A Quiet Place: An In-Depth Study of Mobile Public-to-Private Attacks

Yin Liu
College of Computer Science
Beijing University of Technology
Beijing, China
yinliu@bjut.edu.cn

***Abstract*—Being carried everywhere by end-users, smartphones constantly gather user data, some of which are *private* (e.g., geolocations) and others *public* (e.g., accelerometer readings). To protect against data leakage attacks, most mobile platforms implement permission schemes that restrict access only to private rather than public data. Ironically, such unprotected public data can be craftily exploited by attackers, resulting in an emerging threat of leaking users' private information, which we refer to as a *Mobile Public-to-Private (MP2P) attack*. However, for this particular attack, the current surveys lack details and need more focus on its distinct features, while the proposed attack methods have become increasingly similar. To better understand the MP2P attack, we conducted an in-depth study of academic examples spanning over a decade (2012-2024). Specifically, we first investigate and categorize public resources that can be exploited to leak private information. Then, we systematically explore the standard workflow and methodologies of MP2P attacks. Further, we propose a new MP2P attack method within a typical user scenario, *taking flight*. Based on our experiments with real flight trips, our method successfully infers passengers' flight numbers, departure locations, destinations, and routes. Having deeply analyzed MP2P attacks and unveiled a new attack method, this paper can assist security engineers in devising countermeasures against this attack.**

***Index Terms*—user privacy, data leakage, public resources**

I. Introduction

A. Motivation

Thanks to mobile technologies, smartphones have been featured from simple communication devices to a complex mobile ecosystem. Supported by its colorful services, modern smartphones have become a part of our habits, our lives—even more—"our bodies." However, the deeper the smartphone improves our lives, the more risks it may compromise our privacy. In fact, to ensure various mobile services, a smartphone has to continuously collect our information regardless of whether it is private. Most mobile platforms (e.g., Android and iOS) provide permission schemes that restrict the access of *private* data, such as a user's GPS locations and routes. In contrast, *public* data (e.g., accelerometer readings) can be arbitrarily obtained without any permissions.

Ironically, end-users' private information can still be compromised from the "harmless" public data. Over the last decade, many studies, including surveys (e.g., [1], [2], [14], [32], [37], [44]), attack methodologies (e.g., [24] [46] [43] [27] [17]), and attack detection techniques (e.g., [26] [36] [6] [29] [38]), have showcased this type of attacks, which is referred to as side-channel attacks, linkage attacks, or data leakage attacks. However, as these attack names suggest, the previous surveys focus on a broader problem domain rather than the details and characteristics of this specific attack. Furthermore, current research on attack methodologies and detection techniques tends to focus on homogeneous methods, which can be easier to defend against by using certain types of countermeasures.

Thus, it is essential to study such a mobile-based attack, which uses public data to infer private information. We refer to this attack as a *Mobile Public-to-Private (MP2P) attack*. To deeply comprehend the MP2P attack, we need to answer three research questions:

- RQ1: What type of public data can be exploited for private data leakage, and how would these leakages interact with each other?
- RQ2: What is the typical workflow of MP2P attacks, and what are the most common attack methodologies?
- RQ3: Are there new attack methods or other types of public data that can be used to leak users' private data?

B. Our Work

This paper first conducts an in-depth empirical study of the MP2P attack and then presents a new attack method using previously unexploited types of public data.

To answer RQ1, we deeply studied prior work related to MP2P attacks over a decade (2012-2024), including 6 literature surveys and 36 attack methodology and detection technique papers. Specifically, we first gather public resources from these works, categorize them, and analyze the data types that MP2P attacks can exploit. Second, we associate each public data category with the specific type of private information targeted by MP2P attacks, along with the auxiliary information required for the attack. Additionally, we explore the connections between these types of private information, as one type of private data could compromise other types.

To answer RQ2, we systematically analyzed a significant amount of MP2P attack methods and scenarios. Specifically, we first extract common characteristics of the MP2P attack from the aforementioned papers, summarize the attack's definition, and generalize its workflow. Then, we categorize and compare the attack's specific methodologies.

To answer RQ3, we demonstrated a new MP2P attack method exploiting some public resources that prior work never studied. In particular, this method focuses on a specific scenario — taking flight. That is, it discovers whether a victim is on board and further infers his/her flight number, departure/arrival time, destination, and itinerary by using the public resources along with some auxiliary information online.

C. Contributions

The contribution of this work includes three folds:

1) We investigate and categorize public resources that can be exploited for leaking private information, as well as unveiling the interactions between leaked private data.
2) We conduct an in-depth study of research literature over a decade, revealing typical workflows and methodologies of the *Mobile Public-to-Private (MP2P) attack*.
3) We present a novel method that leverages unstudied public resources and auxiliary online knowledge to infer victims' flight information. An evaluation of two real-world flight trips proves the effectiveness of this method.

We believe this research can encourage mobile users, developers, and researchers to reconsider the privacy of public resources in daily life. Further, it can help airlines' security departments and mobile platform providers consider how to balance the utility and privacy of the "public resources."

II. Survey Methodology

This section outlines our study's challenges and solutions and then introduces our data collection and related surveys.

A. Challenges & Solutions

In previous studies, attacks related to the MP2P attack have been referred to by various names, such as side-channel attacks, linkage attacks, or data leakage attacks. However, a common term has yet to be settled to describe the MP2P attack. As a result, it is challenging to use specific keywords when searching for related research papers.

We were fortunate to find a highly cited paper detailing several MP2P attack methods [52]. Starting with this paper, we employed a snowballing approach to conduct our literature survey. Additionally, we continued to gather new keywords (e.g., public resources, side channels, zero-permission) during the snowballing process, and used these keywords to keep searching for relevant papers.

To consider both the breadth and depth of the research, our study not only analyzes the latest studies but also includes classic ones that are highly cited. Furthermore, we don't limit our review to survey papers, but also include other types of papers on MP2P attacks. Please note that we exclusively select surveys and other papers focusing on the *mobile field* because the MP2P attack targets mobile systems.

B. Data collection

We selected 42 papers spanning over a decade (2012-2024) from hundreds of research papers that were searched. These selected papers can be divided into three categories: attack methodology (31 papers), attack detection technique (5 papers), and survey (6 papers). The first two discuss how an attacker can exploit public resources to leak users' private information and how we can automatically identify which public resources can be exploited. The last one reports the status quo and characteristics of related attacks by reviewing a specific scope of research literature.

C. Related Surveys

As illustrated in Table-I, we summarize related surveys in five aspects (columns 2 - 6). We found that some of them only study a single type of public data (e.g., accelerometer [14], sensors [32], network [1], see the column "Multiple Data Types") or a single mobile system (e.g., Android [2], [44], the column "Multiple Mobile Systems"). Furthermore, only two of them (partially) investigate the interactions between leaked information [14], [32] (the column "Leaked Data Interactions"), or provide a standard workflow or the definition of the attack [1], [37] (the column "Standard Workflow or Definition"). In addition, none of them propose new attack methods (the column "New Attack Method"). Our study, to the best of our knowledge, is the first work that covers all the five aspects mentioned above, presenting a unique perspective that differs from previous surveys.

TABLE I
Summary of Related Surveys
●: YES; ◐: Partially YES; ○: NO

Related Work	Multiple Data Types?	Multiple Mobile Systems?	Leaked Data Interactions?	Standard Workflow or Definition?	New Attack Method?	Year
Xu et al. [44]	●	○	○	○	○	2016
Spreitzer et al. [37]	●	●	○	◐	○	2017
Alqazzaz et al. [2]	●	○	○	○	○	2018
Kroger et al. [14]	○	●	●	○	○	2019
Sikder et al. [32]	○	●	◐	○	○	2021
Agrawal et al. [1]	○	●	○	●	○	2022
Ours	●	●	●	●	●	2024

"Multiple Data Types": does it involve multiple types of public data?
"Multiple Mobile Systems": does it include multiple mobile systems?
"Leaked Data Interactions": does it investigate the interactions between leaked information?
"Standard Workflow or Definition": does it provide a standard workflow or the definition of the attack?
"New Attack Method": does it propose new attack methods?

III. Public Resources, Private Information, and their Interactions

In this section, we first present the exploitable public resources and their linked private information. We then discuss the interactions between the leaked information.

A. Public Resources to Private Information

Table II presents the private information being leaked (the first column), the types of public data that can be exploited to leak this information (the second column), and the exact public data (the third column). This information is summarized from the papers we reviewed (see § II-B).

In general, there are seven types of private information that can be leaked through public data exploitation, including "Location & Routes," "App Behavior & Status" (e.g., installed or running apps, browser/UI states, and all users' activities on those apps), "User & Device Identify" (e.g., user name and

TABLE II
SUMMARY OF PUBLIC RESOURCES AND PRIVATE INFORMATION

Private Info	Public Data Type	Specific Public Data	Ref.
Location & Routes	Sensors	accelerometer	[11] [14] [32]
		magnetometer	[32] [27]
		accelerometer and gyroscope	[16] [32]
		gyroscope, accelerometer, and magnetometer	[22] [32] [51]
		light	[47]
	Processes & System Info	network	[4] [6] [40] [44] [52] [20] [15]
		power	[19] [45] [2]
		other or combined info (usually in /proc files)	[37]
	Peripherals	microphone, speaker	[32]
		speaker	[52]
	Other	photo file's metadata	[29]
	Combinations	timezone,device's address (IP), network status, accelerometer, magnetometer, barometer	[20]
App Behavior & Status	Sensors	magnetometer	[18] [32]
		accelerometer	[14]
		light	[32]
		motion, light, temperature	[32]
		accelerometer, magnetometer, and gyroscope	[42]
		any embedded sensors	[32]
	Processes & System Info	network	[31] [37] [17] [9] [1] [4] [52] [8] [35]
		power	[7] [37]
		memory	[44] [48] [2]
		API Exec Time	[26] [38]
		interuputs	[10]
		other or combined info (usually in /proc files)	[48] [2] [50] [36] [44] [49] [37]
	Peripherals	combined: microphone and magnetometer	[24]
User & Device Identity	Sensors	accelerometer	[14]
	Processes & System Info	network	[6] [29] [37] [1]
		memory	[48]
		other or combined info (usually in /proc files)	[37] [43]
	Peripherals	microphone	[37]
		speaker	[37]
Keystroke	Sensors	accelerometer and gyroscope	[28] [32]
		light	[32]
		accelerometer	[25] [14] [44] [32]
		device orientation or/and accelermetor	[2]
		accelerometer, magnetometer	[32]
		magnetometer	[32]
		gyroscope	[32]
		accelerometer, gyroscope, light	[32]
	Processes & System Info	network	[37]
		power	[45] [2]
		GPU	[46]
		other or combined info (usually in /proc files)	[43] [37] [36]
	Peripherals	combined: microphone, magnetometer	[32] [24]
		camera	[32] [37]
		microphone, camera	[32]
		microphone, speaker	[32]
		microphone	[32] [37]
		combined: microphone and gyroscope	[21] [32]
Credentials	Sensors	accelerometer	[44]
		magnetometer	[37]
	Processes & System Info	CPU	[44]
		other or combined info (usually in /proc files)	[37] [10]
Personal Characteristics	Sensors	accelerometer	[14]
	Processes & System Info	network	[44]
Voice Info	Sensors	gyroscope	[32]
		accelerometer	[32]
	Peripherals	microphone	[32]
		microphone, speaker	[32]

device IMEI/Brand/OS), “Keystroke” (e.g., keystrokes, screen taps, soft-keyboard gestures, and user input), “Crentials” (e.g., password, crypto key, PIN, and unlock pattern), “Personal Characteristic” (e.g., medical conditions, gender, and emotion), and “Voice Info” (e.g., user voice).

Correspondingly, four types of public resources can be exploited to compromise above private information, including “Sensors” (e.g., accelerometer and gyroscope), “Processes & System Info” (e.g., network, power, and other information in the /proc files), “Peripherals” (e.g., microphone and speaker), “Other” (e.g., photo files), and their combinations. Although space constraints make it difficult to detail each row in the table, it is evident that a large amount of public data, typically considered “harmless,” can result in significant breaches of user privacy.

B. Interactions between Leaked Information

Figure 1 illustrates the interactions between the seven types of private information mentioned above. It is important to note that the leaked information is not independent and can be chained to cause a larger privacy breach. Specifically, we summarize these links as follows.

(a) *“Location&Routes”* ↔ *“App Behavior & Status”:* Li et al. used Wifi lists to obtain users’ locations and routes, and then used this information to infer the victim’s daily activities [15]. In contrast, Watanabe et al. first used sensor readings to identify user activities (e.g., walking or on vehicles) and then used this information along with train timetables and maps to infer the user’s routes [42].

(b) *“App Behavior & Status”* → *“User & Device Identify”:* Zhou et al. used mobile data usage values to analyze Twitter apps’ behavior. Then, they leveraged this behavioral data to identify users who posted the tweets [52].

(c) *“App Behavior & Status”* → *“Personal Characteristic”:* Zhou et al. first used network usage statistics to analyze WebMD apps’ behavior. Then, they used this behavioral data to infer the user’s diseases [52].

(d) *“App Behavior & Status”* → *“Crentials”:* Ni et al. used data from the magnetometer and microphone to capture UI switch sequences, which were then combined with keystrokes to deduce the unlock screen passcode [24].

(e) *“Keystroke”* → *“Crentials”:* Owusu et al. utilized keystrokes infered from accelerometer data to successfully crack 59 out of 99 passwords [25].

(f) *“Voice Info”:* In the reviewed papers, we did not find any specific examples of attacks that link “Voice Info” to other private information. However, intuitively, if a user’s voice is compromised, all their private information could be leaked if they mention it while speaking.

IV. Mobile Public-to-Private Attack

In this section, we present the MP2P attack’s definition and standard workflow and then discuss its methodologies.

A. Definition and Standard Workflow

1) Definition: As we named it “Mobile Public-to-Private,” this attack naturally has three major characteristics: “mobile system,” “public resources,” and “private information.” Therefore, the *Mobile Public-to-Private (MP2P) attack* is an attack performed on a mobile system, exploiting public resources to leak private information.

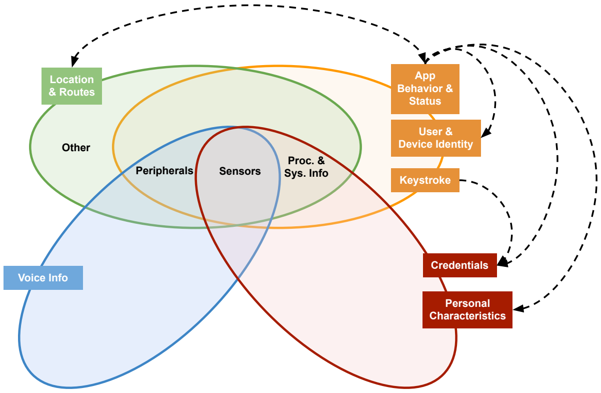


Fig. 1. Interactions between Leaked Information.

Especially, The “private information” refers to data that, if exposed, would directly or indirectly harm the end-users. This information is safeguarded by the mobile system and cannot be accessed without proper permissions. The “public resources” consist of two types of data. The first type is data primarily exploited by attackers, while the second type is auxiliary data that helps in leaking private information. Both types of data are publicly available and can be freely obtained within the mobile system or online.

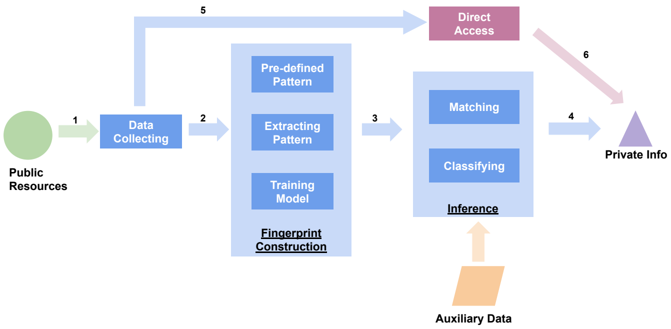


Fig. 2. The standard workflow of MP2P attacks.

2) Workflow: The workflow includes 4 phases (Figure 2):

(a) Collect public resources (usually over a period of time, step-1). For example, one can collect a trace of accelerometer readings over 5 minutes.

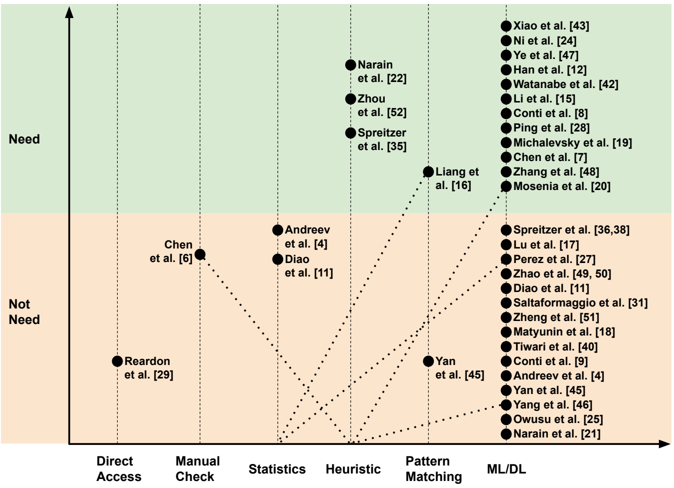
(b) Construct fingerprints of collected public resources by training models or using pattern-extracting methods (e.g., generate a fingerprint from the collected traces to represent a user’s routes, step-2). Besides, an attacker may already have pre-defined patterns.

(c) Use constructed fingerprints as input to uncover private information by pattern-matching or classifying algorithms (e.g., match the occurrence of a given trace via the fingerprints to infer a user’s routes, steps 3 and 4). The auxiliary data (e.g., maps or timestamps) can be used to supply fundamental information for the matching process [11] [20] or to improve the inference accuracy [22].

(d) In some cases, private information can be directly obtained from public resources (steps 5 and 6). For instance, Reardon et al. caution that attackers can directly read geolocation from carelessly shared photo files [29].

B. Attack Methodologies

We have compiled several attack methods from our reviewed papers on attack methodology and detection techniques (see Figure 3). Generally, most attacks utilize machine or deep learning approaches ("ML/DL") due to their ability to effectively fingerprint data. Additionally, some attacks use hybrid methods, combining ML/DL with heuristics or statistics, statistics with pattern matching, and heuristics with manual checks (the dotted line in Figure 3). Furthermore, roughly half of the attacks require auxiliary data, while the other half do not.



X-axis: each type of MP2P attack methods;
Y-axis: "Need" or "Not Need" the auxiliary data in the attack.
(a) we consider the Dynamic Time Warping (DTW) alone as a "Pattern Matching" algorithm. However, if the DTW is integrated with a machine or deep learning process, we think of it as a "ML/DL" approach;
(b) we consider the Hidden Markov Model as a "Statistics" method.

Fig. 3. The summary of methodologies of the MP2P attack.

C. Threat Model

In this section, we generalize the behavior model of attackers and mobile users. Please note that the behaviors associated with our new attack method are discussed in § V-B.

a) Attackers: In the MP2P attack, attackers aim to infer mobile users' private information, such as user identity, specific activities, and locations. However, instead of obtaining this information directly, the attackers can only exploit the public resources that have been published in any form. For example, an attacker can obtain public resources collected by mobile platforms (e.g., Android, iOS) or released by online websites, inferring a user's routes without directly touching any private information, such as GPS data.

b) Mobile Users: Mobile users, as victims of the MP2P attack, will not hide or restrict their public resources from being accessed and collected. For example, a mobile user will not turn off the accelerometer sensor or perturb its value by intentionally shaking the phone. Further, if an attacker uses a particular service to obtain public resources, the mobile user will not kill the service to block the data collection.

V. Getting your flight information

This section illustrates a new MP2P attack method to infer users' specific flight information from their public resources. First, we discuss threats of such flight information. We then model behaviors of attackers and users. Finally, we detail the attack process and demonstrate its effectiveness through a real-world case study.

A. Threats of Flight Information

It is obvious that customers' flight information is so-called "personally identifiable information (PII)" [23], whose exfiltration would expose an individual's identity or lead to unexpected incidents: in May 2021, a Korean idol group was mobbed at an airport by crazy fans who were aware of their flight schedule and destination [12]; In 2019, another idol had to "run for his life" away from the fans waiting for him at an airport [13]. To protect the flight information from being disclosed, airlines always claim their mobile apps or websites are equipped with several technical safeguards [3], [34], [41].

However, these safeguards focus on private information (e.g., booking information) without paying enough attention to the "harmless" public resources. Hence, we believe attackers can work around those safeguards by exploiting the public resources. In the following sections, we present our novel attack method that only relies on public resources, such as cell phone's airplane mode status and gyroscope data, to infer a customer's flight information.

B. Modeling Behaviors

In this section, we introduce the behavior models of attackers, malware, and mobile users, respectively.

a) Malware and Attackers: The malware aims to gather public resources from users' smartphones. Specifically, we assume malware has already been installed in the victim's smartphone. Moreover, it can only access public resources from the phone and send it to attackers. The attackers are looking to find out the victim's flight information, and they can only use the public resources collected by the malware or freely available information online to achieve this.

b) Mobile Users: We assume mobile users never grant permission to access their private information to the malware. However, they will not prevent the malware from touching public resources (e.g., manually shut down the malware or turn off mobile sensors). *Importantly, we assume the user will comply with the rule on the flight — the user will turn on the airplane mode before the flight takes off and turn off the airplane mode after the flight lands.* We use this special user behavior on the flight as the pre-defined behavior pattern. In addition, the user will not restrict the malware's network access, so it can send the collected data to attackers.

C. Our Attack Method

In this section, we first outline our attack. Next, we detail the attack process and demonstrate how it works with an example.

1) Overview: As shown in Figure 4, our attack comprises two phases: "data collection" and "pattern matching & inference." The former relies on the pre-installed malware to continuously collect public resources and send them to an attacker. The latter requires the attacker to infer flight information by synthesizing the collected data, our pre-defined behavior pattern, and other reachable public data.

Please note that in our *MP2P attack*, the *public resources* include airplane mode status and accelerometer/gyroscope readings; the targeted *private information* contains passengers' flight numbers, departure locations, destinations, and routes; the *auxiliary information* refers to online flight schedule.

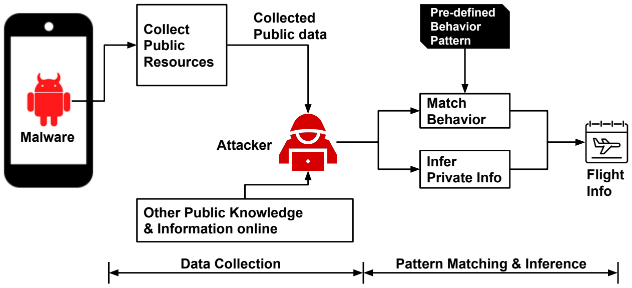


Fig. 4. Attack Overview.

2) Public Data Collection: In this phase, the pre-installed malware auto-launches when the smartphone boots. Then, it starts a background service that continuously collects public resources, including airplane mode status (on/off) and gyroscope/accelerometer readings. Note that most mobile systems (e.g., Android) allow these resources to be obtained without any permissions. To prevent the data-collecting process from being stopped accidentally, we implement a memory-resident malware service. That is, if the service has been killed, it will be restarted either by the mobile system or the malware itself.

As soon as the malware obtains a piece of public data, it converts the data into a vector with a particular form: [date][type][value].

- *date* is the instant time the malware obtains this information (e.g., 20240421 17:36:10);
- *type* is the public data type, including airplane mode, accelerometer, and gyroscope;
- *value* is the pubic data value based on the *type*: when referring to airplane mode, the value will be "1" for "on" and "0" for "off." For the gyroscope/accelerometer, the value refers to a specific sensor reading.

Finally, the malware periodically sends the information vectors to the attacker for further inference of the victim's specific flight information.

3) Pattern Matching & Inference: After receiving the data sent by the malware, the attacker infers the flight information in three steps based on our pre-defined pattern:

a) Matching the user behavior: As discussed in § V-B, the victim always adheres to airplane rules and performs specific behaviors: turn on airplane mode before takeoff and turn off it after landing. Hence, by checking the timestamps of the airplane mode switch, the attacker infers when the plane is taking off or landing. However, although one can approximate a flight's departure and arrival time via those behaviors, some other user activities may generate noise that misleads the inference of flight information. For example, a user may accidentally switch the airplane mode or intentionally change it during a meeting or a movie. To overcome this challenge, we filter the noise out by the following information:

- *Time Spans:* We observe that the world's longest and shortest flights take more than 18 hours and about 2 minutes, respectively [5], [30]. Based on this fact, we set the valid time intervals for switching airplane modes from 1 to 1440 minutes (i.e., 24 hours). For example, if a user accidentally turns on airplane mode and then turns it off immediately (i.e., the time interval is less than 1 minute), the malware will consider it the noise and filter it out of the collected dataset.
- *Accelerometer Readings:* Accelerometer readings can help predict whether a user is on an airplane. Suppose the collected data of airplane mode switches fall within a valid period. However, the accelerometer readings stay outside an empirical range that can reflect the user being in an airplane. In that case, the malware will filter that noise from the dataset.

To sum up, the public data that malware sends to attackers includes (1) timestamps representing possible airplane departure/arrival times and (2) sensor readings (i.e., accelerometer and gyroscope readings) within the valid time spans.

b) Inferring the flight info: After receiving the public data from the malware, the attacker starts inferring the flight information. Specifically, they search online flight databases [33], [39] and match the received timestamps with actual flights' departure/arrival time. However, such timestamps only represent approximate periods, which may not accurately identify a flight and may match multiple flights.

To improve the accuracy, our attack method provides an "enlarge-then-reduce" algorithm: first enlarge the scope of timestamps by $5\% \sim 20\%$ to obtain more airplane candidates; then reduce the scope by identifying the flight direction from gyroscope readings to determine airplanes that match the most. This "enlarge-then-reduce" process is dynamic, and the above ratios may need to be adjusted multiple times to achieve a satisfactory result.

D. Case Study — Real Flight Trips

This section seeks to answer *how effectively our attack works on real flight trips.* We first discuss the experiment setup and then detail the entire attack process and experiment results.

1) Experiment setup: To collect and send public resources to the attacker, we implemented the malware as an Android app and installed it on a Moto G2 with Android Lollipop. We recruited a volunteer to play the victim role and carried the Moto G2 device with the pre-installed malware. The authors played the attacker role and worked on a workstation running Ubuntu 16.04, Intel i7-7500 CPU, and 15.4 GB memory.

To evaluate our attack method in a real-world case, a volunteer took two flights with the setup device. First, he departed on UA-3868 from Roanoke-Blacksburg Regional

Airport (ROA) and arrived at Dulles International Airport (IAD). Then, he flew on UA-915 from IAD to Paris Charles de Gaulle Airport (CDG).

2) Attack in Action:

a) Make sure the victim is on the plane: As mentioned in § V-C, the malware periodically sends the collected public resources to us. Then, as attackers, we start by checking the timestamps of the airplane mode switch. In our case, the data shows the user turned on airplane mode on April 27 at 14:35:43 and later turned it off at 15:20:10. The duration is about 45 minutes. On the same day, the user turned on the airplane mode again at 17:22:50 and turned it off at 00:41:23 the next day. The duration is about 7 hours and 20 minutes. Then, we examined the accelerometer readings and found strong turbulence in the above duration. Hence, we concluded that the user was very likely on the flight.

b) Finding possible routes: Next, we inferred the itinerary based on the above timestamps, which indicate the departure/arrival time. We first enumerated all airports in the U.S. and crawled their flight departure/arrival time from the flightarrivals.com website. Then, we discovered eight routes for the first trip from April 27 at 14:35:43 to 15:20:10.

c) Determine flights and trips: Since we already have eight possible routes for the first trip, there is no need to enlarge them, only to reduce the number. To that end, (i) we realized that the second trip endured about 7 hours following the first one, indicating it was very likely an international flight. Moreover, only IAD provided such international routes after checking destination airports among the eight routes. Hence, we found two valid flights: UA-3868 (from ROA to IAD) and UA 3885 (from AVP to IAD). (ii) We examined the gyroscope readings during this period and found the direction heading northeast then southeast. Fortunately, among all the eight routes, only flight UA-3868 from ROA to IAD matches such direction. Combining the results in (i)(ii), we identified the flight for the first trip as UA-3868 from ROA to IAD.

As we were confident that the second trip's departure airport was IAD (Washington D.C., the U.S.), we checked the international flights, whose departure time was around 17:22 on April 27. Then, we found that only one flight fit the trip duration, UA-915, from IAD to CDG (Paris, France). So far, we have finally obtained all the victim's flight information.

E. Limitations

a) Limits on Behaviors: Our attack method assumes that users will adhere to the airplane rules. In other words, a user will switch the airplane mode when they hear cabin crews ask to turn on the mode or inform them to turn it off safely. On the one hand, if the user denies following the flight's rule or forgets to turn the airplane mode off after landing[1], then it is difficult to infer the departure/arrival time.

On the other hand, if a user shuts down their phone instead of activating airplane mode during a flight, the attacker cannot access the accelerometer or gyroscope readings. Since our method needs those readings to narrow down possible routes, the inference accuracy will be reduced, even if we can obtain the phone's shutdown/boot time without permission.

b) Limits on Timestamps: Our attack method relies on the timestamps of airplane mode switches to determine flights' departure/arrival time. However, users may not change the mode in time, even if they adhere to the airplane rules. For example, when a user hears cabin crews ask to activate the airplane mode, they may do it after a few minutes instead of immediately. To mitigate this limit, our method provides the "enlarge-then-reduce" algorithm that adjusts the obtained timestamps back and forth to match a sufficient number of possible flights.

VI. Conclusions

In this paper, we have delved into the MP2P attack. First, by thoroughly studying academic examples over a decade, we classified public resources that could be exploited to leak private information. Then, we systematically analyzed the typical workflow and attack methodologies of the attack. Further, we presented a novel attack method to infer users' private information in a specific scenario—taking flight. Our real-world experiment proved the method's effectiveness.

Acknowledgements

The author thanks the anonymous reviewers, whose insightful comments helped improve this paper, and also thanks Da Pu and Yongce Chen for their help with the case study in this paper. This research is supported by Beijing Natural Science Foundation (Grant No. 4232019).

References

[1] A. Agrawal, A. Bhatia, A. Bahuguna, K. Tiwari, K. Haribabu, D. Vishwakarma, and R. Kaushik, "A survey on analyzing encrypted network traffic of mobile devices," *International Journal of Information Security*, vol. 21, no. 4, pp. 873–915, 2022.

[2] A. Alqazzaz, I. Alrashdi, R. Alharthi, E. Aloufi, and M. A. Zohdy, "An insight into android side-channel attacks," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018, pp. 776–780.

[3] American Airlines, "Privacy policy," 2021, www.aa.com/i18n/customer-service/support/privacy-policy.jsp.

[4] M. Andreev, A. Klausner, T. Tiwari, A. Trachtenberg, and A. Yerukhimovich, "Nothing but net: Invading android user privacy using only network access patterns," *arXiv preprint arXiv:1807.02719*, 2018.

[5] BudgetAir.com, "Longest and shortest flights in the world," 2019, www.budgetair.com/en_us/blog/longest-and-shortest-flights-in-the-world.

[6] S. Chen, S. Zhao, B. Han, and X. Wang, "Investigating and revealing privacy leaks in mobile application traffic," in *2019 Wireless Days (WD)*. IEEE, 2019, pp. 1–4.

[7] Y. Chen, X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Powerful: Mobile app fingerprinting via power analysis," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[8] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Can't you hear me knocking: Identification of user actions on android apps via traffic analysis," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 2015, pp. 297–304.

[9] ——, "Analyzing android encrypted network traffic to identify user actions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 114–125, 2015.

[10] W. Diao, X. Liu, Z. Li, and K. Zhang, "No pardon for the interruption: New inference attacks on android through interrupt timing analysis," in *IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 414–432.

[11] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang, "Accomplice: Location inference using accelerometers on smartphones," in *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*. IEEE, 2012, pp. 1–9.

[1] It is a rare case since one cannot use the cellular network or make a call when the airplane mode is on.

[12] D. Johnson, "Ateez's members mobbed," 2021, https://shorturl.at/BQz2I.

[13] Koreaboo, "Exo's chanyeol chased by sasaengs at airport, leaves fans speechless," 2019, https://www.koreaboo.com/news/exo-chanyeol-sasaeng-airport.

[14] J. L. Kröger, P. Raschke, and T. R. Bhuiyan, "Privacy implications of accelerometer data: a review of possible inferences," in *Proceedings of the 3rd international conference on cryptography, security and privacy*, 2019, pp. 81–87.

[15] F. Li, X. Wang, B. Niu, H. Li, C. Li, and L. Chen, "Exploiting location-related behaviors without the gps data on smartphones," *Information Sciences*, vol. 527, pp. 444–459, 2020.

[16] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, 2017.

[17] J. Lu and S. Yu, "Application identification based on overlap relationship of concurrent flows and their durations," in *Proceedings of the 2023 International Conference on Communication Network and Machine Learning*, 2023, pp. 88–92.

[18] N. Matyunin, Y. Wang, T. Arul, K. Kullmann, J. Szefer, and S. Katzenbeisser, "Magneticspy: Exploiting magnetometer in mobile devices for website and application fingerprinting," in *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, 2019, pp. 135–149.

[19] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "{PowerSpy}: Location tracking using mobile device power analysis," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 785–800.

[20] A. Mosenia, X. Dai, P. Mittal, and N. K. Jha, "Pinme: Tracking a smartphone user around the world," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 3, pp. 420–435, 2017.

[21] S. Narain, A. Sanatinia, and G. Noubir, "Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning," in *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*, 2014, pp. 201–212.

[22] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," in *2016 IEEE Symposium on Security and Privacy*. IEEE, 2016, pp. 397–413.

[23] A. Narayanan and V. Shmatikov, "Myths and fallacies of" personally identifiable information"," *Communications of the ACM*, vol. 53, no. 6, pp. 24–26, 2010.

[24] T. Ni, X. Zhang, C. Zuo, J. Li, Z. Yan, W. Wang, W. Xu, X. Luo, and Q. Zhao, "Uncovering user interactions on smartphones via contactless wireless charging side channels," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 3399–3415.

[25] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: password inference using accelerometers on smartphones," in *proceedings of the twelfth workshop on mobile computing systems & applications*, 2012, pp. 1–6.

[26] G. Palfinger, B. Prünster, and D. J. Ziegler, "Androtime: Identifying timing side channels in the android api," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 1849–1856.

[27] B. Perez, A. Mehrotra, and M. Musolesi, "Marcopolo: A zero-permission attack for location type inference from the magnetic field using mobile devices." Springer Nature, 2024.

[28] D. Ping, X. Sun, and B. Mao, "Textlogger: inferring longer inputs on touch screen using motion sensors," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2015, pp. 1–12.

[29] J. Reardon, Á. Feal, P. Wijesekera, A. E. B. On, N. Vallina-Rodriguez, and S. Egelman, "50 ways to leak your data: An exploration of apps' circumvention of the android permissions system," in *28th USENIX security symposium (USENIX security 19)*, 2019, pp. 603–620.

[30] Robert Schrader, "The world's shortest scheduled flights," 2019, www.tripsavvy.com/the-worlds-shortest-scheduled-flights-4101228.

[31] B. Saltaformaggio, H. Choi, K. Johnson, Y. Kwon, Q. Zhang, X. Zhang, D. Xu, and J. Qian, "Eavesdropping on {Fine-Grained} user activities within smartphone apps over encrypted network traffic," in *10th USENIX workshop on offensive technologies (WOOT 16)*, 2016.

[32] A. K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A. S. Uluagac, "A survey on sensor-based threats and attacks to smart devices and applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1125–1159, 2021.

[33] M. Smith, D. Moser, M. Strohmeier, V. Lenders, and I. Martinovic, "Undermining privacy in the aircraft communications addressing and reporting system (acars)," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, pp. 105–122, 2018.

[34] Southwest Airline, "Privacy policy," 2021, www.southwest.com/html/about-southwest/terms-and-conditions/privacy-policy-pol.html.

[35] R. Spreitzer, S. Griesmayr, T. Korak, and S. Mangard, "Exploiting data-usage statistics for website fingerprinting attacks on android," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2016, pp. 49–60.

[36] R. Spreitzer, F. Kirchengast, D. Gruss, and S. Mangard, "Procharvester: Fully automated analysis of procfs side-channel leaks on android," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 749–763.

[37] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard, "Systematic classification of side-channel attacks: A case study for mobile devices," *IEEE communications surveys & tutorials*, vol. 20, pp. 465–488, 2017.

[38] R. Spreitzer, G. Palfinger, and S. Mangard, "Scandroid: Automated side-channel analysis of android apis," in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2018, pp. 224–235.

[39] M. Strohmeier, M. Smith, D. Moser, M. Schäfer, V. Lenders, and I. Martinovic, "Utilizing air traffic communications for osint on state and government aircraft," in *2018 10th International Conference on Cyber Conflict (CyCon)*. IEEE, 2018, pp. 299–320.

[40] T. Tiwari, A. Klausner, M. Andreev, A. Trachtenberg, and A. Yerukhimovich, "Location leakage from network access patterns," in *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2019, pp. 214–222.

[41] United Airlines, Inc., "Customer data privacy policy," 2021, www.united.com/ual/en/us/fly/privacy.html#information_auto_collect.

[42] T. Watanabe, M. Akiyama, and T. Mori, "RouteDetector: Sensor-based Positioning System That Exploits Spatio-Temporal Regularity of Human Mobility," in *9th usenix workshop on offensive technologies*, 2015.

[43] Y. Xiao, Y. Jia, X. Cheng, S. Wang, J. Mao, and Z. Liang, "I know your social network accounts: A novel attack architecture for device-identity association," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1017–1030, 2022.

[44] M. Xu, C. Song, Y. Ji, M.-W. Shih, K. Lu, C. Zheng, R. Duan, Y. Jang, B. Lee, C. Qian *et al.*, "Toward engineering a secure android ecosystem: A survey of existing techniques," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–47, 2016.

[45] L. Yan, Y. Guo, X. Chen, and H. Mei, "A study on power side channels on mobile devices," in *Proceedings of the 7th Asia-Pacific Symposium on Internetware*, 2015, pp. 30–38.

[46] B. Yang, R. Chen, K. Huang, J. Yang, and W. Gao, "Eavesdropping user credentials via gpu side channels on smartphones," in *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2022, pp. 285–299.

[47] Q. Ye, Y. Zhang, G. Bai, N. Dong, Z. Liang, J. S. Dong, and H. Wang, "Lightsense: A novel side channel for zero-permission mobile user tracking," in *Information Security: 22nd International Conference, ISC 2019, New York City, NY, USA, September 16–18, 2019, Proceedings 22*. Springer, 2019, pp. 299–318.

[48] X. Zhang, X. Wang, X. Bai, Y. Zhang, and X. Wang, "Os-level side channels without procfs: Exploring cross-app information leakage on ios," in *Proceedings of the Symposium on Network and Distributed System Security*, 2018.

[49] X. Zhao, M. Z. A. Bhuiyan, L. Qi, H. Nie, W. Rafique, and W. Dou, "Trcmp: An app usage inference method for mobile service enhancement," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage: 11th International Conference and Satellite Workshops, SpaCCS 2018, Melbourne, NSW, Australia, December 11-13, 2018, Proceedings 11*. Springer, 2018, pp. 229–239.

[50] X. Zhao, M. Z. A. Bhuiyan, L. Qi, H. Nie, W. Tang, and W. Dou, "Trcmp: A dependable app usage inference design for user behavior analysis through cyber-physical parameters," *Journal of Systems Architecture*, vol. 102, p. 101665, 2020.

[51] H. Zheng and H. Hu, "Missile: A system of mobile inertial sensor-based sensitive indoor location eavesdropping," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3137–3151, 2019.

[52] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt, "Identity, location, disease and more: Inferring your secrets from android public resources," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 1017–1028.