared_prefs/bluetooth_volume_map.xml - contains paired mac address.

/data/user_de/0/com.android.bluetooth/shared_prefs/phonebook_access_permission.xml - contains paired mac address and can be associated to the peoplelog.db which updates contacts to the vehicle when the device is connected via Bluetooth or USB.

/data/data/com.google.android.gms/databases/carservicedata.db - Lists the allowed cars for the device.

/data/data/com.google.android.googlequicksearchbox/app_shared_prefs/SearchSettings.bin - Associations Google account to the vehicle mac address

/data/user_de/0/com.android.bluetooth/databases/bluetooth_db - The metadata table shows paired Bluetooth mac addresses.

References:

[1] Josh Hickman's public Android image: <https://thebinaryhick.blog/2020/02/15/android-10-image-now-available/>

[2] DFIR Summit Presentation: <https://www.youtube.com/watch?v=IGhXsfZXL6g>

DFIR Review

The paper/blog structure and its story telling method makes it easy to read and understand. This research could be very helpful to construct timeline analysis for Bluetooth connections made between devices and is a good addition to the digital forensic community. The images/figures, table, the list of the artifact paths on the device are presented clearly. Reviewers were able to verify the results using open source tools including HexEditor, SQLite DB Browser, Autopsy, and Notepad++.

We must consider the possibility that Bluetooth was turned off on the Android device. This means that a user could be in a vehicle, but there might not have been a connection. The table being looked at in SQLite Wizard was not indicated. One of the reviewers was able to identify the database as Pixel 3\data\data\com.google.android.gms\databases\carservicedata.db.

Future Work

A tool or plugin to extract these artifacts and construct the timeline analysis could be a big help. Future work could also include research on the additional timestamps available in `coffee_preferences.xml` as well as testing to see if multiple entries for different vehicles would exist in `carservicedata.db`. It would also be helpful to determine how long records last before they are overwritten by new records.

Reviewers

- Addisu Afework Birhanu (Methodology Review, Verified Review using Author Provided Datasets)
- Yohannes Yemane Brhan (Methodology Review, Verified Review using Author Provided Datasets)
- Jessica Hyde (Methodology Review, Verified Review using Author Provided Datasets)