Using Data Generated From In-Vehicle Infotainment and iPhone Data as Forensic Evidence

Maria Harrison
CS 8395-52: Digital Forensics
Vanderbilt University
Nashville, TN, USA
maria.harrison@vanderbilt.edu

***Abstract*—The purpose of this paper is to delve into the modern automobile's electronics and data generated from in-vehicle infotainment systems.**

Keywords—in-vehicle infotainment, forensic evidence, digital data

I. Introduction

A source of forensic evidence that is not commonly utilized in investigations is digital evidence acquired from the modern car. Traditionally, digital evidence is acquired from smartphone devices and computers, but as the modern car has evolved, it also generates digital data. This data can contain pertinent information used as forensic evidence if a vehicle was involved in an investigation.

In this paper, I will discuss Charlie Miller and Chris Valasek published research on remote attacks on an in-vehicle infotainment system. Additionally I will conduct research to analyze my iPhone after it was connected to my car's infotainment system via CarPlay, as a method of acquiring digital data from the vehicle.

II. Background

In 2014, Charlie Miller and Chris Valasek successfully carried out a remote attack on an unaltered 2014 Jeep Cherokee.The remote attack displayed the security vulnerabilities from the IVI; Harman/Kardon's Uconnect 8.4AN/RA4 radio. Miller and Valasek research made a forensic copy of the unit's file system, noting the IPL, IFS, ETFS, and MMC file systems. They discovered that the MMC file system ISO was able to be mounted as read/write, allowing for enabling and editing of bootable scripts and services; and the PPS system contains files that write data used as input by other processes.

A. In-Vehicle Network

The modern car's infrastructure is quite complex and has greatly evolved over the past two decades. Every (modern) vehicle has an In-Vehicle Network (IVN). The IVN consists of a Controller Area Network (CAN) bus; the digital network communication system designed for intercommunication within the car. The CAN bus topology is composed of nodes, Electronic Control Units (ECUs). ECUs are embedded systems that control an electrical system/subsystem in the car's functionality like braking, steering, engine control, and object detection. A new car's IVN contains more than 100 ECUs and numerous connection interfaces. The connection interfaces generate, transmit, process, and store digital data. Traditionally, each ECU was designed to perform a single function but as the IVN evolved, ECU consolidation became standard practice. ECU consolidation consists of software defined ECUs that can control multiple functions of a subsystem. A prime example of this is a vehicle's In-Vehicle Infotainment System (IVI).

B. In Vehicle Infotainment System

The In-Vehicle Infotainment System (IVI) is an electrical system that handles GPS, Navigation, Radio, and Audio Multimedia. The modern IVI provides smartphone connectivity through Apple CarPlay, Android Auto, and Bluetooth.

In-vehicle infotainment systems (IVI) and telematic systems can produce digital evidence. Telematic systems send, receive, and store information about a vehicle, while IVI pertains to entertainment features.

In-vehicle Infotainment Systems provide data on:

- GPS Navigation
- Video/CD Players
- Music Streaming
- SMS texting and phone calls
- Hands-free phone calling
- USB and Bluetooth connectivity
- In-car internet
- Wi-Fi

Telematic Systems provide data on:

- Vehicle location by GPS
- Remote access, including: unlocking and locking doors, starting the ignition, temperature controls

- Notification of vehicle collision, vehicle/object detection
- Notification of blindspot monitoring, and lane departure warning
- Control of vehicle speed
- Emergency Services calling
- Vehicle diagnostics and maintenance notifications

III. Investigating Toyota Infotainment System

The Toyota 4Runner is a sport utility vehicle; since its initial market release in 1984, there are 5 total generations. Toyota's 2021 model of the 4Runner is part of the 5th generation category, meaning it shares exterior body design and interior design features with other 4Runner models in this group. As the modern car has evolved, the Toyota 4Runner is notorious for its lack of advanced technology within its in-vehicle network, and minimal driver experience features. Because of this, my research will be focusing on the in-vehicle infotainment system in my 4Runner.

A. Tools

I examined my personal vehicle for this research, a 2021 Toyota 4Runner SUV (4Runner). I chose my personal vehicle due to prior knowledge of the vehicle and its features and access to data. I purchased this as a used vehicle from a dealership in Saint Louis, MO in February 2023. For this research, I focused on the data generated from my 4Runner's in-vehicle infotainment system/stereo system.

Test Vehicle:
2021 Toyota 4Runner SUV vehicle
- SR5 Premium
- All Wheel Drive
- Magnetic Gray

My vehicle was equipped with Toyota's 8.0" touch screen GPS/Navigation system, with the radio and audio multimedia supported by Gracenote.

Toyota 4Runner's GPS/Navigation Stereo Assembly Features
- Toyota Connected Services
- Toyota Safety Connect
- Toyota Service Connect
- Audio Multimedia
- Entune AppSuite
- Android Auto, Apple CarPlay, SiriusXM, and Bluetooth connectivity
- In-vehicle Wi-Fi Connect & Mobile Hotspot

B. Test Environment

There is very limited software for automotive forensics, none of which are open source toolkits. In 2017, The Department of Homeland Security Science & Technology published a fact sheet from the project iVe. Project iVE was a research project with Berla Corporation to create an automotive digital forensics toolkit.

Because of the lack of open source toolkits, my options for this project was to physically remove my stereo (like Miller and Valasek did in their research) to find the CPU chip and make an image of it or to find another method to analyze data from my car's infotainment system.

For the scope of this project, I narrowed my focus to the data from my iPhone's backup. My iPhone can only connect to my car's infotainment system/utilizing CarPlay via a USB-2.0 to lightning cable connection. This option was the least invasive as I could backup my personal iPhone to my computer, and theoretically acquire the digital data written to my phone when I used CarPlay, rather than remove and deconstruct my car radio.

C. Initialization

iPhone backup data is saved as SQLite Database files. The filenames are encoded as a hex string using SHA-1 hash. Apple CarPlay integrates the following apps in its interface: Messages, Phone, Contacts, Spotify, Google Maps, and Apple Maps.

I will be investigating these core folders and their hashed names in hopes to acquire digital data written from my car. The following chart shows possible naming for most folders created from an iPhone backup.

TABLE I. Data Folders

Data Folders		
Interface	***SQLite Naming***	***SHA-1 hash***
SMS	sms.db	3d0d7e5fb2ce288813306e4d4636395e047a3d28
Contacts	AddressBook.sqlitedb	31bb7ba8914766d4ba40d6dfb6113c8b614be442
Calendar	Calendar.sqlitedb	2041457d5fe04d39d0ab481178355df6781e6858
Reminders	Reminders.sqlitedb	
Notes	notes.sqlite	ca3bc056d4da0bbf88b5fb3be254f3b7147e639c
Notes	NoteStore.sqlite	4f98687d8ab0d6d1a371110e6b7300f6e465bef2
Call History	call_history.db	2b2b0084a1bc3a5ac8c27afdf14afb42c61a19ca
Call history	CallHistory.storedata	5a4935c78a5255723f707230a451d79c540d2741
Locations	consolidated.db	4096c9ec676f2847dc283405900e284a7c815836
Web history	History.db	1a0e7afc19d307da602ccdcece51af33afe92c53
Voicemail	voicemail.db	992df473bbb9e132f4b3b6e4d33f72171e97bc7a
Photos	Photos.sqlite	12b144c0bd44f2b3dffd9186d3f9c05b917cee25
Wallpaper (home screen)	HomeBackground.cpbitmap	b97b0c3bc8a6bb221d0849b450fbd92b5d06a301
Wallpaper (lock screen)	LockBackground.cpbitmap	86736007d0166a18c646c567279b75093fc066fe

a. Possible folder names generated during an iPhone backup

IV. Test Steps

A. *Backing up iPhone*

First I backed up my iPhone to my computer and opted to NOT encrypt it.

B. *Copying Backup Data*

Once the backup was complete, I made a copy of this folder named “Backup Copy”. This allowed me to easily use this data with my Windows virtual machine.

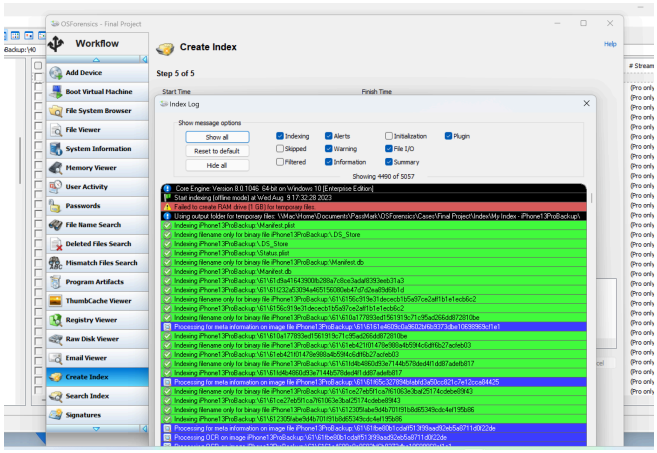
I started my Windows 10 virtual machine and imported the copy of my backup contents to my Windows desktop.

C. *Mount Data in OSForensics Program*

I then launched OSForensics within my Windows VM. Within my desired case, I used the Add Device tool to mount my iPhone backup folder to the case.

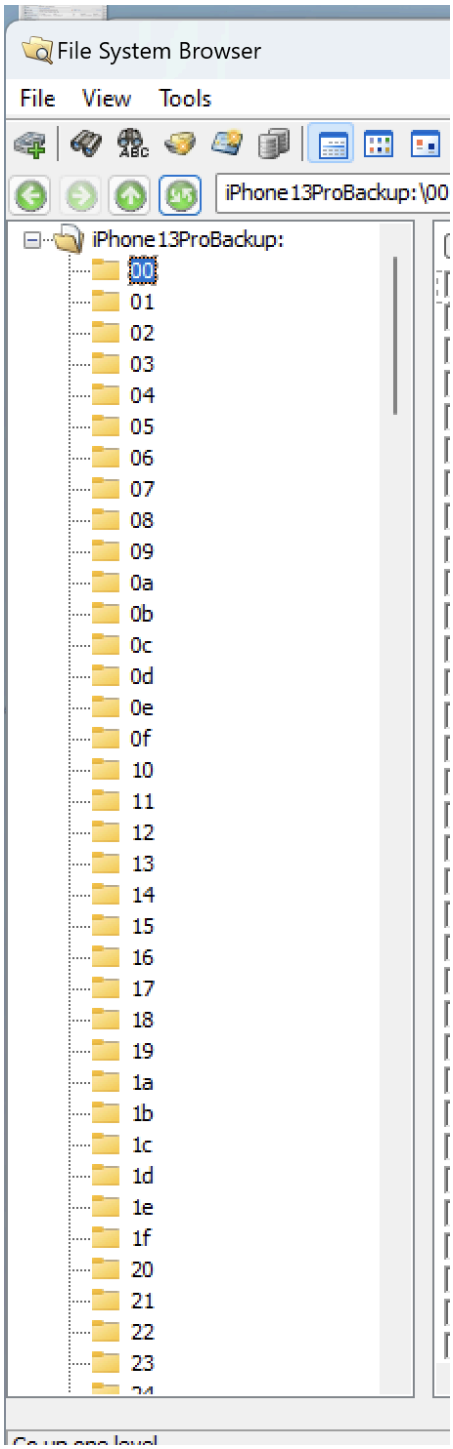
D. *Index Data*

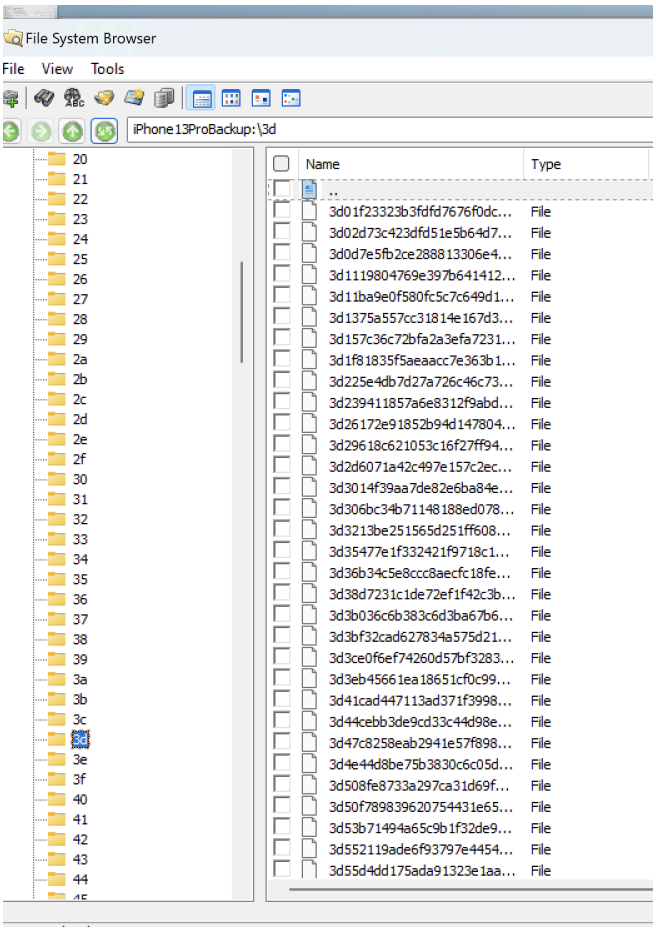
Once my backup folder was mounted, I created an index of it. I used the Create Index tool to successfully create an index. I can now search the index contents.



E. *View Data from Backup*

Using the “File System Browser” and “File Viewer” tools, I was able to view all the folders from this backup.

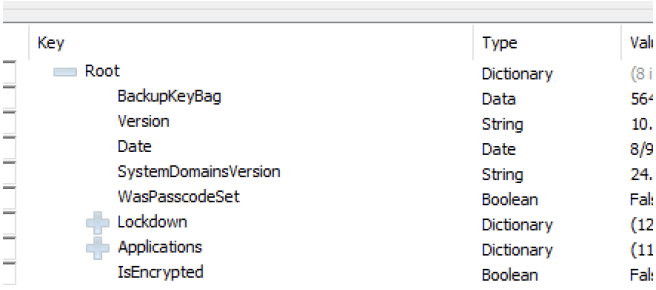




F. Search Files

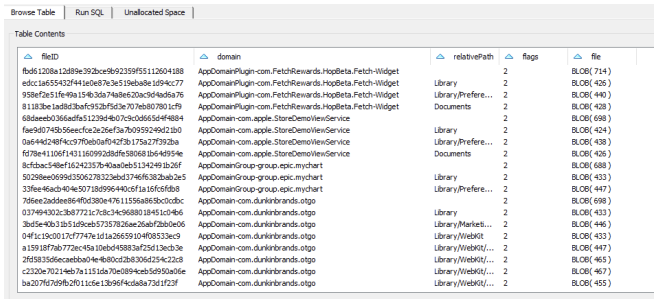
Using the "Search Index" tool, I was able to view the following files.

Manifest.plist



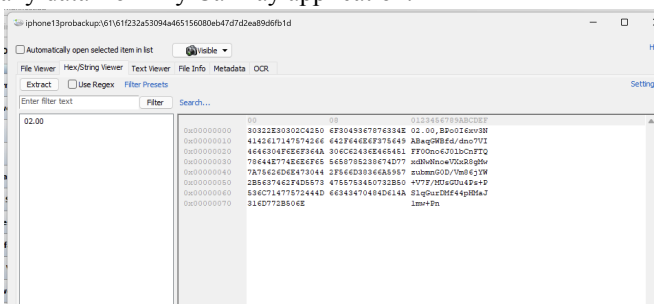
Manifest.db

This file's contents opened in OSForensic's SQLite DB view, allowing me to see a more detailed list of all the contents of this backup.



G. *Analyze Additional Files*

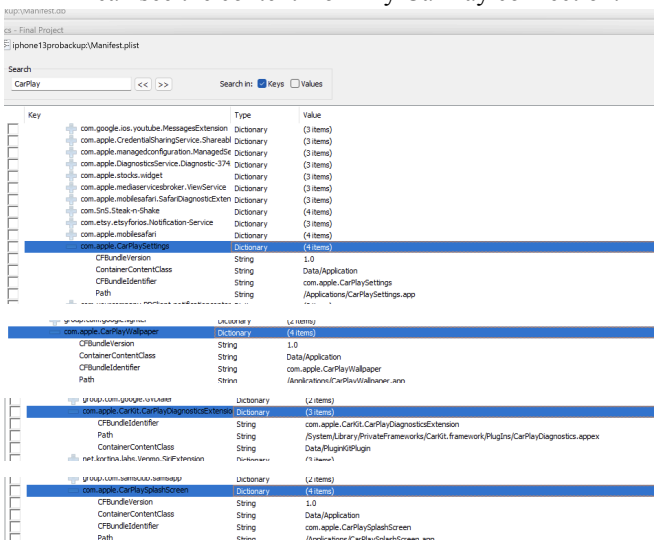
I analyzed more files with the "Search Index" tool, but the majority of them had minimal data and I was not able to find any data from my CarPlay application.



H. *Searching by Keyword*

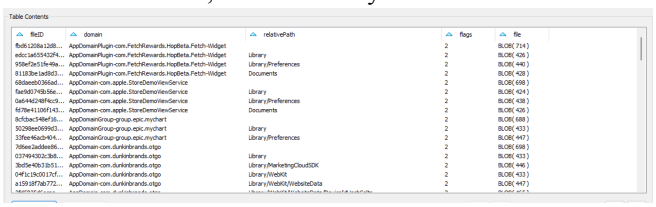
Using the "Search Index" tool, I was able to search the contents of the Manifest.plist file. When searching within the Search Index tool, I searched by keyword. I used the word "Car Play" and searched under the Root/Application directory, finding the following results:

- The ContainerContentClass, CFBundleIdentifier, and Path values were the most prevalent to me as this could help point me to the directory where I can see the content from my CarPlay connection.



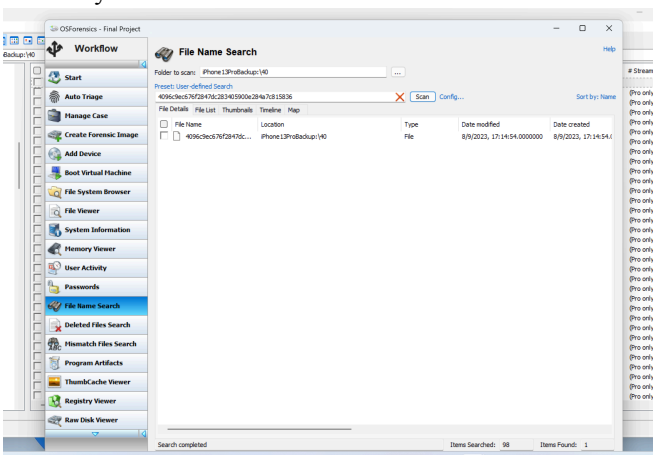
I. *SQLiteDB Tool*

In the "SQLiteDB" tool, I searched the Manifest.db file for a relative path that matched any of the ones shown in Step 7. I returned 0 results searching in the domain and relativePath with the values found from ContainerContentClass, CFBundleIdentifier, and Path in my Manifest.db file.



J. *File Name Search*

I used the "File Name Search" tool as another search method to find digital data from the CarPlay connection. In this tool, I searched for core SQLite Database folders to see if CarPlay data was written there.



K. *File Name Search Results*

From the "File Name Search", I was able to conclude that there were 5 main types of files created during an iPhone backup.

- SQLite Database files
- Plain text plist
- Binary plist files
- Multimedia and text files
- Non-standard data files

L. *Searching Specific Files in File Name Search*

Using "File Name Search", I searched for specific files including Locations, SMS, Call History, and Contacts via their hashed file names.

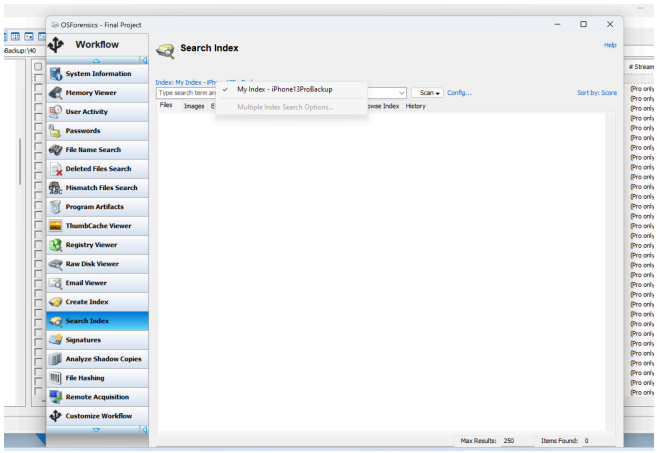
TABLE 2. DATA FOLDERS

Data Folders		
Contents	***Real Content Name***	***Hashed Folder Prefix***
SMS	sms.db	3d
Contacts	AddressBook.sqllitedb	31
Call History	call_hsitory.db	2b
Call History	CallHistory.storedata	5a

V. Findings

While I was able to successfully analyze a forensic copy of my iPhone's backup files, I was not able to find direct data written to my iPhone when it was connected to my car's infotainment system. The CarPlay data I assumed I would find was information like answering the phone call via hands free, using Siri to reply via text, and initiating a phone call from my favorite's contact via Siri. The further I got into this research, I started to realize that this type of data was most likely written to the filesystem in my car's IVN rather than my mobile device that's connected to my car.

Although I wasn't able to acquire specific digital data written to my phone during the CarPlay connection, I was able to conduct a successful analysis of an iPhone 13 Pro backup.



From my findings, I now understand how difficult automotive forensics is and why there is such a strong need for forensic toolkits for car networks.

VI. Works Cited

[1] Miller, Charlie, and Chris Valasek. *Remote Exploitation of an Unaltered Passenger Vehicle*. 2014.

[2] Drozhzhin, Alex. "Black Hat USA 2015: The Full Story of How That Jeep Was Hacked." *Kaspersky.com*, Kaspersky Lab, 7 Aug. 2015, www.kaspersky.com/blog/blackhat-jeep-cherokee-hack-explained/9493/.

[3] K. Strandberg, N. Nowdehi and T. Olovsson, "A Systematic Literature Review on Automotive Digital Forensics: Challenges, Technical Solutions and Data Collection," in IEEE Transactions on Intelligent Vehicles, vol. 8, no. 2, pp. 1350-1367, Feb. 2023, doi: 10.1109/TIV.2022.3188340.

[4] J. Walrand, M. Turner and R. Myers, "An Architecture for In-Vehicle Networks," in IEEE Transactions on Vehicular Technology, vol. 70, no. 7, pp. 6335-6342, July 2021, doi: 10.1109/TVT.2021.3082464.

[5] Le-Khac, Nhien-An, et al. "Smart Vehicle Forensics: Challenges and Case Study." in Future Generation Computer Systems, vol. 109, Aug. 2020, pp. 500–510, https://doi.org/10.1016/j.future.2018.05.081.

[6] Homeland Security, "S&T Project iVe Vehicle Navigation Infotainment System Forensics for Law Enforcement Fact Sheet" in Homeland Security Archived Content, https://www.dhs.gov/publication/st-project-ive-vehicle-navigation-infotainment-system-forensics-law-enforcement-fact

[7] Gomez Buquerin, Kevin Klaus, et al. "A Generalized Approach to Automotive Forensics." *Forensic Science International: Digital Investigation*, vol. 36, Apr. 2021, p. 301111, https://doi.org/10.1016/j.fsidi.2021.301111.

[8] Black Hat USA. "BRIEFINGS - AUGUST 5-6." *www.blackhat.com*, *Black Hat USA 2015*, Aug. 2015, www.blackhat.com/us-15/briefings.html.

[9] https://www.iacpcybercenter.org/wp-content/uploads/2020/09/Vehicle-Data_LECC-Article.pdf