UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Cybersecurity Curriculum

Md Rashadul Islam

Bluetooth-based Tracking Devices: Extraction and Analysis of Digital Forensic Artifacts from Android Applications

Master's Thesis (30 ECTS)

Supervisor(s): Hayretdin Bahsi, PhD
Raimundas Matulevičius, PhD

Tartu 2024

Bluetooth-based Tracking Devices: Extraction and Analysis of Digital Forensic Artifacts from Android Applications

Abstract:

The extraction and analysis of data from user applications installed on mobile devices is the subject of mobile forensics, a rapidly expanding subfield of digital forensics. The need for digital forensics is becoming increasingly apparent with the smartphone market's accelerated expansion and the IoT device industry's rising innovation. IoT devices are, therefore, compatible with smartphones and have become essential. Mobile phones are typically owned by a single individual, making them a valuable source of personal data for forensic examination. Depending on the company's technology and the environment, these Bluetooth tracking devices, widely known as Bluetooth beacons (transmitters), can be linked with the mother device and broadcast positions to their identifier within the range. This beacon uses BLE to transmit a UUID picked by an application or operating system. When it is used with a smart device with a beacon as a tracker application, it can save additional user data with the aid of a user application through Bluetooth and a Wi-Fi or internet connection. As a result, smartphones can serve as a source of forensic evidence when there is a case file about a stalking attack or cyberstalking in court. Therefore, we will examine what artifacts can be gleaned from Bluetooth tracking apps and whether this information can be used as evidence. Based on Android Bluetooth (BLE) trackers, including the Smart Tag Plus, HTC Fetch, Chipolo Classic, and Tile Slim for Samsung Galaxy A13 devices, we analyzed five Bluetooth tracking applications to demonstrate the potential artifacts obtained and how the data extraction approach was carried out. We also suggest which extraction technique is most important for an inquiry to bring more artifacts. Finally, we give a flowchart of our steps to gather information from a Bluetooth-based tracking device.

Keywords:

Digital Forensics, Cyberstalking, Stalking, Bluetooth tracker, BLE beacon, Transmitter, Smart Tag, IoT.

CERCS: P170, Computer science, numerical analysis, systems, control

Bluetooth-põhised jälgimisseadmed: digitaalsete kohtuekspertiisi artefaktide eraldamine ja analüüs Androidi rakendustest

Lühikokkuvõte:

Andmete hõive ja analüüs kasutajarakendustest, mis on paigaldatud mobiilseadmetesse, on teemaks mobiilses kohtuekspertiisis, digitaalse kohtuekspertiisi kiiresti kasvavas alamvaldkonnas. Vajadus digitaalse kohtuekspertiisi järele muutub üha ilmsemaks nutitelefonide turu kiire laienemise ja IoT (Internet of Things, asjade internet ehk nutistu) seadmete tööstuse kasvava innovatsiooni tõttu. Seetõttu on IoT-seadmed nutitelefonidega ühilduvad ja nad on muutunud olulisteks. Mobiiltelefonidel on tavaliselt üks omanik, mis teeb need kohtuekspertiisi uuringute seisukohalt väärtuslikuks isiklike andmete allikaks. Sõltuvalt konkreetse ettevõtte tehnoloogiast ja keskkonnast saab Bluetoothi jälgimisseadmed, tuntud kui Bluetoothi majakad (saatjad), ühendada emaseadmega ja nad võivad edastada enda positsioone neid identifitseerivale seadmele nende levialas. Selline majakas kasutab BLE (Bluetooth Low Energy) tehnoloogiat edastamaks UUID (Universally Unique Identifier, universaalne unikaalne identifikaator), mille rakendus või operatsioonisüsteem on määranud. Kui seda kasutatakse nutiseadmega, millel on jälgimisrakenduseks majakas, saab see kasutajarakenduse abil Bluetoothi ja Wi-Fi või internetiühenduse kaudu salvestada täiendavaid kasutajaandmeid. Selle tulemusena saavad nutitelefonid olla kohtuekspertiisis tõendite allikaks, kui kohtus menetletakse jälitamise või küberjälitamise juhtumeid. Seetõttu uurime, milliseid andmeobjekte saab Bluetoothi jälgimisrakendustest koguda ja kas seda teavet saab kasutada tõendina. Võttes aluseks Androidi BLE jälgimisseadmed, sealhulgas Smart Tag Plus, HTC-Fetch, Chipolo Classic ja Tile Slim Samsung Galaxy A13 seadmetes, analüüsisime viit Bluetoothi jälgimisrakendust, et näidata võimalikke saadavaid andmeid ja viisi, kuidas lähenesime nende andmete kättesaamisele. Samuti soovitame, milline andmete hõivamise meetod on kõige olulisem uurimaks, kuidas koguda rohkem andmeid. Lõpuks anname vooskeemi meie sammudest teabe kogumiseks Bluetoothi- põhisest jälgimisseadmest.

Võtmesõnad:

Digitaalne kohtuekspertiis, küberjälitamine, jälitamine, Bluetoothi jälgija, BLE-majakas, saatja, nutikas silt, asjade internet

CERCS:P170, Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1 Introduction

Smartphone users increasingly use Bluetooth tracker applications to keep track of their personal belongings, including baggage, wallets, and keys. The worldwide personal smart tracker is growing rapidly. According to Acumen Research and Co., the value of the IoT tracker will be $ 1.65 billion within 2030 [1].These apps function by pairing with tiny Bluetooth devices, like tracking tags, which emit a signal that the app can pick up and use to find the lost object. Unfortunately, even though these apps can help track personal property because they gather and transmit location data, they have the potential to be used to track users, endangering their security and privacy [2]. In such a scenario, a forensic examination of Bluetooth tracker apps can shed important light on the threats to one's privacy and security from using them. This kind of analysis extracts data from mobile devices and examines indications of app data storage, network traffic, user interactions, and potential vulnerabilities. Unfortunately, few studies have been done on Bluetooth tracker app forensic analysis thus far because it is a relatively new field of study. However, the limited research that is now accessible has brought to light several security and privacy issues with these apps. For instance, according to a 2019 study by Yue Zhang et al., some Bluetooth tracker apps use insecure communication protocols, potentially exposing users' data to hackers [3]. Given the rising popularity of Bluetooth tracker apps and possible security threats related to their use, there is a need for more excellent studies in this area, with an emphasis on exploring possible forensic artifacts and privacy problems. This thesis intends to examine Bluetooth tracker apps depending on the tracker for Android devices thoroughly for more user related data. This thesis explores the possibility of the information that can be obtained from four applications tested on Android devices Samsung Galaxy A13. These Bluetooth apps are a relatively recent addition to the software landscape regarding how IoT devices communicate with trackers. These programs' ability to keep certain smartphone data types may be crucial for criminal investigations like cyberstalking, theft and loss recovery, accident investigation, missing person case, corporate espionage. Most mobile forensic research has concentrated on social media, messaging tools, automotive maintenance applications, mobile banking applications, cryptocurrency wallets, etc. Since these Bluetooth tracker programs give their users access to the precise and occasionally approximate position of the tracker device, they also keep track of the user's mobile phone number, IMEI number, user name, email address, defined home address, and other addresses, among other things, as well as the last time a tracker was connected to the Bluetooth application on the mobile device. They could be a valuable source of forensic information. This thesis investigates the necessity for research on what kinds of data with evidential value may be extracted from these applications. We put four different tracking tags to the test. Therefore, these four applications, Chipolo Classic, Galaxy Smart Tag Plus, HTC-Fetch, and Tile Slim, may interact with the tracking tag in various ways and save various types of data to the smartphone. Due to the availability in the local market, we chose these four Bluetooth tracker for our study. This type of Bluetooth tracking gadget is relatively new, and as we just indicated, the market is expanding quickly. Digital forensics investigators might need to be made aware of the kind of data that can be gathered from these applications or the best way to do so. Our study offers instructions on gathering and assessing potential artifacts. Triage analysis might also be used to lessen the pressure that forensic investigators are under in a world where there are a lot of mobile applications and IoT devices since bringing every device to the lab is impractical because it would add to their workload. They might need more than the findings they need. An increasing number of devices need to be studied due to the need for more forensic analysts in law enforcement, as described in [4], which has delayed the retrieval of evidence from digital devices. The gathering and evaluation of evidence must be prioritized to increase the efficiency of this process. The significance of triage becomes apparent in this situation, as noted in [5][6]. We cover the triage steps to complete the thesis work, too. Since we want to recover forensic data, our study adheres to digital forensics' fundamental concepts and procedures during the work period. Identification, preservation, recovery, analysis, and presentation of information concerning evidence discovered on computers or digital storage devices are all included in digital forensics [7]. Identification is a vital step in the digital forensics process that involves identifying artifacts from multiple devices made available to forensic investigators at a crime scene. The next step after that is to preserve the discovered artifacts in order to maintain their integrity. This phase, essential in forensic inquiry, includes data transfer from one device to another. The retrieved artifacts become inadmissible in a case if they have been altered. Following that, a study of the recovered artifacts is done to find information that may be submitted and offered as evidence to a court during the case's adjudication [8].

1.1 Research Objective

The widespread use of wearable technology in the smart device industry has enhanced the standardization of digital forensics. People may keep track of their possessions by connecting the devices accessible to their smartphones, where information is exchanged and shown more clearly. Wearable trackers are incredibly well-liked because they continually gather and store data utilizing 8 various applications [9]. In this regard, Bluetooth trackers have emerged as significant tools that enhance public safety and play a crucial role in digital forensics. However, while independent or dependent tracking devices make it easier to monitor objects closely, such technological advancements could be exploited by malicious actors who may insert Bluetooth trackers into someone's personal effects without consent [10]. The presented information offers a novel source of digital evidence that may be admissible in legal proceedings involving both civil litigation and criminal fraud investigations [11]. In digital forensics cases, Bluetooth connections often play a vital role, ranging from those pertaining to location proximity to other types of inquiries. It is imperative to obtain sufficient proof to establish whether an individual was genuinely connected via Bluetooth. Moreover, investigators need the ability to identify unauthorized devices and determine their duration and proximity. Consequently, extracting relevant artifacts from such gadgets is crucial for effectively leveraging this digital evidence. Achieving these outcomes necessitates adequate comprehension of the data related to these devices.

1.2 Research Questions

The extraction of forensic data from messaging and social media applications on smartphones was the topic of earlier publications [12][13][14] by various academics—other studies [15][16][17] concentrated on getting information from infotainment systems connected to smartphones. The main objective of our research is to perform a forensic analysis on Bluetooth-based tracker applications to see what valuable data could be retrieved from them. These data could then be used as evidence or to help with further investigation. The data of interest in our research can be the currently connected user's details, geo-location, time, system details that tracker applications use, Etc. The following chapters present these data in more detail based on Bluetooth tracker tag applications used on Android devices. Possible Bluetooth trackers are (i) Galaxy Smart Tag Plus, (ii) HTC Fetch, (iii) Chipolo Classic, and (iv) Tile Slim for the Android device Samsung Galaxy A13. We see what valuable data can be retrieved from the mobile device related to the individual Bluetooth tracker user application. These data could then be used as evidence or help with further Bluetooth trackers in a stalking attack investigation.

- How can we extract user data from the Bluetooth tracker application on an Android phone, utilizing root access only, while maximizing the collection of forensic artifacts without the need for Android phone decryption?

1.3 Research Goal

Our study aims to assess the admissibility of data extracted from Android applications, specifically focusing on Bluetooth trackers. We investigate potential inconsistencies related to the brands of these trackers (Chipolo Classic, Samsung Smart Tag Plus, HTC-Fetch, and Tile Slim). Additionally, we offer a comprehensive manual extraction guide for forensic analysts and digital investigators. This guide elucidates valuable insights that can be gleaned from Bluetooth tracker applications, including details such as MAC address, last connected locations (latitude and longitude), device FCC-IDs, user-defined names, country information, user emails, IMEI numbers, custom ringtone seetings, tag color codes, user-defined tag names, icons for tag identification, home address, and other associated addresses. Furthermore, we delve into application-specific details concerning tags paired with users' mobile devices.

In digital forensics, the MAC address can be used to infer information about the device using a specific MAC address and determine the physical location of the device [18]. The first three bytes of a MAC address are assigned to a specific device vendor and can be used to infer information about a device using a specific MAC address. Therefore, this information can play an essential role in the investigation process. Forensic geo-location is the act of determining the geographic source of an item based on its observable attributes. For example, soil on a suspect might be compared with soil at the location of a crime [19]. Geo-location artifacts are the most prevalent 10 form of digital forensic evidence. Effective searching and mapping of latitude and longitude data may often lead investigators to hit the most accurate spot [20]. An unique number called the International Mobile Equipment Identity (IMEI) identifier for every mobile device. In mobile forensics, the IMEI number can be used to identify and track a specific mobile device. For example, if a phone is stolen or used in a crime, the IMEI number can be used to track the device and potentially link it to a suspect. The IMEI number may also be used to confirm a mobile device's legitimacy. An email address is often a high-value asset in digital forensics casework. Finding email addresses on a digital device may allow investigators to establish connections between individuals or detect unknown email accounts to the examiners. Also, many online services, such as social networks, rely on email addresses for authentication [21]. Other artifacts, such as device type, time, user-defined name, user-defined addresses, and more, can help the investigator when submitting the evidence.

1.4 Scope

Our research focus on Mobile Forensics, particularly on the applications of locally available Bluetooth-based tracking devices like the Galaxy Smart Tag Plus, Chipolo Classic, Tile Slim, and HTC-Fetch for Android phones. These specific Bluetooth devices were chosen due to their availability in the local market. We aim to analyze the data that these Bluetooth applications generate and store on the device and explore various methods for extracting this data, including manual, logical, and physical extraction.

The goal is to understand the structure of the data and how it can be interpreted. As these devices work with location data, we need further evidence to identify the person who used the device for criminal activity like stalking so that we may present all the artifacts as evidence in a possible case. To generate user data with the help of Bluetooth trackers, internet, Wi-Fi, Bluetooth connection, and GPS-enabled settings are required.

We plan to investigate how the extracted data can be used in a forensic context, which includes identifying potential evidence of criminal activity, tracking user behavior, and understanding user device interaction. We expect to provide valuable insights into the forensic potential of Bluetooth tracker applications on Android phones, contributing to the body of knowledge in digital forensics and potentially informing future investigative techniques and tools. This study worked only on Android phones, and installed the application data from those devices was extracted using forensic methods.

However, we acknowledge that these safeguards are only available to a small proportion of possible victims and are mostly ineffective. There are also concerns about the potential abuse of these gadgets for sinister reasons, and technology companies are working to address these issues. In our study, we did not conduct a volatile memory forensics analysis because there is nothing to dump passwords or credentials from an Android phone and the Bluetooth tracker. We also do not cover cloud forensics in our paper.

1.5 Contribution

Disclosing our data and discoveries allows our work to advance the discipline of forensics. Our results can give forensic investigators another way to gather evidence concerning Bluetooth tracking devices. One study found that it related to Tile-tracking devices based on iOS. Here, the author concentrates only on geolocation and uses commercial tools to retrieve artifacts from the iOS device [23]. Jimmy Briggs et al., in their research, aimed to create an algorithm for a publicly accessible Android application to detect malicious Bluetooth trackers in real-time [24]. Lauren R. Pace et al. their work focus on geolocation coordinates from Tile applications and Tile eco system based on Android, iOS, and Windows systems [56]. In our research, we concentrate more on forensic artifacts, not only geolocation but also user information, used device data, and user movement activity from the installed Bluetooth-based tracker application, which can help ensure that law enforcement agencies can effectively investigate criminal activities involving cyberstalking, theft and loss recovery, accident investigation, missing person cases, corporate espionage, etc.

1.6 Ethical Issue

Every effort has been made to ensure that each participant's privacy is respected in accordance with local and national laws, including, but not limited to. User data sources for research are collected in a controlled operating environment to avoid violating the privacy of individuals in a public land environment. In addition, privacy-related user information is censored by blurring or masking sensitive information with the corresponding blur text box in this document.

1.7 Thesis Structure

The following illustrates the format in which our study will be presented. The topics of each chapter are summarized as follows:

- Chapter 1: The introduction of the topic, related questions and aims to answer them, the reasons for choosing this topic, and the scope and limitations of the thesis.

- Chapter 2: Explanation, theoretical background, different forensic terminology, and literature review.

- Chapter 3: The methodology, experimental setup, generating data, and process flow for collecting data.

- Chapter 4: Data extraction method and presentation of the results.

- Chapter 5: Discussion and Analysis of the Extracted Data, The thesis's conclusion and its recommendation for further research.

2 Background

This chapter includes all the details necessary to comprehend our article and an explanation of the procedures used to retrieve forensic artifacts from smartphones connected to Bluetooth-based trackers for Android applications. In order to offer a reader a better understanding of the work that has previously been done in the domain of mobile forensics, we have also included a presentation of those works.

2.1 Stalking vs Cyberstalking

Staking is a form of harassment in which an individual or group of people follow, monitor, or contact another person repeatedly, often without their knowledge or consent. A stacking attack aims to gather information about the target's whereabouts, routines, and habits to exploit vulnerabilities or commit crimes. According to Mullen et al. [67], stalking is defined as the repeated imposition of unwanted communication on another individual, characterized by systematic activity directed at a specific person and undesirable to the target person. The victim may find it threatening, fear-inducing, and disturbing.

Cyberstalking is the new way to stalk in modern times. Cyberstalking can be carried out in various ways, including physical surveillance, online tracking, and the use of social engineering techniques to obtain personal information. In some cases, stalkers may also use technology such as GPS trackers, hidden cameras, and spyware to monitor their targets [68]. Overall, cyberstalking attacks represent a significant security threat, Individuals as well as entities must take precautions to defend themselves against them.

2.2 Bluetooth Tracking Application

Bluetooth-enabled tracking tags designed for Android and iOS devices utilize Bluetooth Low Energy (BLE) technology, which operates on the radio frequency (RF) band for wireless communication. BLE technology enables the detection and tracking of people, devices, and assets, facilitating indoor positioning use cases such as asset tracking, indoor navigation, and proximity services. In some cases, these tags also employ Ultra Wide Band (UWB) technology for smartphone communication. The Ultra Wide Band (UWB) technology employed by some smart trackers, such as the Apple AirTag and Samsung Smart Tag Plus, facilitates precise positional guidance, with location tracking accuracy reaching down to a few inches. These smart trackers use BLE to communicate with any smartphones that can pick them up in the vicinity. These Bluetooth applications can save user-related data inside the phones and can routinely upload tracker locations to the cloud. This application allows us to see their last known location on a map. It is worth noting that some of these devices may have additional features or variations depending on the specific model or version [27]. The technical aspects of the Bluetooth-based tracker application for Android and iOS devices involve several vital components. Firstly, the application starts communicating with the tracker using the BLE protocol. Here, authentication involves the exchange of passkeys to generate a shared secret key that is used to encrypt the data exchanged between the devices. This process ensures that the devices communicate with the correct device and that the data is secure. This process involves the necessary APIs and libraries to establish a connection and send and receive data. Secondly, the application displays the location of the tracker on a map. This process requires integration with the location service on the mobile device and using an algorithm to estimate the tracker's location based on the strength and direction of the Bluetooth signal. Finally, the application continuously monitors the Bluetooth signal strength and provides notifications and alerts when the tracker is out of range or lost.

Table 1. Comparison between Chipolo Classic [79] [80], Samsung Smart Tag Plus [75] [76], HTC-Fetch [81] [82], and Tile Slim [77] [78].

Feature	Chipolo classic	Samsung Smart Tag Plus	HTC fetch	Tile Slim
Capability	iOS, Android	iOS, Android	Android	iOS, Android
OS version	6.0 or later	8.0 or later	4.3 or later	8.0 or later
Range	200 ft.	120 m.	15 m.	250 ft.
Connectivity	BT 4.0	BT 5.0	BT 4.0	BT 4.0
Technology	BLE	BLE/UWB	BLE	BLE

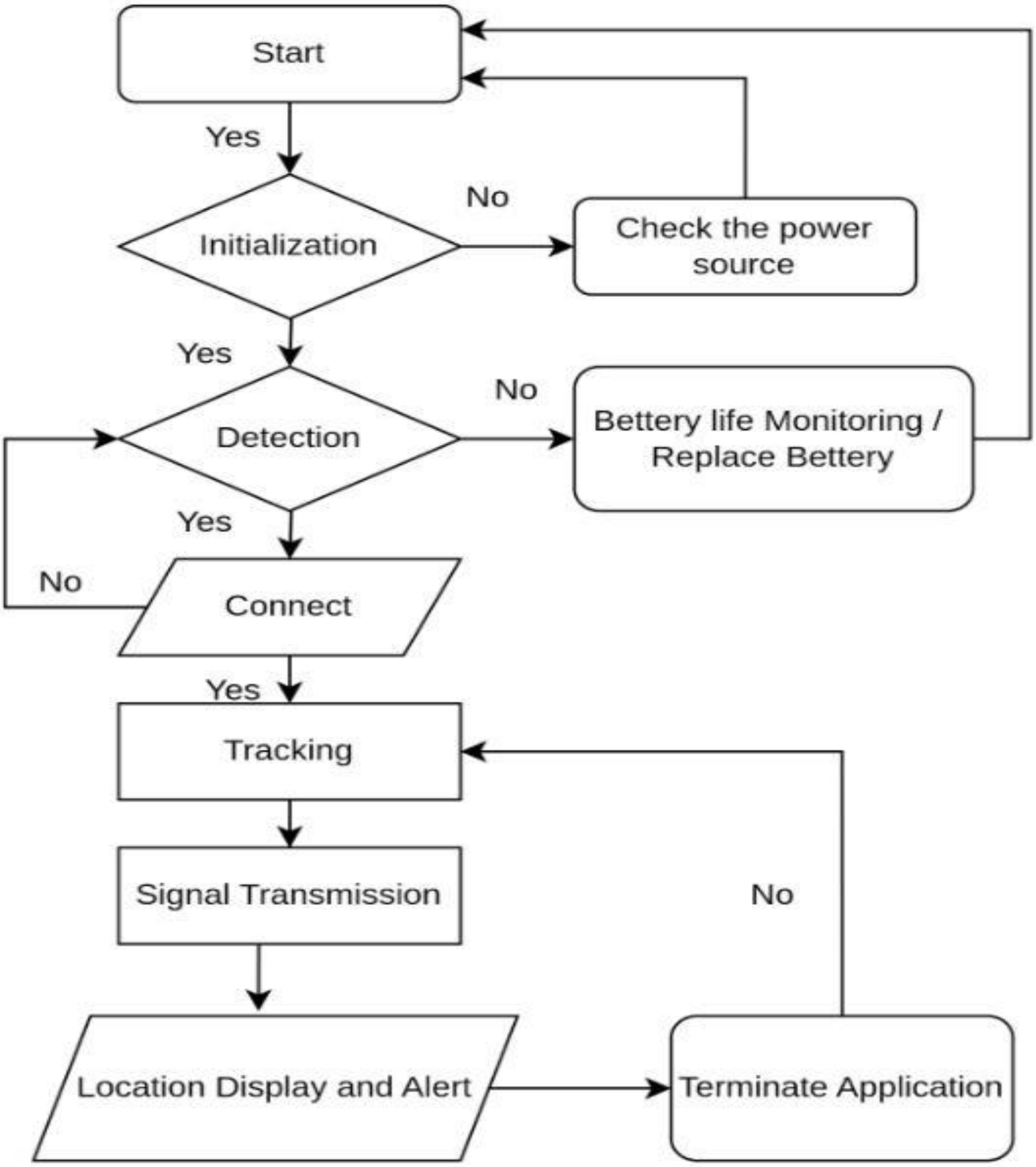


Figure 1. Sample Bluetooth tracking device working process

2.3 Digital Forensic Background

In this study section, we define key terms related to digital forensics. We aim to help readers without a technical or digital forensics background better understand our paper. These forensics terms appear frequently in the following chapters, so it is essential to clarify their meaning to ensure everything is clear. According to Locard's Exchange principal, derived from criminology, asserts that a perpetrator cannot commit a crime or leave a crime scene without leaving behind traces [74]. This principle applies even in the digital world, where interactions between two objects leave mutual traces. Digital forensics is the branch of computer science that deals with acquiring, restoring, and analyzing electronic data from devices with persistent and volatile data storage. A digital forensics investigation aims to answer questions about what happened, where, when, and how. In law enforcement and security assessments, it is also necessary to determine who acted and whether the incident can be repeated in the future. Digital forensics objectively answers these questions by identifying incriminating and exculpatory digital traces. In order to comply with legal standards and be admissible in court as evidence, an investigation must comply with government guidelines published by organizations such as the German Federal Office for Information Security (BSI), the National Institute of Standards and Technology (NIST), the European Council [73], and INTERPOL. The integrity and documentation of the investigation are crucial for forensic analysis to meet court standards. The BSI requires that the investigation process meet standards for acceptance, credibility, reproducibility, integrity, cause and effect, and documentation. Lawful documentation must also include proof of digital traces, whereabouts, and the chain of custody to guarantee the authenticity of the collected data and support the identification of responsible parties in cases of evidence trail integrity violations. Cryptographic hash functions are often used to ensure integrity in digital forensics.

2.3.1 Volatile Memory

Volatile memory is a type of storage that allows the device to read and write information quickly. However, it requires a continuous electric current flow to hold the data. If the power is lost, the data stored in volatile memory is erased [28]. Digital forensics is focusing more and more on the volatile memory of Android phones. Memory analysis is more critical than device and application storage encryption. Memory typically holds the encryption keys of persistent data that is now used in the digital forensic sector and is instantly available data without decryption.

2.3.2 Non-Volatile Memory

Non-volatile memory is a permanent memory area written during the manufacturing of a device and retains its data even after a reboot or power off the device. Modern consumer devices use Electrically Erasable Programmable ROMs (EEPROM), allowing byte-by-byte modification of the memory array. This feature is helpful for firmware updates, among other purposes [29]. The non-volatile memory can store data related to the operating system and application through files, folders, and registry entries. It stores user data such as documents, email, user logs, etc.

2.3.3 Forensics Artifact

Since the term “artifact” lacks a formal definition within the digital forensics domain, V. S. Harichandran et al. proposed a new term: “Curated (digital) Forensics Artifact (CuFA)” [30]. However, we have chosen not to use the term CuFA in our work. For our study, we have adopted one of the proposed stipulations from their list. In our case, an artifact is defined as data that has evidentiary value in a legal proceeding, and making it evidence.

2.3.4 Triage

In digital forensics, the triage procedure is used to find immediately objects that are most likely to contain evidence [31]. The Cyber Forensics Field Triage Process Model (CFFTPM) suggests an on-site or field method for quickly identifying, analyzing and interpreting digital evidence without the requirement of having to take the system(s) or media to the lab for a more thorough inspection or to acquire a complete set of forensic image(s). [32][33].

2.4 Related Work and Literature

In this section of our work, we present some past study works in the domain of mobile and Android forensics. Furthermore, our focus on these works was on their acquisition methods and how we can utilize them in our study. We reviewed related literature that presents various mobile apps, like Android and iOS applications that have been forensically examined. We also review some thesis papers about tracking location or partial research on a few geo-location artifacts. Our study focuses on forensic artifacts that include geo-location and potential data from the installed Bluetooth-based tracker application, which can assist law enforcement agencies in effectively investigating criminal activities involving cyberstalking.

2.4.1 Bluetooth Related Works

MacDermott et al. [52] performed a forensic examination of three fitness wearables: the Garmin Forerunner 110, Fitbit Charge HR, and a low-cost generic HETP fitness tracker. By manually inspecting their associated Windows 10 app (Fitbit) or directly connecting the wearable to a 18 Windows 10 computer, the authors discovered forensically relevant artifacts stored by these devices. The HETP band only interacts with a mobile app, but the author could not access any data that may have been stored within the app. The authors discovered that the Garmin device and the Fitbit Windows 10 app record significant amounts of forensically relevant data. They were able to recover information about the test runs they conducted to populate the devices with data, including GPS location (Fitbit), deleted records (Garmin), group chat or post interactions (Fitbit), and other user and device information (both).

Jimmy Briggs et al. [24] their research paper introduce a general method for detecting maliciously deployed Bluetooth Low-Energy (BLE) trackers, such as Apple AirTag and Tile Finder. These devices can be concealed by stalkers in their targets' clothing or vehicles to track their movements. The authors' algorithm can detect malicious devices in a minute, compared to hours or days for previous algorithms. In a small but innovative validation study, the authors demonstrate that their algorithm has high precision and recall for most existing trackers, although AirTags present additional challenges. The authors also provide an open-source Android application for real-time detection of these devices and discuss the behavior of the AirTag and the risk factors that make it difficult to detect.

2.4.2 Mobile Forensic Related Works

In order to identify some areas for development in the mobile forensic arena during the last seven years as of the time the study was written in 2018, Barmpatsalou et al. [34] looked into the trends of mobile forensics, both existent and imminent. Their report also mentions the mobile forensics investigation procedure that resulted from this effort [35]. In addition, they noted that while each stage of a forensic investigation for mobile devices is crucial, there is a disparity in the amount of time and attention given to each in terms of research, which they attributed to the fact that not all stages are equally crucial for all fields [34]. The collection techniques described in their work are crucial to our research since physical and logical data-collecting methods can be used to extract the artifacts from the smartphone. Gokila Dorai et al. [35] showed, how to locate the application that hides user related information or vault from the iOS devices. A logical acquisition strategy based on system-level data transfer services offered by the makers of Android mobile devices has been studied by P. Feng et al. [36]. They suggests performing a system-level data transfer from the target device, a non-rooted phone, to the intermediary device. This rooted Android smartphone was subsequently utilized for logical acquisition.

Depending on the situation and the devices involved, a strategy including an intermediate device may be helpful. A. Levinson et al. suggested [37] another intriguing technique for acquiring forensic evidence in which they look at third-party installed applications on an Apple mobile device. Depending on the authorization they received, third-party programs can store much information. For instance, forensic analysis of the app can be applied to acquire GPS coordinates if the application has access to the mobile device's Global Positioning System (GPS) feature. If we come into a similar situation, we can utilize a way to get around some access restrictions imposed by the mobile device's proprietary software by looking into relevant third-party applications.

In the article [38], Zhang et al. presented investigations and conclusions on 18 Android vault applications that they had reverse-engineered and examined the artifacts created. Their research revealed that more than 60% of the software had illegible code; 27% of the software prevented the process of reversing because of their built-in libraries on the hidden data; about 55% of the software was obtained without the user having privileged root access or device control; just over 30% of the software stored images without encryption; 44% did not also encrypt videos; nearly 39% of the software saved passwords in plaintext and without encryption; and 27% had password files exchanged with other users. Their research, which was centered on the security of Android applications, showed that compromises might happen within the application, offering forensic professionals a chance to gather crucial evidence throughout an investigation. Our research aims to extract data from smartphone Bluetooth tracker apps in both rooted and non-rooted modes. Unlike their study, our analysis was not be performed from an application's security standpoint.

Aya Fukami et al. [45] discuss the challenges posed by encryption and other security features in modern mobile devices for forensic investigators seeking to extract data. The author explains that traditional forensic data extraction methods have been rendered ineffective due to the robust security measures implemented in modern mobile devices. As a result, the authors propose a novel approach for extracting forensic data from encrypted mobile devices that involves vulnerability exploitation and underdeveloped regulation of encryption and government access to encrypted devices.

In their work [39], Domingues et al. evaluated the evidential data generated in Windows 10 individual PCs when the "Your Phone" system program is installed on the computer. They discovered valuable data, including messages that are 30 days old and the database that houses the phone contacts of connected smartphones, both of which are accessible from the SQLite 3 database in the Windows 10 computer. They also demonstrated that data obtained from their Phone system can become important forensic artifacts if the associated smartphones are not readily available to be examined for evidence. Although they altered it with some Python scripts, we utilize the Sleuth Kit Autopsy tool to evaluate the retrieved data. Similar to the automobile application, we studied for our work, their research also examined an Android version of the program.

2.4.3 Application Forensic Related Works

Nhien-An Le-Khac et al. discussed the forensics investigation of TomTom navigation systems [40]. They present a forensic acquisition and analysis procedure for TomTom AA. Their approach involves a detailed, step-by-step process for retrieving forensic data from different mobile devices. Through a series of driving tests conducted using various mobile devices, they uncovered the significance of the data contained within TomTom AA. Their analysis revealed that while most files were stored in humanreadable format, critical data such as GPS coordinates were stored in a proprietary format that required conversion for analysis. They describe their result and provide an in-depth analysis of data obtained from Samsung mobile phones. They also compare and contrast differences between TomTom PND and TomTom AA devices.

Jason Bays et al. [53] discovered forensic artifacts from third-party locations sharing Android applications from Android and iOS devices. They use commercial forensics tools to analyze forensic evidence.Posie Aagaard et al. [41] illustrate how forensic investigators can readily acquire user personal data generated by the Life360 Android application through device logical files and network traffic forensics. They also demonstrated that only one device would need to be compromised for all Circle users' personal data to be compromised.

Mahajan et al. [42] examined the internal memory of five mobile devices running three different Android operating systems to determine the data types that could be extracted. The researchers focused on information related to "WhatsApp" and "Viber" messages, as well as media files. They found that Android devices store many evidentiary data that forensics experts can obtain. The researchers used the UFED (Cellebrite Universal Forensic Extraction Device) to conduct a File System Acquisition, in which folders and files were extracted. This tool allows them to fully understand the directory and file structure of the Android device [42]. The File System Acquisition method can retrieve information such as media and system files, databases, logs, passwords, contacts, calls, and web history from a device's file system [43]. The methodology used in this study [42] could be helpful for our research, as our goal is to extract data from phones regarding installed Bluetooth tracking applications in a similar manner to how "WhatsApp" and "Viber" were installed.

2.4.4 Cloud Forensic Related Works

Cloud forensics is a relatively new field compared to mobile and Android forensics. With the widespread use of cloud storage to expand the capabilities of mobile devices and applications, finding valuable data in the cloud has become increasingly important. In their work, Choo et al. [44] identified three key areas to focus on when conducting cloud forensics: data on the device, data in transit, and data stored on the servers. This means all three areas must be examined for artifacts during a cloud forensic investigation [44]. Our study focuses on the first area, analyzing Bluetooth tracker applications installed on the mother device to extract information about the Bluetooth tracker owner-related information. As mentioned, the tracker application could store information on the mobile device. Puneet Sharma et al. [46] presented a mobile cloud forensic process incorporating time synchronization, inter and intra-application analysis, and the traditional forensics procedure. The proposed process aims to improve cloud event traceability in a massive cloud environment by utilizing the metadata of potential mobile evidence. The study uses the Android WeChat social network application to show the efficacy of the suggested forensic methodology.

2.4.5 Infotainment Forensic Related Works

Faisal Sumaila et al. [47] discussed the forensic analysis of mobile automotive maintenance applications, which provide drivers with notifications or information about their vehicles' integral parts and its health conditions. These applications collect data from the Engine Control Units (ECU) located in the Controller Area Network (CAN) bus via a dongle connected to the onboard diagnostic-II port of the vehicle. This data is vital for traffic, vehicle insurance, or criminal investigations. The authors highlight the importance of conducting forensic analysis on automotive maintenance applications due to their potential evidentiary value. The paper presents a methodology for conducting forensics analysis on these applications and demonstrates its effectiveness through a case study. In a separate study [48], Whelan et al. examined two different types of infotainment systems to identify valuable forensic artifacts. The research aimed to determine the types of forensic artifacts that could be obtained from in-vehicle infotainment systems. The findings revealed that one infotainment system stored a more significant amount of data compared to the other. To conduct their analysis, the researcher utilizes Berla's iVe tool, a commercially available software commonly used by the military, law enforcement, and selected governmental institutes. While their focus was on infotainment systems within vehicles, their work provided insight into acquisition methods, including logical and physical approaches for data extraction. In our study, we go the same way to collect data from installed applications of Bluetooth trackers on a mobile device.

2.4.6 Triage Related Works

Eric Gentry et al., in their [49] research "SEAKER: A Tool for Fast Digital Forensic Triage," present a digital forensic device named SEAKER (Storage Evaluator and Knowledge Extractor Reader), It enables forensic investigators to triage a large number of digital devices. According to this study, forensic investigators must be able to swiftly analyze many high-capacity digital media devices to determine which ones require more investigation. Triage is a crucial step in the early stages of an investigation and is essential to evaluating the available evidence. Instead of imaging the devices, which takes hours, SEAKER searches for files with names that conform to preestablished patterns. The search is done by monitoring the devices in read-only mode (to preserve evidence) and listing the device's contents. Unlike imaging, this approach takes minutes rather than hours. In their research, Darren Quick et al. [50] presented a framework for performing triage that relies on machine learning rather than human interaction. This necessitates an inference engine adequately feeding, selecting, and categorizing information from multiple sources. Their primary goal was to restrict human input, preventing human mistakes from occurring during triage. It concentrated on developing a framework for artificial intelligence that may be useful for dividing up devices at a live scene. However, their work guides us when we collect artifacts from our work by providing a case study of categorizing child pornography messages on a mobile phone. According to Hosrman [51], the growing amount of digital evidence in criminal cases has resulted in a backlog of unfinished forensic investigations. To address this issue, first responders at crime scenes often use triage to identify and collect only relevant devices, reducing the number of items that need to be examined. The effectiveness of this on-site triage depends on the investigator's understanding and knowledge of the device in question. However, due to a shortage of trained forensics professionals, many personnel need more experience, which can negatively affect the triage process. The authors propose a system for ranking and scoring devices at the scene of an investigation. Their guidelines can be helpful for our thesis during the data acquisition phase. We document the steps we take to collect and analyze data in a guideline that can also serve as a triage process for investigators to follow when they arrive at a crime or incident scene.

2.5 Summary

From the related literature, we found that the researchers showed techniques and frameworks to collect, analyze, and preserve mobile forensic data based on Android and iOS devices. We found the location-based data retrieval study concentrated on GPS coordinates regarding locating users on Android and iOS applications, and they used commercial tools to retrieve digital artifacts. We can retrieve more information and not only location addresses but also user's related data, due to the previous work that has been done in the digital forensic area in Android user applications, triage, infotainment systems, and the cloud.

3 Methods, Experimental Environment, and Data Process

This chapter explains the technique and reasons behind our investigation. Furthermore, it provides a thorough review of the materials and equipment used in our experimental arrangement, as well as for data extraction and analysis. An accompanying process flow diagram, dealing with the sequential processes involved in the data-collecting process relevant to our thesis, is also supplied.

3.1 Methods

Our research utilizes the extraction methods recommended in the work [54] by Ayers et al., which propose a layered approach to extraction or acquisition methods for mobile forensics. We have chosen their methodology because our study focuses on extracting information from a mobile device, and their approach provides guidance for data acquisition. As outlined in [54], it is advisable to conduct data extraction starting from the lower levels and moving upward. According to the researchers' explanation, extracting data using higher-level methods such as the chip-off method (level 4 in the extraction hierarchy) and then reverting to a Hex Dump or level 3 acquisition method may not be feasible, as lower-level tools may not function as intended. Data loss or damage to the phone is risky if the investigator lacks the necessary expertise or does not follow the correct procedures when performing extraction at this level. Consequently, their work emphasizes the importance of employing proper procedures and tools to avoid data loss or alteration. In our study, we initiate data acquisition with the manual extraction method, as it is the first level that requires fewer tools to acquire data from Bluetooth tracker applications. We then progress to the logical method, with the physical method serving as the final level in our extraction methodology. The chip-off method was not utilized in our study due to the potential for data loss or damage to our smartphone. Additionally, the smartphone we used in this study was fully functional, and the chip-off method is typically employed as a final stage when the mobile device is non-operational during the acquisition process.

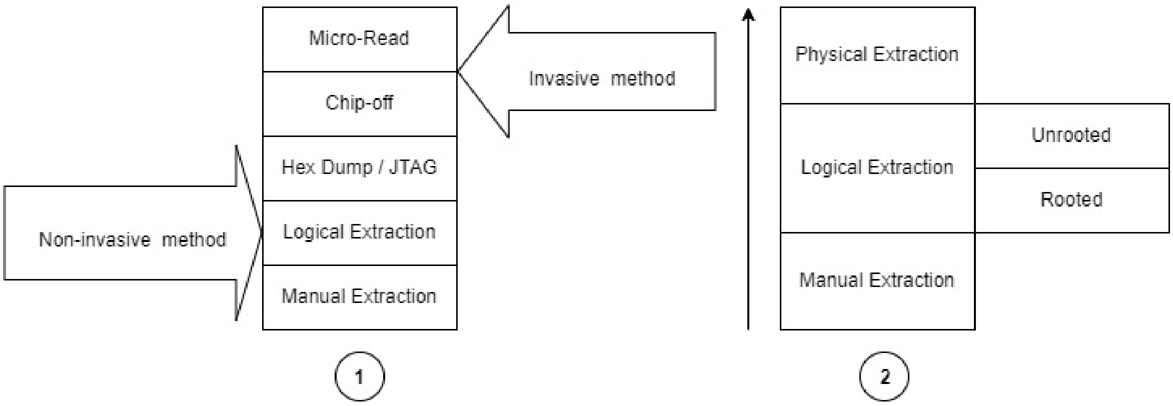


Figure 2. Mobile Device Tool Classification System (1) & Extraction method we used in our data acquisition (2).

3.1.1 Manual Extraction

In digital forensics [54], manual extraction often refers to the process of extracting data from digital devices without the assistance of automated methods. Some instruments, however, may still be utilized to aid in the manual extraction procedure. For example, during the extraction process, an examiner may employ a camera or other recording device to capture the material presented on the device screen. Furthermore, specific manual extraction procedures may entail using emulators or other software tools to generate virtual representations of the target device for examination. As a first-level method, we gather as much information regarding the Bluetooth tracker and any other user-related data with the paired Bluetooth tracker, such as the GPS location of the Bluetooth tracker, the name of the Bluetooth tracker name, the possible MAC address of the tracker, the userspecified tracker icon, the timestamp, and more. During this process, we use the Samsung Galaxy A13 Android phone as a mother device for four Bluetooth trackers: Chipolo Classic, HTC-Fetch, Tile Slim, and Galaxy Smart Tag Plus. All the Bluetooth tracker tags interact with the mother device using its application and Bluetooth Low Energy (BLE). Ayers et al. mentioned some limitations to manual extraction in digital forensics. One limitation is that extracting the necessary information when dealing with large amounts of data can take time and effort [54].

Additionally, there is a risk of accidentally manipulating the data during extraction [54]. Despite these limitations, manual extraction can provide a broad overview of the available information before using specialized tools or software. This method can be helpful for investigators to gather crucial information for optimized forensic triage without the need for special tools. However, caution must be exercised to avoid accidental modifications while examining the interfaces of applications for artifacts.

3.1.2 Logical Extraction

At the logical extraction level, it is possible to execute the extraction process through either a wireless or wired connection between the device and the forensic workstation, as mentioned by Ayers et al. [54]. According to the NIST, Active storage items located on a logical store, such as a file system partition, such as an address book, personal information management data, call logs, text messages, and standalone data files, are copied bit by bit [55]. For our research, firstly, we conduct logical acquisition from non-rooted Android devices. After that, we root the Android device for more privileges at the file system level to get more data.

3.1.3 Physical Extraction

In the last phase of our work, we applied physical extraction techniques to obtain artifacts from Bluetooth tracking applications installed on the mobile device. According to Fukami et al. [45], physical extraction can be challenging due to modern mobile devices' increased encryption and security measures. These security features can turn off many data acquisition methods used historically, and new approaches to acquiring data from modern mobile devices must be explored. Physical extraction entails making a bit-by-bit copy or image of a device's physical storage, according to NIST Special Publication 800-101 from 2014 [54]. Similar to logical extraction but with a physical storage copy instead of a logical one. Hex Dumping and JTAG are two techniques for doing physical extraction that are described by Ayers et al. [54]. Hex Dumping is a noninvasive technique that copies the device's storage to a portion of its memory without requiring the device to be disassembled [54]. Given that the smartphone from which we recovered data was powered on during the procedure, we used this non-invasive technique for our goals. We use a wired (USB-C) connection with the help of ADB tools between the Android device and Windows 10 Pro N system, and we have Kali Linux as a backup host system. Then, use dd to acquire an image of those phones for analysis in Android Studio and Sleuth Kit Autopsy.

3.1.4 Test and Analysis

We run the test procedure for generating data individually for each application related to its Bluetooth tracker tag. We have a screen lock and other credentials when we use our mobile device. For Chipolo Classic and Tile Slim, we have to create a user profile to have user credentials. To compare generated data with the extracted data from installed user applications, we follow the same starting point and same destination on each application. We use Chipolo Classic, HTC Fetch, Tile Slim, and Galaxy Smart Tag Plus on a Samsung device running Android 13. Those tags have their Android-based user applications. Chipolo Classic and Tile Slim are independent trackers and can run on any Android device. HTC Fetch is made for HTC brand Android phones, but this tracker can also run other Android devices. Our Galaxy Smart Tag Plus is made for Android-based Samsung mobile phones, and our mother device for Bluetooth application is the Samsung Galaxy A13, which is perfectly capable and fully functional and available for the Galaxy Smart Tag Plus tracker. This Galaxy Smart Tag Plus has a Smart Things application that tracks Bluetooth beacon devices. We chose our starting point from Majaka 23 Tallinn to our destination point, Peterburi Tee 2 (T1 Mall), Tallinn. We used the different trackers individually in different time slots. We captured screenshots and recorded the dashboard of both mobile devices to capture the starting and destination GPS location, the name of the tags and their defined icon, the timestamp, the name of the user profile info, etc. Of course, all this data depends on the Bluetooth application user interface. So, data may vary depending on the application.

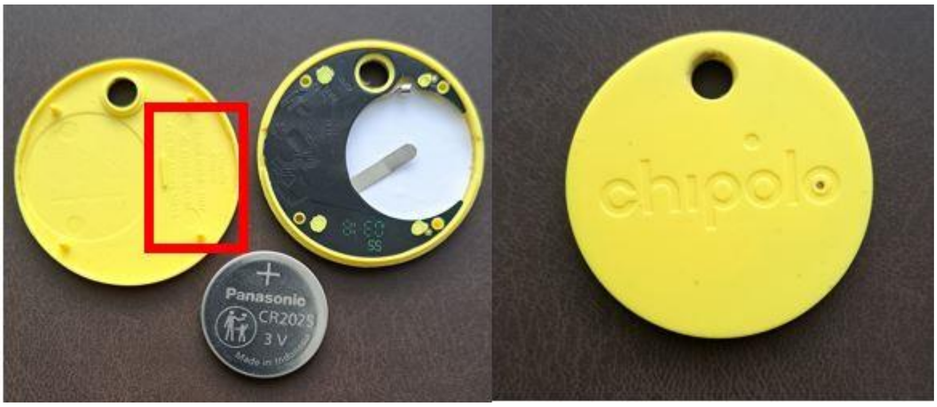
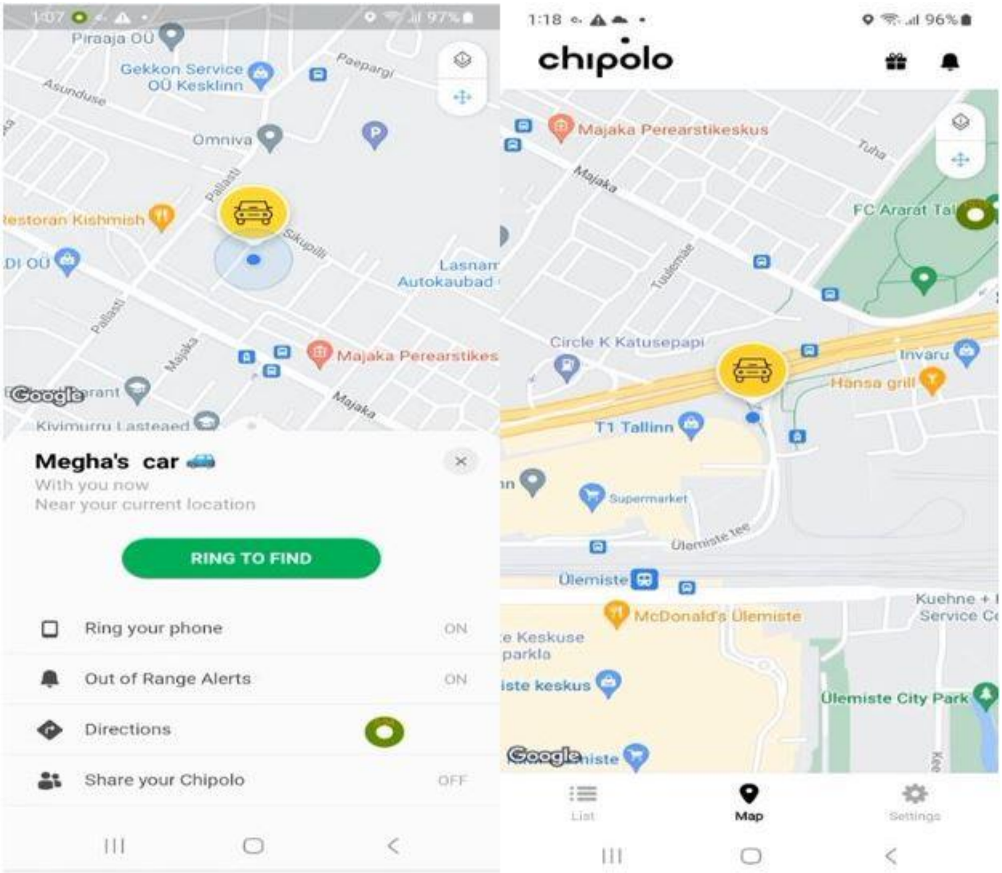


Figure 3. Chipolo Classic (Yellow)

The color of the Bluetooth tag was yellow. With the Chipolo classic Bluetooth tracker, we began our route on August 27, 2023, from Majaka 23 at 01:07 PM (13:07) and arrived at Peterburi Tee 2 (T1 Mall) at 01:18 PM (13:18), as can be seen in the recording[1] . The Chipolo Classic Bluetooth device has a 200-foot range, connects through Bluetooth, and sends an updated position every 15 minutes (most BLE devices)[58]. Opening this gadget and looking at the hardware for additional analysis is straightforward. Although this connection was ongoing, we noticed some latency when updating the live location point. We can observe from the dashboard that the user uses his username and a particular icon for his tag. In the following chapter, we go into further depth and emphasize its key points.

[1] https://youtu.be/rCV03uf-pj8



Megha's car
With you now
Near your current location

Ring your phone ON
Out of Range Alerts ON
Directions
Share your Chipolo OFF

Figure 4. Starting and destination point on the Chipolo application

With the Galaxy Smart Tag Plus, we set out on our next journey, which began at Majaka 23 on August 27th, 2023, at 01:38 PM (13:38) and ended at Peterburi Tee 2 (T1 Mall) at 01:49 PM (13:49)[2] .

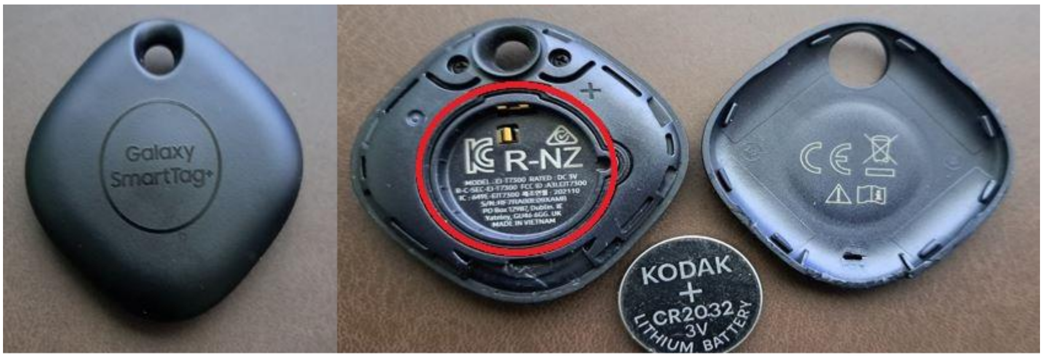
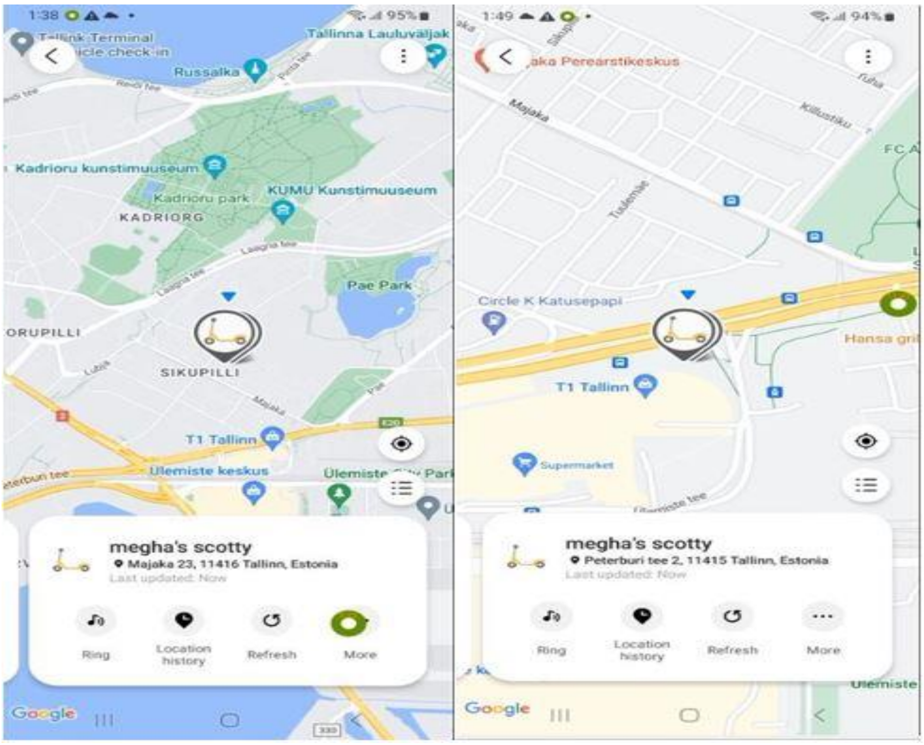


Figure 5. Galaxy SmartTag+ (Black) front and inside

The starting and ending points are the same, but the path is different for each Bluetooth tag. The connected device may interact with this Bluetooth beacon every 15 minutes; its range

[2] https://youtu.be/Bukh-pquhP4

is 120 meters. We applied the refresh button within three to four minutes to check the tag location on the map. The location is being updated on the map. Although a slight delay was detected throughout the experiment, the tag was visible close to the mother device on the map as we kept a set distance to prevent losing contact with the Bluetooth tag.



megha's scotty
Majaka 23, 11416 Tallinn, Estonia
Last updated: Now

megha's scotty
Peterburi tee 2, 11415 Tallinn, Estonia
Last updated: Now

Figure 6. Starting and destination point on the Galaxy Smart tag+ application

In order to obtain additional information from the Bluetooth tracker, which allows us to identify the tracker precisely, we opened the beacon to find the FCC-ID and other hardware related information. We examined further information from the installed program in the next chapter.

Following the route, we begin using the HTC-Fetch tracker at Majaka 23 at 02:06 PM (14:06), and we continue to follow our subject until we get to Peterburi Tee 2 (T1 Mall) at 02:18 PM (14:18)[3] . The range of the HTC-Fetch tracker is up to 15 meters. The user-defined tag name "megha's key" and the designated tag icon "key" are displayed on the HTC-Fetch dashboard. The target point's latitude and longitude are also shown. We can observe the

[3] https://youtu.be/ggDrZds6E0s

street view option on this dashboard, which provides additional vital data that is essential for the triage step.

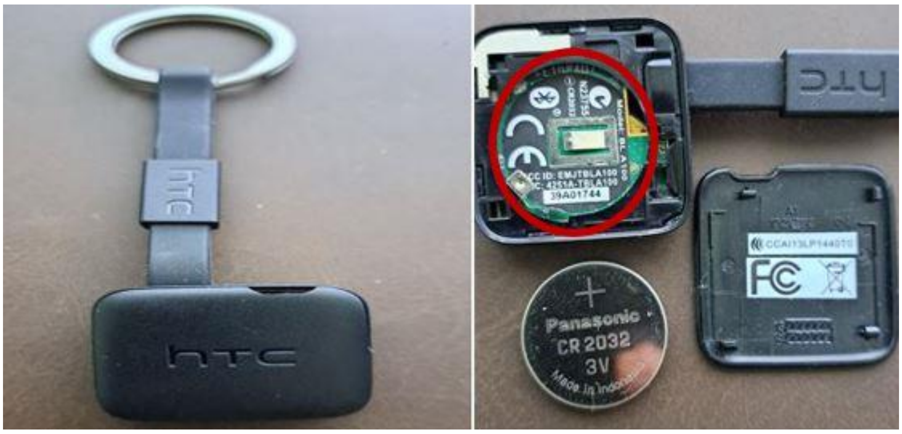
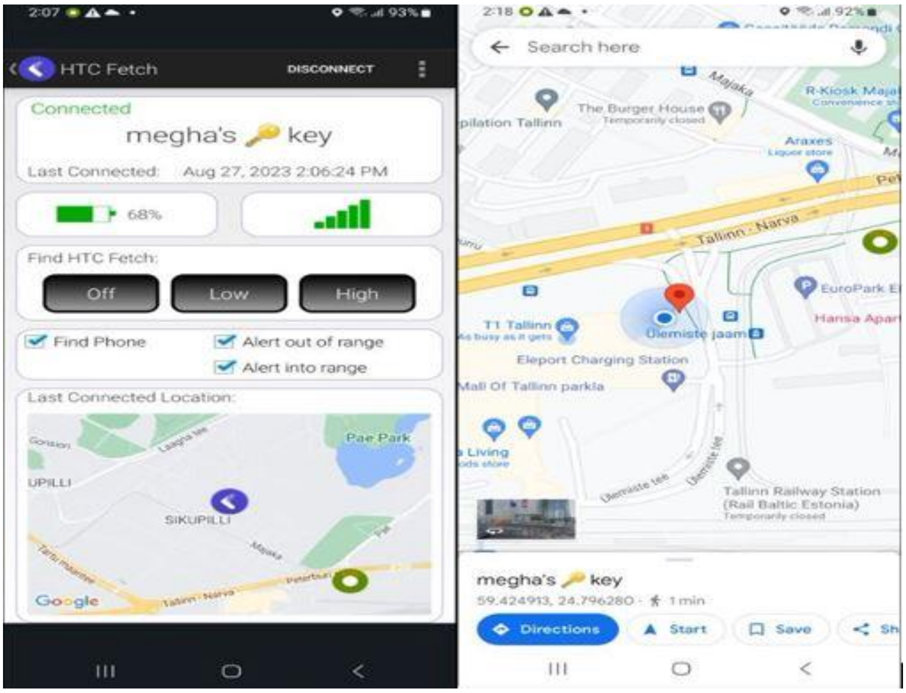


Figure 9. Front and inside of HTC-Fetch


2:07 93%

HTC Fetch DISCONNECT

Connected

megha's 🔑 key

Last Connected: Aug 27, 2023 2:06:24 PM

68%

Find HTC Fetch: Off Low High

☑ Find Phone ☑ Alert out of range ☑ Alert into range

Last Connected Location:

2:18 92%

Search here

megha's 🔑 key

59.424913, 24.796280 · 1 min

Directions Start Save Sh

Figure 10. Starting and destination point on the HTC-Fetch application

During the HTC-fetch tracker observation, we faced the latency of the beacon update time. We found the last connected time from the dashboard with the tag. We tried to update the time on the dashboard, but there was no option for the user to refresh or manually update the time with the new location. We found that only disconnect and reconnect beacon from the application can update the connection time with a new GPS location. For triage analysis, we did not disconnect the device, and we expect the user application to give us the last update time according to the updated GPS location after a 15-minute update time period.

We started our last experiment with the following route with the Tile Slim Bluetooth tracker from 02:34 PM (14:34) to 02:44 PM (14:44) on 27th August 2023, and the starting and destination locations are the same as previously used[4] . This device can communicate within a 250 ft. range.

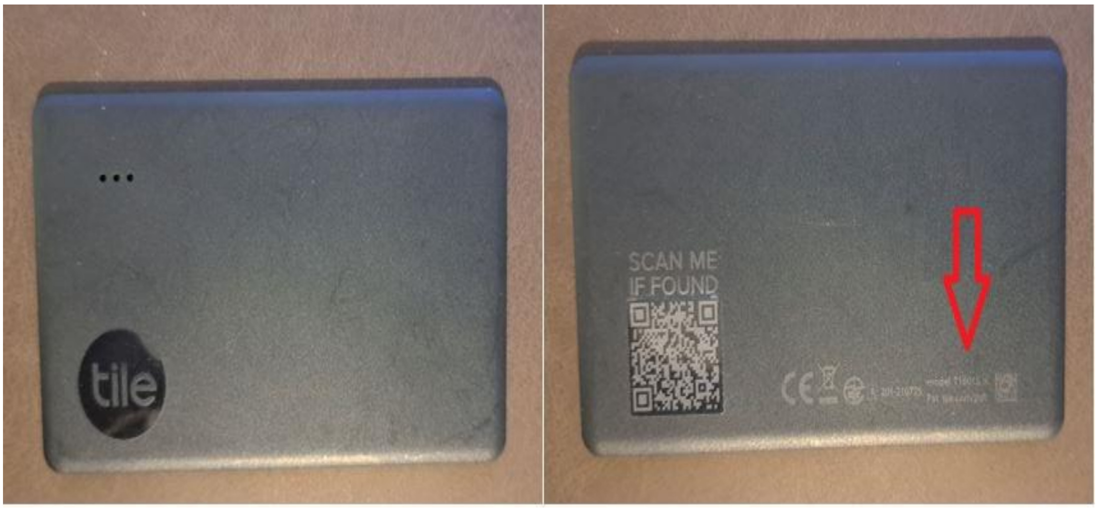


Figure 11. Front sides of Tile Slim tracker

In addition to the model number we found on the device's back, it also has a QR code that the user may use to identify the device as a lost item in the application. The application database file contains fascinating artifacts that we can find. During the test session, we did notice any latency in the map's location updates. The tag symbol and the user's name were shown on the application dashboard. During our data extraction procedure, we anticipate extracting additional valuable data.

[4] https://youtu.be/MVIPe3PdeMs

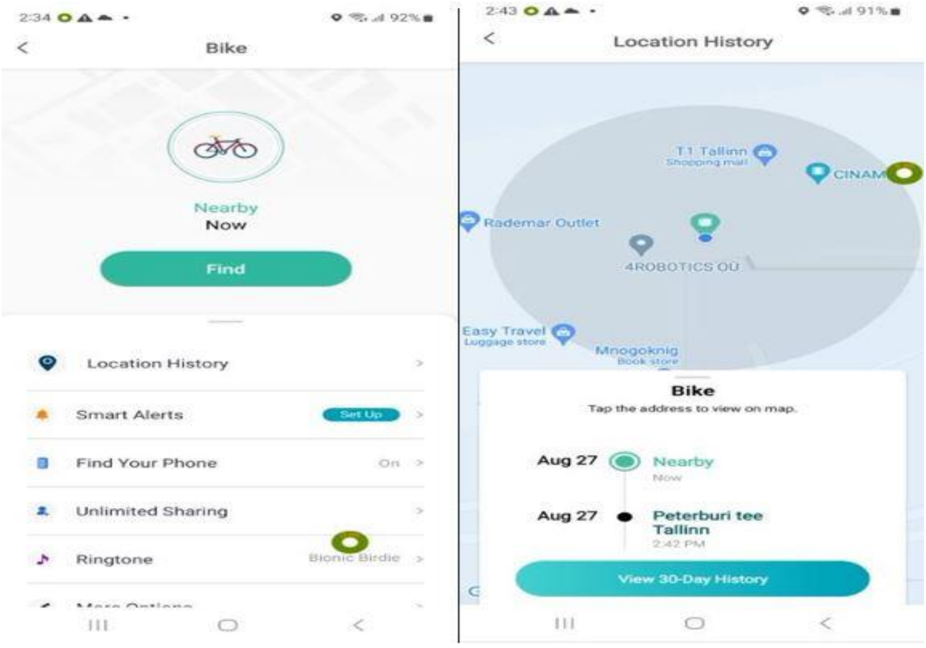


Figure 12. Tile Slim application dashboard and destination point

Table 2. Logs for Bluetooth tracker dataset

Tracker	**Duration (HH:MM:SS)**	**Start point**	**Time**	**Destination**	**Time**	**Movement**	**Tracker-Tag**
Chipolo	00:11:00	Majaka 23	01:07	Peterburi Tee 2	01:18	Walking	Car
Galaxy Smart tag plus	00:11:00	Majaka 23	01:38	Peterburi Tee 2	01:49	Waking	Scotty
HTC-Fetch	00:12:00	Majaka 23	02:06	Peterburi Tee 2	02:18	Walking+Bus	Key
Tile Slim	00:10:00	Majaka 23	02:34	Peterburi Tee 2	02:44	Walking+Bus	Bicycle

3.2 Experimental Plan

We employ Bluetooth tracking devices with Android operating systems to test our Bluetooth tracker. Finding a tracking device might be difficult if it has been intentionally silenced or altered so that it cannot play sound or vibrate. Throughout the studies, we maintained each tag in a same state. Chipolo Classic, Galaxy Smart Tag Plus, Tile Slim, and HTC-Fetch will all be used on the Samsung A13 device. We utilize our subject's bicycle and rucksack and put our Bluetooth tracker in his pocket. Our trackers, however, were marked with distinctive names and identifiable icons. We installed updated Bluetooth tracker applications till August 27, 2023. All Bluetooth tags are connected to their mother device when used for observation. We followed Mr. X from the beginning point to the destination point as our Mr. X (willingly). To preserve Bluetooth communication with the beacon, we retain a safe distance. We confirm that Mr. X was not utilizing the AirGuard protection program [57] or any other Bluetooth tracker during our testing. We discovered that the Bluetooth beacons we use need to update their position status more quickly. To do this, we use manual scanning (refresh location) to go to the target. Typically, our Android system's sensor data requires precise location, like GPS, that we choose for our devices to identify. For those Bluetooth beacon devices to broadcast their location on a map and connect with their mother devices through Bluetooth signal, we activated the Bluetooth and Location service on the Samsung A13 phone.

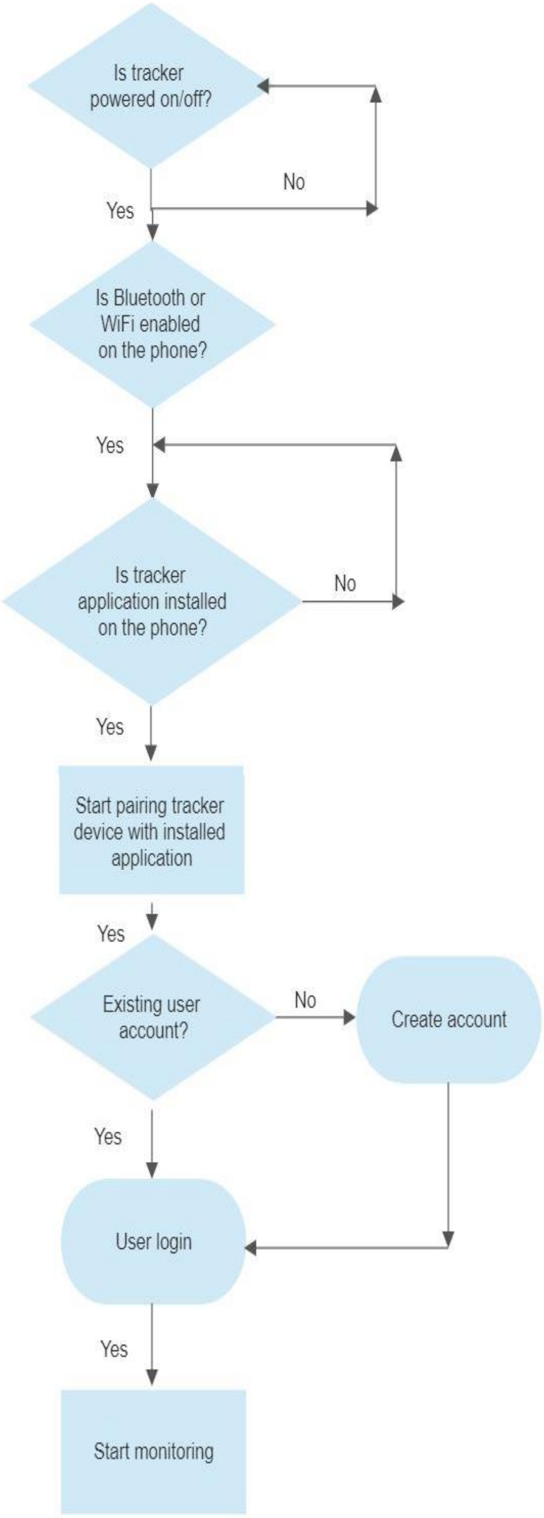

Is tracker powered on/off?
No
Yes
Is Bluetooth or WiFi enabled on the phone?
Yes
Is tracker application installed on the phone?
No
Yes
Start pairing tracker device with installed application
Yes
Existing user account?
No
Create account
Yes
User login
Yes
Start monitoring

Figure 13. Initial process for Bluetooth tracker with mother device

3.2.1 Devices and Toolsets

Now, we are going to provide essential information on the apparatus and resources utilized for data development in our study. We provide more detailed information on the tools and software used in this study for the subsequent data collection procedure, including the app version we used for the Bluetooth tracker and smartphone. Our system for forensic investigation is the computer. Utilize various tools to interact with Windows and Linux systems and collect data from them.

Table 3. List of tools used in our experiment

Operating System	Tools	Version	Reason for
Windows	ADB(android debug bridge)	1.0.41	To interact with Android device
Windows	Android Backup Extractor	V20210224105130	To make a tar file from the extracted file
Windows	Apache Maven	3.8.6	To pack and unpack the jar file
Windows	DB Browser (SQLite)	3.12.2	To see data in a human-readable format
Windows	Android Studio	2022.2.1	To browse the Android device file system
Windows	Realm Studio	14.0.3	To read the realm database file
Windows	Hash Calculator	2.02	To confirm the integrity of the acquired image file/dd file
Windows	Notepad++	8.4.9	To read the string in human-readable format
Android	Magisk Manager	26.1	To root Android phone
Android	nRF Connect	4.26.1	To scan and enumerate BLE tag
Android	BusyBox	1.32.0	To establish a TCP connection between a PC and a mobile device
Windows	7zip	22.01	To extract .tar file

The Windows system handles most of our analytic tasks due to its system-dependent tools, capacity to handle related processes, and simplicity of use case. Kali Linux was also chosen because it has security tools, applications, and commands like dd that we used during the physical extraction phase of our data collection.

Table 4. Devices and Operating systems used for our experiments

Host Machine	Operating System	Version
LENOVO 80J2	Windows 10 Pro N (64-bit)	10.0.19045 Build 19045
ASUS K551L	Kali Linux	2023.1
Samsung Galaxy A13	Android	13
Samsung Galaxy A53 5G	Android	13

The Android Debug Bridge (ADB), which gives investigators a solid way to access and retrieve data from Android smartphones [72] using a computer, is a crucial tool for forensic research. It is an indispensable tool for every forensic investigator due to its adaptability and broad range of skills. The analytical capabilities of the SQLite browser [72] are crucial for locating pertinent data and creating links between the metadata and the SQLite database. For data reconstruction and recovery, data parsing and interpretation, recovering lost data, or metadata analysis. A vital tool in digital forensics is the Android backup extractor. We used an extractor to extract our ADB backup file into a tar file.

Table 5. List of Bluetooth trackers we used for our study

Bluetooth-Tracker	Firmware-Version	Tagged Device/OS	Bluetooth
Chipolo Classic	4.21.0	Samsung Galaxy A13 5G / Android 13	4.0
Galaxy Smart Tag Plus	1.8.13-19	Samsung Galaxy A13 5G / Android 13	5.0
Tile Slim	2.1150.0 (5050)	Samsung Galaxy A13 5G / Android 13	4.0
HTC-Fetch	1.6	Samsung Galaxy A13 5G / Android 13	4.0

3.2.2 Connectivity between Device and Tools

To enable communication between our mobile device and the host computer, we installed the Samsung USB driver firmware on the Windows computer. Applications for Bluetooth trackers have been downloaded from the Google Play Store. To distinguish each tracker, we create a random profile in the Bluetooth application and use a unique tag name. We do not need to make a profile or have any login information to interact with the HTC-Fetch tracker. This gadget is plugand-play in nature. We employ profile and login information for the rest of our Bluetooth tracker to grant users dashboard access. The Samsung gadget supports Bluetooth 5.0, which also includes backward compatibility. Three of our Bluetooth trackers employ Bluetooth 4.0, which enables communication between our Samsung gadget and a Bluetooth 4.0 device. We do not utilize an offline map for our Bluetooth tag; thus, our primary device, a Samsung, was linked to the internet to load the map. For Android, we connect to the host device via a USB-C cable.

3.2.3 Data Collection Process

This session will outline our procedure to produce and gather data. We independently installed programs for the four Bluetooth trackers on the Samsung Galaxy A13 Android phone. Then, using Bluetooth tags, we matched the tracker with the profile we had built online. Depending on the application's needs, we populate the user profile with extra data when the device is linked. After finishing the procedure, the dashboard and Bluetooth tracker notifications are shown, and we can observe active Bluetooth tags on the map.

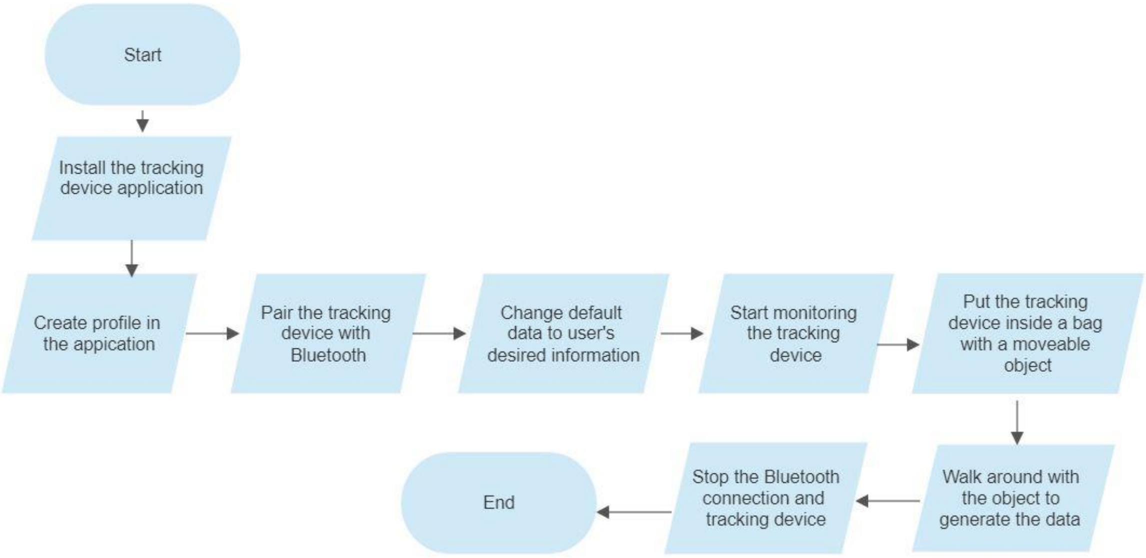

Start
Install the tracking device application
Create profile in the appication
Pair the tracking device with Bluetooth
Change default data to user's desired information
Start monitoring the tracking device
Put the tracking device inside a bag with a moveable object
Walk around with the object to generate the data
Stop the Bluetooth connection and tracking device
End

Figure 14. Sample process diagram for data collection

3.3 Summary

In the mobile device forensic analysis, a comprehensive approach combines various techniques. We applied manual extraction, which involves hands-on examination of devices, while logical extraction focuses on file-level data retrieval. We also applied physical extraction to go deeper into raw access of storage system. Test and analysis validate our findings guided by a well defined experiental plan. Including hardware connectors and software tools facilitate efficiently on data collection process. We maintain seamless connectivity between Bluetooth tracking tags and Android devices during data collection proces.

4 Result and Discussion

We discuss our results by following the mobile forensics acquisition process. At the beginning of our forensics process, we applied manual extraction to Bluetooth tracker applications, and we will discuss all of our findings. We first acquired data from the non-rooted device and then observed data from the rooted device in the logical acquisition process. We apply the physical acquisition process at the end and enumerate the findings.

4.1 Manual Extraction Process

We display the data that we discovered throughout the manual data extraction process on each Bluetooth tracker application's user interface or dashboard. We use the manual extraction procedure to extract as much data as possible. Each Bluetooth tracker application will be detailed along with findings in a subcategorized portion of this chapter.

4.1.1 Chipolo Classic

From the starting point, we start observing Chipolo activity with the help of active Bluetooth and enabled Location settings. The connection of the tracker with the mother device was engaged, and the beacon device location was also seen on the map. So we can navigate through the dashboard of the application easily and can locate the tracker. This situation is essential for the investigator when following the triage procedure. During the triage investigation process, the Forensics investigator has to ensure that the Bluetooth beacon is connected to the tracker application. When people use their data for smartphones to use applications, this information resides on the phone storage for further use or communications. It can be accessed by the forensics investigator when necessary. The exterior shows the model number M45S-2017, FCC ID: 2AD85-M45S[5]. FCC-ID helps us to identify the tag manufacturer [59] immediately. We found the tag's generic name: Chipolo Classic. From the application dashboard, we found the location of the Bluetooth tag. We can see the Home and Work address locations on the other tab of the application. The user email address that he used for registering this tag. We found the tag color is yellow and the firmware version. We are expecting more artifacts during the logical and physical data acquisition phase. More screenshots are attached in the Appendix section of this paper.

[5] https://fccid.io/2AD85

4.1.2 Galaxy Smart Tag Plus

We discovered the model EI-T7300, FCC-ID: A3LEIT7300[6], and S/N: RF7RA00E09XAMB from Galaxy Smart Tag Plus Hardware Observation. The investigator may uniquely identify the device maker using the FCC-ID and serial number, which is crucial information for the triage stage. These data can be obtained from the installed application, and the examiner can relate them together for further confirmation. The user-defined tag name and icon are shown on the application dashboard. We also notice a specific ringtone set on the tag settings. The address on the dashboard is the last position that was updated at 01:49 PM (13:49) at the target point, and it is evident that the tag is attached there. Every updated location is also displayed as a history in the application's history tab.

4.1.3 HTC-Fetch

From the HTC-Fetch application dashboard, we can see that the tag last connected on August 27, 2023, at 02:06:24 PM, and it remained the same when we reached the destination location. We locate the model no: BLA100, FCC ID: EMJTBLA100[7] from the hardware. We tried to refresh manually from the dashboard to see if there were any changes according to the time and location, but there was no option for the user to update the time with the updated site. The dashboard shows the user-defined tracker tag “megha’s key” with a golden key icon. From the beginning of our journey, it showed its location on the dashboard with GPS coordinates till the destination location. This application also offers the plus code address on the application dashboard. Plus code are like street addresses for the places that do not have one. This plus code is based on latitude and longitude and displays a combination of numbers and letters. GPS data can be helpful in criminal investigation and can be extracted and used as digital evidence, as Devon R. Clark et al. mentioned in their research [60]. More information and screenshots are in the Appendix section.

4.1.4 Tile Slim

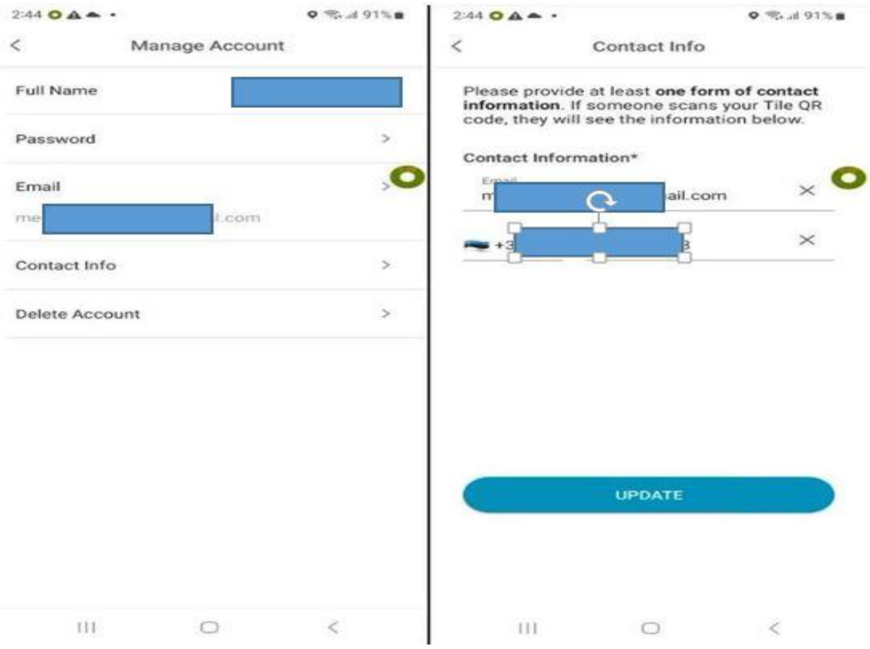
From the Tile Slim, we found the model number T1601S on the back side of the tracker and also from the dashboard, FCC ID: 2ABXLT1601S[8] from the product catalog. The dashboard shows the category of the tag that the user-defined and the type of the tracker. Location history shows the immediate location on the map when the tracker updates its location. The

[6] https://fccid.io/A3LEIT7300

[7] https://fccid.io/EMJTBLA100

[8] https://fccid.io/2ABXLT1601S

application we used was a regular version, not a paid subscription. Location history can be kept in the premium application for up to one month. So, we can retrieve the location history for one day because it will keep a record only for the last day. We obtained the user's full name, email address, mobile phone number, and the tracker connected time with the location address from the application dashboard. We assume that we can acquire more data from the application folder during data acquisition.



2:44 91%

Manage Account

Full Name [redacted]

Password

Email

me[redacted]l.com

Contact Info

Delete Account

2:44 91%

Contact Info

Please provide at least **one form of contact information**. If someone scans your Tile QR code, they will see the information below.

Contact Information*

Email

m[redacted]ail.com

+3[redacted]3

UPDATE

Figure 15. User related information from the Tile Slim dashboard

Table 6. Data from manual extraction phase from each tracker

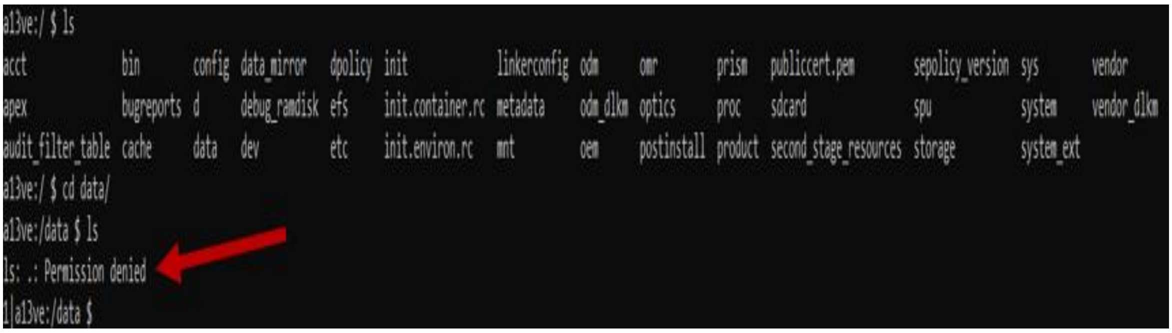
Artifacts	Chipolo Classic	Samsung Smart Tag Plus	HTC-Fetch	Tile Slim
FCC-ID	✓	✓	✓	✓
Tag name	✓	✓	✓	✓
Tag type	✓	✓	✓	✓
User nick name	✓	✓	✓	✓
User full name	✗	✗	✗	✓
Tag icon	✓	✓	✓	✓
User email	✓	✗	✗	✓
User phone number	✗	✗	✗	✓
User physical address	✓	✗	✗	✗
Starting point location	✓	✓	✓	✓
Last updated location	✓	✓	✓	✓
GPS coordinates	✗	✗	✓	✗
Last connected time	✓	✓	✓	✓
Street view	✗	✗	✓	✗
Firmware version	✓	✓	✓	✓

4.2 Logical Extraction Process

The information gleaned from the Samsung A13 handset throughout the logical extraction procedure will be presented in this chapter. We focused on the folders that created the Bluetooth tracker programs established upon installation for this procedure. We use the same logical extraction procedure on both rooted and non-rooted Android devices. We also go through the specific steps that we take on each level.

4.2.1 Data from Non-rooted Phone

In this step, we attempt to collect data from an Android device that was not rooted and was utilized for our logical extraction test. This technique is used to get data from a mobile device and does not necessitate system root access. A user-accessible copy of the file, such as application data from the Android backup, may be made using this procedure. This method does not recover deleted data. Several tools are on the market for logical extraction from an Android smartphone. An effective command-line tool is ADB, where investigators communicate with Android devices. We use the ADB backup process to do logical extraction from the Android device, which allows for data quality assessment [61]. Lwin et al. [62] stated that Android Backup is a viable option for logical extraction, although they recommended using the Magnet Acquire tool. This tool is commercial and not free to use. For our study, we plan to use open-source tools like ADB. We enable the USB debugging mode on the Android phone to collect data from non-rooted Android devices. It allows investigators to access data on the phone. Then, we connect our Samsung Galaxy A13 Android device with a Windows host machine using a USB-C cable and start with CMD using the following command and corresponding screenshots. We have full access to the Samsung A13 mother device, which was used for Bluetooth tracking devices and their applications. This device has no external storage. So, we expect all installed applications to reside in the internal storage system. We have screen lock codes that could be requirements for making backup files and extraction of the backup process. At this stage, we do not have any administrative-level access to the file system of this mobile phone. We cannot list the /data to see the available directory; the permission was denied.



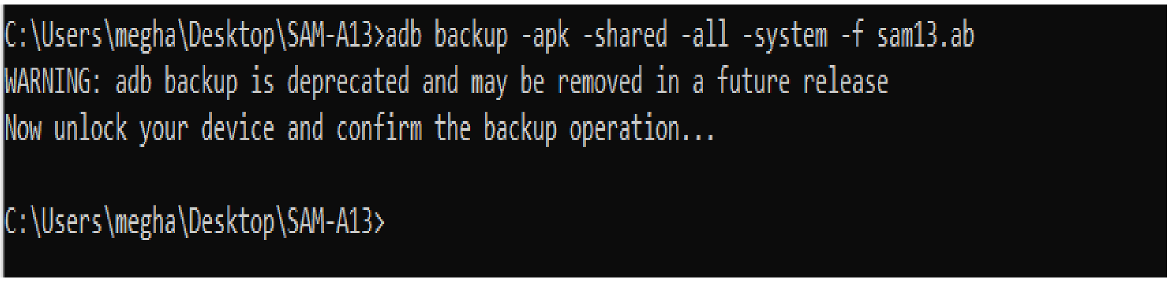
```
a13ve:/ $ ls
acct               bin        config data_mirror   dpolicy init              linkerconfig odm      omr         prism   publiccert.pem         sepolicy_version sys        vendor
apex               bugreports d      debug_ramdisk efs     init.container.rc metadata     odm_dlkm optics      proc    sdcard                 spu              system     vendor_dlkm
audit_filter_table cache      data   dev           etc     init.environ.rc   mnt          oem      postinstall product second_stage_resources storage          system_ext
a13ve:/ $ cd data/
a13ve:/data $ ls
ls: .: Permission denied
1|a13ve:/data $
```

Figure 16. Non-root access on Android device

So, first, we fully backed up the Samsung A13 device by executing the following command.

C:\Users\megha>adb devices	[This shows the attached device in brief]
C:\Users\megha>adb devices –l	[This command gives more specific information about connected device]
C:\User\megha>adb shell	[This gives the shell access but not as root]

C:\User\megha\Desktop\SAM-A13>adb backup –apk –shared –all –system –f sam13.ab



```
C:\Users\megha\Desktop\SAM-A13>adb backup -apk -shared -all -system -f sam13.ab
WARNING: adb backup is deprecated and may be removed in a future release
Now unlock your device and confirm the backup operation...

C:\Users\megha\Desktop\SAM-A13>
```

Figure 17. Android device full backup process on non-rooted device

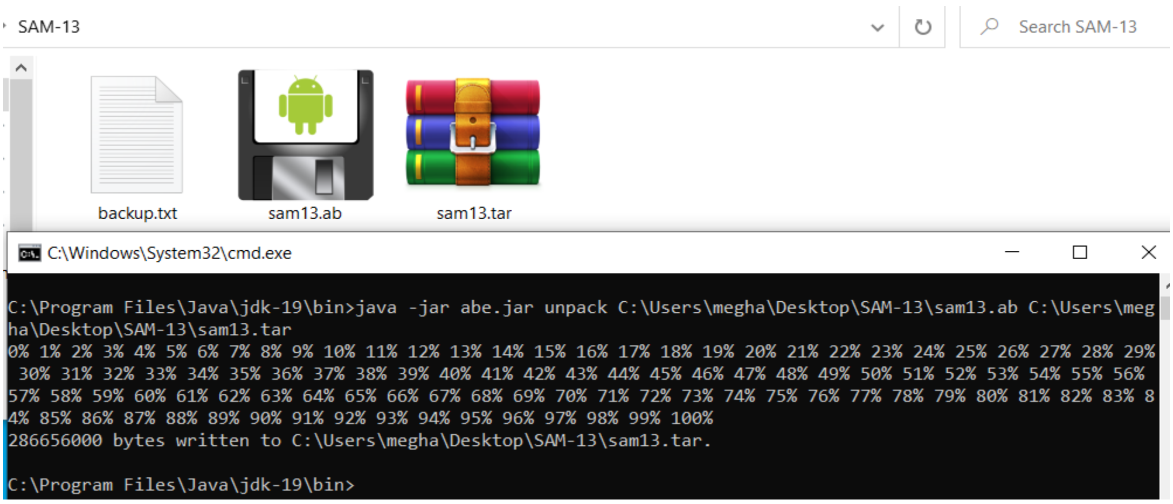
When we apply this command, we receive a prompt on the device screen to confirm the process with the passkey, as the device has a lock code. Because of this, our backup was encrypted; hence, we had to provide a lock code to make a tar file of this backup file. After making the backup file, we make the .ab file into a tar file using the abe.jar file. The abe.jar file is a component of Nikolay Elenkov's open-source project, which enables an investigator to extract an Android backup file (.ab) into a tar file[9] . Due to the DEFLATE[10] algorithm's compression and the possibility of AES encryption, if a password is supplied when generating the backup (correctly producing the original full-size of data or file), the Android Backup file format requires this passkey to make a tar file.

C:\Program Files\Java\jdk-19\bin>java -jar abe.jar unpack C:\Users\megha\ Desktop\SAM13\sam13.ab C:\Users\megha\Desktop\SAM-13\sam13.tar

Now, the sam13.ab file has been successfully extracted into a sam13.tar file. Then, we extract the tar file into a zip file using the 7zip tool.

[9] https://github.com/nelenkov/android-backup-extractor

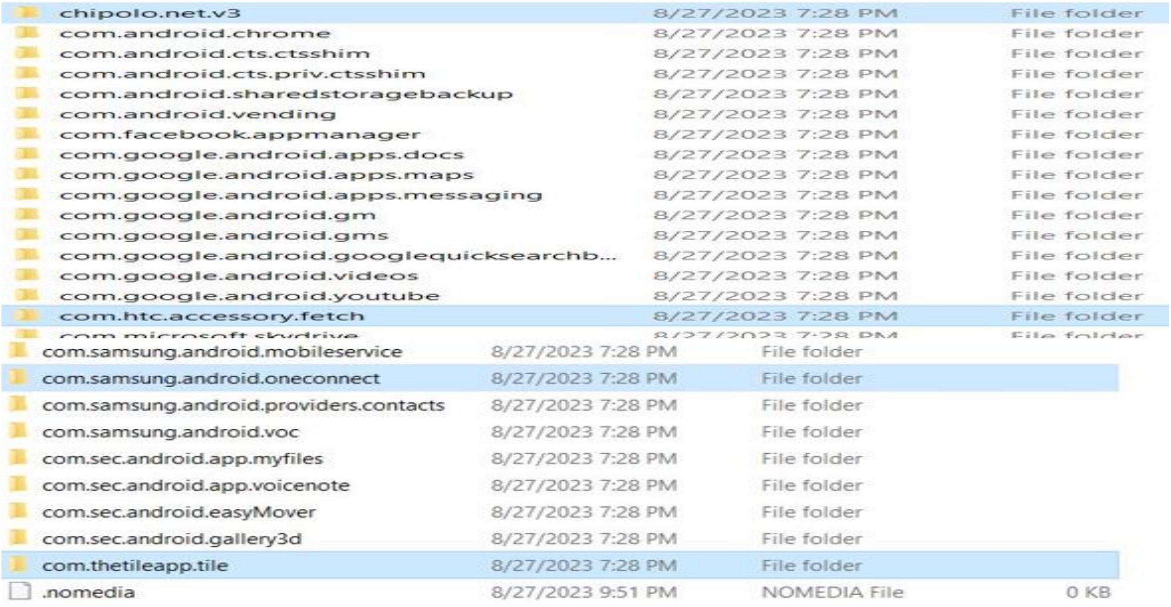
[10] https://en.wikipedia.org/wiki/Deflate



```
C:\Program Files\Java\jdk-19\bin>java -jar abe.jar unpack C:\Users\megha\Desktop\SAM-13\sam13.ab C:\Users\meg
ha\Desktop\SAM-13\sam13.tar
0% 1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 12% 13% 14% 15% 16% 17% 18% 19% 20% 21% 22% 23% 24% 25% 26% 27% 28% 29%
 30% 31% 32% 33% 34% 35% 36% 37% 38% 39% 40% 41% 42% 43% 44% 45% 46% 47% 48% 49% 50% 51% 52% 53% 54% 55% 56%
57% 58% 59% 60% 61% 62% 63% 64% 65% 66% 67% 68% 69% 70% 71% 72% 73% 74% 75% 76% 77% 78% 79% 80% 81% 82% 83% 8
4% 85% 86% 87% 88% 89% 90% 91% 92% 93% 94% 95% 96% 97% 98% 99% 100%
286656000 bytes written to C:\Users\megha\Desktop\SAM-13\sam13.tar.

C:\Program Files\Java\jdk-19\bin>
```

Figure 18. Unpacking the .ab file to .tar file

The "apps" and "shared" folders may be found in the sam13 folder after the sam13.tar file was extracted using 7zip. We discovered the Bluetooth application folders and their subfolders that contain encrypted databases and unknown-format files. We were unable to recover any humanreadable data from those files. Then, we use a rooted device to proceed to our following extraction process.



chipolo.net.v3	8/27/2023 7:28 PM	File folder	
com.android.chrome	8/27/2023 7:28 PM	File folder	
com.android.cts.ctsshim	8/27/2023 7:28 PM	File folder	
com.android.cts.priv.ctsshim	8/27/2023 7:28 PM	File folder	
com.android.sharedstoragebackup	8/27/2023 7:28 PM	File folder	
com.android.vending	8/27/2023 7:28 PM	File folder	
com.facebook.appmanager	8/27/2023 7:28 PM	File folder	
com.google.android.apps.docs	8/27/2023 7:28 PM	File folder	
com.google.android.apps.maps	8/27/2023 7:28 PM	File folder	
com.google.android.apps.messaging	8/27/2023 7:28 PM	File folder	
com.google.android.gm	8/27/2023 7:28 PM	File folder	
com.google.android.gms	8/27/2023 7:28 PM	File folder	
com.google.android.googlequicksearchb...	8/27/2023 7:28 PM	File folder	
com.google.android.videos	8/27/2023 7:28 PM	File folder	
com.google.android.youtube	8/27/2023 7:28 PM	File folder	
com.htc.accessory.fetch	8/27/2023 7:28 PM	File folder	
com.microsoft.skydrive	8/27/2023 7:28 PM	File folder	
com.samsung.android.mobileservice	8/27/2023 7:28 PM	File folder	
com.samsung.android.oneconnect	8/27/2023 7:28 PM	File folder	
com.samsung.android.providers.contacts	8/27/2023 7:28 PM	File folder	
com.samsung.android.voc	8/27/2023 7:28 PM	File folder	
com.sec.android.app.myfiles	8/27/2023 7:28 PM	File folder	
com.sec.android.app.voicenote	8/27/2023 7:28 PM	File folder	
com.sec.android.easyMover	8/27/2023 7:28 PM	File folder	
com.sec.android.gallery3d	8/27/2023 7:28 PM	File folder	
com.thetileapp.tile	8/27/2023 7:28 PM	File folder	
.nomedia	8/27/2023 9:51 PM	NOMEDIA File	0 KB

Figure 19. Extracted folders from backup tar file

Next, we used the “adb pull” command to pull data from the Android device to the Windows system in the path /sdcard/Android/data.

```
C:\User\megha\Desktop\pull-data>adb pull –a /sdcard/Android/data/chipolo.net.v3
```

```
C:\User\megha\Desktop\pull-data>adb pull –a /sdcard/Android/data/com.htc.accessory.fetch
```

```
C:\User\megha\Desktop\pull-data>adb pull –a /sdcard/Android/data/com.samsung.android. Oneconnect
```

```
C:\User\megha\Desktop\pull-data>adb pull –a /sdcard/Android/data/com.thetileapp.tile
```

We retrieved all the Bluetooth application related data directories of the phone. We may have been more effective in this phase by thoroughly examining the extracted files and directories to find any forensically sound artifacts. However, we discovered the same files as we found from the full backup image of a non-rooted phone. Unexpectedly, no critical files can unveil forensic artifacts related to installed Bluetooth applications. Then, we use a rooted device to proceed to our following extraction process.

4.2.2 Data from Rooted Phone

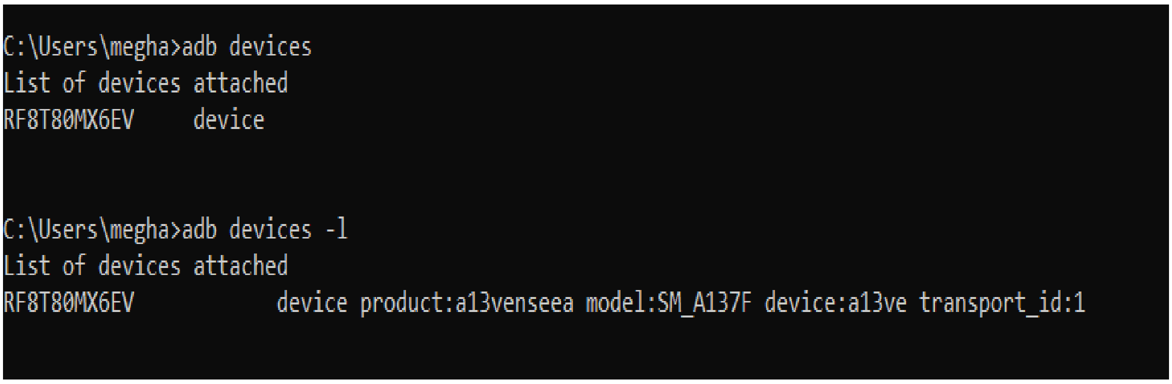
Our Samsung A13 mobile device was not rooted before and had never been used. In order to access our Android smartphone and collect additional data, we need administrator rights to investigate further user information on the installed Bluetooth tracker application. In order to effectively root our Samsung Galaxy A13 mobile, we used a technical blog [63]. An Android smartphone's bootloader must be unlocked to gain root access. Here, we must focus on unlocking the phone's bootloader, which might result in a reset exactly like factory settings. As a result, the Bluetooth tracking application's filled data will not be accessible. Therefore, we completed the entire Android backup process and kept rooting the phone. As previously stated, the backup command is executed while performing logical extraction on a non-rooted device.

```
C:\User\megha\Desktop\SAM-R-backup>adb backup –apk –shared –all –system –f sam13.ab
```

Table 7. Rooting process of our Android phone

Step	Action
1	Before proceeding with the rooting process, it is essential to back up all critical data.
2	Enable USB Debugging: • Go to the "Settings" menu on your Samsung phone. • Scroll down to find "Developer options." • Enable "USB Debugging." If you don't see the "Developer options," go to "About phone" and tap on the "Build number" seven times to unlock it.
3	**Download necessary files:** Download the latest version of Magisk Manager APK from GitHub Project and the appropriate firmware for our Samsung phone model from a trusted source. Also, download the Odin tool from the official website.
4	**Install Magisk Manager:** Transfer and install the downloaded Magisk Manager APK to our phone. It prompted to allow installation from unknown sources.
5	**Extract firmware files:** Extract the downloaded firmware file on the computer using a file extraction tool like 7zip.
6	Boot our Samsung A13 phone into Download Mode: Power off your phone completely. Press and hold the Volume Down + Home + Power buttons simultaneously until we see a warning message. To activate Download Mode, use the Volume Up button.
7	We use a USB-C cable to connect your Samsung phone to the computer. We ensure we install the necessary Samsung USB drivers on our forensic station.
8	**Open Odin tool:** Launch the Odin tool on the computer. We connected our phones successfully. We see an "Added!" message in the Odin tool.
9	**Select firmware files in Odin:** In the Odin tool. Click on the "AP" button and select the extracted firmware file with the ".tar"
10	**Start the flashing process:** Double-check that only the "Auto Reboot" and "F. Reset Time" options are selected in Odin. Then, click the "Start" button to begin the flashing process.
11	**Root with Magisk:** The phone automatically reboots once the flashing process is complete. After the phone boots up, open the Magisk Manager app. It detects our phone is not rooted and prompts us to install Magisk. We follow the on-screen instructions to install Magisk.
12	**Reboot and verify root access:** After installing Magisk, reboot our phone. Once it boots up, open the Magisk Manager app again, then check "Magisk is installed" and "Root access is properly installed."
13	To confirm the “Root” access, we installed another tool, “Root Checker Basic.” It also ensures that the access level is changed to Root.

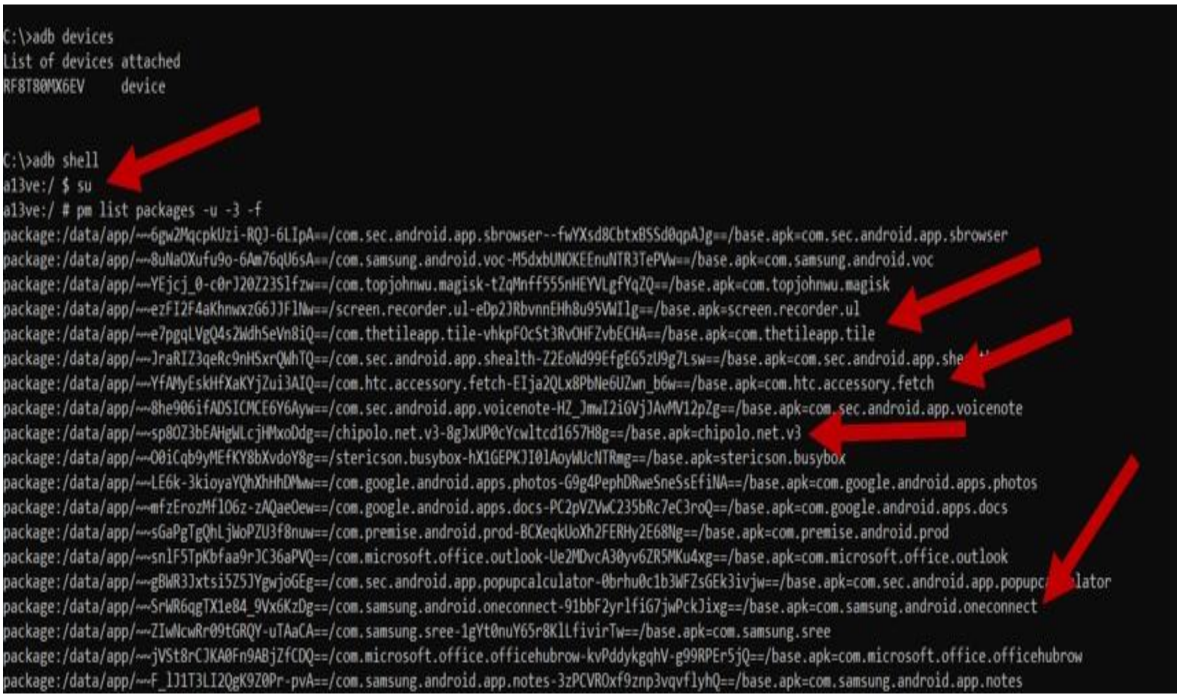
After getting root privileges, we restored our backup, pulled data to the phone, and followed the same procedure we followed for data extracting in logical acquisition for non-rooted phones. This time, we have the Super-User command, which gives us privileged access. This time, the ADB shell gives us more specific information about the connected Android phone's codename and model number. We list on the phone to see the installed third-party applications.



```
C:\Users\megha>adb devices
List of devices attached
RF8T80MX6EV     device


C:\Users\megha>adb devices -l
List of devices attached
RF8T80MX6EV           device product:a13venseea model:SM_A137F device:a13ve transport_id:1
```

Fig 20. Connected Android device details

C:\User\megha>adb shell	[This give the shell access but not root]
a13ve:/ $ su	[SU gives us root privileges with # sign]
a13ve:/ # pm list packages -u -3 –f	[To list all 3rd party installed applications]



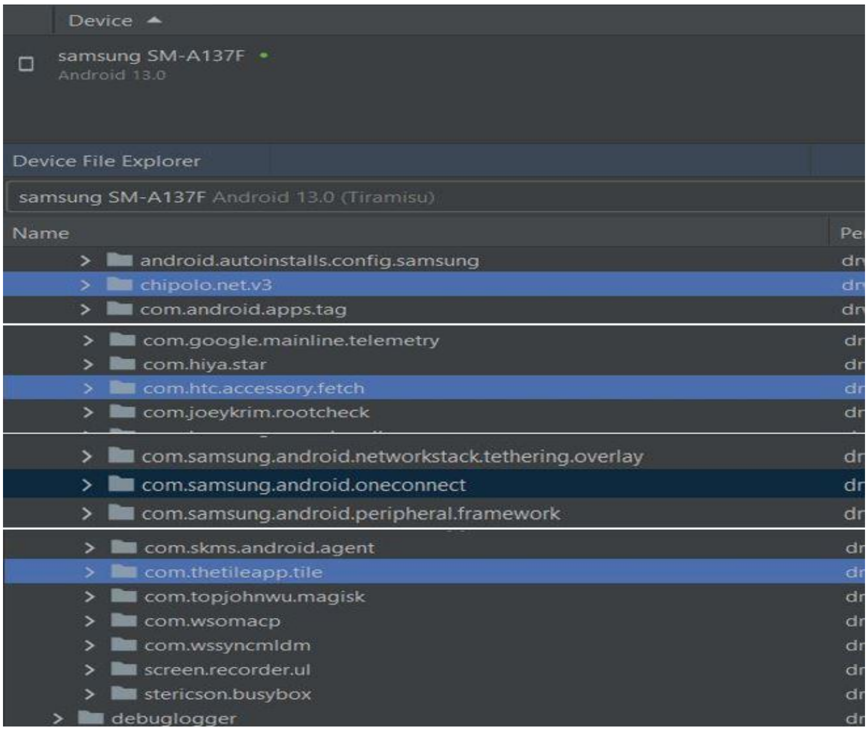
```
C:\>adb devices
List of devices attached
RF8T80MX6EV     device


C:\>adb shell
a13ve:/ $ su
a13ve:/ # pm list packages -u -3 -f
package:/data/app/~~6gw2MqcpkUzi-RQJ-6LIpA==/com.sec.android.app.sbrowser--fwYXsd8CbtxBSSd0qpAJg==/base.apk=com.sec.android.app.sbrowser
package:/data/app/~~8uNaOXufu9o-6Am76qU6sA==/com.samsung.android.voc-M5dxbUNOKEEnuNTR3TePVw==/base.apk=com.samsung.android.voc
package:/data/app/~~YEjcj_0-c0rJ20Z23Slfzw==/com.topjohnwu.magisk-tZqMnff555nHEYVLgfYqZQ==/base.apk=com.topjohnwu.magisk
package:/data/app/~~ezFI2F4aKhnwxzG6JJFlNw==/screen.recorder.ul-eDp2JRbvnnEHh8u95VWIlg==/base.apk=screen.recorder.ul
package:/data/app/~~e7pgqLVgQ4s2WdhSeVn8iQ==/com.thetileapp.tile-vhkpFOcSt3RvOHFZvbECHA==/base.apk=com.thetileapp.tile
package:/data/app/~~JraRIZ3qeRc9nHSxrQWhTQ==/com.sec.android.app.shealth-Z2EoNd99EfgEG5zU9g7Lsw==/base.apk=com.sec.android.app.shealth
package:/data/app/~~YfAMyEskHfXaKYjZui3AIQ==/com.htc.accessory.fetch-EIja2QLx8PbNe6UZwn_b6w==/base.apk=com.htc.accessory.fetch
package:/data/app/~~8he906ifADSICMCE6Y6Ayw==/com.sec.android.app.voicenote-HZ_JmwI2iGVjJAvMV12pZg==/base.apk=com.sec.android.app.voicenote
package:/data/app/~~sp8OZ3bEAHgWLcjHMxoDdg==/chipolo.net.v3-8gJxUP0cYcwltcd1657H8g==/base.apk=chipolo.net.v3
package:/data/app/~~O0iCqb9yMEfKY8bXvdoY8g==/stericson.busybox-hX1GEPKJI01AoyWUcNTRmg==/base.apk=stericson.busybox
package:/data/app/~~LE6k-3kioyaYQhXhHhDMww==/com.google.android.apps.photos-G9g4PephDRweSneSsEfiNA==/base.apk=com.google.android.apps.photos
package:/data/app/~~mfzErozMfl06z-zAQaeOew==/com.google.android.apps.docs-PC2pVZVwC235bRc7eC3roQ==/base.apk=com.google.android.apps.docs
package:/data/app/~~sGaPgTgQhLjWoPZU3f8nuw==/com.premise.android.prod-BCXeqkUoXh2FERHy2E68Ng==/base.apk=com.premise.android.prod
package:/data/app/~~snlF5TpKbfaa9rJC36aPVQ==/com.microsoft.office.outlook-Ue2MDvcA30yv6ZR5MKu4xg==/base.apk=com.microsoft.office.outlook
package:/data/app/~~gBWR3Jxtsi5Z5JYgwjoGEg==/com.sec.android.app.popupcalculator-0brhu0c1b3WFZsGEk3ivjw==/base.apk=com.sec.android.app.popupcalculator
package:/data/app/~~SrWR6qgTX1e84_9Vx6KzDg==/com.samsung.android.oneconnect-91bbF2yrlfiG7jwPckJixg==/base.apk=com.samsung.android.oneconnect
package:/data/app/~~ZIwNcwRr09tGRQY-uTAaCA==/com.samsung.sree-1gYt0nuY65r8KlLfivirTw==/base.apk=com.samsung.sree
package:/data/app/~~jVSt8rCJKA0Fn9ABjZfCDQ==/com.microsoft.office.officehubrow-kvPddykgqhV-g99RPEr5jQ==/base.apk=com.microsoft.office.officehubrow
package:/data/app/~~F_lJ1T3LI2QgK9Z0Pr-pvA==/com.samsung.android.app.notes-3zPCVROxf9znp3vqvflyhQ==/base.apk=com.samsung.android.app.notes
```

Figure 21. adb shell with root access on rooted device

Another critical location is where user data resides in the /data/data partition[11]. This partition, also known as userdata, contains the user's data, such as contracts, messages, settings, and applications that the user installed. Android devices and file systems differ depending on the vendor and their products. In their study, Ömer Faruk Yakut et al. used Android Studio to detect and analyze pirated Android application behavior that targets user data [64]. In our study, we used Android device file explorer, which is built-in with Android Studio. We found exciting files for every Bluetooth application through the Android device file explorer. We are going to describe all of those findings next.

4.2.3 Data from Device Explorer

The only location in the entire file system that apps can write to is each app's/data/data subdirectory. This directory, along with stock applications for location, texting, and calls on all Android devices, makes crucial for forensic analysis [65]. So, we are trying to explore more on the partition /data/data using Android studio, and we have some exciting artifacts that can be present in court as a shred of forensics evidence for a stalking attack case file.



```
> android.autoinstalls.config.samsung      dr
> chipolo.net.v3                           dr
> com.android.apps.tag                     dr
> com.google.mainline.telemetry            dr
> com.hiya.star                            dr
> com.htc.accessory.fetch                  dr
> com.joeykrim.rootcheck                   dr
> com.samsung.android.networkstack.tethering.overlay   dr
> com.samsung.android.oneconnect           dr
> com.samsung.android.peripheral.framework dr
> com.skms.android.agent                   dr
> com.thetileapp.tile                      dr
> com.topjohnwu.magisk                     dr
> com.wsomacp                              dr
> com.wssyncmldm                           dr
> screen.recorder.ul                       dr
> stericson.busybox                        dr
> debuglogger                              dr
```

Figure 25. Device file explorer on Android Studio

[11] https://thesecmaster.com/explore-the-android-file-system-hierarchy-in-depth/#sdcard

Table 8. List of artifacts obtained by logical extraction from rooted phone using device explorer

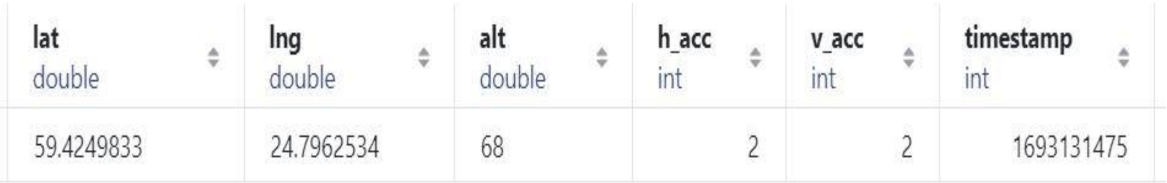
Item	Location	File type	Application	Data
1	/data/data/chipolo.net.v3/cache/log/chipolo_0.log	.txt	Chipolo Classic	MAC address, ID, last update time, tag color, name of the tag, tag icon, GPS location of last seen location, location address
2	/data/data/chipolo.net.v3/files/.com.google.firebase.crashlytics.files.v2:chipolo.net.v3/open-sessions/64EBD98A0346000133B807087517BC0C/report	File	Chipolo Classic	Mobile device information
3	/data/data/chipolo.net.v3/files/default.realm	.realm	Chipolo Classic	User name, email address, MAC address, tag color code, user name with icon, last tag seen time, last GPS location, last seen address location, other location of user with address and GPS position
4	/data/data/chipolo.net.v3/shared_prefs/com.google.maps.api.android.lib6.drd.PREFERENCES_FILE.xml	.xml	Chipolo Classic	Legal country name
5	/data/data/com.samsung.android.oneconnect/shared_prefs/UserCache.xml	.xml	Galaxy Smart tag plus	User name, email, country code
6	/data/data/com.samsung.android.oneconnect/shared_prefs/PLUGIN_PLATFORM_PREFERENCE.xml	.xml	Galaxy Smart tag plus	Legal country name

7	/data/data/com.samsung.android.oneconnect/shared_prefs/FME_SELECTED_DEVICE.xml	.xml	Galaxy Smart tag plus	Device IMEI, user nick name, defined tag, last GPS position
8	/data/data/com.samsung.android.oneconnect/shared_prefs/MovablePluginData.b5ac6b74-14ed-4565-9146-b968006604e1.xml	.xml	Galaxy Smart tag plus	User ring tone, tag-type, firmware version
9	/data/data/com.samsung.android.oneconnect/shared_prefs/LaunchDarkly_TqmAUzY4nh7nFIXxQjfIVFbB2Il00ZeNrRZaprYOPqs=.xml	.xml	Galaxy Smart tag plus	Last connected time
10	/data/data/com.samsung.android.oneconnect/shared_prefs/com.google.android.gms.appid.xml	.xml	Galaxy Smart tag plus	Application version
11	/data/data/com.samsung.android.oneconnect/shared_prefs/FirebaseHeartBeatW0RFRkFVTFRd+MTozNDI2NjgwNjg3MjQ6YW5kcm9pZDo4OTYwOWM5MjVhOWUwZDc2.xml	.xml	Galaxy Smart tag plus	Last connected date
12	/data/data/com.htc.accessory.fetch/shared_prefs/HTCFETCHPREFS.xml	.xml	HTC-Fetch	Tag name, MAC address of Tag, last seen date and time
13	/data/data/com.htc.accessory.fetch/shared_prefs/com.google.maps.api.android.lib6.drd.PREFERENCES_FILE.xml	.xml	HTC-Fetch	Legal country
14	/data/data/com.thetileapp.tile/files/tile_log/com.thetileapp.tile2023-08-27.log	.txt	Tile Slim	MAC address of Tag, mobile device info, Tag name, Tag starting time.
15	/data/data/com.thetileapp.tile/shared_prefs/TilePrefs.xml	.xml	Tile Slim	User email address, phone number, application version, user

				profile name, last GPS location
16	/data/data/com.thetileapp.tile/shared_prefs/FirebaseHeartBeatW0RFRkFVTFRd+MTo5ODUxMzEzMTU0Mzc6YW5kcm9pZDplNDc1NTAzOTVkMDExMTgy.xml	.xml	Tile Slim	Last used date.
17	/data/data/com.thetileapp.tile/shared_prefs/com.google.maps.api.android.lib6.drd.PREFERENCES_FILE.xml	.xml	Tile Slim	Country domain
18	/data/data/com.thetileapp.tile/shared_prefs/com.appboy.storage.device_cache.v3.37a6259cc0c1dae299a7866489dff0bd.xml	.xml	Tile Slim	Phone model information, country time zone
19	/data/data/com.thetileapp.tile/no_backup/androidx.work.workdb	.db	Tile Slim	Tag last connected time
20	/data/data/com.thetileapp.tile/files/dcs_logs/priority_c_logs/events_c.log	.log	Tile Slim	App version, mobile device information

4.2.4 Acquired Data Item from Chipolo Classic

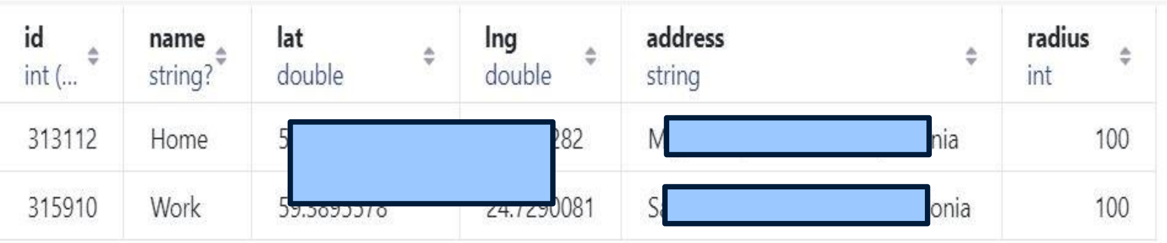
Using Android device explorer with the help of Android Studio, we retrieve interesting XML files and Realm database files. In Android forensics analysis, XML and Realm can contain valuable information about the device, applications, and uses. XML and Realm files stored structured data, user preferences, and application data. We use Notepad ++ to read the log and XML files and the Realm browser for the Realm database file. As we can see from the application dashboard, that tag user saves his home and office address.



lat double	lng double	alt double	h_acc int	v_acc int	timestamp int
59.4249833	24.7962534	68	2	2	1693131475

Fig 26. location and time in realm database file from Chipolo Classic application

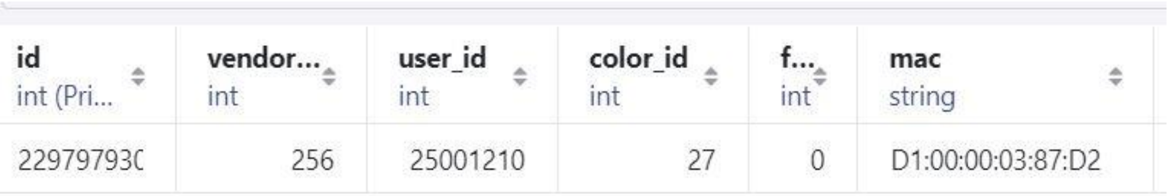
On the Samsung A13 device, we selected accurate location settings for using a map and turned on GPS location. The calibration of an Android smartphone also affects the precision of pinpointing on a map. Prior to data collection, our equipment identifies medium-range calibration during the GPS accuracy test. During the manual extraction phase from the dashboard of all the applications, we observed that the Bluetooth tags were shown on the map location with the physical address and time when we reached the destination location, and the tags were actively connected. We are interested in the connected tag on the location with time, which could help the investigator reconstruct the event with time and location in a stalking case.



id int (...	name string?	lat double	lng double	address string	radius int
313112	Home	[redacted]	[redacted]	[redacted]	100
315910	Work	[redacted]	[redacted]	[redacted]	100

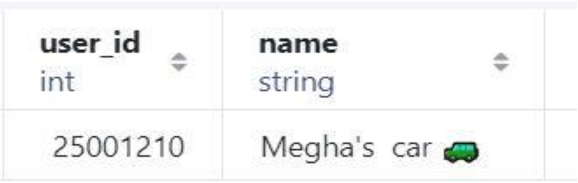
Fig 27. GPS location in realm database file from Chipolo Classic application

As we identify the stalker as the Bluetooth tracker user, we get his home and workplace addresses. We found artifacts from the uncovered realm database file. The Chipolo Classic tracker was yellow in color, as we can see (Fig 3, 4), and we found this information in the database file also. This tag was assigned by the stalker with the nickname "megha's car," it is also clearly shown in the application's recovered database with the provided icon. We discovered the stalker's email address from the program dashboard, as seen in the retrieved file. This tag's serial number is located in the user interface and matched with the tag's MAC address in the database table.



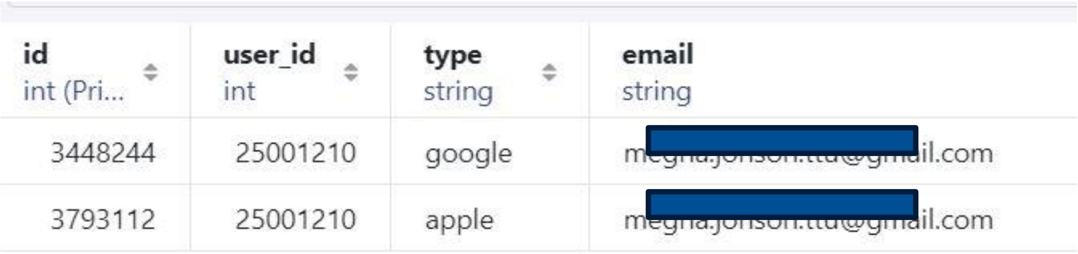
id int (Pri...	vendor... int	user_id int	color_id int	f... int	mac string
22979793C	256	25001210	27	0	D1:00:00:03:87:D2

Fig 28. User_id, color_id and MAC address of Chipolo Classic



user_id int	name string
25001210	Megha's car 🚙

Fig 29. User_id tag name of the tracker and icon

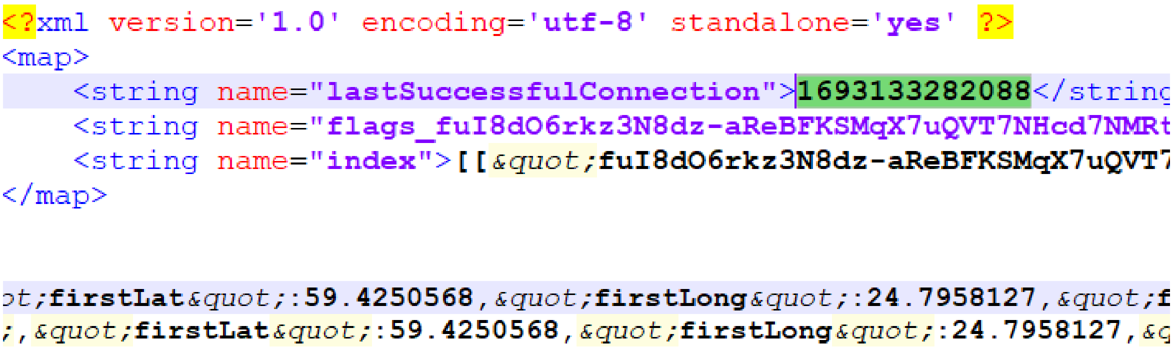


id int (Pri...	user_id int	type string	email string
3448244	25001210	google	[redacted]
3793112	25001210	apple	[redacted]

Fig 30. User email address for Chipolo Classic

4.2.5 Acquired Data Item from Galaxy Smart Tag Plus

We found the Galaxy Smart Tag Plus application folder as oneconnect. From the application dashboard, we found the destination address (Figure 6), which was the last seen location on the map connected with the tag, but we found the latitude and longitude from the XML file. The time is shown in UNIX epoch time format. This kind of Bluetooth device saves only the last updated location address.

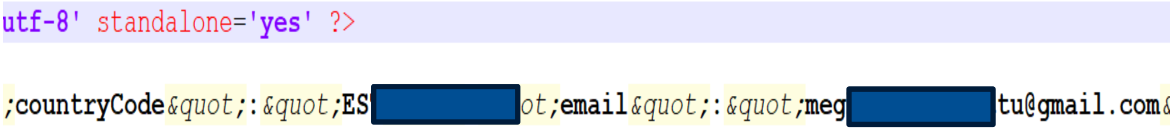


```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="lastSuccessfulConnection">1693133282088</string
    <string name="flags_fuI8dO6rkz3N8dz-aReBFKSMqX7uQVT7NHcd7NMRt
    <string name="index">[[&quot;fuI8dO6rkz3N8dz-aReBFKSMqX7uQVT7
</map>
```

```
ot;firstLat&quot;:59.4250568,&quot;firstLong&quot;:24.7958127,&quot;f
;,&quot;firstLat&quot;:59.4250568,&quot;firstLong&quot;:24.7958127,&q
```

Fig 31. GPS location and last seen time from Samsung Smart Tag Plus application

We also found the country code that tag resides in. The email address that the stalker used to sign up for the tracker and his full name. This application stores more information about the user and its mother device. We added more screenshots to the appendix section.

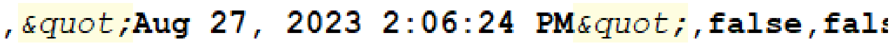


```
utf-8' standalone='yes' ?>

;countryCode&quot;:&quot;ES[redacted]ot;email&quot;:&quot;meg[redacted]tu@gmail.com&
```

Fig 32. Email address, country code

4.2.6 Acquired Data Item from HTC-Fetch

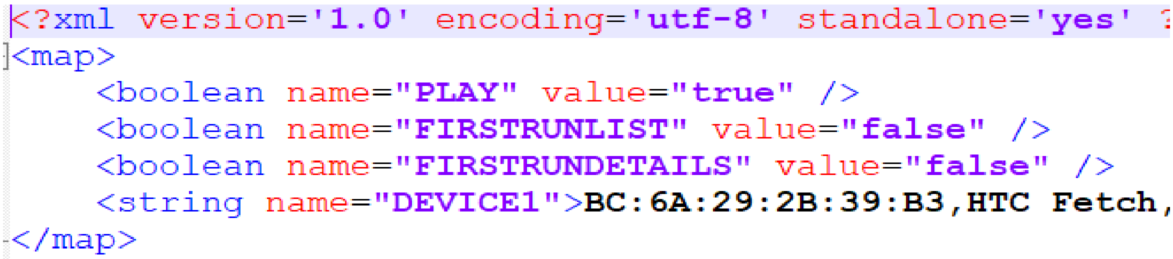
From the HTC fetch, we can recover the last connected time and date in human-readable format. We do not find any GPS coordinates or physical address in the application files, as we can see from the application dashboard (Fig 10).



```
,&quot;Aug 27, 2023 2:06:24 PM&quot;,false,fal
```

Fig 33. last connected date and time from HTC-Fetch application

The configuration file in the application folder contains the MAC address. The suspects and their devices may be uniquely identified by their MAC addresses, which are accepted as reliable evidence. Other Bluetooth trackers we utilize are less engaging than the HTC-Fetch program dashboard. During the manual extraction procedure, we observed the tracker position with a street view option and a plus code from the dashboard.



```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?
<map>
    <boolean name="PLAY" value="true" />
    <boolean name="FIRSTRUNLIST" value="false" />
    <boolean name="FIRSTRUNDETAILS" value="false" />
    <string name="DEVICE1">BC:6A:29:2B:39:B3,HTC Fetch,
</map>
```

Fig 34. MAC address from the HTC-Fetch application

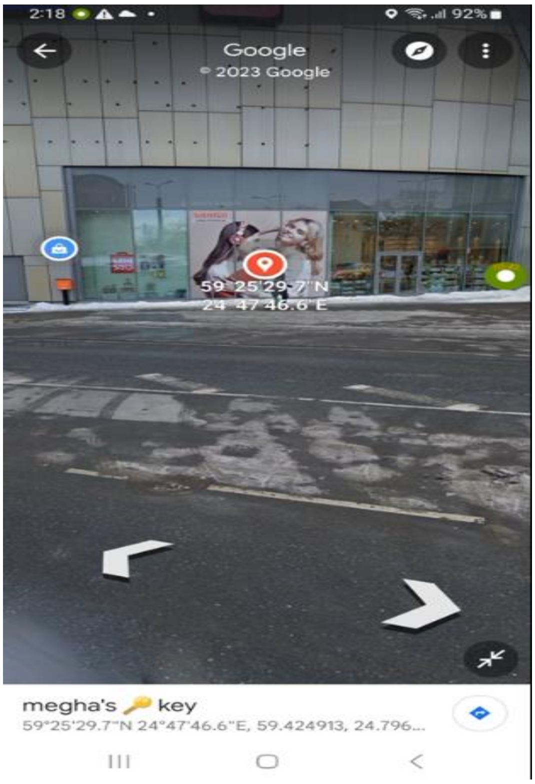
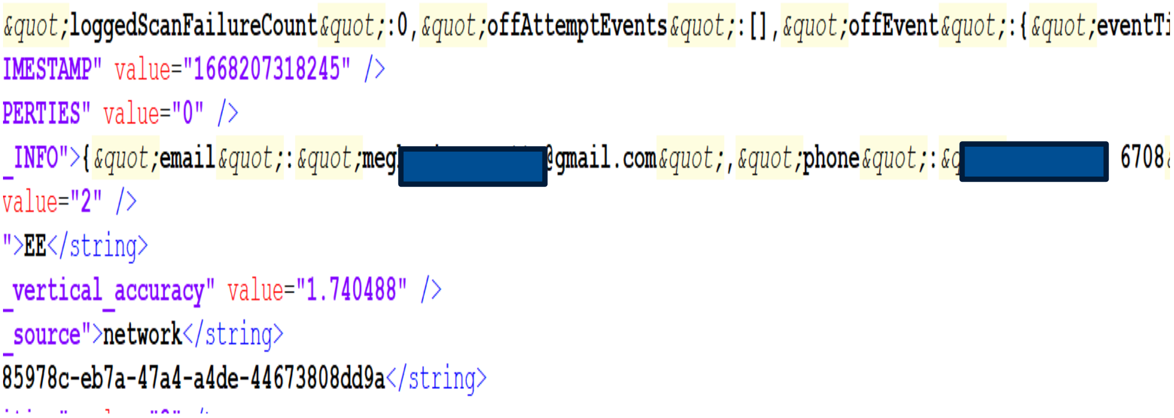


Fig 35. Last seen GPS location by HTC-Fetch tracker with streetview

4.2.7 Acquired Data Item from Tile Slim

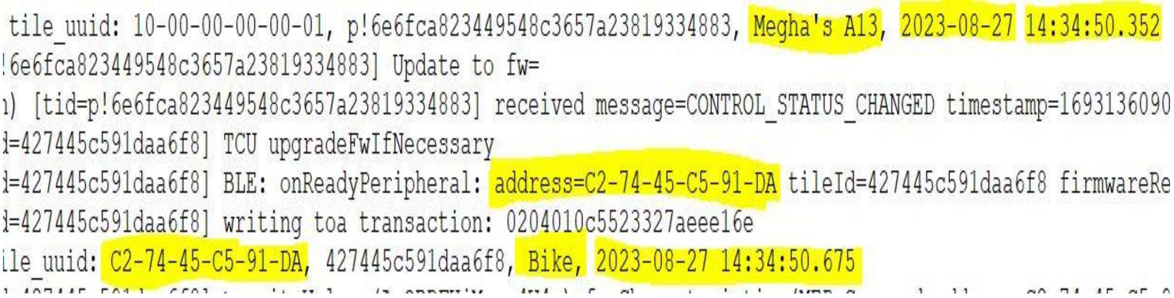
The Tile Slim application dashboard (Fig 12), during manual extraction, application shows the tag location on the map, and it is within 250 ft. range but not any specific physical address. The user name, cell phone number, email address, and country location are all stored by this application. Email addresses can be utilized to locate specific individuals or accounts that have engaged in suspicious activity, like stalking. Investigators can construct a timeline of activities and connect the participation to the suspects by tracking the email addresses connected to a specific device or account.



```
&quot;loggedScanFailureCount&quot;:0,&quot;offAttemptEvents&quot;:[],&quot;offEvent&quot;:{&quot;eventTi
IMESTAMP" value="1668207318245" />
PERTIES" value="0" />
_INFO">{&quot;email&quot;:&quot;meg[redacted]@gmail.com&quot;,&quot;phone&quot;:&q[redacted] 6708&
value="2" />
">EE</string>
_vertical_accuracy" value="1.740488" />
_source">network</string>
85978c-eb7a-47a4-a4de-44673808dd9a</string>
```

Fig 36. User email, phone number, country domain, and last tag seen time

We also found the MAC address of this tracker and defined the tag name with the starting time and date in the application's log file. All of these might serve to dispel any doubts about the proof gathered previously to support the case. Forensics experts may find all of this valuable information in their investigation to learn more about stalkers.



```
tile_uuid: 10-00-00-00-00-01, p!6e6fca823449548c3657a23819334883, Megha's A13, 2023-08-27 14:34:50.352
!6e6fca823449548c3657a23819334883] Update to fw=
1) [tid=p!6e6fca823449548c3657a23819334883] received message=CONTROL_STATUS_CHANGED timestamp=1693136090
d=427445c591daa6f8] TCU upgradeFwIfNecessary
d=427445c591daa6f8] BLE: onReadyPeripheral: address=C2-74-45-C5-91-DA tileId=427445c591daa6f8 firmwareRe
d=427445c591daa6f8] writing toa transaction: 0204010c5523327aeee16e
ile_uuid: C2-74-45-C5-91-DA, 427445c591daa6f8, Bike, 2023-08-27 14:34:50.675
```

Fig 37. MAC address of the tracker, tag name, phone model, app version and starting date/time

4.3 Data from Physical Extraction

Our study will not employ invasive techniques to get data from the physical acquisition phase. We have not used equipment meant for professionals, and the Android device we are looking at is still functional and has not damaged any hardware. So, we do not need to apply JTAG or Chip-off techniques. According to digital forensics researchers, if the situation is that the seized device is damaged or a non-invasive method cannot be used, an expert can apply the JTAG or Chip-off method. This approach takes more time and requires more specialized knowledge and tools. The dd tools may be used for physical acquisition [66]. The dd command is a device for transferring and converting data. Forensic investigators frequently use this tool to make a bit-by-bit replica of the storage system. We utilized this command-line tool on our Windows PC.

ADB tools are necessary to interface with Android devices[12] because we will use our Android phone with a Windows system. As our phone is already rooted, we can install BusyBox on Android phones. In the context of digital forensics, BusyBox helps acquire data from the Android device during the Android device forensics investigation. Netcat can read/write data across the network connection. To send data from the Android system to the Windows system, we connect our phone by USB-C cable, and the TCP protocol will be engaged with a customized port. Netcat will help us in this scenario. First, we run "adb devices" with confirmation from the Android device to allow the connection, and it confirms that our device is connected to the Windows system. Then we ran "adb shell" and changed the user to root permission using the "su" command. The "cat /proc/partitions" command gives us the whole partition layout, and we try to copy the mmcblk0 block using the following command in the first terminal, and it will work as a source.

Step 1: On the Android side (ADB shell*)*

```
a13ve:/ # dd if=/dev/block/mmcblk0 | busybox nc –l –p 9999
```

[For the Android side where l (listening) p (port) on 9999 port number]

The second terminal will work for the Windows system as a destination. Here, we apply the command to receive our data block.

```
# adb forward tcp:9999 tcp:9999
```

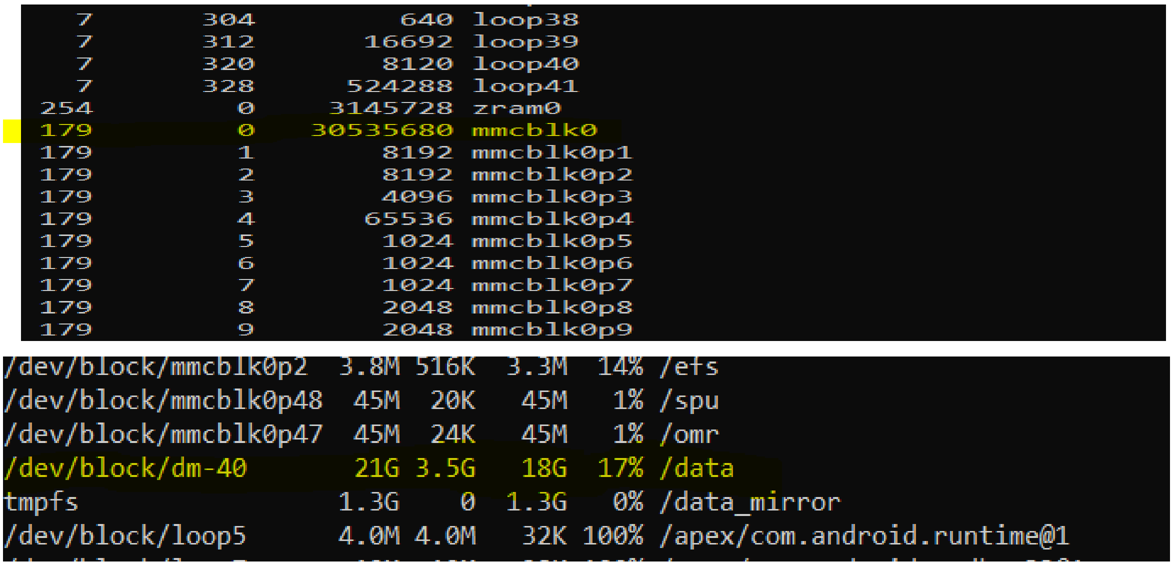
[All the data from adb will forwarded to port 9999 using tcp protocol]

Step 2: On the Windows station (Windows CMD prompt)

```
# nc 127.0.0.1 9999 > /home/squid/Desktop/ SAM-Physical-dd/mmcblk0.dd
```

[netcat listen 9999 port on localhost and save mmcblk0.dd file on the user-directed path]

[12] https://developer.android.com/tools/releases/platform-tools



```
   7        304         640 loop38
   7        312       16692 loop39
   7        320        8120 loop40
   7        328      524288 loop41
 254          0     3145728 zram0
 179          0    30535680 mmcblk0
 179          1        8192 mmcblk0p1
 179          2        8192 mmcblk0p2
 179          3        4096 mmcblk0p3
 179          4       65536 mmcblk0p4
 179          5        1024 mmcblk0p5
 179          6        1024 mmcblk0p6
 179          7        1024 mmcblk0p7
 179          8        2048 mmcblk0p8
 179          9        2048 mmcblk0p9
```

```
/dev/block/mmcblk0p2  3.8M 516K  3.3M  14% /efs
/dev/block/mmcblk0p48  45M  20K   45M   1% /spu
/dev/block/mmcblk0p47  45M  24K   45M   1% /omr
/dev/block/dm-40       21G 3.5G   18G  17% /data
tmpfs                 1.3G    0  1.3G   0% /data_mirror
/dev/block/loop5      4.0M 4.0M   32K 100% /apex/com.android.runtime@1
```

Fig 38. mmcblk0 (whole block device) and dm-40(only user data partition)

Step 1: On the Android side (ADB shell)

a13ve:/ # dd if=/dev/block/dm-40 | busybox nc –l –p 9999

# adb forward tcp:9999 tcp:9999

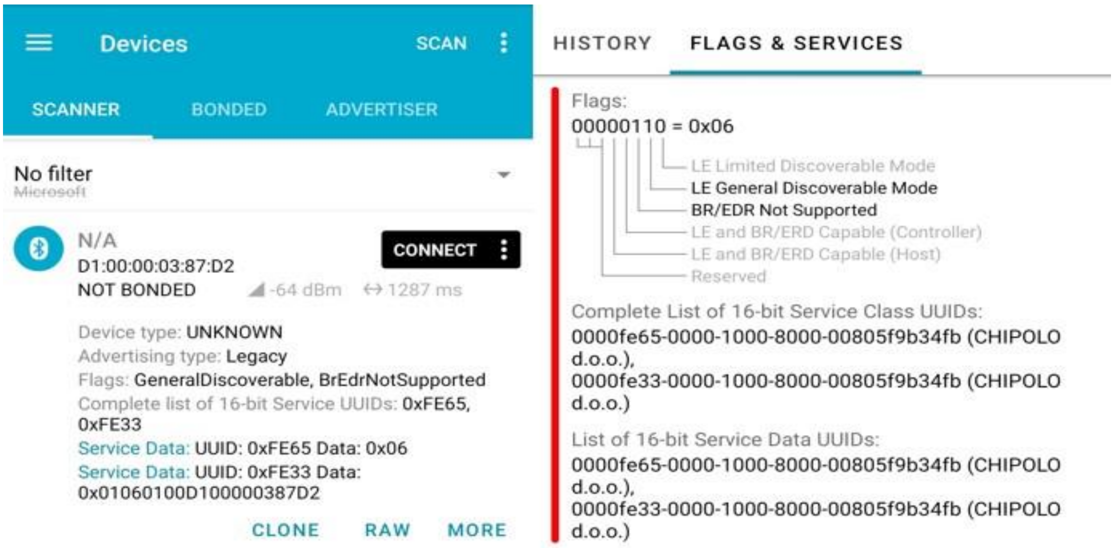
Step 2: On the Windows station (Windows CMD prompt)

# nc 127.0.0.1 9999 > /home/squid/Desktop/ SAM-Physical-dd/dm-40.dd

Android devices and file systems differ depending on the vendor and their products. In our case, we copied mmcblk0, which is known as the whole Android storage block, and the dm-40 block. For our Android, user data partition "/data "resides in the dm-40 block (fig. 38). To preserve the integrity of our acquired dd image, we generate Md5 and SHA1 hash values for our forensic image on the Windows machine using HashCalc tools. We then use Sleuthkit Autopsy, a well-known open-source program, on our Windows computer. The first thing we do is add our disk image (mmcblk0.dd image) as a data source and set up the ingest modules using "file type identification, keyword search, email parser, Encryption detection, Interesting Identifier, Android Analyzer (aLEAPP), GPX parser, Android Analyzer." Although the size of our obtained mmcblk0.dd picture was 29.1 GB, we were unable to recover any data related to our Bluetooth application. After that, we proceeded with our second data source, dm-40.dd image, which is 21.3 GB, was added to the Autopsy, and the same ingest module was applied to analyze it. Reiber Lee states [69] that default encryption is applied from Android 6.0 to upwards. Our Samsung Galaxy A13 runs on Android 13, which has the most updated OS and is, by default, encrypted.

4.4 Data from Tracker

Johannes K Becker et al., in their research [70], mentioned that Bluetooth Low Energy devices are advertising their presence. It was addressed in the Bluetooth core specification 4.0, and three of our trackers have Bluetooth 4.0 core specifications (Table 5), and one of them has a 5.0 specification. Open source Bluetooth Low Energy tools (Table 4), we installed Nordic Semiconductor's nRF[13] connect Android application on another Android device, Samsung A53, to scan and monitor our target Bluetooth tracker to capture MAC address.



N/A
D1:00:00:03:87:D2
NOT BONDED -64 dBm ↔ 1287 ms

Device type: UNKNOWN
Advertising type: Legacy
Flags: GeneralDiscoverable, BrEdrNotSupported
Complete list of 16-bit Service UUIDs: 0xFE65, 0xFE33
Service Data: UUID: 0xFE65 Data: 0x06
Service Data: UUID: 0xFE33 Data: 0x01060100D100000387D2

Flags:
00000110 = 0x06
LE Limited Discoverable Mode
LE General Discoverable Mode
BR/EDR Not Supported
LE and BR/ERD Capable (Controller)
LE and BR/ERD Capable (Host)
Reserved

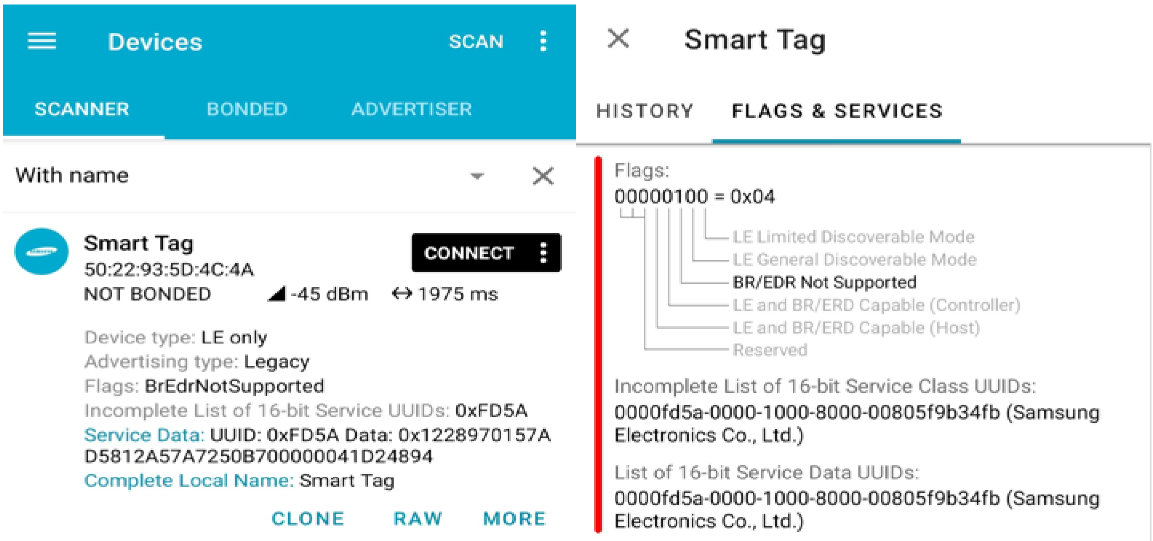
Complete List of 16-bit Service Class UUIDs:
0000fe65-0000-1000-8000-00805f9b34fb (CHIPOLO d.o.o.),
0000fe33-0000-1000-8000-00805f9b34fb (CHIPOLO d.o.o.)

List of 16-bit Service Data UUIDs:
0000fe65-0000-1000-8000-00805f9b34fb (CHIPOLO d.o.o.),
0000fe33-0000-1000-8000-00805f9b34fb (CHIPOLO d.o.o.)

Fig 39. nRF scanning result of Chipolo Classic

nRF connect successfully captured the MAC address of Chipolo Classic. It is also showing us its vendor name by 16-bit service UUID. By Bluetooth SIG[14] proprietary database, we can locate the vendor identification.

[13] https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-mobile
[14] https://studylib.net/doc/25935852/16-bit-uuid-numbers-document



Smart Tag
50:22:93:5D:4C:4A
NOT BONDED -45 dBm ↔ 1975 ms

Device type: LE only
Advertising type: Legacy
Flags: BrEdrNotSupported
Incomplete List of 16-bit Service UUIDs: 0xFD5A
Service Data: UUID: 0xFD5A Data: 0x1228970157A D5812A57A7250B700000041D24894
Complete Local Name: Smart Tag

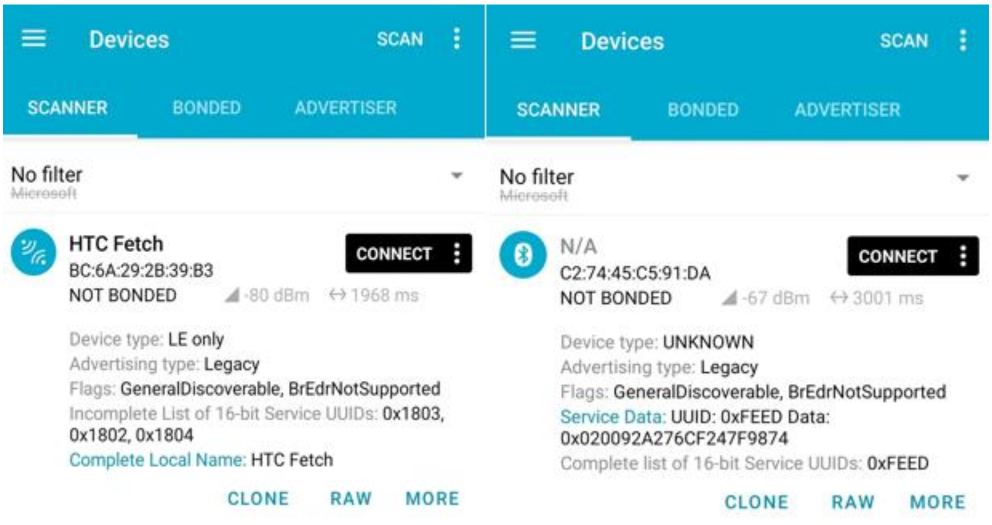
Flags:
00000100 = 0x04
LE Limited Discoverable Mode
LE General Discoverable Mode
BR/EDR Not Supported
LE and BR/ERD Capable (Controller)
LE and BR/ERD Capable (Host)
Reserved

Incomplete List of 16-bit Service Class UUIDs:
0000fd5a-0000-1000-8000-00805f9b34fb (Samsung Electronics Co., Ltd.)

List of 16-bit Service Data UUIDs:
0000fd5a-0000-1000-8000-00805f9b34fb (Samsung Electronics Co., Ltd.)

Fig 40. nRF scanning result of Galaxy smart tag plus

When we scan Galaxy Smart Tag Plus, it shows its MAC with the name and vendor name available in the 16-bit UUID and Bluetooth SIG database. Galaxy tag uses Bluetooth 5.0 architecture and a random MAC address mechanism, so we captured a MAC address when the device was seized. The following MAC addresses are captured from HTC-Fetch and Tile Slim. They have static MAC addresses as they have Bluetooth 4.0 specifications. The MAC address was captured and matched what we discovered on a rooted phone during logical extraction.



HTC Fetch
BC:6A:29:2B:39:B3
NOT BONDED -80 dBm ↔ 1968 ms

Device type: LE only
Advertising type: Legacy
Flags: GeneralDiscoverable, BrEdrNotSupported
Incomplete List of 16-bit Service UUIDs: 0x1803, 0x1802, 0x1804
Complete Local Name: HTC Fetch

N/A
C2:74:45:C5:91:DA
NOT BONDED -67 dBm ↔ 3001 ms

Device type: UNKNOWN
Advertising type: Legacy
Flags: GeneralDiscoverable, BrEdrNotSupported
Service Data: UUID: 0xFEED Data: 0x020092A276CF247F9874
Complete list of 16-bit Service UUIDs: 0xFEED

Fig 41. nRF scanning result of HTC-Fetch and Tile Slim

4.5 Summary

We have collected as much user-related data as possible from the application dashboard for each Bluetooth tracker during the manual extraction process. Each Bluetooth tracker was actively connected during manual acquisition and showed up on the map. After that, we proceeded with the logical extraction method with the non-rooted phone, and we were not successful in retrieving anything. We took a full backup of our Android device after that. We hashed our full backup image to preserve integrity and then got administrative access by rooting the phone then restored our full backup data to the phone. Using the device explorer in Android Studio, we found plenty of forensic artifacts related to the user of the Bluetooth tracker application. We proceed with the physical data extraction process to get more data from the raw image of the Android storage system. We hashed our Android DD image file to ensure integrity. We use Autopsy forensic toolkits to analyze the DD image, and we faced encryption here. As our device is running on Android 13, it is by default encrypted, and we received nothing here. From the tracker side, we retrieve the MAC address only, as this kind of Bluetooth tracker does not have storage to store user data but broadcasts the MAC address only.

5 Discussion and Analysis

This chapter cover our research findings during each acquisition phase of our study. We summarize the data by organizing it into applications and the acquisition techniques used to get this item. In mobile forensics, it is expected that the evidence from the Android device can be found in installed third-party application files that can be accessed using standard forensics manual and logical extraction methods. These applications typically have relatively moderate-level data collection processes and analysis functions. The security features of the Android system are developing day by day. Security measures make it necessary to utilize expert tools to collect forensic artifacts using logical and physical extraction methods. Our study's most successful data acquisition phase was logical extraction from the rooted phone and matched with the data we acquired during the manual extraction phase.

In our study, we used the most recent version of the Android 13 operating system, and neither logical acquisition from non-rooted mobile devices nor physical acquisition from rooted devices revealed any artifacts. However, when data recovery is required, physical extraction is advised by the expert. Commercial tools can be used for artifact collection, and the result can vary. Otherwise, using an open-source tool like Device Explorer, someone with intermediate technical skills can reveal important artifacts in stalking-related cases involving Bluetooth applications. We emphasize the triage process to help investigators understand what artifacts may be retrieved from Bluetooth applications and how that data might be recovered. During manual extraction, we observed that all our Bluetooth beacon devices were actively shown on the map of each application dashboard. Sometimes, with GPS coordinates and physical location address. It is essential to notice that the location setting is enabled and the Bluetooth connection is active on the mother device. This observation will help to understand the investigator when analyzing manual extraction from the Bluetooth application dashboard during the triage procedure.

Bluetooth tracking devices broadcast their updated location on the application dashboard. This broadcast procedure depends on a specific time interval. Different beacons have different time interval sets to update their location on the map. We also observe the slight latency in updating location with time on each application. This latency is also found in the log file during logical extraction using an Android studio device explorer on our rooted phone. During the data generation for our experimental process, the tag connected time was not updated automatically from the starting point to the destination point. In digital forensic analysis, date and time are crucial artifacts. It allows investigators to correlate event and incident timeline construction. On the HTC-Fetch dashboard, during the manual extraction process, we found that the user can update the connection time using disconnect and then connect or wait for the default update period. Though we observed that the display time was not updated on the HTC-Fetch dashboard, we found the tag's updated location on the map with street view. Street view can help investigators verify the physical location associated with GPS data collected from Bluetooth tracker applications. This situation can provide a visual context to the digital evidence and allow investigators to understand the environment in which stalking took place.

We retrieved the last tag connected time from the XML file of the HTC-Fetch application folder on the rooted phone. Chipolo Classic, Galaxy Smart Tag Plus, and Tile Slim, these Bluetooth applications, show that tags are connected, and they update their location and prompt on the map for user reference. This application cannot disconnect the Bluetooth devices except if the user does not remove or stop the application manually. If the Bluetooth tracking device is involved in the stacking incident, map or GPS information can be correlated with data and time to determine the movement of the device's owner's intention. We observed that the date and time artifacts are easily acquirable during manual extraction from the application dashboard. By analyzing this data, investigators can identify the locations where the devices were detected and when they were active. Investigators can create a timeline of the events and map out the movement of the devices and the stalker, and this helps to establish patterns of behavior and identify potential suspects.

The investigator should think about performing manual extraction as the first stage when they arrive at a crime scene or seize the equipment utilized in stalking incidents in order to collect some essential information immediately. This procedure might help them decide on the spot if they need to take the phone and Bluetooth tag to the lab for a closer examination in search of more evidence. This triage at the event scene might yield first leads or evidence that could significantly improve the case, as we have seen from the manual extraction process that the information saved by each Bluetooth application was different. This step also helps reduce backlogs at the laboratory. The timestamp we retrieved from the XML file for every Bluetooth tracker was in UNIX epoch time format. We use an online epoch converter[15] to make a human readable format.

[15] https://www.epochconverter.com/

Table 9. Summary of artifacts from Bluetooth application on each acquisition phase

Extration phase	**Application**	**Location**	**Artifacts**
Manual extraction	Chipolo Classic	Application dashboard	• start time • location on the map • tag item name • user name and icon • tag item • tag color • ring tone • enabled location • tag near location • saved location address • tag serial number • user email address • tag version • active tag on the map • tag app version • last seen time and GPS coordinate.
	Galaxy Smart Tag Plus	Application dashboard	• start time • tag item • current location address • tag icon • active tag on the map • user email address • last tag connected time • last tag location address • app version • tag connected with mother device
	HTC-Fetch	Application dashboard	• start time • current GPS position • streetview • tag icon • tag name • date and time • plus code for map location • last updated GPS coordinate • active tag on the map • app version
	Tile Slim	Application dashboard	• start time • tag connected with mother device • active tag on the map • tag icon • tag type • ring tone

			• updated time and date • tag locaton address • user name • user full name • user email address • user phone number • tag serial number and version • tag fcc-id
Logical extraction (non rooted device)	Chipolo Classic	backup image	nothing recovered
	Galaxy Smart Tag Plus	backup image	nothing recovered
	HTC-Fetch	backup image	nothing recovered
	Tile Slim	backup image	nothing recovered
Logical extraction (rooted device)	Chipolo Classic	/data/data/chipolo.net.v3/cache/log/chipolo_0.log	mother device information
		/data/data/chipolo.net.v3/shared_prefs/FirebaseHeartBeatW0RFRkFVTFRd+MTo4Mzk0NzExMzkyOTphbmRyb2lkOjZlOWYwYThiNDRmZDBlN2Q.xml	last tag seen date
		/data/data/chipolo.net.v3/files/default.realm	MAC address, GPS location, Start time, last seen time, user name, user email address, user saved location address , user ID, tag color,

			tag icon
		/data/data/chipolo.net.v3/shared_prefs/com.google.android.gms.signin.xml	use given name, family name, display name
		/data/data/chipolo.net.v3/shared_prefs/com.google.maps.api.android.lib6.drd.PREFERENCES_FILE.xml	tag located caountry
	Galaxy Smart Tag Pro	/data/data/com.samsung.android.oneconnect/shared_prefs/FME_SELECTED_DEVICE.xm	user nick name, device IMEI, tag type, last GPS position
		/data/data/com.samsung.android.oneconnect/shared_prefs/UserCache.xml	user full name, user email, user country code
		/data/data/com.samsung.android.oneconnect/shared_prefs/MovablePluginData.b5ac6b74-14ed-4565-9146-b968006604e1.xml	user ring tone, firmware version
		/data/data/com.samsung.android.oneconnect/shared_prefs/FirebaseHeartBeatW0RFRkFVTFRd+MTozNDI2NjgwNjg3MjQ6YW5kcm9pZDo4OTYwOWM5MjVhOWUwZDc2.xml	last tag seen date
		/data/data/com.samsung.android.oneconnect/shared_prefs/FME_LOCATION_GETTING_TIME.xml	last tag connected time

		/data/data/com.samsung.android.oneconnect/shared_prefs/ LaunchDarkly_TqmAUzY4nh7nFIXxQjfIVFbB2Il00Ze NrRZaprYOPqs=.xml	last successful connection
	HTC-Fetch	/data/data/com.htc.accessory.fetch/shared_prefs/HTCFETCHPREFS.xml	MAC address, tag name, last tag seen time, last tag seen date
		/data/data/com.htc.accessory.fetch/shared_prefs/com.google.android.gms.analytics.prefs.xml	tag start time
		/data/data/com.htc.accessory.fetch/shared_prefs/com.google.maps.api.android.lib6.drd.PREFERENCES_FILE.xml	tag located country
Physical extraction	Chipolo Classic	backup image	nothing recovered
	Galaxy Smart Tag Plus	backup image	nothing recovered
	HTC-Fetch	backup image	nothing recovered
	Tile Slim	backup image	nothing recovered

Bluetooth low energy tracker devices usually do not receive data but continuously broadcast their presence. These Bluetooth devices do not require an internet connection to communicate with their paired primary device. Chipolo Classic, HTC-Fetch, and Tile Slim (Table 1) have 4.0 Bluetooth, which is regularly broadcast in the precise format and has a static MAC address. Galaxy Smart Tag Plus uses 5.0 (Table 1) Bluetooth specification, which uses a random MAC address mechanism. The network-capable equipment is given a specific identification number called a MAC address. By obtaining the MAC address of the BLE tracking device, investigators can identify the specific device used by the stalker, which can help to trace its primary device and its owner. We retrieve the MAC address of each tracker device during the logical acquisition from the Bluetooth application folder on the rooted phone. Forensics investigators can collect the Bluetooth tags' MAC address during the triage acquisition phase by scanning these tags. During the MAC address capture of one of our Bluetooth tags, Galaxy Smart Tag Plus, we noticed that the device uses a random MAC address to anonymize itself. MAC often works as a unique identifier in the forensics investigation. If any device with a random MAC address mechanism is seized for forensic investigation, the examiner should pay extra attention to it. The examiner should capture the MAC address within the period of the triage acquisition phase.

If the device has a random MAC address, then the application of that Bluetooth device will not store the MAC address inside the log file or encrypt the MAC address. We did not find the MAC address from the Galaxy Smart Tag Plus application directory during our logical acquisition. Forensic examiners can pay attention to those Bluetooth trackers using a randomized MAC address mechanism.

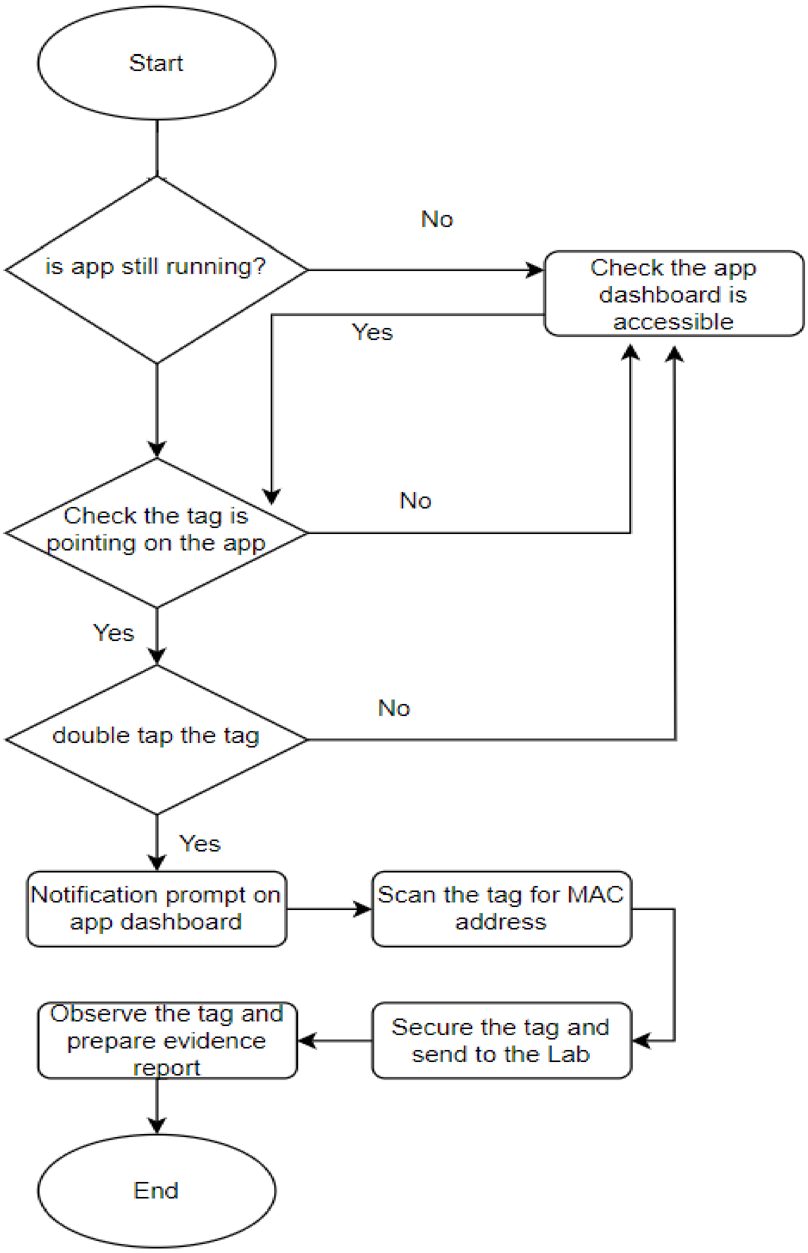

Start
is app still running?
No
Check the app dashboard is accessible
Yes
Check the tag is pointing on the app
No
Yes
double tap the tag
No
Yes
Notification prompt on app dashboard
Scan the tag for MAC address
Observe the tag and prepare evidence report
Secure the tag and send to the Lab
End

Fig 39. The MAC address collection process during triage stage

During the triage acquisition phase, an investigator can investigate inside tag hardware (Figure 3, 5, 9, 11). Analysts can collect FCC-ID from the hardware part. FCC-ID can help the forensics expert uniquely identify the vendor of the tag manufactured and relate to the logs installed by Bluetooth applications. During the data generation stage, our Bluetooth tracker updates their location address with latency, specifically at the end location. The dashboard time and the time we get from the logs file slightly lagged in seconds. So, when the forensics examiner seized the smartphone involved in the stalking situation, he should pay attention to the device seized time and the tracker's last connected time. So, the examiner can draw a timeline between the connected tracker with the installed Bluetooth application and the stalker's movement. Chipolo Classic, Galaxy Smart Tag Plus, and Tile Slim require a user profile; hence, the user must use his name, email address, and cell phone number. All of this information is available for manual acquisition from the application dashboard. We could find this information from the rooted device in the logical extraction phase (Table 9). These artifacts can act as unique identifiers for individuals and can be found in other communication history. Examiners can pay attention to identifying and analyzing stalker behavior and communication patterns. HTC-Fetch, one of our Bluetooth trackers, can be used without user information, such as a user name, email address, or phone number. This tracker works as power, pair, and locate. So, we expected that we might have fewer artifacts in every acquisition phase. We observed the HTC-Fetch application dashboard during manual acquisition, which provided us with more interactive artifacts on the dashboard. The triage phase helps the examiner to obtain these exciting data. HTC-Fetch additionally, we encounter fewer artifacts when logically acquiring data from rooted smartphones. Our observation during the analysis of the data that we retrieved from the logical extraction phase from the rooted device is that Chipolo Classic, Galaxy Smart Tag Plus, and Tile Slim applications save more information regarding their mother device (Table 9). These data can help investigators understand the person's involvement in a stalking case. We are trying to draw a possible stalking scenario and data collection guidelines (Figure 40) for the forensic investigator. We prepare a stalking scenario to explain the steps as an example. During the data generation with the Bluetooth tracker, Mr. X is followed by Mr Y. The investigator arrived at the scene and seized the device when the Bluetooth tracker was actively connected to the mother device and was showing on the application dashboard.

Step 1: The investigator would initially examine the Bluetooth traffic between the connected device and the tracker (MAC address collection).

Step 2: The data on the dashboard that was attached to the Bluetooth tracker would subsequently be analyzed by the investigator (manual acquisition).

Step 3: The investigator will also examine internet communication using the Bluetooth tracker. If the tracker has a cloud profile, this might show further information on stalker and tracker usage.

Step 4: After gathering sufficient information, the investigator would begin the data preservation process, secure the device, and transport it to the forensic lab.

Step 5: If the investigator needs further information on the Bluetooth tracker application and the stalker, the logical acquisition is followed by, if necessary, physical acquisition.

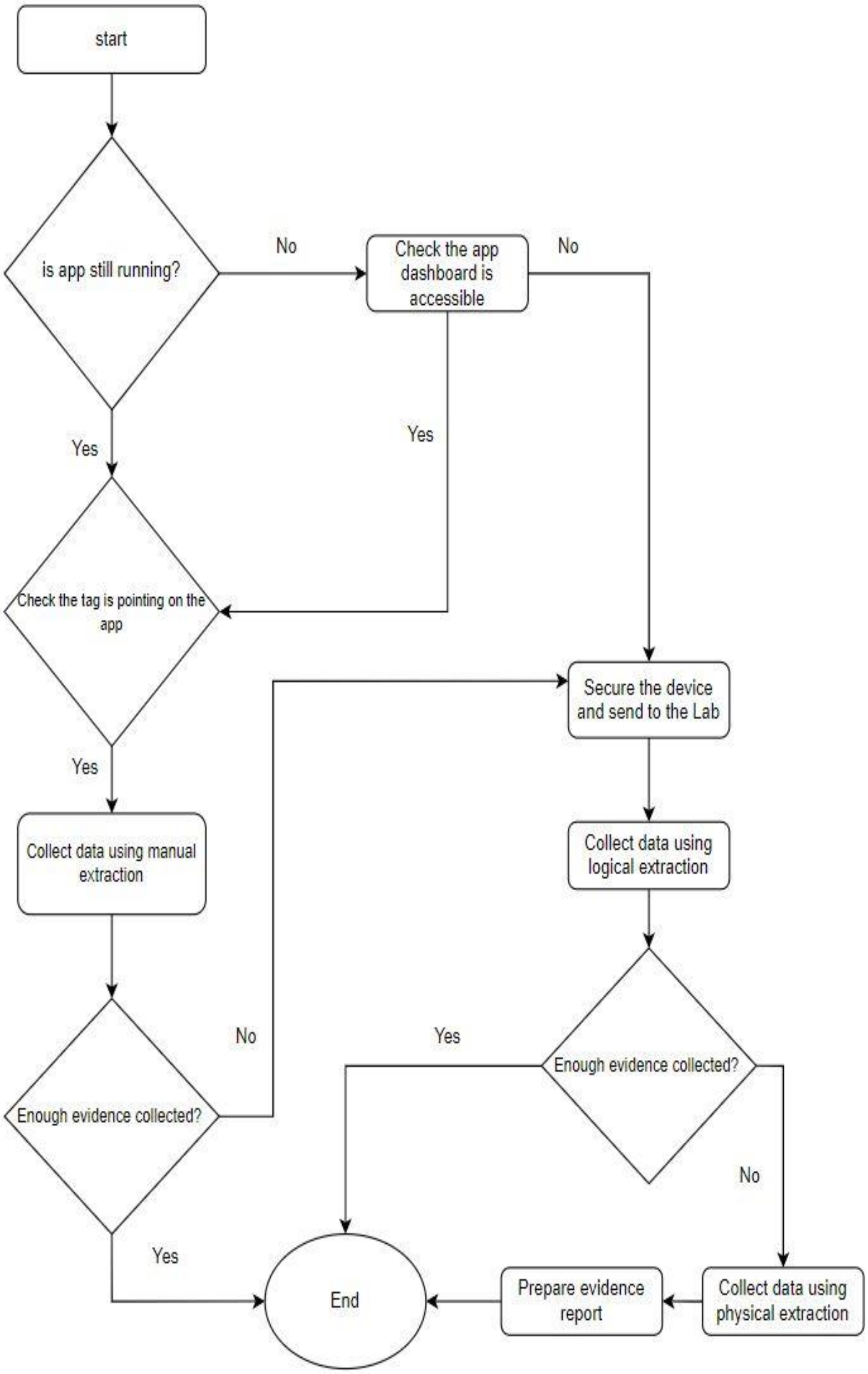

start
is app still running?
No
Check the app dashboard is accessible
No
Yes
Yes
Check the tag is pointing on the app
Secure the device and send to the Lab
Yes
Collect data using manual extraction
Collect data using logical extraction
No
Yes
Enough evidence collected?
Enough evidence collected?
No
Yes
End
Prepare evidence report
Collect data using physical extraction

Fig 40. Possible data collection procedure for stalking case

5.1 Additional Discussion

The Android 13 operating system we employed in our investigation, we searched for and could not locate any reliable open-source decryption tools for the Samsung Galaxy A13. In the field of digital forensics, regarding collecting and analyzing artifacts from Android devices with the help of the decryption technique, the researcher used the TWRP tool to decrypt the device for further analysis [47][17][71]. We do not find any TWRP[16] decryption tool version that supports our Samsung A13 device. The required tools are available for forensic analysts; if they want to use open-source tools, then observing the market about required source tools is better. Observing all the data acquisition phase stages, we understand that the triage acquisition phase significantly impacts the stalking scenario. At this stage, the analyst can see that the Bluetooth tracker is still connected with the application and showing actively on the dashboard. When forensic experts seize the device and the tracker, they can even confirm the first step in the triage stage by tapping on the tag and the application dashboard. So, both devices notify each other that they are connected.

5.2 Limitation

We restricted our research to four particular Bluetooth tracking devices based on Android and the applications used by their users. Bluetooth trackers are designed from the ground up to keep location data private and secure. No location data or user data is physically stored inside the tracker [25] [26]. These kinds of tracker devices do not receive and send any extra data. This Bluetooth tracker only broadcasts a unique identifier that can detected by nearby devices. So, having more data except the MAC address was only possible for our study. Lastly, the mobile forensic tools and handsets employed in this experiment were chosen based on the authors' availability.

[16] https://twrp.me/Devices/Samsung/

5.3 Future Work

These Bluetooth tracker devices are improving constantly, adding new features, functionality, and security. Android system is also developing its security features day by day. On the other hand, reverse engineering is also enriching and become a crucial aspect of digital forensic science. Forensic examiners can retrieve encryption keys and decrypt the system using reverse engineering techniques by understanding a device's internal security structure and operation. Forensic examiners can acquire forensic artifacts from the seized device when they retrieve the decryption key. We plan to apply reverse engineering-based acquisition from the installed Bluetooth tracker application data from encrypted Android and iOS devices.

5.4 Conclusion

In conclusion, our study on the Bluetooth tracker application has demonstrated the significant role of digital forensics in the context of cyberstalking. We obtained data through manual and logical acquisition methods, focusing on crucial artifacts such as the MAC address, date, time, and GPS location. We successfully gathered these artifacts from the phone involved in the stalking scenario. We also highlighted the critical points for the triage phase, which can significantly reduce the workload for investigators in stalking cases related to Bluetooth trackers.

Although we noticed slight variations in the time data, forensic examiners can accept these as the latency of the device's seizure time. The consequences of these discoveries are significant, extending beyond the realm of digital forensics into areas such as cybersecurity, law enforcement, and legal jurisprudence. They underscore the importance of ethical considerations and legal boundaries in using such technologies. Future research could focus on refining the process of artifact collection, improving the interpretive algorithms, and developing robust measures to protect the privacy and security of individuals.

As technology continuously evolves, so will the strategies and tools used in digital forensics and the fight against cyberstalking. In closing, exploring digital artifacts from Android phones' Bluetooth tracker applications has provided a new lens to view and combat cyberstalking. It is a testament to the power of technology in shaping our understanding of complex social issues and enhancing our ability to address them.

References

[1] “Smart Tracker - Global Market and Forecast Till 2030.” Accessed: Nov. 04, 2023. [Online]. Available: https://www.acumenresearchandconsulting.com/smart-tracker-market

[2] “Police reports suggest a larger pattern of AirTag stalking,” Engadget. Accessed: Nov. 04, 2023. [Online]. Available: https://www.engadget.com/apple-airtag-stalking-police-reports-190022315.html

[3] Y. Zhang, J. Weng, R. Dey, and X. Fu, “Bluetooth Low Energy (BLE) Security and Privacy,” in *Encyclopedia of Wireless Networks*, X. (Sherman) Shen, X. Lin, and K. Zhang, Eds., Cham: Springer International Publishing, 2019, pp. 1–12. doi: 10.1007/978-3-319-32903-1_298-1.

[4] B. Hitchcock, N.-A. Le-Khac, and M. Scanlon, “Tiered forensic methodology model for Digital Field Triage by non-digital evidence specialists,” *Digit. Investig.*, vol. 16, pp. S75–S85, Mar. 2016, doi: 10.1016/j.diin.2016.01.010.

[5] E. Casey, M. Ferraro, and L. Nguyen, “Investigation Delayed Is Justice Denied: Proposals for Expediting Forensic Examinations of Digital Evidence*,” *J. Forensic Sci.*, vol. 54, no. 6, pp. 1353–1364, 2009, doi: 10.1111/j.1556-4029.2009.01150.x.

[6] D. Wilson-Kovacs, “Effective resource management in digital forensics: An exploratory analysis of triage practices in four English constabularies,” *Polic. Int. J.*, vol. 43, no. 1, pp. 77–90, Jan. 2019, doi: 10.1108/PIJPSM-07-2019-0126.

[7] A. Al-Dhaqm, S. A. Razak, R. A. Ikuesan, V. R. Kebande, and K. Siddique, “A Review of Mobile Forensic Investigation Process Models,” *IEEE Access*, vol. 8, pp. 173359–173375, 2020, doi: 10.1109/ACCESS.2020.3014615.

[8] P. Krishna Reddy, A. Sureka, S. Chakravarthy, and S. Bhalla, *Big Data Analytics: 5th International Conference, BDA 2017, Hyderabad, India, December 12-15, 2017, Proceedings*. 2017. doi: 10.1007/978-3-319-72413-3.

[9] S. A. Butt, T. Jamal, M. A. Azad, A. Ali, and N. S. Safa, “A multivariant secure framework for smart mobile health application,” *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 8, p. e3684, 2022, doi: 10.1002/ett.3684.

[10] M. Levitt, “AirTags are being used to track people and cars. Here’s what is being done about it,” *NPR*, Feb. 18, 2022. Accessed: Nov. 04, 2023. [Online]. Available: https://www.npr.org/2022/02/18/1080944193/apple-airtags-theft-stalking-privacy-tech

[11] A. K. Agrawal, A. Sharma, S. R. Sinha, and P. Khatri, “Forensic of an unrooted mobile device,” *Int. J. Electron. Secur. Digit. Forensics*, vol. 12, no. 1, pp. 118–137, Jan. 2020, doi: 10.1504/ijesdf.2020.103882.

[12] N. A. Aziz, F. Mokhti, and M. N. M. Nozri, “Mobile Device Forensics: Extracting and Analysing Data from an Android-Based Smartphone,” in *2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec)*, Oct. 2015, pp. 123–128. doi: 10.1109/CyberSec.2015.32.

[13] C. Anglano, M. Canonico, and M. Guazzone, “Forensic analysis of Telegram Messenger on Android smartphones,” *Digit. Investig.*, vol. 23, pp. 31–49, Dec. 2017, doi: 10.1016/j.diin.2017.09.002.

[14] F. A. Awan, “Forensic examination of social networking applications on smartphones,” in *2015 Conference on Information Assurance and Cyber Security (CIACS)*, Dec. 2015, pp. 36–43. doi: 10.1109/CIACS.2015.7395564.

[15] D. Jacobs, K.-K. R. Choo, M.-T. Kechadi, and N.-A. Le-Khac, “Volkswagen Car Entertainment System Forensics,” in *2017 IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 699–705. doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.302.

[16] A. Almogbil, A. Alghofaili, C. Deane, T. Leschke, A. Almogbil, and A. Alghofaili, “Digital Forensic Analysis of Fitbit Wearable Technology: An Investigator’s Guide,” in *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, Aug. 2020, pp. 44–49. doi: 10.1109/CSCloud-EdgeCom49738.2020.00017.

[17] S. Ebbers, F. Ising, C. Saatjohann, and S. Schinzel, “Grand Theft App: Digital Forensics of Vehicle Assistant Apps,” in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, in ARES ’21. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 1–6. doi: 10.1145/3465481.3465754.

[18] R. Beverly, S. Garfinkel, and G. Cardwell, “Forensic carving of network packets and associated data structures,” *Digit. Investig.*, vol. 8, pp. S78–S89, Aug. 2011, doi: 10.1016/j.diin.2011.05.010.

[19] N. S. Grantham *et al.*, “Global Forensic Geolocation with Deep Neural Networks,” *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 69, no. 4, pp. 909–929, Aug. 2020, doi: 10.1111/rssc.12427.

[20] A. Irons, “Measurement Issues with Geo-Positional Forensics,” *Meas. Control*, vol. 45, no. 10, pp. 311–314, Dec. 2012, doi: 10.1177/002029401204501004.

[21] P. Domingues, M. Frade, and J. M. Parreira, “Filtering Email Addresses, Credit Card Numbers and Searching for Bitcoin Artifacts with the Autopsy Digital Forensics Software,” in *Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018)*, A. M. Madureira, A. Abraham, N. Gandhi, C. Silva, and M. Antunes, Eds., in Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, pp. 318–328. doi: 10.1007/978-3-030-17065-3_32.

[22] “Stalking Concerns Raised by Bluetooth Tracking Technologies: In Brief.” Accessed: Nov. 04, 2023. [Online]. Available: https://crsreports.congress.gov/product/pdf/R/R47035

[23] V. Gazeau and Q. Liu, “Catch Me if You Can: Analyzing Geolocation Artifacts Left by the Tile Application on iPhones,” 2020.

[24] J. Briggs and C. Geeng, “BLE-Doubt: Smartphone-Based Detection of Malicious Bluetooth Trackers,” in *2022 IEEE Security and Privacy Workshops (SPW)*, May 2022, pp. 208–214. doi: 10.1109/SPW54247.2022.9833870.

[25] “Introducing New Samsung Galaxy SmartTag2: A Smart Way to Keep Track of Important Things in Your Life,” Samsung US Newsroom. Accessed: Nov. 04, 2023. [Online]. Available: https://news.samsung.com/us/introducing-new-samsung-galaxy-smarttag2-a-smart-way-to-keep-track-of-important-things-in-your-life/

[26] “AirTag | Precision Tracking, Anti-Stalking, Accessories,” AppleInsider. Accessed: Nov. 04, 2023. [Online]. Available: https://appleinsider.com/inside/airtags

[27] Inpixon, “Bluetooth RTLS: BLE Location Tracking & Positioning | Inpixon.” Accessed: Nov. 04, 2023. [Online]. Available: https://www.inpixon.com/technology/standards/bluetooth-low-energy

[28] H. Nyholm *et al.*, “The Evolution of Volatile Memory Forensics,” *J. Cybersecurity Priv.*, vol. 2, no. 3, Art. no. 3, Sep. 2022, doi: 10.3390/jcp2030028.

[29] P. Olivo and E. Zanoni, “Flash Memories: An Overview,” in *Flash Memories*, P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, Eds., Boston, MA: Springer US, 1999, pp. 1–35. doi: 10.1007/978-1-4615-5015-0_1.

[30] V. S. Harichandran, D. Walnycky, I. Baggili, and F. Breitinger, “CuFA: A more formal definition for digital forensic artifacts,” *Digit. Investig.*, vol. 18, pp. S125–S137, Aug. 2016, doi: 10.1016/j.diin.2016.04.005.

[31] V. Jusas, D. Birvinskas, and E. Gahramanov, “Methods and Tools of Digital Triage in Forensic Context: Survey and Future Directions,” *Symmetry*, vol. 9, no. 4, Art. no. 4, Apr. 2017, doi: 10.3390/sym9040049.

[32] M. Rogers, J. Goldman, R. Mislan, T. Wedge, and S. Debrota, “Computer Forensics Field Triage Process Model,” *J. Digit. Forensics Secur. Law*, 2006, doi: 10.15394/jdfsl.2006.1004.

[33] J. G. Jiang, B. Yang, S. Lin, M. X. Zhang, and K. Y. Liu, "A Practical Approach for Digital Forensic Triage," *Appl. Mech. Mater.*, vol. 742, pp. 437–444, 2015, doi: 10.4028/www.scientific.net/AMM.742.437.

[34] K. Barmpatsalou, T. Cruz, E. Monteiro, and P. Simoes, "Current and Future Trends in Mobile Device Forensics: A Survey," *ACM Comput. Surv.*, vol. 51, no. 3, p. 46:1-46:31, May 2018, doi: 10.1145/3177847.

[35] G. Dorai, S. Aggarwal, N. Patel, and C. Powell, "VIDE - Vault App Identification and Extraction System for iOS Devices," *Forensic Sci. Int. Digit. Investig.*, vol. 33, p. 301007, Jul. 2020, doi: 10.1016/j.fsidi.2020.301007.

[36] P. Feng, Q. Li, P. Zhang, and Z. Chen, "Logical acquisition method based on data migration for Android mobile devices," *Digit. Investig.*, vol. 26, pp. 55–62, Sep. 2018, doi: 10.1016/j.diin.2018.05.003.

[37] A. Levinson, B. Stackpole, and D. Johnson, "Third Party Application Forensics on Apple Mobile Devices," in *2011 44th Hawaii International Conference on System Sciences*, Jan. 2011, pp. 1–9. doi: 10.1109/HICSS.2011.440.

[38] X. Zhang, I. Baggili, and F. Breitinger, "Breaking into the vault: Privacy, security and forensic analysis of Android vault applications," *Comput. Secur.*, vol. 70, pp. 516–531, Sep. 2017, doi: 10.1016/j.cose.2017.07.011.

[39] P. Domingues, M. Frade, L. M. Andrade, and J. V. Silva, "Digital forensic artifacts of the Your Phone application in Windows 10," *Digit. Investig.*, vol. 30, pp. 32–42, Sep. 2019, doi: 10.1016/j.diin.2019.06.003.

[40] N.-A. Le-Khac, M. Roeloffs, and M.-T. Kechadi, "Forensic Analysis of TomTom Navigation Application." arXiv, Apr. 11, 2017. doi: 10.48550/arXiv.1704.03524.

[41] P. Aagaard, B. Dinyarian, O. Abduljabbar, and K.-K. R. Choo, "Family locating sharing app forensics: Life360 as a case study," *Forensic Sci. Int. Digit. Investig.*, vol. 44, p. 301478, Mar. 2023, doi: 10.1016/j.fsidi.2022.301478.

[42] A. Mahajan, M. S. Dahiya, and H. P. Sanghvi, "Forensic Analysis of Instant Messenger Applications on Android Devices," *Int. J. Comput. Appl.*, vol. 68, no. 8, pp. 38–44, Apr. 2013, doi: 10.5120/11602-6965.

[43] R. Das, "Evidence acquisition in mobile forensics | Infosec." Accessed: Nov. 04, 2023. [Online]. Available: https://resources.infosecinstitute.com/topics/digital-forensics/evidence-acquisition-mobile-forensics-2

[44] K.-K. R. Choo, C. Esposito, and A. Castiglione, "Evidence and Forensics in the Cloud: Challenges and Future Research Directions," *IEEE Cloud Comput.*, vol. 4, no. 3, pp. 14–19, 2017, doi: 10.1109/MCC.2017.39.

[45] A. Fukami, R. Stoykova, and Z. Geradts, "A new model for forensic data extraction from encrypted mobile devices," *Forensic Sci. Int. Digit. Investig.*, vol. 38, p. 301169, Sep. 2021, doi: 10.1016/j.fsidi.2021.301169.

[46] P. Sharma, D. Arora, and T. Sakthivel, "Enhanced Forensic Process for Improving Mobile Cloud Traceability in Cloud-Based Mobile Applications," *Procedia Comput. Sci.*, vol. 167, pp. 907–917, Jan. 2020, doi: 10.1016/j.procs.2020.03.390.

[47] F. Sumaila and H. Bahsi, "Digital forensic analysis of mobile automotive maintenance applications," *Forensic Sci. Int. Digit. Investig.*, vol. 43, p. 301440, Sep. 2022, doi: 10.1016/j.fsidi.2022.301440.

[48] C. Whelan, J. Sammons, B. McManus, and T. Fenger, "Retrieval of Infotainment System Artifacts from Vehicles Using iVe," *J. Appl. Digit. Evid.*, vol. 1, no. 1, p. 30, Jul. 2018.

[49] E. Gentry and M. Soltys, "SEAKER: A mobile digital forensics triage device," *Procedia Comput. Sci.*, vol. 159, pp. 1652–1661, Jan. 2019, doi: 10.1016/j.procs.2019.09.335.

[50] D. Quick and K.-K. R. Choo, "Background and Literature Review," 2018, pp. 5–45. doi: 10.1007/978-981-10-7763-0_2.

[51] G. Horsman, “The COLLECTORS ranking scale for ‘at-scene’ digital device triage,” *J. Forensic Sci.*, vol. 66, no. 1, pp. 179–189, 2021, doi: 10.1111/1556-4029.14582.

[52] Á. MacDermott, S. Lea, F. Iqbal, I. Idowu, and B. Shah, “Forensic Analysis of Wearable Devices: Fitbit, Garmin and HETP Watches,” in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Jun. 2019, pp. 1–6. doi: 10.1109/NTMS.2019.8763834.

[53] J. Bays and U. Karabiyik, “Forensic Analysis of Third Party Location Applications in Android and iOS,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 1–6. Accessed: Nov. 04, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9093781

[54] R. Ayers, S. Brothers, and W. Jansen, “Guidelines on mobile device forensics,” National Institute of Standards and Technology, NIST SP 800-101r1, May 2014. doi: 10.6028/NIST.SP.800-101r1.

[55] “Mobile Device Forensic Tool Specification, Test Assertions and Test Cases”. Nov. 04, 2023. [Online]. Available: https://www.nist.gov/document/cftt-document-mobile-device-forensic-tool-test-specification-v31-feb-2021

[56] L. R. Pace, L. A. Salmon, C. J. Bowen, I. Baggili, and G. G. Richard, “Every step you take, I’ll be tracking you: Forensic analysis of the tile tracker application,” *Forensic Sci. Int. Digit. Investig.*, vol. 45, p. 301559, Jul. 2023, doi: 10.1016/j.fsidi.2023.301559.

[57] A. Heinrich, N. Bittner, and M. Hollick, “AirGuard - Protecting Android Users from Stalking Attacks by Apple Find My Devices,” in *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, in WiSec ’22. New York, NY, USA: Association for Computing Machinery, May 2022, pp. 26–38. doi: 10.1145/3507657.3528546.

[58] J. Finn, “How Far Does An AirTag Reach? Distance & Range Explained,” ScreenRant. Accessed: Nov. 04, 2023. [Online]. Available: https://screenrant.com/apple-airtag-bluetooth-distance-range-reach-explained/

[59] F. ID, “FCC ID Search,” FCC ID. Accessed: Nov. 04, 2023. [Online]. Available: https://fccid.io/

[60] D. R. Clark, C. Meffert, I. Baggili, and F. Breitinger, “DROP (DRone Open source Parser) your drone: Forensic analysis of the DJI Phantom III,” *Digit. Investig.*, vol. 22, pp. S3–S14, Aug. 2017, doi: 10.1016/j.diin.2017.06.013.

[61] N. Y. P. Lukito, F. A. Yulianto, and E. Jadied, “Comparison of data acquisition technique using logical extraction method on Unrooted Android Device,” in *2016 4th International Conference on Information and Communication Technology (ICoICT)*, May 2016, pp. 1–6. doi: 10.1109/ICoICT.2016.7571934.

[62] H. H. Lwin, W. P. Aung, and K. K. Lin, “Comparative Analysis of Android Mobile Forensics Tools,” in *2020 IEEE Conference on Computer Applications(ICCA)*, Feb. 2020, pp. 1–6. doi: 10.1109/ICCA49400.2020.9022838.

[63] A. Thomas, “How To Root Samsung Galaxy A13 5G? (SM-A136U, SM-A136U1, SM-A136W & SM-A136B),” SamsungSFour.Com. Accessed: Nov. 04, 2023. [Online]. Available: https://www.samsungsfour.com/tutorials/root-galaxy-a13-5g.html

[64] Ö. F. Yakut and F. Ertam, “A Digital Forensics Analysis for Detection of The Modified COVID-19 Mobile Application,” in *2020 5th International Conference on Computer Science and Engineering (UBMK)*, Sep. 2020, pp. 1–5. doi: 10.1109/UBMK50275.2020.9219416.

[65] “View on-device files with Device Explorer | Android Studio,” Android Developers. Accessed: Nov. 04, 2023. [Online]. Available: https://developer.android.com/studio/debug/device-file-explorer

[66] M.-R. Boueiz, “Importance of rooting in an Android data acquisition,” in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, Jun. 2020, pp. 1–4. doi: 10.1109/ISDFS49300.2020.9116445.

[67] P. E. Mullen, M. Pathé, and R. Purcell, “Stalking: new constructions of human behaviour,” *Aust. N. Z. J. Psychiatry*, vol. 35, no. 1, pp. 9–16, Feb. 2001, doi: 10.1046/j.1440-1614.2001.00849.x.

[68] katharina.kiener-manu, “Cybercrime Module 12 Key Issues: Cyberstalking and Cyberharassment.” Accessed: Nov. 04, 2023. [Online]. Available: //www.unodc.org

[69] L. Reiber, *Mobile forensic investigations: a guide to evidence collection, analysis, and presentation*, Second edition. New York: McGraw-Hill Education, 2019.

[70] J. K. Becker, D. Li, and D. Starobinski, “Tracking Anonymized Bluetooth Devices,” *Proc. Priv. Enhancing Technol.*, 2019, Accessed: Nov. 04, 2023. [Online]. Available: https://petsymposium.org/popets/2019/popets-2019-0036.php

[71] J. G. Murias, D. Levick, and S. McKeown, “A forensic analysis of streaming platforms on Android OS,” *Forensic Sci. Int. Digit. Investig.*, vol. 44, p. 301485, Mar. 2023, doi: 10.1016/j.fsidi.2022.301485.

[72] M. Moreb, *Practical forensic analysis of artifacts on iOS and Android devices : investigating complex mobile devices*. Apress, 2022. Accessed: Nov. 04, 2023. [Online]. Available: https://libcat.arlingtonva.us/ExternalEContent/283704

[73] “Digital Forensics Laboratory Management and Procedures Guide” Accessed: Nov. 04, 2023. [Online]. Available: https://rm.coe.int/16806b3058

[74] E. Mistek, M. A. Fikiet, S. R. Khandasammy, and I. K. Lednev, “Toward Locard’s Exchange Principle: Recent Developments in Forensic Trace Evidence Analysis,” *Anal. Chem.*, vol. 91, no. 1, pp. 637–654, Jan. 2019, doi: 10.1021/acs.analchem.8b04704.

[75] “About the Galaxy SmartTag+,” Samsung uk. Accessed: Oct. 16, 2023. [Online]. Available: https://www.samsung.com/uk/support/apps-services/about-the-galaxy-smarttag-plus/

[76] J. A. last updated, “Samsung Galaxy SmartTag Plus review,” TechRadar. Accessed: Oct. 16, 2023. [Online]. Available: https://www.techradar.com/reviews/samsung-galaxy-smarttag-plus

[77] “Tile Slim: How It Works & Everything You Need to Know | Wallet Tracker,” Tile eCommerce. Accessed: Oct. 16, 2023. [Online]. Available: https://www.tile.com/blog/tile-slim-thin-bluetoothtracker-for-flat-surfaces-devices

[78] “Tile Slim 1-Pack (2022), Black | Bluetooth Wallet Tracker.” Accessed: Oct. 16, 2023. [Online]. Available: https://www.tile.com/product/black-slim

[79] “Which Chipolo do I have?,” Chipolo - Support. Accessed: Oct. 16, 2023. [Online]. Available: https://support.chipolo.net/hc/en-us/articles/115000186205-Which-Chipolo-do-I-have-

[80] “Officially supported devices,” Chipolo - Support. Accessed: Oct. 16, 2023. [Online]. Available: https://support.chipolo.net/hc/en-us/articles/212172905-Officially-supported-devices

[81] “HTC Fetch Review,” PCMAG. Accessed: Oct. 16, 2023. [Online]. Available: https://www.pcmag.com /reviews/htc-fetch

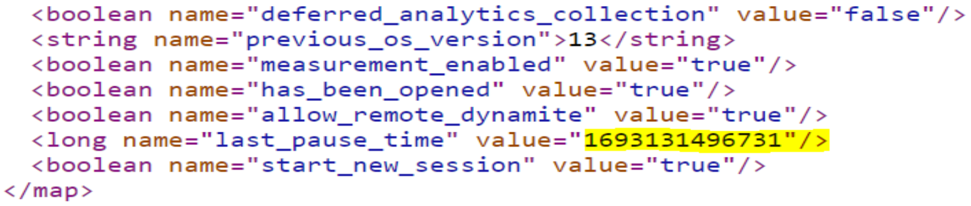
[82] “HTCdev - OpenSense SDK | Bluetooth Low Energy.” Accessed: Oct. 16, 2023. [Online]. Available: https://www.htcdev.com/devcenter/opensense-sdk/bluetooth-smart/htc-fetch/

List of abbreviations and terms

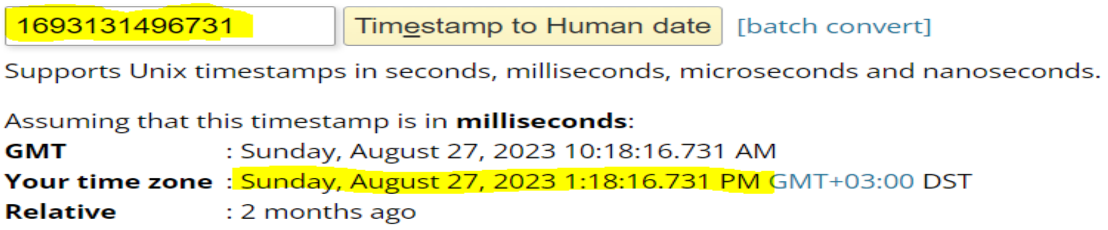
BLE	Bluetooth Low Energy
UWD	Ultra Wide Band
RF	Radio Frequency
EEPROM	Electrically Erasable Programmable ROM
ROM	Read Only Memory
FCC-ID	Federal Communications Commission IDentification
XML	Extensible Markup Language
ADB	Android Debug Bridge
IMEI	International Mobile Equipment Identity
MAC	Media Access Control
WiFi	Wireless Fidelty
GPS	Global Positioning System
JTAG	The Joint Test Action Group
UNIX	UNiplexed Information Computing System
DEFLATE	Data Compression Technique
iOS	iPhone Operating System
GDPR	General Data Protection Regulation
IRB	Institutional Review Board
API	Application Programming Interface
NIST	National Institute of Standards & Technology
INTERPOL	The Internationla Criminal Police Organization
BSI	German Federal Office for Information Security
CuFA	Curated (digital) Forensics Artifact
CFFTPM	Cyber Forensics Field Triage Process Model
UFED	Cellebrite Universal Forensic Extraction Device
CAN	Controller Area Network
SEAKER	Storage Evaluator and Knowledge Extractor Reader

I. Appendix

(Chipolo Classic)



```
  <boolean name="deferred_analytics_collection" value="false"/>
  <string name="previous_os_version">13</string>
  <boolean name="measurement_enabled" value="true"/>
  <boolean name="has_been_opened" value="true"/>
  <boolean name="allow_remote_dynamite" value="true"/>
  <long name="last_pause_time" value="1693131496731"/>
  <boolean name="start_new_session" value="true"/>
</map>
```



1693131496731 | Timestamp to Human date | [batch convert]

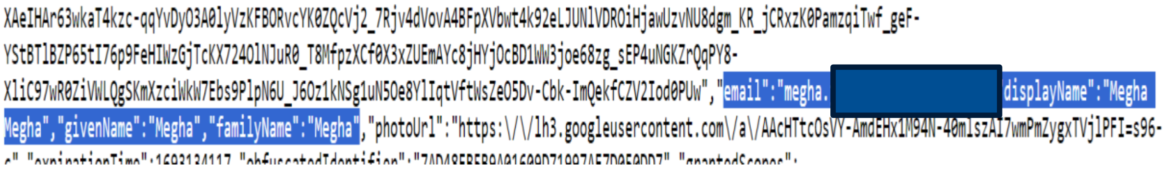
Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **milliseconds**:

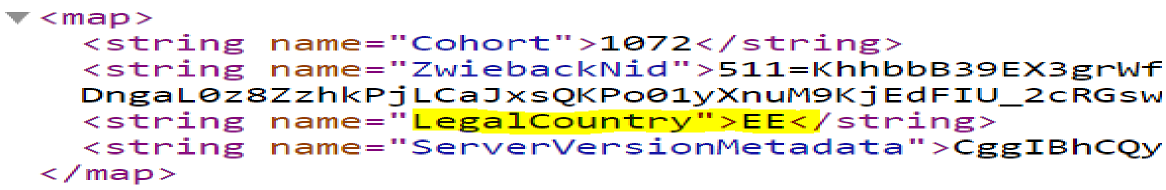
GMT : Sunday, August 27, 2023 10:18:16.731 AM

Your time zone : Sunday, August 27, 2023 1:18:16.731 PM GMT+03:00 DST

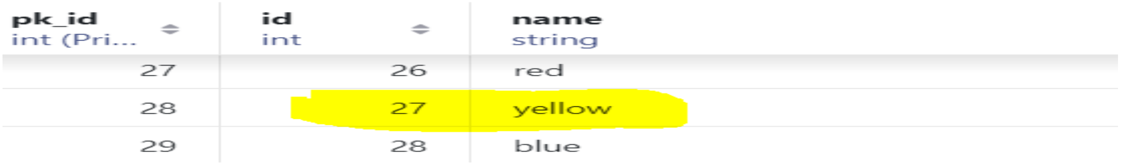
Relative : 2 months ago



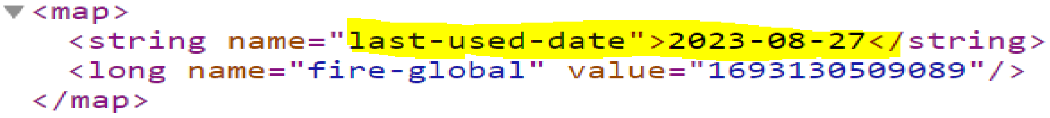
```
XAeIHAr63wkaT4kzc-qqYvDyO3A0lyVzKFBORvcYK0ZQcVj2_7Rjv4dVovA4BFpXVbwt4k92eLJUNlVDROiHjawUzvNU8dgm_KR_jCRxzK0PamzqiTwf_geF-
YStBTlBZP65tI76p9FeHIWzGjTcKX724OlNJuR0_T8MfpzXCf0X3xZUEmAYc8jHYjOcBD1WW3joe68zg_sEP4uNGKZrQqPY8-
XliC97wR0ZiVWLQgSKmXzciWkW7Ebs9PlpN6U_J6Oz1kNSg1uN5Oe8YlIqtVftWsZeO5Dv-Cbk-ImQekfCZV2Iod0PUw","email":"megha.[redacted]displayName":"Megha
Megha","givenName":"Megha","familyName":"Megha","photoUrl":"https:\/\/lh3.googleusercontent.com\/a\/AAcHTtcOsVY-AmdEHx1M94N-40mlszAI7wmPmZygxTVjlPFI=s96-
c","expirationTime":1693134117,"obfuscatedIdentifier":"7AD48FB5B9A01609D71997A57D050DD7","grantedScopes":
```



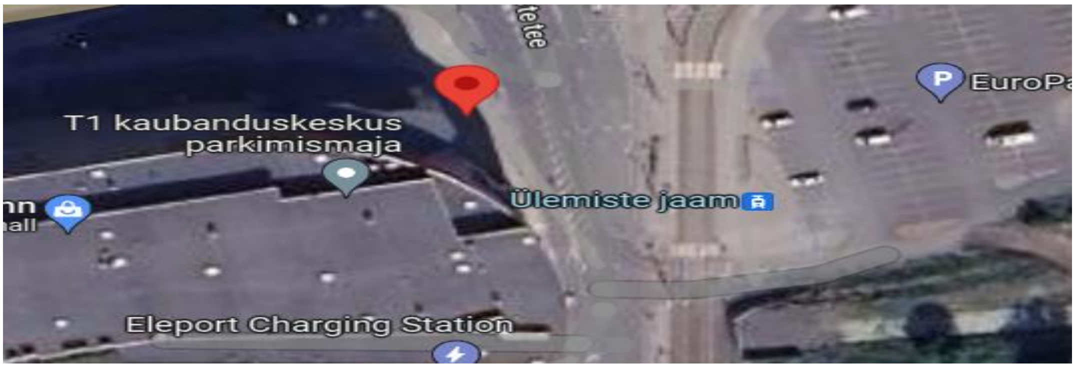
```
▼ <map>
    <string name="Cohort">1072</string>
    <string name="ZwiebackNid">511=KhhbbB39EX3grWf
    DngaL0z8ZzhkPjLCaJxsQKPo01yXnuM9KjEdFIU_2cRGsw
    <string name="LegalCountry">EE</string>
    <string name="ServerVersionMetadata">CggIBhCQy
  </map>
```



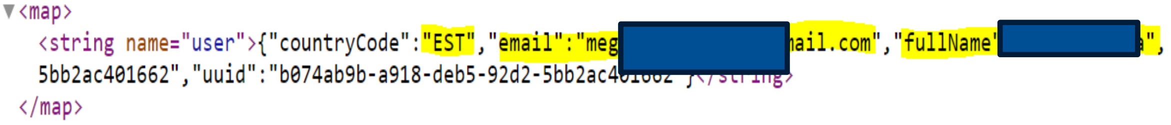
pk_id int (Pri...	id int	name string
27	26	red
28	27	yellow
29	28	blue



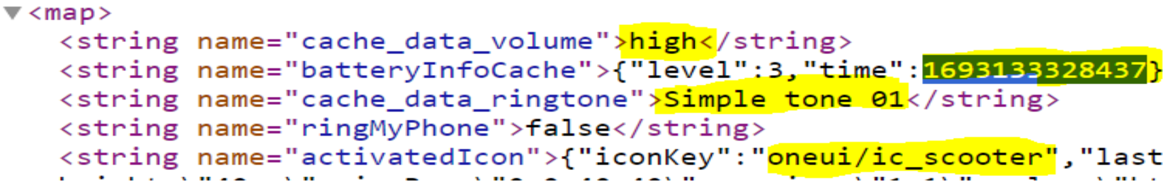
```
▼ <map>
    <string name="last-used-date">2023-08-27</string>
    <long name="fire-global" value="1693130509089"/>
  </map>
```



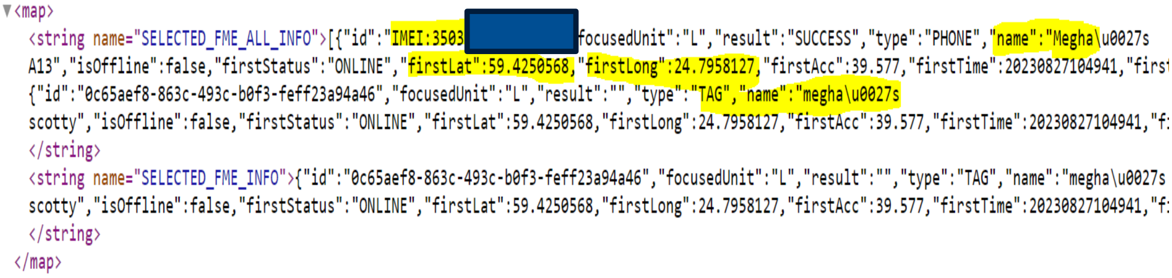
(Galaxy Smart Tag Plus)



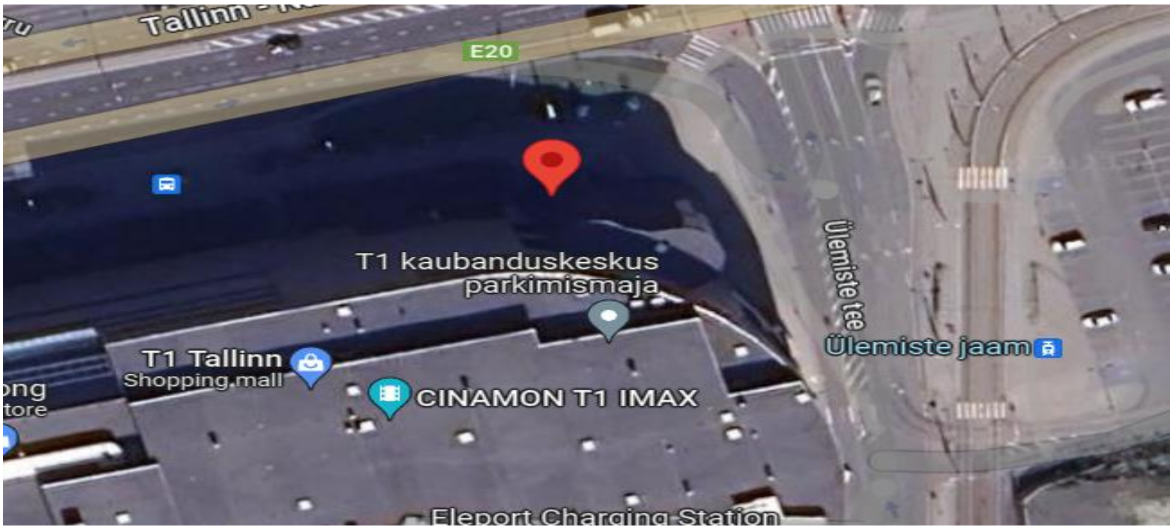
```
▼<map>
   <string name="user">{"countryCode":"EST","email":"meg[redacted]ail.com","fullName"[redacted]",
   5bb2ac401662","uuid":"b074ab9b-a918-deb5-92d2-5bb2ac4 }</string>
 </map>
```

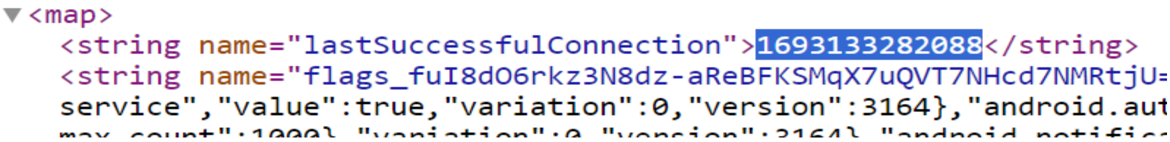


```
▼ <map>
    <string name="cache_data_volume">high</string>
    <string name="batteryInfoCache">{"level":3,"time":1693133328437}
    <string name="cache_data_ringtone">Simple tone 01</string>
    <string name="ringMyPhone">false</string>
    <string name="activatedIcon">{"iconKey":"oneui/ic_scooter","last
```

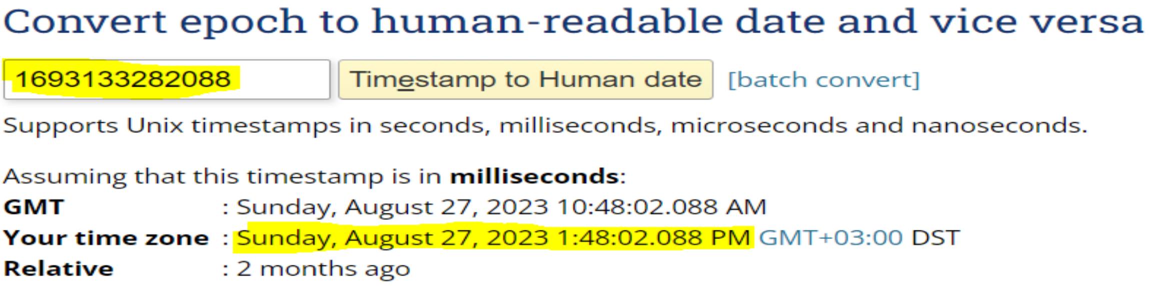


```
▼<map>
   <string name="SELECTED_FME_ALL_INFO">[{"id":"IMEI:3503[redacted]focusedUnit":"L","result":"SUCCESS","type":"PHONE","name":"Megha\u0027s
   A13","isOffline":false,"firstStatus":"ONLINE","firstLat":59.4250568,"firstLong":24.7958127,"firstAcc":39.577,"firstTime":20230827104941,"first
   {"id":"0c65aef8-863c-493c-b0f3-feff23a94a46","focusedUnit":"L","result":"","type":"TAG","name":"megha\u0027s
   scotty","isOffline":false,"firstStatus":"ONLINE","firstLat":59.4250568,"firstLong":24.7958127,"firstAcc":39.577,"firstTime":20230827104941,"f:
   </string>
   <string name="SELECTED_FME_INFO">{"id":"0c65aef8-863c-493c-b0f3-feff23a94a46","focusedUnit":"L","result":"","type":"TAG","name":"megha\u0027s
   scotty","isOffline":false,"firstStatus":"ONLINE","firstLat":59.4250568,"firstLong":24.7958127,"firstAcc":39.577,"firstTime":20230827104941,"f:
   </string>
 </map>
```





```
▼ <map>
    <string name="lastSuccessfulConnection">1693133282088</string>
    <string name="flags_fuI8dO6rkz3N8dz-aReBFKSMqX7uQVT7NHcd7NMRtjU=
    service","value":true,"variation":0,"version":3164},"android.aut
```

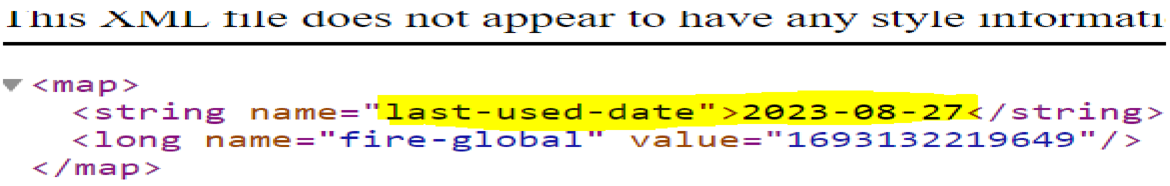


Convert epoch to human-readable date and vice versa

1693133282088 | Timestamp to Human date | [batch convert]

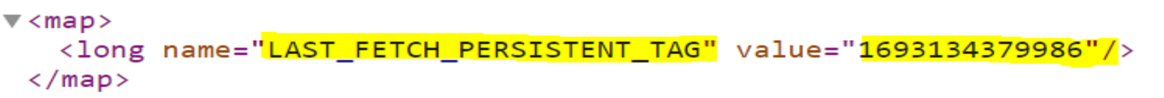
Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **milliseconds**:
GMT : Sunday, August 27, 2023 10:48:02.088 AM
Your time zone : Sunday, August 27, 2023 1:48:02.088 PM GMT+03:00 DST
Relative : 2 months ago

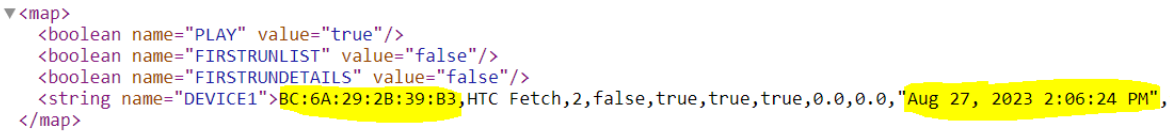


```
This XML file does not appear to have any style informati
▼<map>
    <string name="last-used-date">2023-08-27</string>
    <long name="fire-global" value="1693132219649"/>
  </map>
```

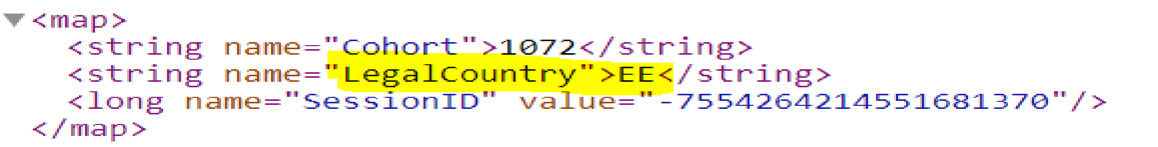
(HTC-Fetch)



```
▼<map>
    <long name="LAST_FETCH_PERSISTENT_TAG" value="1693134379986"/>
  </map>
```

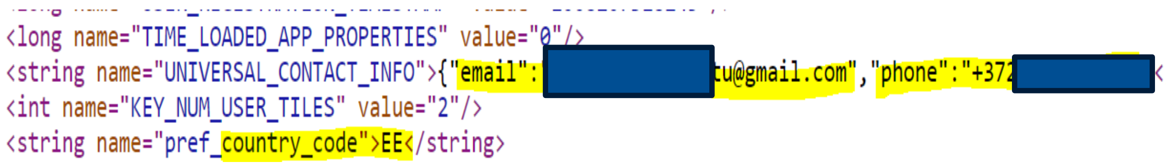


```
▼<map>
   <boolean name="PLAY" value="true"/>
   <boolean name="FIRSTRUNLIST" value="false"/>
   <boolean name="FIRSTRUNDETAILS" value="false"/>
   <string name="DEVICE1">BC:6A:29:2B:39:B3,HTC Fetch,2,false,true,true,true,0.0,0.0,"Aug 27, 2023 2:06:24 PM",
 </map>
```

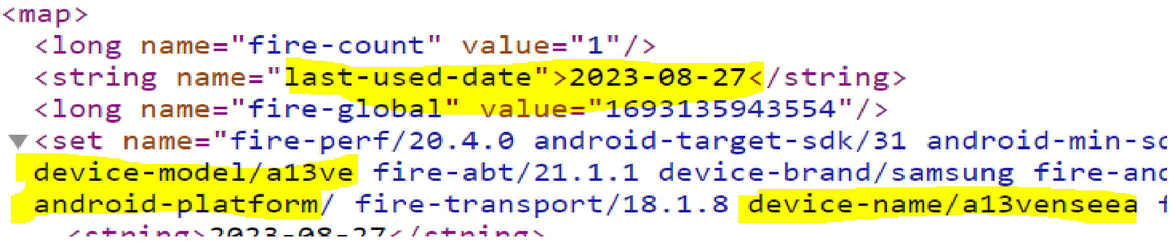


```
▼<map>
    <string name="Cohort">1072</string>
    <string name="LegalCountry">EE</string>
    <long name="SessionID" value="-7554264214551681370"/>
  </map>
```

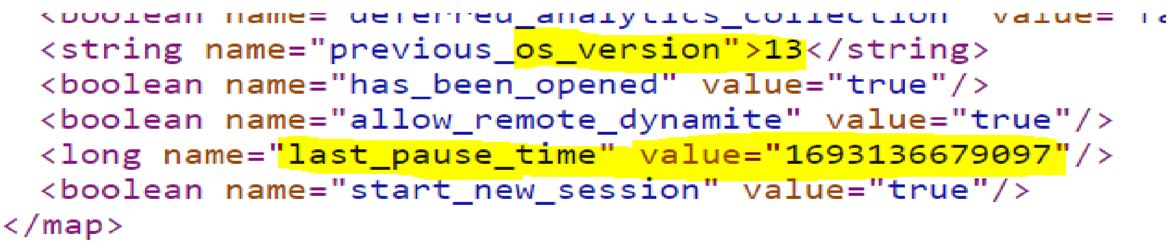
(Tile Slim)



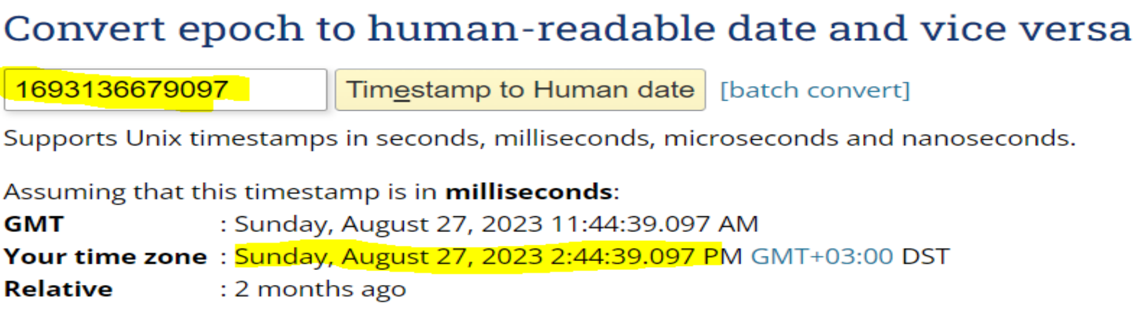
```
<long name="TIME_LOADED_APP_PROPERTIES" value="0"/>
<string name="UNIVERSAL_CONTACT_INFO">{"email":[redacted]u@gmail.com","phone":"+37[redacted]<
<int name="KEY_NUM_USER_TILES" value="2"/>
<string name="pref_country_code">EE</string>
```



```
<map>
  <long name="fire-count" value="1"/>
  <string name="last-used-date">2023-08-27</string>
  <long name="fire-global" value="1693135943554"/>
▼<set name="fire-perf/20.4.0 android-target-sdk/31 android-min-sc
  device-model/a13ve fire-abt/21.1.1 device-brand/samsung fire-anc
  android-platform/ fire-transport/18.1.8 device-name/a13venseea f
  <string>2023-08-27</string>
```



```
  <boolean name="deferred_analytics_collection" value="fa
  <string name="previous_os_version">13</string>
  <boolean name="has_been_opened" value="true"/>
  <boolean name="allow_remote_dynamite" value="true"/>
  <long name="last_pause_time" value="1693136679097"/>
  <boolean name="start_new_session" value="true"/>
</map>
```

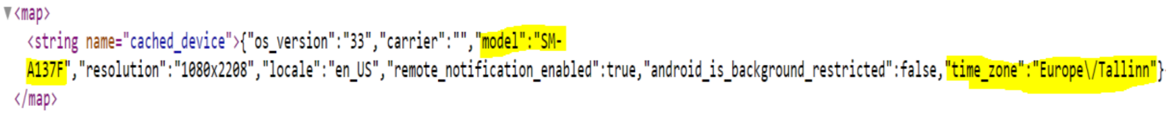


Convert epoch to human-readable date and vice versa

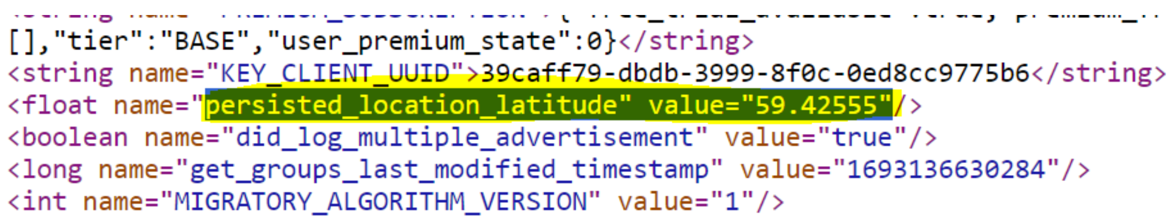
1693136679097 | Timestamp to Human date | [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

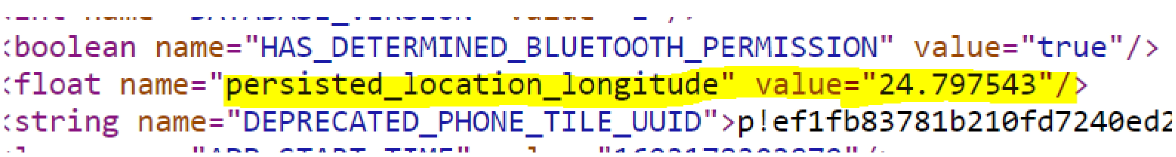
Assuming that this timestamp is in **milliseconds**:
GMT : Sunday, August 27, 2023 11:44:39.097 AM
Your time zone : Sunday, August 27, 2023 2:44:39.097 PM GMT+03:00 DST
Relative : 2 months ago



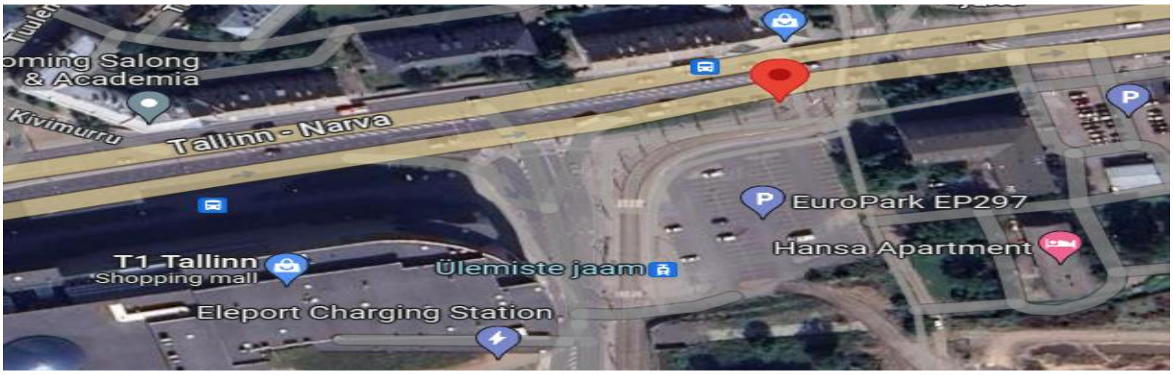
```
▼<map>
  <string name="cached_device">{"os_version":"33","carrier":"","model":"SM-
  A137F","resolution":"1080x2208","locale":"en_US","remote_notification_enabled":true,"android_is_background_restricted":false,"time_zone":"Europe\/Tallinn"}
 </map>
```

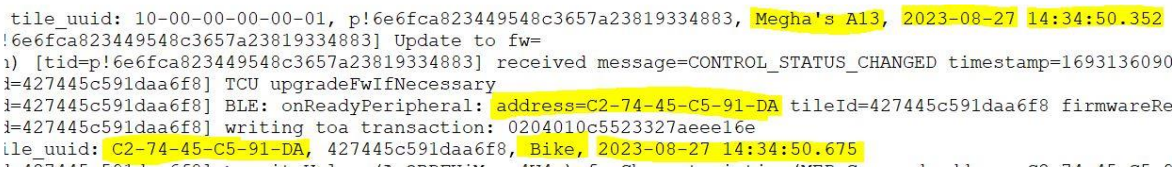


```
[],"tier":"BASE","user_premium_state":0}</string>
<string name="KEY_CLIENT_UUID">39caff79-dbdb-3999-8f0c-0ed8cc9775b6</string>
<float name="persisted_location_latitude" value="59.42555"/>
<boolean name="did_log_multiple_advertisement" value="true"/>
<long name="get_groups_last_modified_timestamp" value="1693136630284"/>
<int name="MIGRATORY_ALGORITHM_VERSION" value="1"/>
```

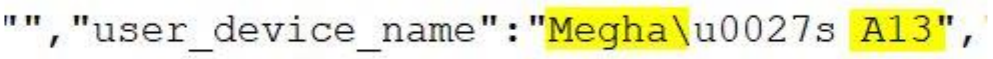


```
<boolean name="HAS_DETERMINED_BLUETOOTH_PERMISSION" value="true"/>
<float name="persisted_location_longitude" value="24.797543"/>
<string name="DEPRECATED_PHONE_TILE_UUID">p!ef1fb83781b210fd7240ed2
```





```
tile_uuid: 10-00-00-00-00-01, p!6e6fca823449548c3657a23819334883, Megha's A13, 2023-08-27 14:34:50.352
!6e6fca823449548c3657a23819334883] Update to fw=
)) [tid=p!6e6fca823449548c3657a23819334883] received message=CONTROL_STATUS_CHANGED timestamp=1693136090
d=427445c591daa6f8] TCU upgradeFwIfNecessary
d=427445c591daa6f8] BLE: onReadyPeripheral: address=C2-74-45-C5-91-DA tileId=427445c591daa6f8 firmwareRe
d=427445c591daa6f8] writing toa transaction: 0204010c5523327aeee16e
ile_uuid: C2-74-45-C5-91-DA, 427445c591daa6f8, Bike, 2023-08-27 14:34:50.675
```



```
"","user_device_name":"Megha\u0027s A13",
```

II. License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, **Md Rashadul Islam**,

1. grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Bluetooth-based Tracking Devices: Extraction and Analysis of Digital Forensic Artifacts from Android Applications,

supervised by Hayretdin Bahsi, PhD
Raimundas Matulevičius, PhD

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Md Rashadul Islam

Tartu, 01/03/2024