


Article

Breaking Users' Mobile Phone Number Based on Geographical Location: A Case Study with YY

Hongzhou Yue *, Huaping Guo and Xingpo Ma 

School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China; hpguo@xynu.edu.cn (H.G.); maxingpo@xynu.edu.cn (X.M.)

* Correspondence: yuehz@xynu.edu.cn; Tel.: +86-182-0123-7593

Received: 6 July 2018; Accepted: 4 August 2018; Published: 6 August 2018



Abstract: Geographical location and mobile phone numbers are important parts of user privacy and lots of studies have been working on the privacy leakage problems of these two aspects. However, no researchers have ever studied the security problems that can be caused by the interaction between them. We show a new form of network attack in this paper by making full use of the relationship between them; namely, we try to break a user's mobile phone number with the aid of a user's geographical location that has been broken. We study the phenomenon of exposing a user's geographical location and parts of their phone number that exist in a series of popular software products, and the possibility of the user's mobile phone number to be broken is discussed. We choose one of the software (the largest entertainment webcast platform in China—YY) as the research object. First, taking advantage of a series of security vulnerabilities that exist in YY, a user's accurate geographical location is broken by the trilateration localization algorithm. Then, their mobile phone number attribution can be inferred according to their geographical location. Next, a mobile phone number test set is constructed according to the mobile phone segment allocation made by the three carriers (telecommunication operators) and the exposed parts of the user's phone number. Finally, a brute-force method is used to break the user's mobile phone number. The great effect of a user's geographical location on breaking a mobile phone number is proved by experiments, and security precaution suggestions are given at the end of the paper.

Keywords: geographical location; mobile phone number; privacy leakage; brute force; security vulnerability; webcast platform

1. Introduction

Privacy leakage has always been an important topic in the field of information security. A user's geographical location and phone number are also the type of privacy that is heavily protected by developers. However, in recent years, there have been many incidents of privacy leakage on the Internet in these two aspects. In terms of geographical location privacy, a user's geographical location is often obtained by mobile ads, a mobile operating system platform, and app (application) developers who are devoted to providing location-based functions and services [1,2]. Some researchers found that, due to the lack of effective means of protection, when some apps provide nearby users' location information for a user, the nearby users' accurate geographical location can be leaked [3–6]. Some researchers even revealed that the access control strategies of location information in many mobile operating systems are ineffective or inefficient, leading to users being faced with the threat of geographical location information being stolen by malicious apps [7].

In terms of phone number privacy, with the popularity of mobile phones, a mobile phone number is used by more and more network systems as a means of binding users' identity [8]. User privacy stored in the system database, such as users' phone numbers, is often stolen by attackers who

successfully invade the system database [9,10]. Some malicious mobile phone apps even aim at stealing phone numbers stored in the address book [11,12]. Besides, some researchers found that the use of a mobile phone number as one of the user's login credentials will face many security threats, such as the leakage of mobile phone numbers [13–15], identity camouflage [16], and the disclosure of users' identity information [17,18].

Although users' geographical location privacy and mobile phone number privacy are being paid increasing attention by researchers, few people care about the problems that may be caused by the combination of the two privacy leakage problems. In fact, as the allocation of mobile phone numbers is related to geographical location, the leakage of geographical location can help break users' mobile phone numbers. This paper first studies the exposure of geographical location and mobile phone number in some software products that are widely used by people; then, it takes the largest entertainment webcast platform in China, YY, as an example, and the role of geographic location leakage for mobile phone number breaking is shown.

The contribution of this paper includes the following three aspects:

(1) Users' geographical location privacy is effectively associated with mobile phone number privacy. A new form of network attack breaking users' mobile phone numbers with the aid of users' geographical location is proposed. We apply this method of breaking users' mobile phone number into one of the most popular entertainment webcast platforms "YY" in China, and break any YY user's mobile phone number. We extend the application scope of the traditional user location positioning method based on trilateration localization algorithm and upgrade its harm, and the harm of privacy leakage of user's geographical location is further enlarged.

(2) An effective exploration of the brute-force technique is carried out, and a more practical brute-force technique of a user's mobile phone number is proposed. Through the query of the user's mobile phone number with a mask and inferring mobile phone number attribution according to the user's geographical location, the user's mobile phone number can be reduced to a certain range, the test set of brute force is reduced, and the efficiency of breaking users' mobile phone number is increased. It makes the brute-force technique of users' mobile phone number more practical.

(3) According to the technique of breaking a user's mobile phone number based on geographical location, the corresponding security defense methods are proposed, which can be used as a reference for developers to protect users' privacy.

2. Problem Presentation

2.1. App Exposes Users' Geographical Location and Parts of Mobile Phone Number

There are many software products providing location-based services on the Internet and many of them have problems with leaking users' geographical location and mobile phone numbers [3–6]. Table 1 lists the exposure condition of user geographic location and mobile phone number in seven popular software products in China. All of them have more than 100 million users (by May 2018), so they are highly representative.

In terms of exposing a user's geographical location, there are mainly two cases. The first case is that users directly fill in their location information on their personal homepages and it is usually the information of the province, the city, etc., such as Sina Weibo, Alipay, QQ, and YY. The second case is user geographical location leakage caused by some functions of the system, such as citywide user searches and nearby user searches. These functions enable us to directly get software users in the same area as us, and the province and city in which the user is located can be regarded as the same as us. We can even make use of some software vulnerabilities to retrieve more user location information; for instance, if the server side of software does not effectively identify the coordinate camouflage behavior of the client side, we can send camouflage coordinates to the server to get more user information near different geographical locations. Besides, we can also use the method of trilateration localization [19] to calculate a user's precise geographical coordinates. The function of a

user's latest login geographical location on YY can return to the searcher the distance between the target user and the searcher. If the server side does not carry out effective defense measures, the accurate geographical location of any user can be calculated through trilateration localization method.

Table 1. The exposure of user's geographical location and mobile phone number of seven apps.

App	Ways of Exposing Geographical Location	Number of Digits Be Exposed	Ways of Exposing Mobile Phone Number
Sina Weibo	Personal homepage	6	Retrieve password
Alipay	Personal homepage	5	Retrieve password
KuGou	Citywide user search	6	Retrieve password
QQ	Personal homepage, nearby user search	3	Retrieve password
Baidu Account	Citywide user search	5	Retrieve password
360 Account	Citywide user search	6	Retrieve password
YY	Personal homepage, location of the user's recent login	5	Retrieve password, recharge for others

In terms of exposing parts of a user's mobile phone number in order to give users a hint and prevent users who are using or have used more than one mobile phone from forgetting their registered mobile phone numbers, functions of retrieving a password in many systems display parts of a user's mobile phone number to them. Column 3 of Table 1 shows the number of digits of a mobile phone number that are exposed by each app. In addition to exposing parts of a user's mobile phone number on the function of retrieving a password, YY has a similar problem on the function of recharging for others to prevent the recharge error. These exposed phone numbers are actually visible to everyone, and anyone can get parts of a user's mobile phone number by entering the user account of a specified user to the system and making use of the vulnerable functions.

2.2. Discussion on the Possibility of Breaking the Mobile Phone Number

For the seven apps introduced in Section 2.1, there might be another function in an app, and the relationship between mobile phone number and user ID can be obtained, as the two cases shown in Figure 1. In the first case, the client sends a mobile phone number to the server, and the server can return the corresponding user ID. In the second case, for a specified user ID, the client sends the user's possible phone number to the server to verify its correctness, and the server returns information to tell the client whether the phone number is correct. Once there is such a function, if the server side lacks effective restrictions, we can use the enumeration method (brute-force method) to try a variety of possibilities of the user's mobile phone number. This process can be done manually or automatically, and the exposed parts of a mobile phone number can undoubtedly help us reduce the scope of trial attempts.

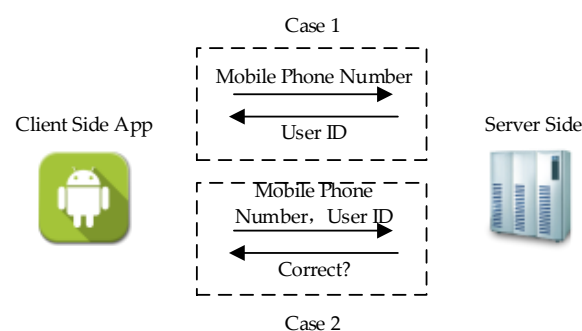


Figure 1. The leakage of the relationship between mobile phone number and user ID.

Without the exposure of mobile phone numbers, the number of attempts needed to break a user's mobile phone number follows Equation (1). There are only limited possibilities for the first three digits of the mobile phone number, as shown in Table 2; therefore, there is no need to make 1000 attempts for the first three digits of a mobile phone number.

$$\text{Number of attempts} = \{\text{Possible combinations of the first three digits}\} \times 10^8 \quad (1)$$

If the exposed digits are the first three digits of the mobile phone number, then the number of combinations of the first three digits of the mobile phone number that needs to be explored can be reduced. However, if n digits are exposed in the last eight digits of the mobile phone number, the number of attempts needed to break a user's mobile phone number follows Equation (2).

$$\text{Number of attempts} = \{\text{Possible combinations of the first three digits}\} \times 10^{(8-n)} \quad (2)$$

Some developers may think that the digits of mobile phone number are so much, and even if some digits are exposed, the cost of breaking a mobile phone is very high, which keeps attackers away. However, they ignore another problem; namely, in addition to the exposed parts of mobile phone number, the leaked user geographical location can also help an attacker to speculate on some digits of a mobile phone number, and reduce the test set of a mobile phone number for brute force.

Table 2. Distribution of the first three digits of mobile phone number of the three carriers.

Carrier	First Three Digits of Mobile Phone Number
China Unicom	139, 138, 137, 136, 135, 134, 159, 158, 157, 150, 151, 152, 147, 188, 187, 182, 183, 184, 178
China Mobile	130, 131, 132, 156, 155, 186, 185, 145, 176
China Telecom	133, 153, 189, 180, 181, 177

Let us take a look at the structure of the mobile phone number, as shown in Figure 2. The first three digits indicate carriers, and the fourth to seventh digits indicate the mobile phone number attribution (prefecture-level or above-level city). For instance, the first seven numbers, “1309636”, are the section numbers of China Unicom of Chengdu. “130” is the carrier section number, which indicates that this section number belongs to China Unicom. “9636” is the location-related number, which indicates that this section number belongs to Chengdu, China. If we know the geographical location of the owner of a mobile phone number, we can approximate that the owner's common geographical location is the attribution of the mobile phone number, so we can basically determine the range of the fourth to seventh digits of the mobile phone number.

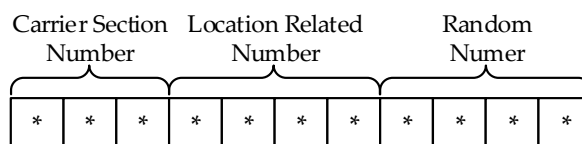


Figure 2. Structure of the mobile phone number in China.

Some online inquiry websites for the attribution of mobile phone numbers can provide assistance to mobile phone number breaking. When we provide the first seven digits to some of the websites, we can get the attribution of a mobile phone number at the prefecture-level city or above. We can use the method of a web iterative query, downloading the information regarding all possibilities of the first seven digits and their corresponding attribution, and setting up a “section number and city belong” (SNCB) table.

Therefore, according to the leaked user's geographical location, by querying the SNCB table, we can limit the user's first seven mobile phone number digits to a certain extent and construct the test

set of the first seven digits (TS-7D). The construction process of TS-7D is shown in Figure 3. Besides, if the exposed parts of a user's mobile phone number belong to the first seven digits, the test set of the first seven digits can be further reduced.

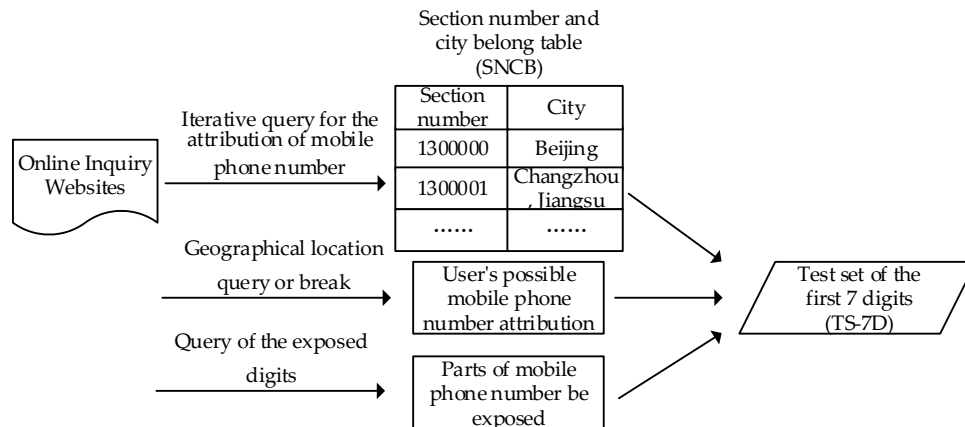


Figure 3. The construction process of the test set of the first seven digits (TS-7D).

After the test set of the first seven digits is constructed, the number of attempts needed to break a user's mobile phone number follows Equation (3); n in the equation represents the exposed number of digits in the last four digits.

$$\text{Number of attempts} = \{\text{Possible combinations of the first 7 digits}\} \times 10^{(4-n)} \quad (3)$$

As a result, the number of trial attempts to break the target user's mobile phone number is greatly reduced, and we can use the method shown in Figure 4 to break it. All the possibilities of the 11 digits are constructed by combining the seven digits in TS-7D with the exposed digits in the last four digits. Each 11-digit mobile phone number is sent to the server side; if one of them is correct, then the user's mobile phone number is successfully broken and the break work is over. Otherwise, if the user's mobile phone number is not successfully broken until all the possibilities of the 11 digits are tested, we take the following measures: For Case 1, we can build a collection for the user IDs returned by the server, one of which may be a user's auxiliary account ID, and we can use the method of auxiliary account breaking to break the user's mobile phone number (we will introduce this method in Section 5.3). For Case 2, it means that the break work has failed, and it may result from the wrong judgment of the attribution of the mobile phone number.

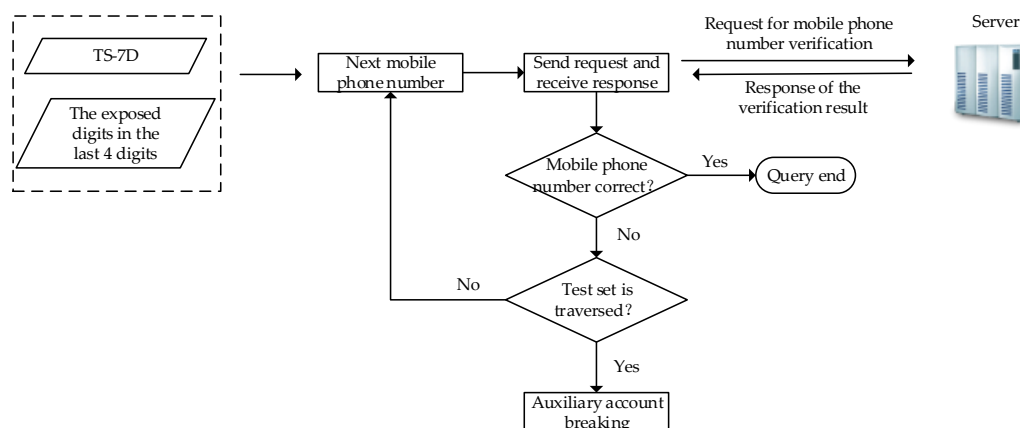


Figure 4. The process of breaking a user's mobile phone number.

In the following sections, we will take the largest entertainment webcast platform in China, YY, as an example, showing the great role of geographic location leakage for mobile phone number breaking.

3. Background Knowledge of YY

3.1. Usage of YY

The current webcast platforms in China can be divided into five types (<http://server.zol.com.cn/629/6296408.html>): entertainment webcast, news information webcast, education webcast, life webcast, and vulgarization webcast that is unpopular but exists. Among them, entertainment webcasts are the most popular and fastest-growing type of webcast at present, and can be mainly divided into two kinds—talent webcast and game webcast. YY is currently the largest entertainment webcast platform in China, and the number of registered YY users has reached 1 billion (This number does not represent 1 billion people, because the same person may use multiple YY auxiliary accounts.), while the monthly active users reached 122 million (<https://www.phb123.com/keji/hulianwang/8860.html>). YY users include two types, webcaster and audience, and because the security issues proposed in this article are applicable to all YY users, we do not make a strict distinction between webcaster and audience in the later description.

3.2. Channel and User Identity Level

A YY channel is a webcaster's webcast room, and for each YY channel, the identity of YY users on this channel can be classified into different levels. Users of different identity levels have different authorities on this channel, which include user-management authority and voice-management authority. The webcaster is the owner of the corresponding YY channel with the highest authority of user management in the channel. The channel owner can set some users as administrator users, who can manage the nonadministrator users. According to the difference of management authority, the administrator user can also be divided into the channel general administrator, the full-channel administrator, and the subchannel administrator. The nonadministrator user can also be divided into many identity levels, including member, guest, and tourist.

Each YY user can have different identity levels in multiple YY Channels, and a user's identity level in each YY channel can be queried by any other user, which is not part of personal privacy. However, a user's channel information can approximately describe user's preferences. Therefore, in Section 5.3, we will use the method of channel matching to break a user's auxiliary account.

3.3. YY Account

Users can watch YY webcasts through three platforms: PC-end software, mobile apps, and webpages. All of these three platforms provide login and registration interfaces, and YY user login and registration methods are shown in Table 3. As can be seen from Table 3, though there are many kinds of login methods for YY platforms at present, the main registration method is mobile phone number registration, and the registration interface of mobile phone number is the first registration interface of each platform displayed to the user. The account registered by mobile phone number can log into YY by that mobile phone number, and the account registered by username or other information cannot log in YY by the mobile phone number. Because mobile phone numbers are easy to remember, most YY users will choose their mobile phone numbers to register and log in to YY (we will prove this conclusion in the experiment in Section 7.3).

Many YY users will register more than one YY account; it is shown in Section 3.1 that YY's registered users reach 1 billion, and it is also a reflection of many YY users' registration of multiple YY accounts. The registration of a YY account needs a mobile phone verification code. In addition, in order to secure the YY account, users are asked to bind their YY accounts with their mobile phone numbers.

A mobile phone number can be used to apply for and bind up to 10 YY accounts. This convenience also encourages users to apply for more YY accounts.

Table 3. YY user login and registration methods.

Platform	Login Method	Registration Method
PC-end Software	YY account number, mobile phone number, username, email	Mobile phone number, username, email
Mobile App	YY account number, mobile phone number, Username, email, Weibo, WeChat, QQ, mobile phone verification code	Mobile phone number
Webpage	YY account number, mobile phone number, username, email, Weibo, WeChat, QQ	Mobile phone number, email

Although a user may have multiple YY accounts, usually only one account is used. We call a user's commonly used YY account the main account, while the noncommonly used YY account is called the auxiliary account. In order to facilitate study and presentation in the following sections, the YY account of a user to be broken is called the main account of this user, and the other accounts owned by the user are called the auxiliary accounts (this definition does not affect the break work of a user's geographical location and mobile phone number).

4. Vulnerability Details

4.1. Leakage of Users' Geographical Location

When we use the YY mobile phone app to enter a YY user's personal homepage, we will find the user's latest login location and login time displayed on their homepage, as shown in the left half of Figure 5. However, this login location and time only correspond to a YY mobile app login, not to a PC or webpage login. In order to protect user privacy, YY has some reservations about the information displayed on the user interface. For instance, when the inquirer is far away from the queried user, only the province where the queried user is located is shown in the personal homepage, while when the inquirer is not far from the queried user, distance data (distance between the inquirer and the queried user) with a precision of 10 m are shown. But we found through reverse analysis and network-packet analysis that, regardless of the distance between the inquirer and the queried user, the server will return the distance data with a precision of 1 m to the app client. This process is shown in Figure 6.

In addition, the "nearby webcaster" function of the YY mobile app also has the problem of exposing distance data, and the principle of which is similar to the "user's latest login location" function, as shown in the right half of Figure 5. When a webcaster begins broadcasting on the YY mobile phone app, their geographical location coordinates will be sent to the server side. When the audience uses the "nearby webcaster" function to query the nearby webcasters, the server uses these coordinates to calculate the distance between the webcaster and the audience, and sends all the webcaster information near the audience, which includes the distance between the webcaster and the audience to the client app of the audience. Although the mobile app interface shows distance data with a precision of 10 m, we found the precision of the distance data returned to the mobile app by the server side is 1 m by reverse analysis and network-packet analysis.

By our test, we found that there is geographical location leakage vulnerability in the function of "user's latest login location" as well as the function of "nearby webcaster", and the method of trilateration localization can be used to calculate the exact location coordinates of any YY user. The concrete realization method is that when we query the geographical location information of a YY user, we change the client coordinates that sent to the server side three times, and the distance data returned by the server is received. Then, the three client coordinates and the corresponding distance data are used as the input of the trilateration localization algorithm, and then the geographical location

coordinates of the queried user can be calculated. Because the YY server lacks distance ambiguity measures and preventive measures for client coordinate camouflage, the user's geographical location can be easily leaked.

Besides that, the province and city where the user is located can also be retrieved by querying the location information (province, city) filled in by the user in their personal homepage. Whether a user's geographical location is gotten by the calculation of trilateration localization, or by querying the location information shown in the user's personal homepage, it can be used as a source to deduce user's mobile phone number from the fourth to seventh digits.

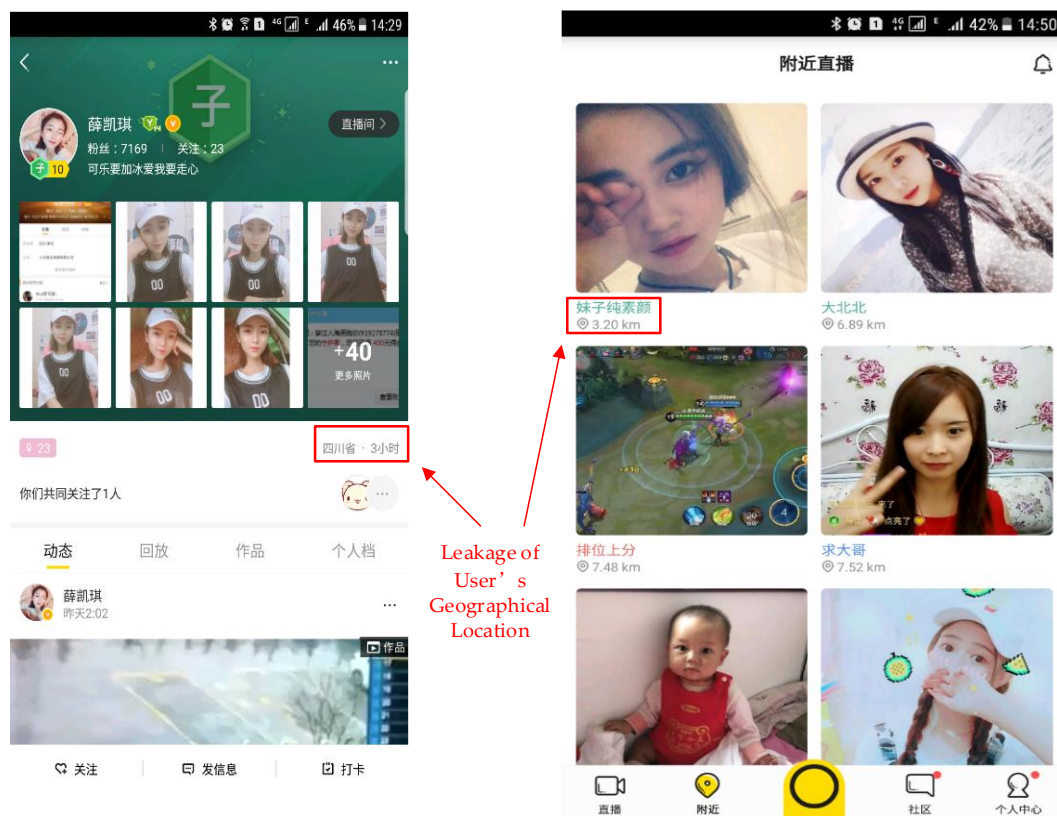


Figure 5. Interface display of leakage of user's geographical location.

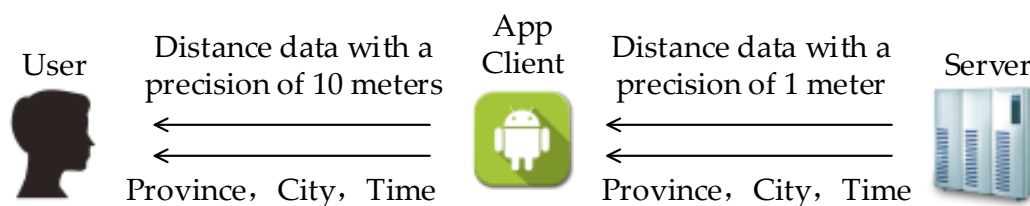


Figure 6. Displayed precision and actual precision of distance data.

4.2. Mobile Phone Number with a Mask

Both YY's "recharge for others" and "retrieving password" functions have the problem of leaking parts of a user's mobile phone number (e.g., "130*****88"), and the exposed five digits are parts of a user's password-protection mobile phone number. For the YY account registered with a mobile phone number, its password-protection mobile phone number is usually the registered mobile phone number. We can get five digits of any YY user's mobile phone number by these two function modules, and in Section 2.2 we learned that with the help of the exposed five digits, the attacker can effectively reduce

the test set of a mobile phone number and reduce the workload of mobile phone number breaking. The user interfaces of these two function modules are shown in Figure 7.

YY's web recharge function can be used to recharge for others. After entering the YY account number and the recharge amount, and sending a recharge request to the server, the server will return some information of the user, such as the password-protection mobile phone number with a mask and the email, to let the user confirm. We speculate that the purpose of this design is to prevent users from charging errors.



Figure 7. User interfaces of leakage of parts of a user's mobile phone number.

YY's password-retrieving function currently only supports retrieving a user's password by the mobile phone verification code. After entering the YY account number and sending a password-retrieval request to the server, the server will return the mobile phone number (password-protection mobile phone number bound by the YY account number) with a mask. Only after the user enters the complete mobile phone number does the server send the SMS verification code to the user. We speculate that the purpose of this design is to prevent the user from maliciously sending a password-retrieval request, causing the server to bear the cost of sending the SMS, and causing the user to be disturbed and misled by the wrong SMS.

4.3. Leakage of the Relationship between Mobile Phone Number and User ID

After getting a user's geographical location and parts of their mobile phone number, what we are missing is a function that can be used to get the relationship between mobile phone number and user ID, introduced in Section 2.2; this function we successfully found in YYPay.

YYPay is an online payment platform providing YY users with online recharge, transaction management, online payment, and other services. Any YY user can log into YYPay in the same way as logging into YY. However, we found through reverse analysis and network-packet analysis that when we log into YYPay with a mobile phone number, if this mobile phone number had been registered by a YY user (this user can log into YY or YYPay with this registered mobile phone number), the server will return this user's ID to the app client of YYPay whether the password is correct or not—the process is shown in Figure 8. If the login password is correct, the server will return the user ID and login credentials to the client to indicate the user's legitimate identity. If the password is wrong, the server will only return the user ID to the client. After the client receives the user ID, it will then send the error report to the server. In addition, if this mobile phone number had not been registered by any YY user, the server will not return the user ID to the client. When the login attempts reach a certain number of times, the server will return a verification code to prevent a password brute-force attack. However, the server returns the user ID while returning the verification code, so the verification code

can only prevent a user's password from being broken, but cannot effectively prevent the leakage of relationship between mobile phone number and user ID.

As a result, Case 1 mentioned in Section 2.2 appears; that is, the client sends a phone number to the server and the server can return the corresponding user ID. Then we need to see if there is effective access restriction on the server. By our test, we found that there was login number restriction vulnerability in the login function module of YYPay, which can be used to make numerous login attempts. The server side has eight IPs that can accept and process login requests from the client, and we speculate that the purpose of this design is to disperse the request data from the client and reduce the load of the server side. When a certain number of requests are sent to one of the IPs, the server returns a hint of "IP risk" and refuses to process the login request, but it is quickly recovered in a very short time. What is more, when we send requests to the server with the method of eight server IP rotation, the situation of the server refusing to process login requests becomes less likely.

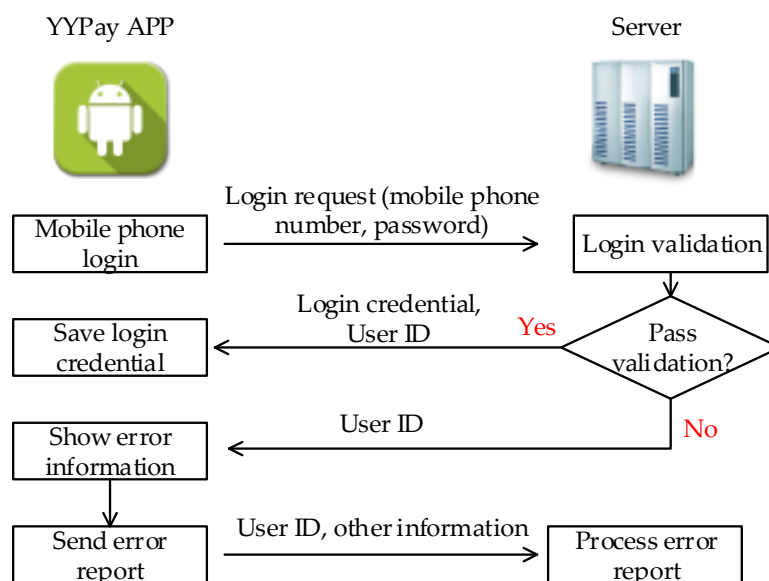


Figure 8. Login model of YYPay.

Therefore, all the conditions mentioned in Section 2.2 that are needed to break a user's mobile phone number based on geographical location are satisfied. The test set of mobile phone number can be constructed by making use of a user's exposed geographical location and parts of their mobile phone number. We can use the method of brute force to frequently send requests with each mobile phone number in the test set to the server to break a YY user's mobile phone number.

5. Breaking Method

5.1. Overall Breaking Process

The overall process of breaking a YY user's mobile phone number is shown in Figure 9. First, according to the introduction of Section 4.1, a user's geographical location can be either retrieved by the calculation of trilateration localization (making use of the geographical location leakage vulnerabilities in the functions "user's latest login location" and "nearby webcaster"), or by querying the location information shown in the user's personal homepage. Then, we approximated that the YY user's common geographical location is the attribution of their mobile phone number. By looking up the SNCB table, we constructed the TS-7D, and with the help of an exposed mobile phone number with a mask, the TS-7D was expanded to build a test set of 11 digits of a mobile phone number. Finally, by making use of the vulnerability of the leakage of relationship between mobile phone number and user ID, we used the method of brute force to send requests with each mobile phone number in the test

set to the server and have the user ID returned from the server. If the returned user ID is the user ID of what we queried, then the user's mobile phone number is successfully broken. Otherwise, we used the method of auxiliary account breaking based on channel matching and geographical location matching to find out if the user ID was the auxiliary account number of the user, and, if it was, the user's mobile phone number could also be broken.

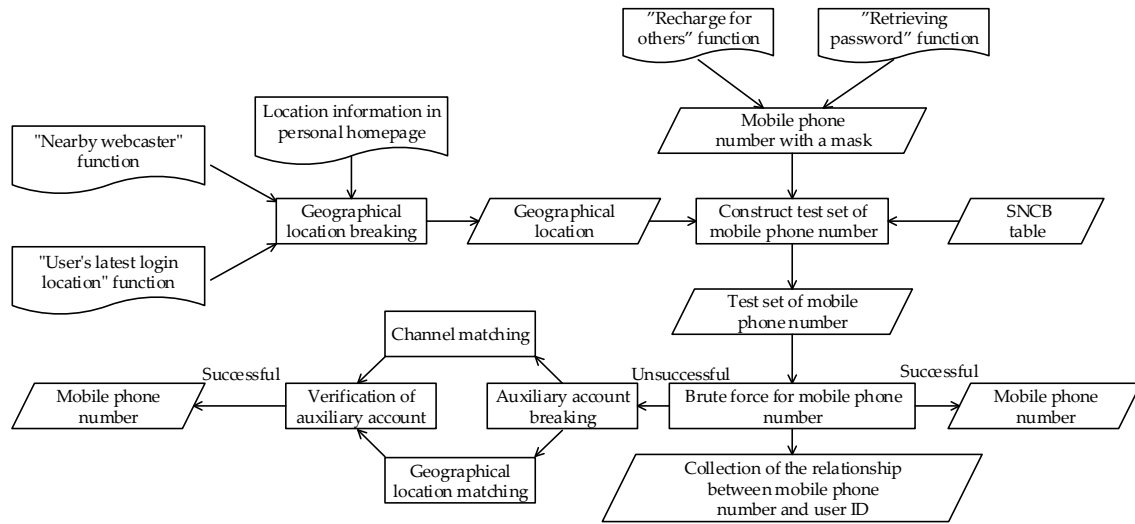


Figure 9. Overall process of breaking a YY user's mobile phone number.

5.2. Location Calculation

By our test, we found that the YY server uses the spherical distance-calculation method to calculate the spherical distance between two coordinates. We briefly introduce the distance-calculation method through the example shown in Figure 10.

Point B in Figure 10 represents a pole of the Earth, e.g., the North Pole. $A_1(m_1, n_1)$ and $A_2(m_2, n_2)$ are two points on the sphere, and m and n represent latitude and longitude, respectively. Point O represents the center of the Earth, and the radius of the Earth is expressed by R . According to the cosine formula, Equation (4) can be obtained:

$$\angle A_1OA_2 = \arcsin \left(\frac{\sin(m_2) \times \sin(m_1) + \cos(m_2) \times \cos(m_1) \times \cos(n_2 - n_1)}{\cos(m_1) \times \cos(n_2 - n_1)} \right) \quad (4)$$

Therefore, the distance l_{12} between A_1 and A_2 can be expressed as Equation (5):

$$l_{12} = \frac{\angle A_1OA_2}{180} \times \pi \times R \quad (5)$$

For the spherical distance-calculation method used by YY, the location coordinates of the target point can be calculated by the method of trilateration localization when the coordinates of the three points and the distances to the target point are known. We use D to represent the target point, the coordinates of the three points are $A_1(m_1, n_1, l_1)$, $A_2(m_2, n_2, l_2)$ and $A_3(m_3, n_3, l_3)$, and the l represents the distance to the target point.

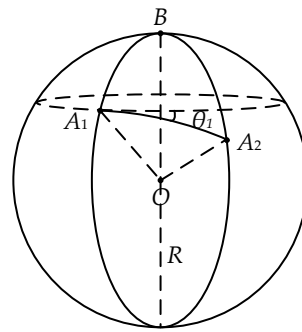


Figure 10. Example of the spherical distance-calculation method.

In fact, only two of the three points can be used to lock the target point in two places. As shown in Figure 11, according to symmetry, we can determine the two possible locations of point D — $D(m_d, n_d)$ and $D'(m'_d, n'_d)$. Therefore, we can first calculate D and D' by A_1 and A_2 , which are verified and chosen by A_3 .

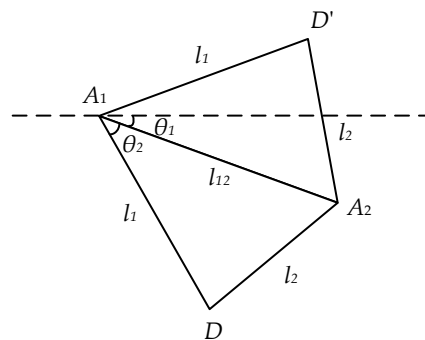


Figure 11. Location calculation based on symmetry.

The dotted line in Figure 11 represents the latitude line passing through A_1 , θ_1 represents the deviation angle of A_2 relative to A_1 on the latitude line, and θ_2 represents the angle between A_1A_2 and A_1D . The calculation steps are as follows:

(1) Calculating the value of θ_1 . For ease of expression, we may consider that A_1 , A_2 , and A_3 in Figure 11 correspond to A_1 , A_2 , and A_3 in Figure 10. According to the spherical sine formula, Equation (6) can be obtained:

$$\frac{\sin \angle A_2OB}{\sin \angle A_2A_1B} = \frac{\sin \angle A_1OB}{\sin \angle A_1A_2B} = \frac{\sin \angle A_1OA_2}{\sin \angle A_1BA_2} \quad (6)$$

Therefore, we can calculate the value of θ_1 by $\angle A_1OA_2$, as shown by Equations (7) and (8):

$$\angle A_2A_1B = \arcsin \left(\frac{\cos(m_2) \times \sin(n_2 - n_1)}{\sin \angle A_1OA_2} \right) \quad (7)$$

$$\theta_1 = |\angle A_2A_1B - 90| \quad (8)$$

(2) Calculating the value of θ_2 . By knowing the coordinates of A_1 and A_2 , the spherical distance l_{12} between A_1 and A_2 can be calculated. Then, the value of θ_2 can be obtained by the cosine theorem, as shown by Equation (9).

$$\theta_2 = \arccos \left(\frac{l_1^2 + l_{12}^2 - l_2^2}{2l_1l_{12}} \right) \quad (9)$$

(3) Calculating the value of $D(m_d, n_d)$ and $D'(m'_d, n'_d)$, as shown by Equations (10) and (11):

$$\begin{cases} \Delta m = \frac{l_1 \cos(\theta_1 + \theta_2) \cdot 180}{R \cos(m_1 \cdot \frac{\pi}{180}) \cdot \pi} \\ \Delta n = \frac{l_1 \sin(\theta_1 + \theta_2) \cdot 180}{R \cdot \pi} \\ m_d = m_1 + \Delta m \\ n_d = n_1 + \Delta n \end{cases} \quad (10)$$

$$\begin{cases} \Delta m' = \frac{l_1 \cos(\theta_1 - \theta_2) \cdot 180}{R \cos(m_1 \cdot \frac{\pi}{180}) \cdot \pi} \\ \Delta n' = \frac{l_1 \sin(\theta_1 - \theta_2) \cdot 180}{R \cdot \pi} \\ m'_d = m_1 + \Delta m' \\ n'_d = n_1 + \Delta n' \end{cases} \quad (11)$$

(4) Determining the target point in D and D' . Taking $A_3(m_3, n_3, l_3)$ as a reference point, according to the calculation method of spherical distance, the point which is closer to l_3 in D and D' is the target point.

5.3. Mobile Phone Number Breaking and Auxiliary Account Breaking

We can get the first three and the last two digits of any YY user's mobile phone number due to the leakage problem of mobile phone numbers with a mask, introduced in Section 4.3. If we use the method of brute force to break a user's mobile phone number, we need to loop through the middle six digits, which requires up to one million cycles. However, as we take the method of breaking a user's mobile phone number based on their geographical location, the scale of the test set can be greatly reduced. According to the introduction in Section 2.2, we can construct the TS-7D by querying the SNCB table, and because the last two digits of a user's mobile phone number are exposed, for each record in TS-7D we only need to iterate through the eighth and ninth digits, which requires up to 100 cycles. For instance, Figure 12 shows the construction schematic of the test set of the masked phone number "130*****88". Therefore, the test set of 11 digits of a mobile phone number (TS-11D) is constructed by extending each record in TS-7D for 100 times.

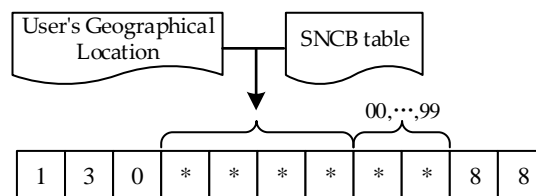


Figure 12. Construction schematic of the test set of mobile phone number.

We sent login requests to the server with each mobile phone number in TS-11D, getting the user ID returned by the server, and saved the collection of the relationship between mobile phone number and user ID locally (we use M to represent this collection for convenience of description). If the returned user ID is the ID of the queried YY user, the mobile phone number is successfully broken. Otherwise, with the help of collection M , we use the method of auxiliary account breaking to query whether the user ID returned by the server is an auxiliary account of the queried YY user.

The method of auxiliary account breaking we use includes channel matching and geographical location matching, and the detailed methods are shown as follows:

(1) Channel Matching

The channel-matching method is based on the principle that the YY channels that the same user likes to visit have some similarities. According to the introduction of Section 3.2, each YY user can have different identity levels in multiple YY Channels, and a user's channel information can approximately

describe a user's preferences. Therefore, if two accounts frequently visit the similar YY channels and even have the same identity levels on some specific channels, we think the two accounts are likely to be owned by the same user, and they are the relationship between main account and auxiliary account.

A YY user's identity level information in each YY channel is shown in a user's personal homepage, which is not part of personal privacy and can be queried by any other user. We use $G_i = \{g_1, g_2, \dots, g_n\}$ to represent the list of channels shown in the personal homepage of YY account S_i . For a channel g_i , the identity level of S_i on channel g_i is represented as $m_i = m_G(g_i)$. The list of identity levels corresponding to S_i is $\{m_1, m_2, \dots, m_n\}$. For YY accounts S_1 and S_2 , we take the intersection $G_1 \cap G_2$ of their corresponding channel lists, and for channel $g_i \in G_1 \cap G_2$, we represent the channel similarity between G_1 and G_2 on the common channel g_i as L_{g_i} ; the value of the channel similarity is shown as Equation (12):

$$L_{g_i} = \begin{cases} a, & m_{G_1}(g_i) = m_{G_2}(g_i) \\ b, & m_{G_1}(g_i) \neq m_{G_2}(g_i) \end{cases} \quad (a > b) \quad (12)$$

Equation (12) indicates that, if G_1 and G_2 have the same identity level on the common channel g_i , channel similarity is a , and if the identity levels are different, channel similarity is b . The value of a and b is set manually, and the value of a must be greater than b . Therefore, we can get the account similarity between YY account S_1 and S_2 , shown as Equation (13):

$$L_{G_1 G_2} = \sum L_{g_i}, \quad g_i \in G_1 \cap G_2 \quad (13)$$

We took each account in order from collection M , querying its channel information, calculating its channel similarity with the queried user account, and taking out several YY accounts with the highest similarity as the auxiliary account to be verified. Finally, we verified the correctness of the auxiliary account by the retrieving-password function shown in Figure 7 (one step is asking the user to enter their complete mobile phone number).

(2) Geographical Location Matching

Geographical location matching is based on the principle that the latest-login geographical locations of the same person have some similarities. We first use the trilateration localization method introduced in Section 5.2 to calculate the latest-login geographical location coordinates of the queried YY account. Next, we disguise the coordinates of the client as these coordinates, and for each YY account in collection M , we, in turn, send a request to the server to query the distance between the YY account's latest login location and the client. Then, we sort the distances corresponding to each YY account from small to large, and a few YY accounts with the smallest distances will be taken as the auxiliary accounts to be verified. Finally, we verify the correctness of the auxiliary account by the retrieving-password function shown in Figure 7.

However, not all auxiliary accounts satisfy the similarity we define; for instance, a YY user's auxiliary account may not have any channel information, its channel information may be very different from the main account, or the geographical location of a user when logging into YY by an auxiliary account is different from the main account. As a result, our auxiliary account breaking will be unsuccessful in these cases. Therefore, the similarity we define here simply describes a universal rule and this rule can be used to break a YY user's mobile phone number as much as possible. Besides, by our test, only five mobile phone numbers at most are allowed to be lost every day to verify the correctness of the auxiliary accounts by the retrieving-password function, so we cannot build the test set of more than five auxiliary accounts to be verified, which may also cause missing a user's real auxiliary account, resulting in the failure of mobile phone number breaking.

6. Technical Implementation

In this section, we will introduce the main technical implementation based on the breaking method introduced in Section 5, and the main implementation codes that were uploaded to Github

(<https://github.com/yuehongzhou/YYCrack>). Though the vulnerabilities have been repaired by the manufacturers, we believe that these codes can still provide a reference to researchers who are studying similar security problems.

6.1. Login Process Simulation

The login process of YYPay can be successfully analyzed by the reverse-analysis tool JEB (<https://www.pnfsoftware.com/>), and we simulated this login process with Java. According to the introduction of Section 4.3, when we log into YYPay with a mobile phone number, if this mobile phone number has been registered by a YY user, the server will return this user's ID to the app client of YYPay whether the password is correct or not. Therefore, we use randomly generated passwords in our login attempts, because we do not need to pay attention to the correctness of the password.

We also introduced the problem of login number restriction in Section 4.3. The server side has eight IPs that can accept and process login requests from the client, and when a certain number of requests are sent to one of the IPs, the server returns a hint of "IP risk" and refuses to process the login request, but it is quickly recovered in a very short time. For instance, Figure 13 is a time-series diagram of the login attempts of our login process to log into a single-server IP within 10 min. The horizontal axis represents the time segment, the blue sequence represents when the server processed client requests normally, and the red sequence represents when the server refused to process a client's request. We found that when login requests are frequently sent to the server in a specific time, the server presents a discontinuous state to process the requests, and the time of normally processing the request and that of refusing to process the request are roughly equal. In order to facilitate analysis and comparison, the login process used a single thread, and the experiments in the subsequent chapters regarding time and performance of the login process were also based on the single thread of the login process.

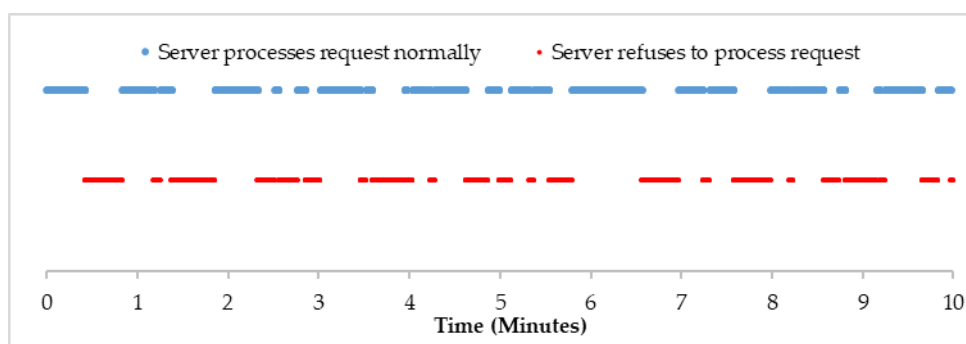


Figure 13. Time-series diagram of the login attempts of a single-server IP.

In order to improve login efficiency, we adopted the login method of 8 server IP rotation. When a server IP refuses to process the login request, we change to the next server IP. The experimental results show that this method can significantly improve the efficiency of the login attempts. Figure 14 is the time-series diagram of the login attempts using the method of 8 server IP rotation, and from the figure we can see that the cases of the server refusing to process the request are rarely happening.

In order to verify that, with the growth of continuous login time, our simulated login process would not be affected by the login restriction measures of the YYPay server, we extended the time of login attempts to 1 h to count the number of successful and unsuccessful login attempts, and observe the relationship between the success login rate and the time. The experimental results are shown in Figure 15. In our experiment, we divided the total time into 6 periods, counting the number of successful and unsuccessful login attempts in each period. As can be seen from Figure 15, the successful login rate is not related to the continuous login time, which indicates that the server did not take effective measures to restrict the login behaviors of the client. Therefore, our breaking process of mobile phone number would not fail due to the long-time login attempts.

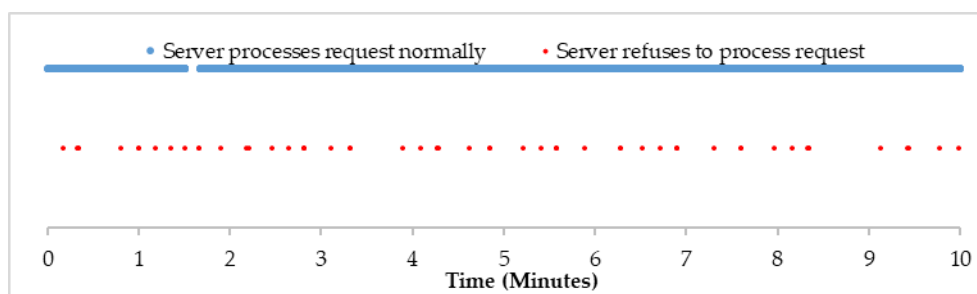


Figure 14. Time-series diagram of the login attempts of 8 server IP rotation.

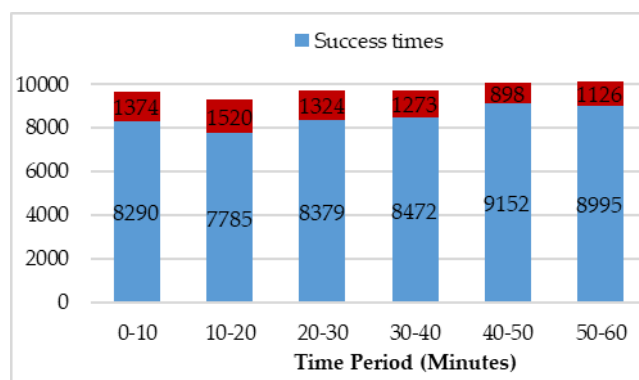


Figure 15. Statistics of the login effect within one hour of 8 server IP rotation.

Within 1 h, the login request was successfully processed 51,073 times, unsuccessfully processed 7515 times, and the number of successful times accounted for 87.17% of the total login attempts. An average of 851.22 mobile phone numbers can be tested within a minute, and according to this proportion, one million mobile phone numbers would take about 19.58 h to complete login attempts. That is to say, without using a user's geographical location to assist in mobile phone number breaking, it would take up to 19.58 h to break the mobile phone number of any YY user. However, we think this time is still very long and this breaking method is low in applicability, while the use of a user's geographical location to assist mobile phone number breaking would significantly improve the speed of the breaking, which we will verify in Section 7.3 in detail.

6.2. Information Query Technique Based on Method-Hook

In Section 6.1, we introduced that the entire login process of YYPay can be simulated by writing an independent login process, but some of the modules of YY are difficult to simulate because of the difficulty of reverse analysis. For instance, in order to prevent reverse analyzing or improve the efficiency of the program, the developer uses native code to write some core programs in the Android mobile phone app, and it is difficult to reverse and simulate the native code. Therefore, in order to automatically query some information from YY, we use the technology of a method-hook to invoke YY's own program and achieve our purpose of information inquiry. For instance, we use the information-query technique based on method-hook for the YY user's channel information query and the latest login-location information query, and implement the method-hook in the Android version of YY App based on the Xposed framework.

For a YY user's channel information query, part of the program logic before the method-hook is shown in the solid part of Figure 16. The "sendRequest" method in the Delvik layer calls the "request-send" function in the native layer through the JNI mechanism, and the "request-send" function is responsible for data encryption and sending requests to the server. After receiving the reply

from the server, the “response-receive” function in the Native layer decrypt the reply and passes the decrypted information to the “onEvent” method in the Delvik layer through the JNI mechanism.

The method-hook technique we adopted is shown in Figure 16 and the dotted line represents the new program logics brought by method-hook. We hooked the “sendRequest” method so that the app could read a user ID from the user ID set (user IDs that may be auxiliary account IDs of a user ID to be tested) stored in the SD card of the mobile phone when the “sendRequest” method is called, and query the channel information corresponding to the user ID. We also hooked the “onEvent” method so that after receiving the channel information corresponding to a user ID returned by the server, the channel information could be stored in the SD card.

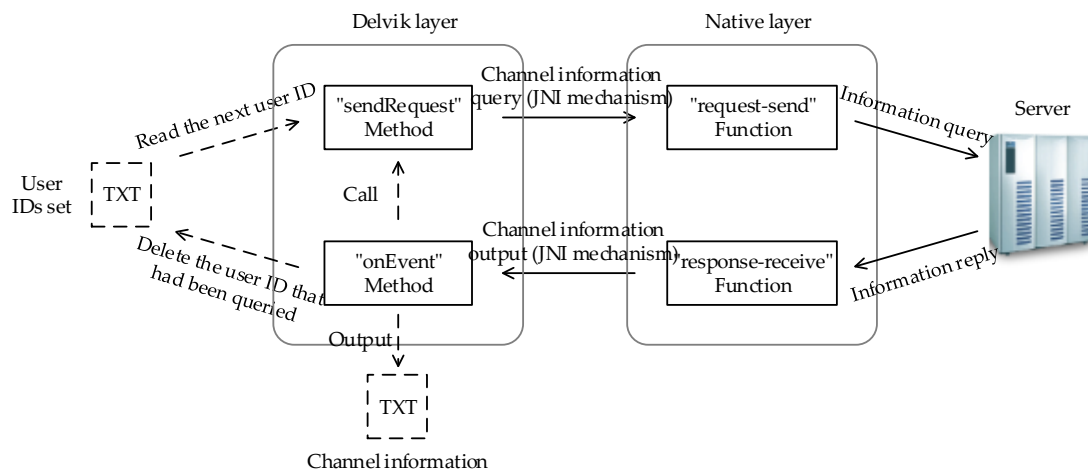


Figure 16. Model of channel information query.

For a YY user’s latest login location information query, we adopted the similar method-hook technique. The camouflaged client-coordinate file and the queried user ID file are stored in the SD card, and through hooking the corresponding methods inside the YY app, the program of the YY app could read the camouflaged coordinates and the user ID to be tested, and send the request to the server to realize the information query of the user’s latest login location. The concrete process will not be explained in detail here.

6.3. Implementation of Trilateration Localization

In Section 6.2, we mentioned that we used the method-hook technique to query a YY user’s latest login location information, but in Section 4.1 we also introduced that the data returned by the server only included the user’s latest login time and the distance data, and not the accurate location coordinates of the target YY user. Therefore, we had to calculate the accurate location coordinates of YY users according to the trilateration localization algorithm introduced in Section 5.2.

According to the introduction of Section 5.2, one time of calculation of trilateration localization needs at least three coordinates and their distances to the target point. Therefore, before each trilateration localization calculation, our location-calculation program had to change the camouflaged client coordinates stored in the SD card three times, and send requests with the camouflaged client coordinates to the server to get the distance data.

However, by our test, we found that one time of trilateration localization calculation is often inaccurate to locate a target user; for instance, Figure 17 shows the variation of the localization error with the increase of the average distance between the three points and the target point. We carry out experiments in 10 distance sections, and at every distance, we do at least 100 times of trilateration localization and calculate the average localization error.

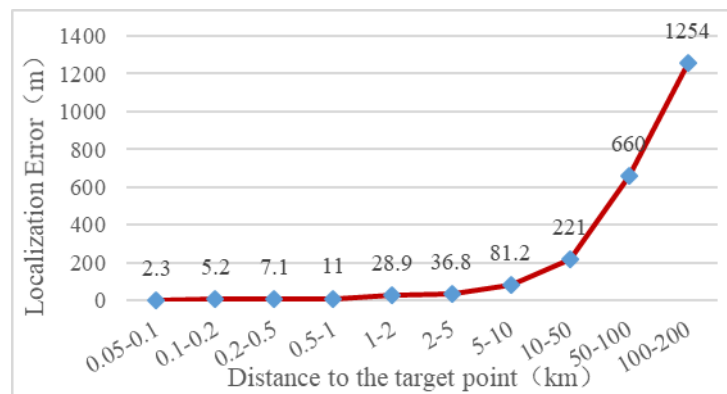


Figure 17. The relationship between distance and localization error.

The results of the experiment show that the greater the distances between the three points and the target point, the greater the localization error of the trilateration localization; the main reason for the localization error is the deviation of distance used in trilateration localization (we do not analyze the detailed reasons here). Therefore, in order to locate the target user more accurately, we adopt more trilateration localization attempts, gradually reducing the distance to the target user. The concrete process is shown in Figure 18. After the distance Dis_i corresponding to probe point coordinates Coo_i is obtained, the coordinates of the next probe point Coo_{i+1} are obtained by shifting Coo_i to a certain random direction for the distance of the $Dis_i/10$. When three probe points and their distance to the target point are obtained, we do the calculation of trilateration localization. The result coordinates $ResultCoo$ of each trilateration localization calculation is used as the initial coordinates of the next trilateration localization calculation. We set up a threshold $errorDistance$ for distance deviation: when the distance between a point (probe point or result point) and the target point is less than $errorDistance$, we consider that this point is the target point, and the target YY user is located successfully.

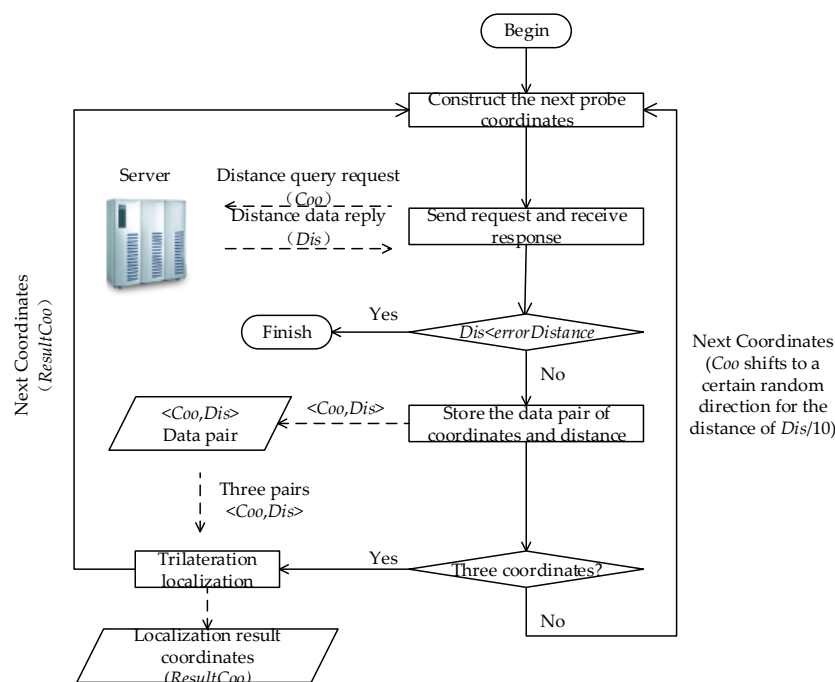


Figure 18. Flowchart of trilateration localization.

7. Breaking Effect

In this section, we will introduce the effect of breaking a YY user's mobile phone number based on geographical location. All of the test sets and experimental results were uploaded to Github (<https://github.com/yuehongzhou/YYCrack>) to provide a reference for the researchers who are interested in our work. In order to protect users' privacy, we kept some sensitive data, such as location coordinates and mobile phone numbers, secret.

7.1. Test Set Construction

In order to verify the validity of breaking a user's mobile phone number based on geographical location, we randomly selected 100 YY webcasters to test. The reason why we chose the webcasters as the test object was that they have higher public attention, and the consequences of their geographical location and mobile phone number being broken are more serious. In addition, the principle of choosing a webcaster is that the amount of fans is over 10,000 (by querying the personal homepage, the amount of fans of the webcaster can be retrieved), because we believe that the privacy of a webcaster with high popularity is more attractive to malicious attackers. Figure 19 is a sectional statistical chart of the fans of the 100 webcasters; for instance, the number of webcasters with fans between 200,000 and 500,000 is 13.

We also made statistics on the time when the webcasters had recently used the YY mobile app to log into YY. Only 12 of the 100 webcasters have no record of their recent login by the YY mobile app, and the time distribution for the 88 other webcasters' recent login by the YY mobile app is shown in Figure 20. It can be seen from Figure 20 that most of the webcasters logged into YY by the YY mobile app in a day. Therefore, it can be concluded that the webcasters frequently use the YY mobile app to log into YY, which reflects the popularity of the YY mobile app from the side, and proves that the method of using a user's latest login location as the user's common location is practical.

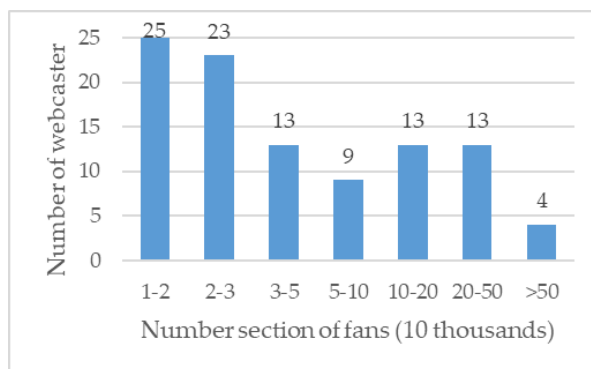


Figure 19. Sectional statistical chart of the fans of the 100 webcasters.

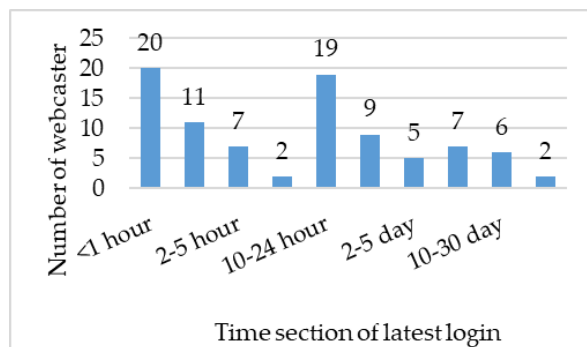


Figure 20. Time distribution for the 88 webcasters' latest login on the YY mobile app.

7.2. Localization Effect

We used the method of trilateration localization to calculate the latest login location of the 88 webcasters, and the localization effect is shown in Figure 21.



Figure 21. Geographical location distribution of the 88 webcasters' latest login.

According to the introduction of Section 4.1, the location information (province, city) filled in by the user in their personal homepage can also be used to deduce a user's common geographical location. However, whether we adopted the user's latest login location or the location information shown on their personal homepage, it may lead to a mistake in judgment of the attribution of the user's mobile phone number. For instance, out of the 100 webcasters, 12 of them had no latest-login records on the YY mobile app and we would not be able to use the latest login locations to infer the attributions of their mobile phone numbers. For the 88 webcasters who we were able to get the latest-login locations from the YY mobile app, their latest login locations were not necessarily the locations of the attributions of their mobile phone numbers. Therefore, we had to consider a user's latest login location on the YY mobile app and the location information shown in the personal homepage. We took the method of combing the two locations and giving priority to the user's latest login location. If a user's mobile phone number could not be broken by using their latest login location as the attribution of their mobile phone number, then we used the location information shown in the personal homepage.

Table 4 shows our statistics on the latest login location on the YY mobile app and the location information shown on the personal homepage of the 100 webcasters. It can be seen from Table 4 that there were 28 webcasters whose location information on their personal homepages is empty, including two webcasters whose latest login locations cannot be retrieved. That is to say, we could not break their mobile phone number based on their geographical location for these two webcasters. For the other 26 webcasters, we could only judge the attributions of their mobile phone numbers by their latest login locations on the YY mobile app. The 10 other webcasters whose latest login locations could not be retrieved had location information shown on their personal homepage; therefore, we used this location information as the attributions of their mobile phone numbers. There are 62 webcasters whose latest login locations could be retrieved and the location information shown on their personal homepages was not empty, but 24 of them had different location information between the latest login location and the location information shown on the personal homepage. For this situation, we took the method of combing the two locations and giving priority to users' latest login location on the YY mobile app.

Table 4. Location information statistics of the geographical locations of the 100 webcasters.

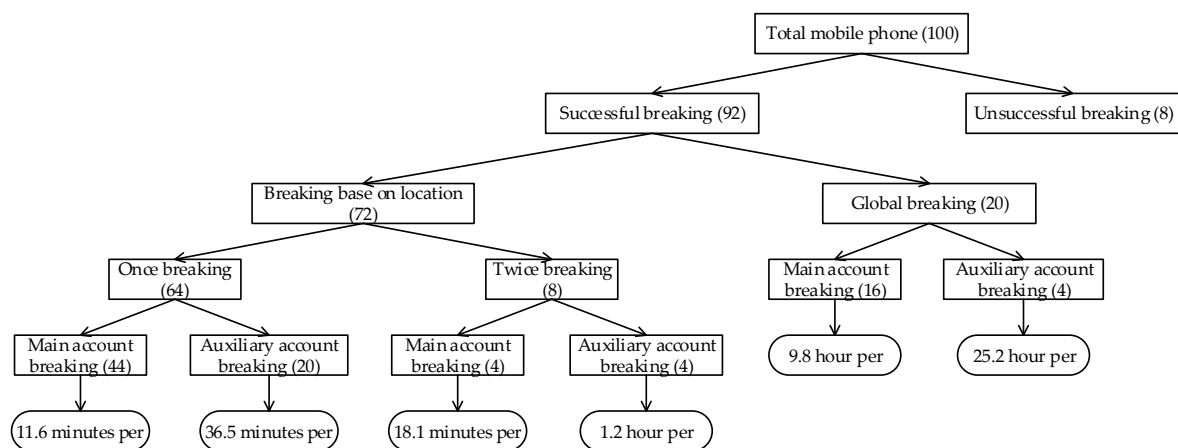
$L_2 \backslash L_1$	$L_1 = \text{Null}$	$L_1 \neq \text{Null}$	Total
$L_2 = \text{Null}$	26	38 ($L_1 = L_2$) 24 ($L_1 \neq L_2$)	88
$L_2 \neq \text{Null}$	2	10	12
Total	28	72	100

Note: L_1 represents user's location information shown in the personal homepage, and L_2 represents user's latest login location on the YY mobile app.

7.3. Effect of Mobile Phone Number Breaking

Figure 22 shows the effect of mobile phone number breaking for the 100 webcasters, and it can be seen from the figure that 92 webcasters' mobile phone numbers were successfully broken and only eight of them were not broken. The reasons why they were not broken include: (1) The webcaster's YY account was not registered by mobile phone number, but registered by other methods, such as email or username, and (2) failure of auxiliary account breaking. Without the participation of YY users, we cannot make a detailed analysis of the concrete reasons for the failure.

For the 92 mobile phone numbers that were broken, 72 of them were broken with the assistance of geographical location, which accounts for 78.26%, and this success rate indicates that the method of breaking a user's mobile phone number based on geographical location has reached an ideal effect. The global breaking shown in Figure 22 means that, for the YY account's mobile phone number that was unsuccessfully broken by the assistance of geographical location, we took the method of breaking the six hidden digits in the middle of the mobile phone number by way of pure brute force (up to one million cycles).

**Figure 22.** The effect of mobile phone number breaking.

We should not only consider a user's latest login location by YY app, but also the location information shown on the personal homepage; therefore, we divided our breaking method into once breaking and twice breaking. Once breaking means breaking a user's mobile phone number by the user's latest login location, while twice breaking means by a user's latest login location and location information shown on the personal homepage. It obviously needs only once breaking if the latest login location or the location information on the personal homepage is empty, while it may need twice breaking if both of them are not empty (e.g., 24 webcasters whose latest login locations are different from the location information in personal homepage). We gave priority to the latest-login location, and if the break work failed by using a webcaster's latest login location as the attribution of

the mobile phone number, we would do the second breaking by using the location information shown on the personal homepage. It can be seen from Figure 22 that among the 72 webcasters whose mobile phone numbers were broken by the assistance of geographical location, 64 webcasters' mobile phone numbers are broken by once breaking, which accounts for 88.89%, while eight webcasters' mobile phone numbers are broken by twice breaking, which accounts for 11.11%.

The main account breaking shown in Figure 22 means breaking a YY user's mobile phone number without auxiliary account breaking. It can be seen from Figure 22 that 64 of the 92 mobile phone numbers that were broken were based on main-account breaking, which accounts for 69.57%, and this reflects that most of the webcasters' YY accounts are registered by mobile phone number, which increases the possibility of leakage of mobile phone numbers. It also validates our conclusion in Section 3.3 that most YY users will choose their mobile phone numbers to register and log into YY. 28 of the 92 mobile phone numbers that were broken were based on auxiliary account breaking, which accounts for 30.43%, and because auxiliary-account breaking needs to query each user ID's channel information and geographical location (user ID in collection *M* introduced in Section 5.3), and do the work of channel matching and geographical-location matching, it costs more time. For instance, it can be seen from Figure 22 that, for the 64 webcasters' mobile phone numbers that were broken by once breaking, the mean time cost for main-account breaking was 11.6 min, while auxiliary-account breaking was 36.5 min. The situation of global breaking and twice breaking was similar to this.

From the perspective of the speed of mobile phone number breaking, the speed of breaking a user's mobile phone number based on geographical location is faster than global breaking. The minimum mean time cost is 11.6 min for the breaking work base on geographical location, while it is 9.8 h for global breaking, and the speed difference is obvious. In addition, for the more time-consuming auxiliary-account breaking, the maximum mean time cost is 1.2 h for the breaking work base on geographical location, while it is 25.2 h for global breaking, and the speed difference is also very obvious. This shows the great advantage of the method of breaking a user's mobile phone number based on geographical location.

8. Security Precaution Suggestions

In this paper, we proposed a new form of network attack, breaking a user's mobile phone number based on geographical location, and took the largest entertainment webcast platform in China "YY" as the test object to test the effectiveness of our methods. The related security vulnerabilities were reported to the manufacturers and have been repaired in time, and we also received thanks from the manufacturers. Experimental results show that this form of network attack is practical and feasible. As similar security problems exist in many software products, in order to effectively prevent this form of attack and protect the users' information security, we put forward some suggestions for security precautions here:

(1) Protect users' geographical location privacy

We previously mentioned that the attacker could retrieve a user's geographical location from two aspects; namely, the location information shown in the personal homepage and the geographical location leaked by some functions of the software (e.g., the "nearby user" function). For the former, a user can completely decide whether to show their geographical location in the personal homepage. For the latter, due to the lack of effective security precautions, a user's location could be calculated by attackers through trilateration localization. Many software products have not even set a functional option to let users prohibit location displaying, by which users can decide whether to allow other users to find them through the software (e.g., a user prohibits location displaying and makes the other users unable to find them with the "nearby user" function). Besides, we think the developer should identify the client's coordinate camouflage behavior effectively, or take effective distance ambiguity measures

to fuzzily process the distance data returned by the server, which makes it difficult for the attacker to calculate a user's accurate location coordinates through the method of trilateration localization.

(2) Restrict the external exposure of user's mobile phone number

Some software products expose parts of a user's mobile phone number just for enhancing user experience, rather than intentionally leaking user's privacy, and developers are often unaware of the dangers. Not exposing any mobile phone number digits may lead to poor user experience for some functions. For instance, if no phone number hints are given in the password-retrieving function, users who have more than one phone number may forget their registered mobile phone number, leading to a failure in getting the password back. Therefore, developers must take account of two aspects—user experience and information security. Our suggestion is that developers should expose the digits of mobile phone numbers to users as little as possible, and try to expose some of the first seven digits of the mobile phone number. Because these digits are related to carrier and geographical location and their possible combinations are already very limited, exposing some of their digits has little contribution to reducing the test set of mobile phone numbers, which is used for brute force. On the contrary, the exposure of each digit in the last four digits will reduce the test set for 90%, which has a big contribution to reducing the test set. Therefore, developers should try to avoid the exposure of the last four digits of a mobile phone number.

(3) Take effective access control to the client

According to the introduction of Section 2.2, the method of breaking a user's mobile phone number based on geographical location needs the attacker to use the enumeration (brute-force) method to frequently send requests to the server to try a variety of possibilities for a user's mobile phone number. Therefore, the server should be able to effectively identify the client's malicious behavior of frequent request sending, and restrict it to prevent user's mobile phone number from being broken.

9. Related Work

Geographical location and mobile phone number are important parts of user privacy and they have always been the key topics for academic studies. Lots of studies have been on the privacy leakage problems about these two aspects.

In terms of the study of geographical location privacy, some researchers studied the “nearby user” function in many social networks, and studied the problem of leakage of users' location privacy. Li et al. [3] found the privacy-leakage problem of users' geographical location in three popular location-based social networks (LBSNs) including Wechat, Skout, and Momo, and proposed the localization methods. Polakis et al. [4] systematically assessed the effectiveness of the defense techniques that are used by many LBSNs to prevent a user's geographical location from being leaked by trilateration attacks, provided the theoretical foundation for formalizing the problem under different proximity models, and designed practical attacks for each case. Xue et al. [5] investigated the user's location privacy leakage problem in location-based social discovery (LBSD) services reporting distances in discrete bands. Using number theory, they found that by strategically placing multiple virtual probes with contrived fake GPS locations, one can nevertheless pinpoint user locations in band-based LBSD. Hoang et al. [6] investigated three popular location-based dating apps, Grindr, Jack'd, and Hornet, and they found that though these three apps adopted some location obfuscation measures and could even disable the “show distance” function, users' geographical location could still be localized. It can be concluded that most of the existing studies on the user location privacy leakage problem only pay attention to users' geographical location, but do not extend to solve other aspects of user's privacy by making use of a user's geographical location privacy that could be leaked. On the contrary, our work makes further use and expansion of the location privacy leakage problem, breaking a user's mobile phone number based on their geographical location.

In terms of studies on phone number privacy, Cheng et al. [13] studied the security problems of the “address book-matching” function in many smartphone apps, and proposed a novel method to abuse such functions to automatically collect user profiles. They also integrated profiles gathered from different messenger applications and provided insights by performing consistency and authenticity analysis on user profile fields. Kim et al. [14] studied the function of “using user’s phone number to find friends” in KakaoTalk, which is the most widely used instant-messaging service in Korea, and demonstrated that there are multiple ways of collecting victims’ personal information, such as their (display) names, phone numbers, and photos, which can be potentially misused for a variety of cybercriminal activities. Gupta et al. [15] explored the feasibility, automation, and scalability of a variety of targeted attacks that can be carried out by abusing phone numbers, and demonstrated a novel system that takes a phone number as an input, leveraging information from different apps to target the victim on a chosen attack channel. However, most of the existing work merely queries user privacy information based on users’ mobile phone number, but not users’ mobile phone number based on the information that has been obtained; in this paper, we studied the method of breaking a user’s phone number privacy based on their other privacy information (geographical location).

10. Conclusions and Future Work

Geographical location and mobile phone number are important contents of a user’s privacy information and they have important connections. This paper makes full use of the relationship between the two privacy aspects, and proposes a new form of network attack using users’ leaked geographical location to assist the breaking of mobile phone numbers. What is more, we successfully tested and verified this new form of network attack through a popular software product (the largest entertainment webcast platform in China—YY). We hope that, through this study, software developers will be made aware of the potential security threats and to take effective measures to protect the security of users’ privacy information.

The security problems studied in this paper are mainly concentrated on Chinese software products. We did not study whether the world’s popular apps have similar security problems because the features of mobile phone numbers in each country are different and we are not familiar with the mobile phone segment allocation made by the carriers in each country. Therefore, in future work, we will expand our study to global web services and discuss the feasibility of our method of breaking user’s mobile phone numbers for the world’s popular apps.

Author Contributions: Conceptualization, H.Y.; Methodology, H.Y.; Software, H.G.; Validation, H.G. and X.M.; Formal Analysis, H.Y.; Investigation, H.G.; Resources, X.M.; Data Curation, X.M.; Writing-Original Draft Preparation, H.Y.; Writing-Review & Editing, H.G.; Visualization, X.M.; Supervision, H.Y.; Project Administration, H.Y.; Funding Acquisition, H.Y.

Funding: This research was funded by Nanhu Scholars Program for Young Scholars of XYNU.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Andrienko, G.; Gkoulalas-Divanis, A.; Gruteser, M. Report from Dagstuhl: The liberation of mobile location data and its implications for privacy research. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2013**, *17*, 7–18. [[CrossRef](#)]
2. Kushwaha, A.; Kushwaha, V. Location based services using android mobile operating system. *Int. J. Adv. Eng. Technol.* **2011**, *1*, 14–20.
3. Li, M.; Zhu, H.; Gao, Z.; Chen, S.; Yu, L.; Hu, S.; Ren, K. All your location are belong to us: Breaking mobile social networks for automated user location tracking. In Proceedings of the 15th ACM International Symposium on Mobile ad Hoc Networking and Computing, Philadelphia, PA, USA, 11–14 August 2014; pp. 154–196.

4. Polakis, I.; Argyros, G.; Petsios, T.; Sivakorn, S.; Keromytis, A.D. Where's Wally?: Precise user discovery attacks in location proximity services. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 817–828.
5. Xue, M.; Liu, Y.; Ross, K.W.; Qian, H. I know where you are: Thwarting privacy protection in location-based social discovery services. In Proceedings of the 2015 IEEE Conference on Computer Communications Workshops, Hong Kong, China, 26 April–1 May 2015; pp. 179–184.
6. Hoang, N.P.; Asano, Y.; Yoshikawa, M. Your neighbors are my spies: Location and other privacy concerns in dating apps. In Proceedings of the 18th International Conference on Advanced Communication Technology, Pyeongchang, Korea, 31 January–3 February 2016; pp. 715–721.
7. Fawaz, K.; Feng, H.; Kang, G.S. Anatomization and protection of mobile apps' location privacy threats. In Proceedings of the 24th USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015; pp. 753–768.
8. Hallsteinsen, S.; Jorstad, I.; Thanh, D.V. Using the mobile phone as a security token for unified authentication. In Proceedings of the 2nd International Conference on Systems and Networks Communications, Cap Esterel, France, 25–31 August 2007; p. 68.
9. Bertino, E.; Sandhu, R. Database security—Concepts, approaches, and challenges. *IEEE Trans. Dependable Secur. Comput.* **2005**, *2*, 2–19. [[CrossRef](#)]
10. Halfond, W.G.J.; Viegas, J.; Orso, A. A classification of SQL-injection attacks and countermeasures. In Proceedings of the 2006 IEEE International Symposium on Secure Software Engineering, McLean, VA, USA, 13–15 March 2006.
11. Fleizach, C.; Liljenstam, M.; Johansson, P.; Voelker, G.M.; Mehes, A. Can you infect me now?: Malware propagation in mobile phone networks. In Proceedings of the 5th ACM Workshop on Recurring Malcode, Alexandria, VA, USA, 2 November 2007; pp. 61–68.
12. Felt, A.P.; Finifter, M.; Chin, E.; Hanna, S.; Wagner, D. A survey of mobile malware in the wild. In Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, Chicago, IL, USA, 17 October 2011; pp. 3–14.
13. Cheng, Y.; Ying, L.; Jiao, S.; Su, P.; Feng, D. Bind your phone number with caution: Automated user profiling through address book matching on smartphone. In Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, Hangzhou, China, 8–10 May 2013; pp. 335–340.
14. Kim, E.; Park, K.; Kim, H.; Song, J. Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk. *Comput. Secur.* **2015**, *52*, 267–275. [[CrossRef](#)]
15. Gupta, S.; Gupta, P.; Ahamad, M.; Kumaraguru, P. Exploiting phone numbers and cross-application features in targeted mobile attacks. In Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices, Vienna, Austria, 24 October 2016; pp. 73–82.
16. Schrittwieser, S.; Fruehwirt, P.; Kieseberg, P.; Leithner, M.; Mulazzani, M.; Huber, M.; Weippl, E. Guess who is texting you? Evaluating the security of smartphone messaging applications. In Proceedings of the 19th Annual Network & Distributed System Security Symposium, San Diego, CA, USA, 5–8 February 2012.
17. Kim, E.; Park, K.; Kim, H.; Song, J. I've got your number: Harvesting users' personal data via contacts sync for the KakaoTalk Messenger. In Proceedings of the 15th International Workshop on Information Security Applications, Jeju Island, Korea, 25–27 August 2014.
18. Gupta, S. Emerging threats abusing phone numbers exploiting cross-platform features. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, San Francisco, CA, USA, 18–21 August 2016; pp. 1339–1341.
19. Murphy, W.; Hereman, W. *Determination of a Position in Three Dimensions Using Trilateration and Approximate Distances*; Technical Report: MCS-95-07; Department of Mathematical and Computer Sciences, Colorado School of Mines: Golden, CO, USA, 1995.

