Forensic Science International: Digital Investigation 48 (2024) 301691

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS EU 2024 - Selected Papers from the 11th Annual Digital Forensics Research Conference Europe

Grand theft API: A forensic analysis of vehicle cloud data

Simon Ebbers [a,*], Stefan Gense [a], Mouad Bakkouch [a], Felix Freiling [b], Sebastian Schinzel [a]

[a] Münster University of Applied Sciences, Stegerwaldstraße 39, Steinfurt, 48565, Germany
[b] Friedrich-Alexander-Universität Erlangen–Nürnberg (FAU), Martensstr. 3, Erlangen, 91058, Germany

ARTICLE INFO

Keywords:
Vehicle forensics
Cloud forensics
Mobile forensics

ABSTRACT

Modern vehicles such as cars, trucks and motorcycles contain an increasing number of embedded computers that continuously exchange telemetry data like current mileage, tire pressure, expected range and geolocation to the manufacturer's cloud. Vehicle owners can access this data via Vehicle Assistant Apps (VAA). Naturally, this data is of increasing interest to law enforcement in criminal investigations. While manufacturers must comply with local laws requiring them to hand over the data of suspects upon the issuance of a warrant, this process can be time-consuming and cause an additional delay in a case. Making use of novel API-based access methods in cloud forensic investigations, we present a method to get permanent access to a vehicle's cloud data by directly accessing cloud servers given suspects' credentials. We analysed a set of 23 different VAAs and pointed out the potentially accessible data categories. With our proof of concept tool `gta.py` in combination with six provided vehicles from BMW, Dacia, Ford, Hyundai, Mercedes and Tesla, we verified the accessibility of the data categories. Our findings demonstrate that the API-based forensic acquisition and analysis of vehicle cloud data provides important insights to be considered in future digital forensic investigations of vehicles.

1. Introduction

Modern vehicles give the impression of being computers on wheels. On-board IT systems consist of dozens of computers such as embedded control units, entertainment systems, fleet management systems, and maintenance monitoring sensors. Controller Area Network (CAN) buses or plain Ethernet using TCP/IP combine them into networked IT systems which use Global System for Mobile Communication (GSM) modules to connect to the Internet. These systems offer users remote control capabilities while also gathering telemetry data, including driving behaviour, vehicle component health status, and fuel consumption. Some of this data is stored on-board, some is uploaded to the vehicle manufacturer's cloud systems, and some of it is available in mobile apps that the vehicle owners can install on their mobile phones.

Naturally, this data is highly valuable to law enforcement agencies for various purposes, such as locating stolen vehicles, identifying suspects in vehicles, verifying witness statements or alibis, addressing insurance matters, and gathering detailed information about traffic accidents. Consider a simple case where the police have arrested suspects and the telemetry data of their car is a relevant piece of evidence. Law enforcement now has two possibilities to access this data. The standard technique has been to extract the data directly from the car, e. g. by connecting directly to the respective on-board control units. The success of this operation highly depends on the specific situation, for example, the physical availability of the car, the presence of stored data, and the duration for which data is stored.

Another and increasingly used possibility to access this data is to recover it from manufacturer cloud systems. Naturally, this method relies on the cooperation of the vehicle manufacturers, which can be cumbersome despite legal authority. Ideally, law enforcement would be able to access the data in the same way as the legitimate user. Given situations where authorities gain lawful temporary access to a suspect's smartphone containing apps with valid credentials for the suspect's vehicle, we investigate how law enforcement can access telemetry data from vehicle manufacturer cloud environments.

1.1. Related work

Until today, the main focus of vehicle forensics has been the investigation of the vehicles themselves (Gomez Buquerin, Corbett and Hof (2021)). The case study of Jacobs et al. (2017) demonstrates such a forensic analysis based on an entertainment system. Companies like Berla (2023) offer a sophisticated set of tools to perform data extraction off major vehicle brands. Furthermore, Jacobs et al. (2017) point out the

* Corresponding author.
E-mail address: simon.ebbers@fh-muenster.de (S. Ebbers).

https://doi.org/10.1016/j.fsidi.2023.301691

possibilities of mobile traffic analysis. These methods usually carry over to state-of-the-art vehicles with modern assistant systems like the Tesla autopilot (Hof et al., 2021; Gomez Buquerin et al., 2022).

As observed by Gomez Buquerin and Hof (2021), there are more data sources to consider than the vehicle itself. This is due to the differences between the stakeholders and their interests in accessing data on the vehicle. As discussed by Ebbers et al. (2021), the Original Equipment Manufacturer (OEM) and the user's smartphone could store relevant data. The routes of a vehicle might be stored on the smartphone tracked by the VAA. Furthermore, the smartphone which was connected to a car might give a hint that points to the driver of the car (Mahr et al., 2022). But shown by Ebbers et al. (2021), some applications store this useful information only in cached files on the smartphone. Therefore, the smartphone is not a data storage device but a gateway to access the cloud storage offered by the OEM. This observation leads to a novel challenge for digital forensic acquisition in vehicle forensics, similar to *cloud forensics*.

In their seminal work in cloud forensics, Roussev et al. (2016) identified the Application Programming Interface (API)-based forensic acquisition approach of cloud drives. This approach is by nature limited by the API and requires labour-intensive reverse engineering to get to the API if it is not publicly documented. Furthermore, the API-based forensic acquisition highly depends on its authentication and authorisation methods (Pawlaszczyk et al., 2022). The given open source tools kumodd and CLOUDxTRACT developed by Roussev et al. (2016) and Pawlaszczyk et al. (2022) require the credentials to acquire the data of the targeted cloud. These limitations were addressed by Priyanka et al. (2021) using the extracted authentication tokens after the analysis of a physical image of an Android device to acquire cloud data in a forensically sound manner.

1.2. Contributions

Overall, the literature pertaining to vehicle forensics strongly suggests that the API-based forensic acquisition of VAAs might be a valuable addition to the methods currently used in vehicle forensics. To fill this research gap, this paper describes a method and presents an API-based forensic acquisition tool to acquire and preserve the accessible data of current VAAs. The central insights obtained from this work are, for example, the storage location of required access tokens of the analysed apps in the mobile operating systems and what kind of data can be acquired by the different APIs of current VAAs.

We are not aware of any work that has specifically addressed the issue of retrieving cloud data from vehicle manufacturers using credentials or tokens extracted from a smartphone. The contributions of this paper are therefore manifold:

1. We introduce a generic method to access vehicle cloud data using credentials or tokens extracted from a forensically analysed smartphone (Section 3).
2. We implement and provide[1] a Proof of Concept (POC) that implements the method (Section 4).
3. We analyse a set of 23 different VAAs focusing on the accessible data categories (Section 5).
4. We provide insights into available data from the six manufacturers BMW, Dacia, Ford, Hyundai, Mercedes and Tesla (Section 6).

2. Background

Modern cars use a variety of digital communication technologies to enable different features and functions for the users. In order to understand which requirements are necessary to get access to the vehicle data, Section 2.1 describes the procedure for the user. Section 2.2 provides an explanation of the fundamental protocol that is crucial for making an API-based acquisition.

2.1. Get connected to your vehicle

To get in touch with the provided technologies to monitor the status and maintenance of the vehicle, the user has to create an account on the manufacturer's web page from inside of the VAA. To do so, an email address respectively a phone number is required to create a password and fill in a form with some further data about the user like name and address. The required data differ only slightly between the manufacturers.

To link a car to the new account there are different options depending on the manufacturer. In more modern cars, for example on a latest Mercedes, a QR-Code shown on the infotainment system can be scanned by a logged-in user with the VAA. With some manufacturers like BMW, the cars can be linked to the user account by submitting the Vehicle Identification Number (VIN). After this, there are a variety of necessary second steps to confirm this request. With an Audi or BMW for example, an interaction with the infotainment system is necessary, where the user has to confirm a transaction number. To get access to a Mercedes or Tesla, the user must authorise themselves at a Mercedes partner or send a picture of their ID using the Mercedes Me app. After submitting the VIN through the VAA the user has to activate data sharing by pushing a dedicated button inside the car. After that, the user has to synchronise the car with the app by starting the engine, turning it off and starting it again. The last step is done by submitting the total mileage of the car through the app. To claim the ownership of a Tesla to an account, the user has to provide the VIN, a picture of their driver's licence, passport or personal ID as well as proof of temporary admission, a picture of the admission, a picture of the certificate of ownership or a picture of the purchase contract. These examples show that there is no standardised procedure to link a user account to a vehicle.

2.2. The underlying authorisation protocol

From a technical point of view, Open Authorisation 2.0 (OAuth 2.0) is used for API authorisation. As shown in Fig. 1, the roles defined by the standard RFC 6749 can be described in the setting of VAAs as follows (Hardt, 2012): The resource owner, the entity that is capable of granting access to a protected resource is the user. The data is provided by the resource server, the backend of the OEM. The vehicle communicates its shared data to the backend of the manufacturer via a GSM module. Some of this data is provided via the API to the authorised user through the VAA. The VAA is the client, that wants to access protected resources provided by the resource server.

The VAA can request desired permissions from the authorisation server. The authorisation protocol is run in the background through the manufacturer's authorisation server. The user account and the VIN of the vehicle are decisive for the authorisation. Scopes are used to define the access levels and limit access through the use of a token. The authorisation server authenticates the resource owner and delivers access tokens and refresh tokens. With this delivered access token, the client can then access the shared data allowed by the resource owner from the resource server. In addition to requests to the resource server for information, some remote control features provide a direct connection to the vehicle. Therefore, the device must either establish a connection to the backend with the application via an API or directly connect to the vehicle, for example via Bluetooth Low-Energy (not shown in the figure). The refresh token can be used to request a new access token and optionally a new refresh token from the authorisation server. This is necessary if the access token has expired or become invalid. The refresh token also has limited validity, which is usually set higher than that of the access token. As a further grant type, the use of JSON Web Tokens (JWT) is also permitted in the standard RFC 7523 (Jones et al., 2015).

[1] https://github.com/FHMS-ITS/Grand_Theft_API.

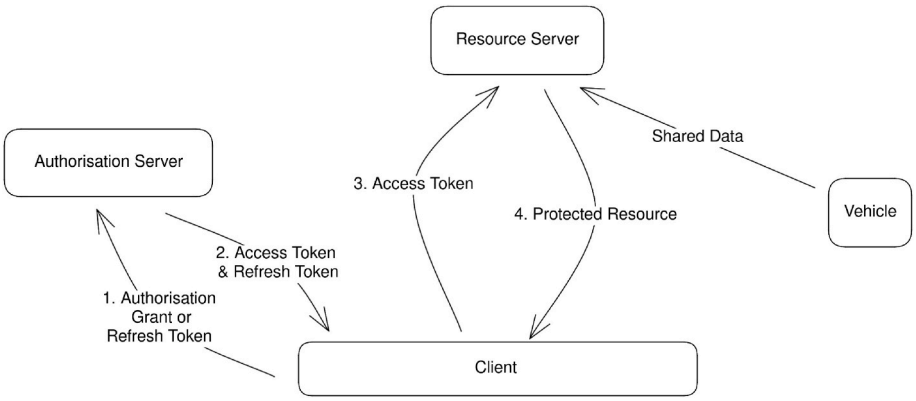

Fig. 1. Visualisation of the OAuth 2.0 standard in the scope of vehicle assistant apps and the resulting data flow.

3. Method

We now describe a generic method to identify and access data stored by vehicle manufacturers in the cloud. In accordance with security literature best practices, we describe the *scenario* we consider for the usage of the developed methods in Section 3.1. In Section 3.2, we present a method to identify the capabilities of the VAAs. We illustrate a method to identify the technical needs to access the manufacturer API. In Section 3.3, we describe a generic method to identify the tokens on an Android smartphone to access the resource server. We set out the requirements to address the forensic soundness of the scenario in Section 3.4.

3.1. Scenario

The scenario in this paper assumes investigators have lawful access to suspects' smartphones. The investigators aim to obtain access to telemetry data from vehicles, including timestamps of driving instances, geographical locations, periods of rest and even the ability to utilise remote control functionalities. While we assume that the investigators strictly adhere to the respective local laws, they stand against the standard security procedures introduced by smartphone vendors to protect the data of their customers.

For locked smartphones, the ability of investigators to directly access telemetry data or relevant credentials for such data depends on the security measures implemented on the particular device. Note that some investigators have access to advanced smartphone unlocking tools (Celebrite GmbH, 2023). Access to unlocked smartphones usually requires that the smartphones be snatched from the suspects while the suspects are actively using them. Other scenarios include suspects having insecure passcodes, the suspect reusing known credentials from other services (credential stuffing), or the investigators acquiring the credentials through inventory data requests, interrogations or other side-channels (Cronin et al., 2021; Shumailov et al., 2019; Yang et al., 2022; Nerini et al., 2023).

3.2. Identification of the APIs and their capabilities

To identify the capabilities of VAAs and the information needed to use the different APIs we considered different Open Source Intelligence (OSINT) sources. We searched for official app descriptions and official API documentation as the best starting point. Furthermore, we looked for community-driven projects related to the APIs. Our findings are broad but do not claim any form of completeness and might lack technical detail.

To verify the advertised capabilities, there is a need for a technical way to identify the vehicle assistant APIs. Sniffing traffic between client and server should not be effective because of Transport Layer Security (TLS). Therefore, a Man-In-The-Middle (MITM) setup was used. To read the cleartext messages within the traffic we had to address further challenges. First of all, to decrypt the encrypted traffic the client has to trust the certificate authority of the used open source software *mitmproxy* (Hils, 2023).

Therefore we prepared an Android with a system certification authority certificate of our MITM proxy so that we could decrypt TLS-protected traffic on the fly (Hils, 2023). In doing so, write access to the system partition of the Android operating systems was needed. We solved this issue by gaining root access to the Android operating systems. This resulted in further challenges because of different app security best practices like root detection and certificate pinning (Android Developers, 2023). Before an app starts, it does several checks if the Android device it is running on is rooted or not. This could result in a notification to the user and furthermore in closing the app. With the Frida toolkit (Ravnås, 2023) and the possibility to hijack a specified function, it was possible to bypass the implemented root detection. A further challenge we needed to address was the mentioned certificate pinning which itself prevents MITM scenarios. Through certificate pinning, the application only trusts a limited set of certificates to ensure secure client-server communication using TLS. To let the VAA trust our MITM proxy certificate we also hijacked the respective function with the Frida toolkit and bypassed it. All in all the workflow with our *mitmproxy* setup was as follows:

1. Prepare an Android device by rooting and installing the Frida toolkit
2. Download and install the VAA from Google Play
3. If necessary, bypass root detection and certificate pinning with Frida
4. Login to an account that is connected to a car
5. Test every feature of the app
6. Analyse the network traffic to understand the API

The analysis of the network traffic makes it possible to determine in-depth information such as the end points, header information and API requests.

We assume that the APIs of the VAAs are identical between the applications of the different operating systems. However, a similar setup for analysing VAAs for other operating systems such as iOS is also conceivable.

3.3. Token identification

To access protected resources from the resource server a valid access token or a refresh token is necessary. By design, these tokens are stored on the client. Therefore we had to identify the location of the required tokens. Based on the digital forensics approach presented by Mohay et al. (2003), we first had to secure and then analyse the data. We used a rooted Android device to log in to a previously registered account of a

VAA. After that, we acquired the internal storage of the rooted Android device with the Android Debug Bridge (ADB). We then analysed the acquired data with freely available Linux-based tools to find the generated tokens of the VAAs.

3.4. Forensic soundness

The evidential value of digital evidence is based on many aspects of the evidence acquisition process (Fröwis et al., 2020). This is particularly relevant in cloud acquisition because of the uncertainty of data access through reliance on the cloud provider (Hammer et al., 2023). By accessing the API endpoints directly, the data integrity can be kept as high as possible. A forensically sound API-based acquisition requires a transparent execution. To meet these requirements, all relevant parameters, including headers and HTTP methods are readily visible for every acquisition. This enables traceability of the complete acquisition process and results in extensive logging. Direct communication with the API allows access to the raw data of the resource server and is independent of an application's graphical user interface. Furthermore, direct API communication surpasses the capabilities of a smartphone app with its possibility to have complete control over the communication between the client and the resource server, the ability to work with tokens from external sources and access data periodically if necessary. In the next section, we present a POC with the described requirements.

4. Proof of concept

With the knowledge we obtained as described in Section 3.2, we developed a Python tool that serves the purpose of accessing data from the vehicle assistant APIs. The POC, called `gta.py`, attempts to perform an API-based cloud acquisition while taking the requirements of Section 3.4 into account. Acting as a client in the OAuth 2.0 standard, `gta.py` supports several car manufacturers' APIs, including those from BMW, Dacia, Ford, Hyundai, Mercedes and Tesla.

The command line tool is divided into multiple modules, outlined in the accompanying documentation. Using the `gta.py` menu, users are guided through the featured manufacturers and their capabilities. The source code of each vehicle class has three main sections: authorisation requests, API requests and helper functions. Authorisation requests are responsible for obtaining an access token either through the use of credentials or an already available refresh token and are therefore communicating with the authorisation server. By design, `gta.py` is independent of the source of the tokens. Recurring requests for obtaining an access token using the refresh token ensure long-term accessibility. API requests contain the information to request protected resources from the resource server. The necessary headers are provided by the helper functions. The requests are provided by the class ReqCreator, while the class LogReqs logs into specified files. This makes it possible to track which data was requested from which resource server, using which access data, at what time and with which detailed request. The requested data are also enriched with the corresponding time stamp as well as a hash value of the response and are stored in JSON format.

With its flexible architecture and comprehensive template, `gta.py` facilitates expansion and integration of additional manufacturers. To get a more detailed output the user can opt to use the verbose option. This will print information about accounts, VINs and the validity of access- and refresh-tokens into the command–line interface. In order to execute the `gta.py` tool, a Python 3 runtime environment or higher and pip3 is required. The tool is designed to be cross-platform, so there are no special requirements regarding the operating system.

5. Experimental setup

The following experimental setup allows the verification of the potentially accessible data through the manufacturer's VAAs. We define generalised data categories in Section 5.1 to improve comparability with the data the manufacturer potentially makes accessible. To validate the potentially accessible data category, we acquired data using the API with our POC. For each of the considered applications and connected vehicles, we describe the API-based acquired data in Section 5.2. The API-based acquired data, which is labelled *verified accessible* data from now on, can then be compared with the *potential accessible* data.

5.1. Categories of data

Using OSINT we identified the potentially accessible data described by advertising texts from Apple's and Google's app stores. Despite considerable efforts, we could not find publicly available documentation for the vehicle assistant APIs. However, several community-driven projects on the APIs such as the ioBroker project TA2k (2023) or the tesla-api project by Dorr (2023) exist which add more technical details to the advertising texts. The identified potentially accessible data of the VAAs can be generalised in the following categories.

5.1.1. User data

Since a user account for every considered VAA is needed, all of them provide data that is linked to the user. The available data depends on the user's input. At a minimum, an email address or a phone number which is required for the authorisation as described in Section 2.1 should be provided. It can be assumed that further personal data will be requested by the manufacturers to identify their customers.

This data category links the user or users to the vehicle and could help gather further information about a suspect. Moreover, data from this category might be interesting in scenarios where further suspects are unknown but there is access to the suspect's vehicle. In a scenario where a token is used to acquire the data, this data category could obtain parts of the credentials such as email address or phone number.

5.1.2. Descriptive data

All considered VAAs provide information that describes or identifies the connected vehicle. The descriptive data ranges from vehicle colour, model description and body type to remote connection capabilities. Furthermore, it can be assumed that more technical identifiers like the VIN are provided.

Data from this category helps to identify a vehicle in a scenario where a suspect's mobile device was forensically analysed but the vehicle is missing. The fact that one user account can be connected with multiple vehicles might help to identify further vehicles.

5.1.3. Current status

All of the considered VAAs provide a current state of the vehicle. While the provided information depends on the capabilities of the connected car, information like the current mileage, the current location and fuel or charging state in combination with an expected mileage are advertised by all manufacturers. Some of the manufacturers provide further information including, control messages of oil and tires, the state of all doors (locked/unlocked) and upcoming inspections.

From the investigator's viewpoint the continuous acquisition of the current status data leads to surveillance of the vehicle connected to the API and might open up new possibilities in observation scenarios.

5.1.4. Health data

The health data category includes information about the tire pressure, oil status, warranty information as well as historical and upcoming inspections. In addition, some apps allow users to link a workshop to their account to book repair work or inspections through the app.

The information summarised by this data category might be interesting for investigation if it is possible to find indications of a poorly maintained vehicle and might help to identify the habits of the user.

5.1.5. Remote control features

Remote control functions are heavily advertised but highly depend on the connected vehicle's capabilities. Therefore a huge variety in this data category can be assumed. The featured remote capabilities might be locking and unlocking doors, changing the air conditioning system, turning on the light, honking or even starting the engine and leaving a parking space.

Accessing the remote control features of a suspect's vehicle might be the most invasive scenario but could be considered in hit-and-run scenarios.

5.1.6. Trip data

Few apps offer the user a trip log feature. The data stored in the logbook contains historical information such as the start and end point of a journey or, in some cases, a complete record of the journey, statistics such as speed and distance travelled, and whether the journey was private or business. All of these data are summarised by the trip data category.

Accessing trip information gives the investigator great insight into where the vehicle might have been and at what time. This information can be very important in checking a suspect's alibi or the discovery of new crime scenes.

5.1.7. Charging and refuelling data

Some of the considered manufacturers provide charging and refuelling information via the VAAs. More specifically, the applications advertise that the users can track the amount and cost of fuel for statistical purposes. In addition, hybrid or electric-powered vehicles also provide scheduled charging features.

The data generalised by this data category might help the investigator to identify the habits of the user and can hold position data of charging/refuelling points.

5.2. Measurement set

A representative set of manufacturers and their applications was considered based on a statistic on the number of new passenger car registrations in the European Union by manufacturer in 2021 and 2022 (ACEA, 2023). Without claim of completeness, we extended this list with several additional manufacturers. The complete list of 23 manufacturers for which the API was investigated is shown in Table 1. Furthermore, the table contains the provided vehicles for our testing purposes. We did not have sufficient resources to obtain a car from all manufacturers so cars were investigated opportunistically with consent from friends and colleagues. For each VAA of a provided vehicle we identified the token location on Android 13 running on a Samsung Galaxy S9 Plus.

Table 1
List of the considered manufacturers and provided vehicles with the apps and their versions.

Manufacturer	Vehicle	App Name	App Version
Audi	–	myAudi	4.16.0
BMW	X1 U11	My BMW	3.3.1
Citroen	–	My Citroen	1.38.1
Dacia	Spring	My Dacia	5.1.9
Fiat	–	Fiat	1.67.3
Ford	Kuga '13	FordPass	4.26.0
Honda	–	My Honda+	5.16.11
Hyundai	Tucson NX4	Hyundai Bluelink Europe	2.0.9
Kia	–	Kia Connect	2.1.9
Land Rover	–	Land Rover Remote	2.14.0
Mazda	–	MyMazda	8.4.3
Mercedes	GLA H247	Mercedes me	1.32.0
Mini	–	Mini	3.3.1
Nissan	–	NissanConnect Services	2.4.0
Opel	–	myOpel	1.38.1
Peugeot	–	MyPEUGEOT	1.38.1
Renault	–	My Renault	5.1.9
Seat	–	Seat Connect App	1.10.0
Skoda	–	MySkoda	6.1.1
Tesla	Model 3	Tesla	4.20.70
Toyota	–	MyT by Toyota	4.16.0
Volkswagen	–	We Connect	5.15.2
Volvo	–	Volvo Cars	5.26.1

6. Experimental results

The experimental results focus on the defined data categories. The results are divided into *potential accessibility* in Section 6.1 and *verified accessibility* in Section 6.2. A more detailed analysis of the acquired data is provided in Section 6.3, followed by the identified token locations in Section 6.4. For better comparability, the provided result tables in the following subsection are formatted identically. The columns with the defined data categories are sorted according to the frequency of occurrence. Because the usage of the considered VAA requires a user account and connected vehicle, they all provide user and descriptive data. Furthermore, all considered VAAs provided at minimum a current status of the vehicle as well as health data and remote control features in various specifications. In contrast to user and descriptive data, the latter categories increasingly depend on the linked vehicle. The assumption was made that fewer vehicles possess these features. Some manufacturers offer trip data and even fewer often charging or refuelling data to the user of the VAA. During the experiment execution we noted that these categories are sometimes outsourced to another app and are therefore less likely to be seen.

6.1. Potential accessibility

To demonstrate the potential of the API-based acquisition of vehicle cloud data, we present an overview of the *potential accessible* category for each manufacturer in Table 2. A checkmark represents a set of data from the data category that the manufacturer potentially provides via the VAA for its vehicle. A checkmark in brackets is set when the manufacturer potentially provides a category through a specific app but not through the general app of the manufacturer. If the manufacturer does not provide a category through an app, the cells are left empty.

6.2. Verified accessibility

Using the POC `gta.py`, an API-based acquisition was performed with the provided vehicles. The acquired data shown in Table 3 verifies the *potential accessible* category of a manufacturer. In contrast to the *potential accessibility*, the *verified accessibility* of a data category is given if data can be technically accessed for *some* specific car type.

The distinction between *potential* and *verified accessibility* results in a different consideration of accessibility. Therefore *potential accessibility* does not imply *verified accessibility* or vice versa. A cell is left empty if we could not access the data of the category. This might be the case if the provided vehicle does not support a necessary feature for example.

6.3. Individual results

In the following, a selected set of the API-based acquired data is explained in more detail. The data was acquired through `gta.py`.

6.3.1. User data

All considered manufacturers communicated data required for authentication such as username, email address or phone numbers. Some of the considered APIs such as the one provided by Mercedes communicate further information such as first and last name, address, birthday or height. Data from further individuals could also be collected if the main user authorised their use of the vehicle via the Hyundai API. The remote control capabilities of the authorised users through the analysis of the BMW API were also noted. The analysis of the acquired data from the provided Tesla showed a key–value pair that might

Table 2
Potentially accessible data categories for each considered manufacturer, identified with the help of OSINT.

Manufacturer	User Data	Descriptive Data	Current Status	Health Data	Remote Control Features	Trip Data	Charging and Refuelling Data
Audi	✓	✓	✓	✓	✓	(✓)	
BMW	✓	✓	✓	✓	✓		✓
Citroen	✓	✓	✓	✓	✓	✓	
Dacia	✓	✓	✓	✓	✓		
Fiat	✓	✓	✓	✓	✓		✓
Ford	✓	✓	✓	✓	✓	✓	✓
Honda	✓	✓	✓	✓	✓	✓	
Hyundai	✓	✓	✓	✓	✓	✓	(✓)
Kia	✓	✓	✓	✓	✓	✓	(✓)
Land Rover	✓	✓	✓	✓	✓	✓	(✓)
Mazda	✓	✓	✓	✓	✓		
Mercedes	✓	✓	✓	✓	✓	(✓)	(✓)
Mini	✓	✓	✓	✓	✓		✓
Nissan	✓	✓	✓	✓	✓	✓	(✓)
Opel	✓	✓	✓	✓	✓	✓	✓
Peugeot	✓	✓	✓	✓	✓	✓	✓
Renault	✓	✓	✓	✓	✓		
Seat	✓	✓	✓	✓	✓	✓	(✓)
Skoda	✓	✓	✓	✓	✓	(✓)	(✓)
Tesla	✓	✓	✓	✓	✓		✓
Toyota	✓	✓	✓	✓	✓	✓	
Volkswagen	✓	✓	✓	✓	✓	(✓)	(✓)
Volvo	✓	✓	✓	✓	✓		

✓ potential accessibility
(✓) potential accessibility by separate app
no accessibility

Table 3
Verified accessible data categories for each provided vehicle, which were acquired with the help of `gta.py`.

Manufacturer	User Data	Descriptive Data	Current Status	Health Data	Remote Control Features	Trip Data	Charging and Refuelling Data
BMW	✓	✓	✓	✓			✓
Dacia	✓	✓	✓	✓			
Ford	✓	✓	✓	✓			
Hyundai	✓	✓	✓			✓	
Mercedes	✓	✓	✓				
Tesla	✓	✓	✓	✓	✓		

✓ verified accessibility
no verified accessibility

contain not yet delivered vehicles. In response to the correlating request done by `gta.py`, Dacia added a creation timestamp of the connected user profile as well as a last modified timestamp.

6.3.2. Descriptive data

Besides the expected model description, colours and capabilities of the vehicles the acquisition of the Integrated Circuit Card Identifier (ICCID) and an identifier of the installed head unit of the BMW X1 by using the `gta.py` POC was also possible. To get a better understanding of the appearance, Dacia and Ford for example responded with URLs with stock pictures of the connected vehicles. Furthermore, the nicknames of the vehicles of Ford and Tesla could be acquired.

6.3.3. Current status

Current status data such as the current mileage or the expected range was accessible by all tested APIs. In addition, the state of each window and door as well as the charging state of the provided BMW was accessible. The provided Dacia and Tesla also communicated information about the heating, ventilation, and air conditioning system. The current location of the vehicle with the verified geographic coordinate parameters was also accessible via the API of BWM, Dacia, Hyundai and Tesla. All the data were provided in combination with a timestamp.

6.3.4. Health data

Some of the provided vehicles like Tesla communicated the current tire pressure. As graduation of this data, the BMW API send a control message to monitor tire pressure and oil levels. Furthermore, a list of upcoming services with a time or mileage limit for the BMW was accessible. The corresponding response of the Mercedes API also includes the chosen contract workshop.

6.3.5. Remote control features

During our research, the remote control features of the Tesla Model 3 were tested and verified. It was possible to open the charging port of the vehicle. More interestingly, the door-locking and unlocking features are accessible via `gta.py`.

6.3.6. Trip data

Only the provided Hyundai featured trip data via API. It was possible to access this category via `gta.py` including the journey time, distance travelled, average and maximum speed from the last three months. Information about the travel route with location data could not be determined. A comparison with the VAA showed that it also did not provide the information. Furthermore, it could be noted that Mercedes has created a separate app for trip data. With the help of the experimental setup, it was possible to determine that the separate logbook app calls a different API than the Mercedes me app.

6.3.7. Charging and refuelling data

The provided BMW X1, which is a hybrid vehicle, logged its charging sessions. These logs are accessible via the API. A timestamp, the charging duration and a resolved address of the charging location for

every charging process as well as other charging statistics like assumed cost and charged kilowatt-hours were accessible. It was possible to access these data points for the whole lifecycle of the car.

6.4. Token location

To identify the token location of the VAAs, the dataset of the Android device was analysed. It can be assumed that the access and refresh tokens can be found in the corresponding app folder under `/data/data/` in the Android file system. An extraction of the access and refresh tokens could be performed from a logfile inside the My BMW app folder `de.bmw.connected.mobile20.row`. The tokens of the BMW API were stored in cleartext while the other apps stored them in encrypted or obfuscated form. It can be assumed that the access and refresh tokens of the Ford application are stored inside the `shared_prefs` folder in an XML file called `encryptedValues`. This file contains key–value pairs where the keys are labelled as `REFRESH_TOKEN_KEY` and `CUSTOMER_AUTH_TOKEN_KEY`. The related values were encrypted or obfuscated. The analysis of the Mercedes me folder also suggests that the tokens are stored inside the `shared_prefs` folder within the XML file `account_prefs`. Both the key and values were stored in encrypted or obfuscated form. The Dacia, Hyundai and Tesla access and refresh token locations are presumably stored in encrypted databases. These challenges can be addressed with `gta.py` by creating the access and refresh tokens with the credentials.

7. Evaluation and discussion

The results demonstrated that it is possible to perform an API-based acquisition of vehicle cloud data using credentials as well as access tokens. The capability of this method is indicated by the identified potentially accessible data categories of manufacturers. It can be assumed that its capability will be strengthened by the progress of digitisation in the field of vehicles.

The potential of an API-based acquisition of vehicle cloud data could be proven by the *verified accessible* data categories of six provided vehicles. While the tested APIs provided expectable user data, this category is no less relevant. On the one hand, the data obtains parts of the credentials if the investigator uses the access token of a VAA. On the other hand, further authorised users with access to the vehicle could be identified by the API-based acquisition of the provided Hyundai Tucson. The description of the head unit might help during the operational preparation to check and prepare the toolsets to acquire the data of the head unit, for example. Furthermore, the communicated ICCID within the descriptive data category could be compared with the data of the GSM cell of a crime scene to prove a suspect's alibi wrong. The results provide evidence that a continuous acquisition of the current status information leads to surveillance of the vehicle connected to the API. From the investigator's perspective, this could open up new possibilities in observation scenarios. In addition, the data of the current state was provided with a timestamp which is not shown in the user interface of the app. This is another argument in favour of an API-based acquisition and not a prepared system with the same application. The verified accessible health data might be an indicator of a poorly maintained vehicle and could support a thesis in the investigation of an accident. The presented remote control feature with the capability to lock and unlock doors might be worth considering in a hit-and-run scenario. From the investigator's perspective, the trip data give great insights into a suspect's location at a specific time. While this can be very important to post mortem investigations, the results could not verify the accessibility of expected location data but statistical data. In this way, it is possible to determine on which day at what time the vehicle was moved at what average and maximum speed. However, it is not possible to determine the location. Interestingly, the connection between refuelling or charging information and a geolocation as seen in the acquisition of the BMW X1 might be helpful to prove someone's alibi at a crime scene.

The analysis of the Android file system to identify the location of the access and refresh token provide evidence of the possibility of accessing vehicle cloud data without knowing the credentials of a suspect. Forensic analysis is made more difficult by the encryption and obfuscation of the tokens in the smartphone file system. Given the fact that access to the data offers great potential from the point of view of investigators, it is not surprising that storing the tokens as cleartext is not the best practice. Investigating authorities could potentially bypass these obstacles by getting the necessary tokens legally issued by the manufacturers.

The limitations of the present studies include the strong dependency on vehicle capabilities as well as the capabilities of the provided APIs. A further limitation of the study is the limitation of cloud acquisition because it relies on the provider to hand the unchanged data over. Moreover, only a spot check could be carried out due to the small number of vehicles available.

8. Conclusion and future work

The findings of this paper show that data generated by modern vehicles are not only stored within the vehicle itself but are also made accessible by manufacturer's APIs. The listing of the potentially accessible data category of the 23 considered manufacturers indicate the capability of the data made accessible by the described method. The API-based acquisition and analysis of vehicle cloud data show that the accessible data is diverse and addresses various guiding forensic questions. It could demonstrate that permanent access to the VAA account can be achieved by suspects' credentials or temporary access to suspects' smartphones.

The present results not only confirm the research question about the usefulness of API-based acquisition for digital forensics in the field of vehicles, but also offer the opportunity to think about new tactical applications in a hit-and-run scenario through the accessible remote control features. It can be assumed that in the future more remote control features will be available in more vehicles and the amount of accessible data will increase. In summary, the API-based forensic acquisition and analysis of vehicle cloud data provides insights to be considered in future digital forensic investigations in the field of vehicles.

Further research is needed to identify the token location on iOS and to make these encrypted or obfuscated tokens accessible for the discussed API-based forensic acquisition. To use these newly provided possibilities for forensic purposes for legal proceedings it is necessary to validate the integrity and therefore the meaningfulness of the data. As a result, future forensic research should address the user's access and manipulation capabilities. Moreover, trucks and motorbikes should also be considered in the future. To point out which data sources can address which guiding questions a comparison of vehicle, smartphone and cloud data is necessary.

References

ACEA, 2023. Anzahl der Neuzulassungen von Personenkraftwagen in der Europäischen Union nach Hersteller in den Jahren 2021 und 2022 [graph]. URL: https://de.statista.com/statistik/daten/studie/1197750/umfrage/marktan teile-ausgewaehlter-autohersteller-in-europa/.

Berla, 2023. Vehicle forensics. https://berla.co/category/vehicle-forensics/.

Celebrite GmbH, 2023. Celebrite UFED. https://cellebrite.com/.

Cronin, P., Gao, X., Yang, C., Wang, H., 2021. Charger-surfing: exploiting a power line side-channel for smartphone information leakage. In: Bailey, M., Greenstadt, R. (Eds.), 30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021. USENIX Association, pp. 681–698. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/cronin.

Developers, Android, 2023. App Security Best Practices. https://developer.android.com/topic/security/best-practices.

Dorr, T., 2023. Tesla JSON API. URL: https://github.com/timdorr/tesla-api.

Ebbers, S., Ising, F., Saatjohann, C., Schinzel, S., 2021. Grand theft app: digital forensics of vehicle assistant apps, 1–30:6. In: Reinhardt, D., Müller, T. (Eds.), ARES 2021: the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, August 17-20, 2021, ACM, p. 30. https://doi.org/10.1145/3465481.3465754, 10.1145/3465481.3465754.

Fröwis, M., Gottschalk, T., Haslhofer, B., Rückert, C., Pesch, P., 2020. Safeguarding the evidential value of forensic cryptocurrency investigations. Digit. Invest. 33, 200902 https://doi.org/10.1016/j.fsidi.2019.200902, 10.1016/j.fsidi.2019.200902.

Gomez Buquerin, K.K., Hof, H.J., 2021. Identification of automotive digital forensics stakeholders. In: SECURWARE 2021 : the Fifteenth International Conference on Emerging Security Information, Systems and Technologies.

Gomez Buquerin, K.K., Corbett, C., Hof, H., 2021. A generalized approach to automotive forensics. Digit. Invest. 36 (Suppl. ment), 301111 https://doi.org/10.1016/j.fsidi.2021.301111, 10.1016/j.fsidi.2021.301111.

Gomez Buquerin, K., Bayerl, D., Hof, H.J., 2022. Digital forensics investigation of the tesla autopilot file system. In: Yee, G. (Ed.), SECURWARE 2022: the Sixteenth International Conference on Emerging Security Information, Systems and Technologies. IARIA, pp. 82–87.

Hammer, A., Ohlig, M., Geus, J., Freiling, F., 2023. A functional classification of forensic access to storage and its legal implications. In: Proceedings of the 12th International Conference on IT Security Incident Management & IT Forensics.

Hardt, D., 2012. The OAuth 2.0 authorization framework. RFC 6749. URL: https://www.rfc-editor.org/info/rfc6749.

Hils, M., 2023. About certificates. URL: https://docs.mitmproxy.org/stable/concepts-certificates/.

Hof, H., Bayerl, D., Gomez Buquerin, K.K., 2021. Überwachung in modernen Fahrzeugen. Datenschutz Datensicherheit 45, 399–403. https://doi.org/10.1007/s11623-021-1459-5, 10.1007/s11623-021-1459-5.

Jacobs, D., Choo, K.R., Kechadi, M.T., Le-Khac, N., 2017. Volkswagen car entertainment system forensics. In: 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, Australia, August 1-4, 2017, IEEE Computer Society, pp. 699–705. https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.302, 10.1109/Trustcom/BigDataSE/ICESS.2017.302.

Jones, M.B., Campbell, B., Mortimore, C., 2015. JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. https://doi.org/10.17487/RFC7523. RFC 7523. URL: https://www.rfc-editor.org/info/rfc7523.

Mahr, A., Serafin, R., Grajeda, C., Baggili, I., 2022. Auto-parser: Android auto and apple carplay forensics. Digital Forensics and Cyber Crime. https://doi.org/10.1007/978-3-031-06365-7_4.

Mohay, G., Anderson, A., Collie, B., de Vel, O., McKemmish, R., 2003. Computer and Intrusion Forensics. Artech House, United States of America.

Nerini, M., Favarelli, E., Chiani, M., 2023. Machine learning for PIN side-channel attacks based on smartphone motion sensors. IEEE Access 11, 23008–23018. https://doi.org/10.1109/ACCESS.2023.3253288, 10.1109/ACCESS.2023.3253288.

Pawlaszczyk, D.D., Bochmann, M., Engler, P., Klaver, C., Hummert, C., 2022. API-based evidence acquisition in the cloud - a survey. Open Research Europe 2, 69. https://doi.org/10.12688/openreseurope.14784.1.

Priyanka, V.S., Kumar, S.S., Kumar, S.V.J., 2021. A forensic methodology for the analysis of cloud-based Android apps. In: 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS), IEEE. https://doi.org/10.1109/fabs52071.2021.9702691.

Ravnås, O.A.V., 2023. Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers. URL: https://frida.re.

Roussev, V., Barreto, A., Ahmed, I., 2016. Api-based forensic acquisition of cloud drives. URL:. In: Peterson, G.L., Shenoi, S. (Eds.), Advances in Digital Forensics XII - 12th IFIP WG 11.9 International Conference, New Delhi, IndiaJanuary 4-6, 2016, Revised Selected Papers. Springer, pp. 213–235. https://doi.org/10.1007/978-3-319-46279-0_11, 10.1007/978-3-319-46279-0_11.

Shumailov, I., Simon, L., Yan, J., Anderson, R., 2019. Hearing Your Touch: A New Acoustic Side Channel on Smartphones. CoRR abs/1903.11137. URL: http://arxiv.org/abs/1903.11137. arXiv:1903.11137.

TA2k, 2023. iobroker.mercedesme. URL: https://github.com/TA2k/ioBroker.mercedesme.

Yang, B., Chen, R., Huang, K., Yang, J., Gao, W., 2022. Eavesdropping user credentials via GPU side channels on smartphones, in: Falsafi, B., Ferdman, M., Lu, S., Wenisch, T.F. (Eds.), ASPLOS '22: 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, 28 February 2022 - 4 March 2022, ACM. pp. 285–299. URL: https://doi.org/10.1145/3503222.3507757, doi:10.1145/3503222.3507757..