I've Got Your Number:

Harvesting users' personal data via contacts sync for the KakaoTalk messenger

Eunhyun Kim[1], Kyungwon Park[1], Hyoungshick Kim[1], and Jaeseung Song[2]

[1]Department of Computer Science and Engineering, Sungkyunkwan University, Korea
[2]Department of Computer and Information Security, Sejong University, Korea

Abstract. Instant messaging (IM) is increasingly popular among not only Internet but also smartphone users. In this paper, we analyze the security issue of an IM application, KakaoTalk, which is the most widely used in South Korea, with a focus on automated friends registration based on contacts sync. We demonstrate that there are multiple ways of collecting victims' personal information such as their names, phone numbers and photos, which can be potentially misused for a variety of cyber criminal activities. Our experimental results show that a user's personal data can be obtained automatically (0.26 seconds on average), and a large portion of KakaoTalk users (around 73%) uses their real names as display names. Finally, we suggest reasonable countermeasures to mitigate the discovered attacks, which have been confirmed and patched by the developers.

Keywords: Automated friends registration; Contacts sync; Enumeration attack; Information leakage; Security; Privacy; Smartphone; KakaoTalk

1 Introduction

Instant messaging (IM) has become a popular communication service for people who want to stay in touch with their family, friends and business colleagues since there is no cost (or low cost) to use IM services other than an Internet data plan that most users already have for their smartphones or PCs. However, IM services (e.g., `WhatsApp`, `iMessage` and `Skype`) have become the target of continuous cyber attacks such as spam, phishing and the misuse of personal data due to their growing popularity. For example, spammers might want to create rogue user accounts to effectively share their advertisements with IM users.

In this paper, we particularly focus on the discussion of security concerns raised by the automated friend registration (or recommendation) feature used by default for `KakaoTalk` (http://www.kakao.com/talk/en) which is the most widely used IM service in South Korea. Once this feature is enabled, the newly added phone numbers from the address book in a user's mobile phone are periodically uploaded to the `KakaoTalk` server in order to maintain the list of the user's friends up to date by automatically registering friends based on their `KakaoTalk` accounts associated to the added phone numbers. This automatic

process is based on the intuition that address book contacts in a mobile phone might be the people that the phone owner wants to communicate with.

This automated friend registration feature with phone numbers sufficiently convenient for an easy way to manage IM friends. However, we argue here that this feature leak critical information about IM users even if `KakaoTalk` is trying to prevent it. An adversary can attempt to collect victims' personal data such as their `KakaoTalk` accounts, names, phone numbers and even photos via the contacts sync for `KakaoTalk` messenger. This is because `KakaoTalk` accounts can be collected with only the phone numbers associated them. The collected `KakaoTalk` accounts (and their personal information) might be used for a variety of cyber criminal activities such as spam, phishing and rogue accounts — it can be beneficial for spammers to collect active phone numbers with the phone owners' real names; similar problems can arise if automated friend recommendation services based on users' personal data (e.g., a unique identifier such as phone numbers) are used.

Schrittwieser et al. [5] reported a similar security flaw named *enumeration* in several smartphone messaging applications (e.g., `WhatsApp`, `Viber` and `Tango`). We here extend their work by presenting new *enumeration* attacks that targeted the `KakaoTalk` service which already have some countermeasures unlike the other applications such as `WhatsApp`. Surely, if the contacts export function is explicitly provided, it is not difficult to implement an efficient *enumeration* attack. `KakaoTalk` originally allowed users to export their friends' information into a text file, but this function was recently removed for security reasons. In this paper, we show that users' names and phone numbers can still be obtained without the contacts export function by automatically generating a specific sequence of touch events and examining the heap memory that is used for the `KakaoTalk` process. Our key contributions can be summarized as follows:

- First, we introduce new *enumeration* attacks that targeted `KakaoTalk` and examine their feasibility and efficiency in practice. We collected more than 50,000 users' personal data and analyzed the data. The best attack method takes 0.26 seconds on average to obtain the information about a user's name and phone number.
- Second, we show the impacts of these attacks by analyzing the collected user profile information. Our experimental results show that 36,817 out of 50,567 samples 72.8%) look like users' *real* names.
- Third, we suggest reasonable countermeasures to mitigate such *enumeration* attacks, which have been confirmed and patched by the developers.

The rest of this paper is organized as follows. In Section 2, we explain how the automated friend registration process in `KakaoTalk` works to provide a better understanding of *enumeration* attacks. Then we present the three *enumeration* attacks that targeted `KakaoTalk` to collect `KakaoTalk` user's personal data in Section 3. In Section 4, we introduce the implementations for *enumeration* attacks and evaluate their feasibility and efficiency by conducting experiments in the real-world environment. We present a discussion on countermeasures to mit-

igate *enumeration* attacks and ethics in Section 5. Related work is discussed in Section 6. Finally, we conclude in Section 7.

2 Automated Friends Registration in KakaoTalk

KakaoTalk is the most widely used free IM in South Korea — it currently has over 145 million registered users worldwide, including 93% of smartphone users in South Korea [3]. The KakaoTalk service was originally developed as a mobile application (similar to WhatsApp) for smartphones such as Android and iOS devices, but the PC and Mac versions of KakaoTalk applications were also recently released.

To encourage a user to find and add other users as his/her KakaoTalk friends, there are three ways: (1) searching for a user by KakaoTalk ID, (2) using a quick response (QR) code and (3) automatic syncing address book contacts with the corresponding KakaoTalk accounts. When a user wants to add a specific KakaoTalk user as a KakaoTalk friend, the user's KakaoTalk ID or the related QR code can be used. However, the most popular way is to use the automated friends registration option. In fact, this feature is turned on by default and can be disabled for only those who do not want to use this.

Once the automatic sync feature is enabled, the contacts in the phone owner's address book are added to the list of her KakaoTalk friends without manual intervention if the phone number of them are associated with KakaoTalk accounts. This process is shown in Figure 1. The newly added phone numbers (step 1) from the address book are uploaded to the KakaoTalk server (step 2); the KakaoTalk server tries to find the KakaoTalk accounts with the phone numbers matched to the received phone numbers from the phone owner's KakaoTalk application and returns those to the KakaoTalk applications running on the requested user's devices such as smartphone and PCs (step 3) to update the list of her KakaoTalk friends with new friends (step 4). This automatic process is based on the intuition that address book contacts in a mobile phone might be the people that the phone owner wants to communicate with.

Interestingly, the KakaoTalk service does not provide the newly added friends' original display names, which are registered to the KakaoTalk server, via the automated friends registration process. Therefore, their names are displayed on the KakaoTalk application as the contact names in the address book rather than their original display names which are kept confidential. We surmise that this naming policy has been established to protect users' personal data from *enumeration* attacks which attempt to collect the KakaoTalk users' names and phone numbers with enumerated the (possibly) entire phone number range. Since the display names are not synced, the owner of a phone number cannot be identified even when there exits a KakaoTalk account associated with the phone number.

Therefore, in designing a new *enumeration* attack against KakaoTalk, the main hurdle we had to overcome was to obtain the information about KakaoTalk accounts' original display names without any knowledge about the account hold-

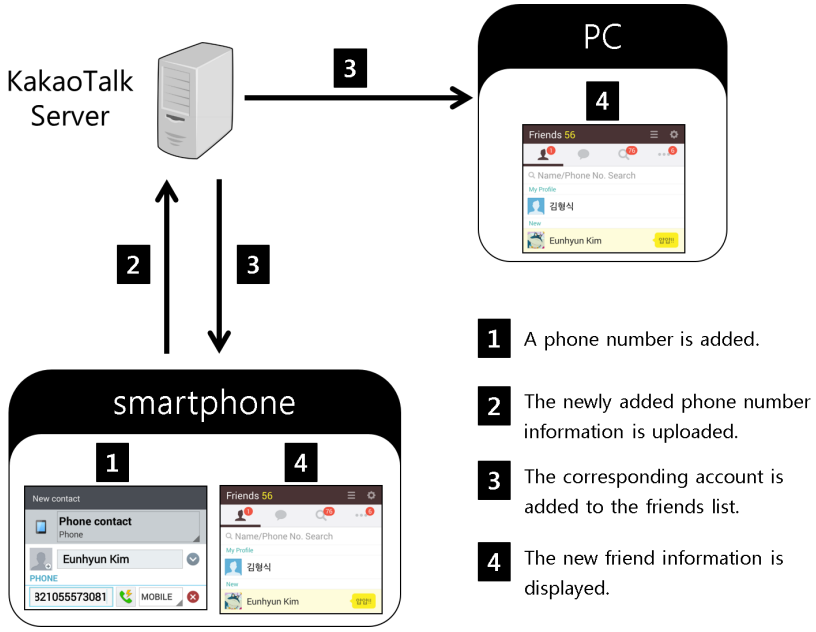


Fig. 1. The process of automated friends registration in `KakaoTalk`

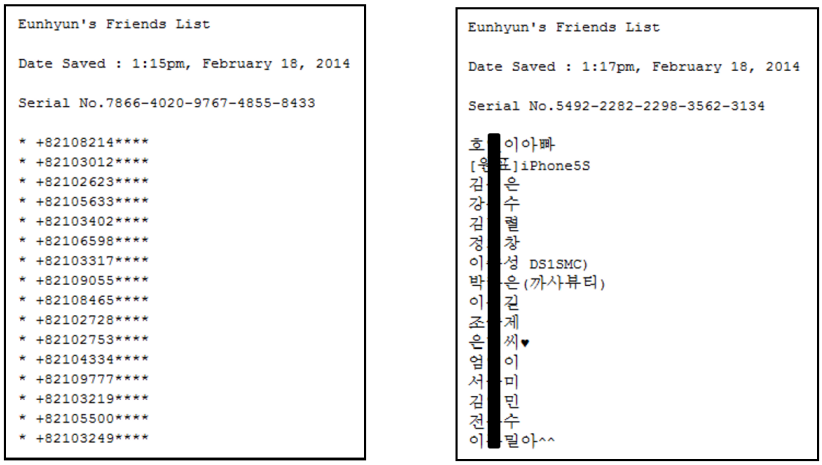
ers. We will discuss the details of the proposed *enumeration* attacks in the next sections.

3 Enumeration Attacks via Contacts Sync

To conduct an *enumeration* attack via contacts sync, an attacker generates a range of phone numbers in a valid format and adds the generated phone numbers into the attacker's address book to collect valid phone numbers with their (display) names via the automated friends registration feature in `KakaoTalk`. The collected information might be effectively used for spam and phishing attacks. To make matters worse, victims do not find that their personal data were leaked by an *enumeration* attack since users can be added without their explicit consents in `KakaoTalk`.

In this paper, we introduce the following three *enumeration* attacks and discuss their advantages and disadvantages:

- **Use of the export function in `KakaoTalk`:** an attacker saves the added `KakaoTalk` users' personal data as a file and/or exports it to email.
- **Use of Optical Character Recognition (OCR) software:** an attacker uses OCR software to extract users' display names.
- **Use of a debugging tool:** an attacker uses a debugging tool to extract users' display names from the memory of a running `KakaoTalk` application.



(a) Exported phone numbers (b) Exported original display names

Fig. 2. Examples of exported contacts information by using the export feature

3.1 Use of the export function in KakaoTalk

The KakaoTalk application allows a user to save the user's friend list as a text file and export the file via email as well. This feature is particularly useful if a user acquires a new device for the KakaoTalk service since the user can restore her KakaoTalk friends from the backup file. The first *enumeration* attack is to simply use this feature.

As described in Section 2, the KakaoTalk service uses the contact names in the address book as the names displayed on the KakaoTalk application rather than their original display names. Therefore, if an attacker exports the list of friends obtained by an *enumeration* attack into a file, the attacker can obtain those numbers with the unwanted *pseudo* names arbitrarily assigned by the attacker instead of their original display names in KakaoTalk. Figure 2(a) shows an example of the exported phone numbers with the arbitrary name of '*'.

However, we found that when a KakaoTalk friend's phone number is removed from the address book, the friend's display name on the KakaoTalk application is changed to his/her original name by default via contacts sync. That is, when the friend's phone numbers are removed from the address book, an attacker can export the list of friends' original display names although their phone numbers are removed. Figure 2(b) shows an example of the list of KakaoTalk friends' original display names alone in the exported file.

Therefore, the following *enumeration* attack against KakaoTalk can be implemented by repeatedly exporting the friends list two times:

Step 1. A range of phone numbers in a valid format is generated and added into the address book.

Step 2. By using the export feature, the list of `KakaoTalk` friends' phone numbers is exported (see Figure 2(a)).
Step 3. The added phone numbers are removed from the address book.
Step 4. The registered friends' display names on the `KakaoTalk` application are changed to their original names via contacts sync.
Step 5. By using the export feature, the list of `KakaoTalk` friends' original display names is exported (see Figure 2(b)).
Step 6. The combination of these two lists of phone numbers and their `KakaoTalk` names is stored as the output of the attack.

As a countermeasure against *enumeration* attacks, however, the `KakaoTalk` service removes the last four-digits of each friend's phone number by masking them with a sequence of asterisk ('`****`') characters so that the user's private phone number is possibly protected (see Figure 2(a)). At first glance, this seems secure and reasonable, in reality can do little or nothing to actually achieve improved security.

A simple trick can be used to successfully bypass this defensive mechanism. When a range of phone numbers is generated, an attacker can generate the phone numbers having the same last four-digits but (uniquely) different remaining digits. For example, the attacker can collect active numbers with '`3333`' as the last four-digits by generating phone numbers from '`+82-10-0000-3333`' to '`+82-10-9999-3333`' and entering those number into the attacker's address book. Although the information about '`3333`' is hidden since the last four-digits are masked, the attacker can easily recover this information by replacing '`****`' with '`3333`'.

3.2 Use of OCR software

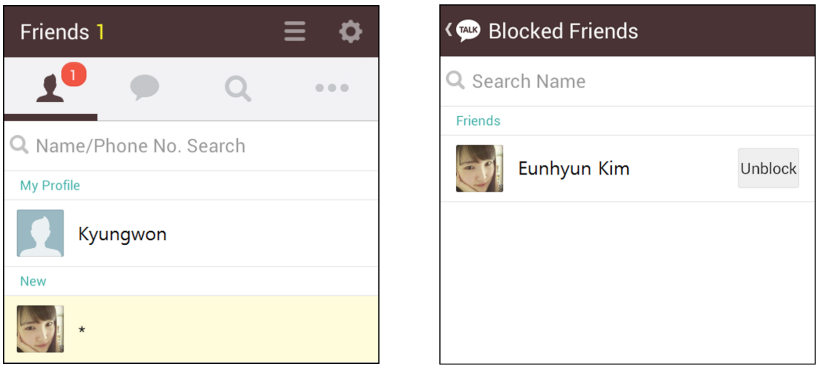
Although an efficient *enumeration* attack can be implemented by using the export feature, `KakaoTalk` recently removed this feature for security reasons (e.g., to prevent *enumeration* attacks).

Without the export feature, however, we can still collect `KakaoTalk` users' names and phone numbers by using another *enumeration* attack. We found that a victim's original display name can be shown by a specific sequence of user interactions.

When a user is blocked, the user's original name is displayed in the list of blocked friends. This vulnerability was discovered by generating possible user actions in a brute force manner. Figure 3 shows an example of this situation. We can see that a user's display name of '`*`' is replaced with '`Eunhyun Kim`' when her account is blocked.

In the second *enumeration* attack, our main idea is to extract a user's original display name using OCR software after blocking the user. The *enumeration* attack can be implemented as follows:

Step 1. A phone number in a valid format is generated and added into the address book.



(a) before blocking (b) after blocking

Fig. 3. An example of the displayed user name in the list of blocked friends

Step 2. The list of friends is synchronized with the added phone number for the automated friends registration.
Step 3. After the synchronization, it is checked whether the new `KakaoTalk` friend (associated with the added phone number) is added. This test can be easily implemented by checking the color of the pixel at the specified location in the captured image of the friends list. If a new user is added into the list of friends, the tested pixel should be yellow.
Step 4. If there exists a new friend, the friend is blocked in sequence to find his/her original display name.
Step 5. The screen of blocked friends is captured.
Step 6. The victim's display name is extracted from the captured image by using OCR software.
Step 7. The combination of the entered phone number and the recognized display name is stored as the output of the attack.
Step 8. This process is repeated with another phone number over and over, until there is no new phone number.

3.3 Use of a debugging tool

The `KakaoTalk` service allows a user to change their friends' display names on the user's `KakaoTalk` application according to the user's needs and preferences. This implies that a text object (i.e., `TextView`) should be used in the application to handle the display name of the friend's name in a flexible manner.

The third *enumeration* attack is based on the use of this feature. An attacker can retrieve the text of the object handling a victim's display name by using a debugging tool to track memory allocation of the object since there exist several debugging tools (e.g., DDMS: Dalvik Debug Monitor Server) which enable us to capture a snapshot of the volatile memory used by an Android application.

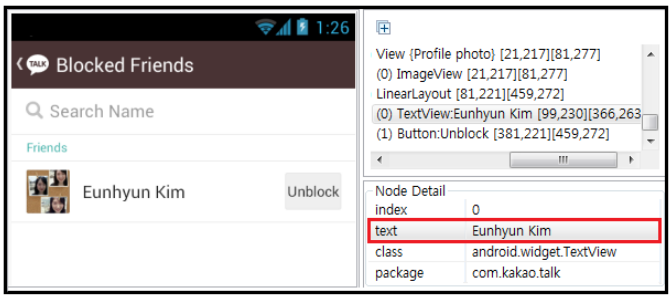


Fig. 4. An example of the victim's name in a memory dump

In the memory dump for the `KakaoTalk` application, the application's objects (e.g., text, image, list and etc.) and their properties can be retrieved. Figure 4 shows that a blocked friend's display name can be accessed from the associated `TextView` object. In the memory dump of the blocked friends layout, the text property of this object includes a blocked friend's original display name (highlighted in a red box).

This *enumeration* attack can be implemented in a similar way as the 'use of OCR software' except that a debugging tool is used to directly retrieve the victim's display instead of using OCR software. Instead of using OCR software, a debugging tool is used to directly retrieve the victim's display name.

4 Experiments

In this section, we describe the *enumeration* attacks described in Section 3 to show their feasibility against the `KakaoTalk`'s countermeasures and evaluate their performance in a real-world setting.

4.1 Implementation

For the *enumeration* attack using the export function, we used a Google Nexus S (with a 1 GHz CPU and 512MB RAM) running the Android 4.1.1 Jelly Bean. In Android, the contacts in the address book can simply be modified by sending an intent from the attacker's application.

For the other *enumeration* attacks, however, we additionally used a Windows PC (with an Intel core i5 CPU and 4GB RAM) running the 64-bit Windows 7, equipped with a non-congested 100 Mbit/s WiFi connection to a LAN that was connected to the Internet via a Gigabit-speed link. The Windows PC was needed to use OCR software or a debugging tool.

For the *enumeration* attack using OCR, we used the OCR service provided by NAVER lab (http://t.lab.naver.com/ocr/) since most `KakaoTalk` users' display names are written in Korean and NAVER lab's OCR service supports Korean with a high accuracy. To improve the accuracy of text recognition in the OCR

Table 1. Performance of *enumeration* attacks

	Export	**OCR**	**Debugging**
Time	0.26 sec	51.07 sec	32.56 sec
Accuracy	1.00	0.32	1.00

service, we increased the resolution of the captured image and cropped the image to remove unnecessary image area (e.g., the blank space).

For the *enumeration* attack using a debugging tool, we used a debugging tool called DDMS, which is commonly used for debugging a process, to track the thread and heap information on the `KakaoTalk` application.

4.2 Attack results

With these implementations, we tested the sequentially generated 101,000 phone-like numbers and collected 50,567 `KakaoTalk` user data (about 50.1%) in an automatic manner. When we conducted these experiments, the `KakaoTalk` server did not prevent us from synchronizing these numbers and blocking them in sequence. However, our intention is not to collect users' private data but test the feasibility of the attacks. We discuss the ethics in Section 5.

We analyzed the performance of the above three attack implementations by measuring the number of `KakaoTalk` users collected during a time period by each attack implementation.

Unsurprisingly, the 'use of the export function' (**Export**) outperformed the other *enumeration* attacks in terms of speed. In this attack, the average time required for a user's data is about 0.26 seconds. However, `KakaoTalk` recently removed the export feature for security reasons. Hence, this attack is no longer available.

In this situation, a more obvious recommendation would be to use the debugging tool (**Debugging**) since this approach does not rely on the `KakaoTalk`'s export feature and is significantly better than the 'use of OCR software' (**OCR**) in terms of speed and reliability; when we use the 'use of a debugging tool', the average time required for a user's data is about 32.56 seconds — this enable us to collect around 2,654 `KakaoTalk` users' personal data within a day — while the average time required for a user's data is about 51.07 seconds when we use the 'use of OCR software'.

Finally, we analyze how many `KakaoTalk` users set real names as their display names in user profiles. When we manually checked the collected 50,567 `KakaoTalk` user data, we found 36,817 (72.8%) users set real names as their display names. This implies that serious invasions of privacy might be raised since the collected users' personal data (e.g., phone numbers, status messages and profile pictures) can be effectively associated with their real identities. Probably, this private data can be used to design sophisticated spam and phishing attacks.

5 Discussion

In this section we discuss about potential countermeasures and ethical issues. First, we suggest the three reasonable countermeasures to mitigate *enumeration* attacks in different aspects. We also address the ethical issues in this section since we collected user information without permissions.

5.1 Countermeasures

We describe several potential mechanisms to mitigate the leakage of private information in `KakaoTalk`. Since these methods are tackling different aspects of the system design, instead of recommending a single approach, we discuss potential countermeasures without any preferences.

- **Detecting anomalous behaviors:** An *enumeration* attack blindly tries to collect user information by automatically sending a lot of queries to a server, which are totally different from normal users' behaviors — in general, the activities generated by an *enumeration* attack would be periodically repeated. Wang et al. [7] already introduced a system to detect automated activities in an online social network using server-side event models. They found that automated bots have shown significantly different behaviors (e.g., sending too many friend requests and spam) from human users during a session. Therefore, we can also build a similar system to detect the queries used in an enumeration attack with a server-side model.
- **Minimizing information leakage:** According to our experiments (see Section 4 for details), when the automated friends registration is processed, the `KakaoTalk` server tries to synchronize friends' personal data stored on the server-side database with the local database of a `KakaoTalk` application running on a user's (i.e., attacker's) device. This might be helpful for some users, but is not necessarily required to complete the automated friends registration process. We suggest that the `KakaoTalk` service should not provide any personal information (such as display name, profile picture and etc.) about new friends right after new friends are added via automated friends registration. For usability, instead, some user data (e.g., profile picture) can be synchronized after verifying that they actually know each other (e.g., after having a first chat).
- **Changing the registration policy:** Unlike other social network services (e.g., Facebook and LinkedIn) and IMs (e.g., Skype), `KakaoTalk` users can add their friends without their consents. If a user simply adds a phone number together with the corresponding contact name, the `KakaoTalk` service automatically adds the contact as the phone owner's `KakaoTalk` friend. This is a very convenient feature and helps `KakaoTalk` increase the number of users for a short-period. However, as we described in this paper, this feature now can be used for *enumeration* attacks. Therefore, in order to mitigate this type of vulnerabilities, `KakaoTalk` should consider changing the current friend registration mechanism to an invitation-based one.

5.2 Ethical issues

The main motivation of our experiments is not to obtain personal information data or to use collected data for commercial or illegal purpose. We conduct research work in order to discover vulnerabilities from a popular smartphone IM and develop reasonable countermeasures to mitigate the discovered vulnerabilities. Therefore, we reported the discovered design flaws to the `KakaoTalk` developers, who acknowledged these flaws, patched them and released an updated version.

Soon after we reported the discovered vulnerabilities, `KakaoTalk` has released the patch fixing the vulnerabilities. We again tested the updated `KakaoTalk` and confirmed that a user's real name is not revealed anymore by the proposed attacks in this paper. However, we have found a side-effect from the patch that still revealing the user's real profile name. This is mainly because the patch has not been analyzed enough to guarantee the correct fixing the vulnerabilities.

In summary, our motivation is to open and discuss the vulnerabilities about *enumeration attacks* in the automated friends registration process. The countermeasures for our attacks are suggested in the above subsection.

6 Related work

Schrittwieser et al. [5] analyzed the security of popular IM applications (e.g., `WhatsApp`, `Viber` and `Tango`) and particularly introduced an *enumeration* attack to collect active phone numbers. They showed the feasibility of the attack by collecting 21,095 valid phone numbers that are using the `WhatsApp` application within less than 2.5 hours. We extend this work by introducing several *enumeration* attacks targeting `KakaoTalk`, which is widely used in South Korea.

A similar problem related to *enumeration* attack was already reported in social networks. Balduzzi et al. [1] showed the feasibility of an *enumeration* attack that automatically queries about e-mail addresses to collect a list of valid e-mail addresses by uploading them to the friend-finder feature of `Facebook`. Based on the return value of Facebook, they were able to determine the status of an email address. They tested about 10.4 million e-mail addresses and identified more than 1.2 million user profiles associated with these addresses.

Gross et al. [2] showed that user profiles in online social networks can be misused in ways that present an abuse of personal privacy. They observed that 77.7% of users were stalked because of the disclosure of their profiles. Wanying Luo et al. [4] presented a group-key based social network service such that users' real identities can be revealed to only authorized group members who own a valid group key.

Smale et al. [6] analyzed online users' profile names (or display names) by categorizing them into several types: name, activity, advertisement, opinion, feeling and etc. According to their observations, about 42.4% of users in an IM service used their real names as profile names (i.e., name: about 32.4% and a modification of name: about 10%).

7 Conclusion

This paper examines the security issues (i.e., three enumeration attacks) that arise in an IM service named `KakaoTalk`, which is the most widely used in South Korea and suggests potential countermeasures that have been designed to mitigate them. Our test results show that `KakaoTalk` users' personal data (such as phone numbers, display names and profile pictures) can effectively be collected in an automatic manner. Since a large number of `KakaoTalk` users (about 73%) are using their real name as display names, serious invasions of privacy might be raised by the discovered *enumeration* attacks.

Although, we currently limited our attack experiments in `KakaoTalk` alone, we believe this type of attacks can also be applicable to other social networks and IM applications. For future work, we are planning to extend the proposed techniques (i.e., enumeration attacks and countermeasures) to other social network and IM applications.

8 Acknowledgements

This research was partly supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1010) supervised by the NIPA (National IT Industry Promotion Agency) and is funded in part by the ICT R&D program (2014-044-072-003 , 'Development of Cyber Quarantine System using SDN Techniques') of MSIP/IITP.

References

1. Marco Balduzzi, Christian Platzer, Thorsten Holz, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. Abusing social networks for automated user profiling. In *Recent Advances in Intrusion Detection*, pages 422–441. Springer, 2010.
2. Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.
3. Jeeshan Khan. KakaoTalk Launches Official Mac App. http://tropicalpost.com/kakaotalk-launches-official-mac-app/, 2014.
4. Wanying Luo, Qi Xie, and Urs Hengartner. Facecloak: An architecture for user privacy on social networking sites. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 3, pages 26–33. IEEE, 2009.
5. Sebastian Schrittwieser, Peter Frühwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, and Edgar Weippl. Guess who's texting you? evaluating the security of smartphone messaging applications. In *NDSS '12: Proceedings of the 19th Annual Network & Distributed System Security Symposium*, 2012.
6. Stephanie Smale and Saul Greenberg. Broadcasting information via display names in instant messaging. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 89–98. ACM, 2005.
7. Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y. Zhao. You Are How You Click: Clickstream Analysis for Sybil Detection. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC'13, pages 241–256, 2013.