COMPUTERS & SECURITY XXX (2015) 1–9

Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk

Eunhyun Kim [a], *Kyungwon Park* [a], *Hyoungshick Kim* [a,*], *Jaeseung Song* [b]

[a] *Department of Computer Science and Engineering, Sungkyunkwan University, Republic of Korea*
[b] *Department of Computer and Information Security, Sejong University, Republic of Korea*

ARTICLE INFO

Article history:
Received 13 December 2014
Received in revised form
21 March 2015
Accepted 18 April 2015
Available online xxx

Keywords:
Finding friends with phone numbers
Enumeration attack
Information leakage
Privacy
Instant messaging
KakaoTalk

ABSTRACT

Users' phone numbers are popularly used for finding friends in instant messaging (IM) services. In this paper, we present a new security concern about this search feature through a case study with KakaoTalk which is the most widely used IM in Korea. We demonstrate 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 cyber–criminal activities. Our experimental results show that a user's personal data can be obtained automatically (0.26 s on average). The results also indicate that a large portion of KakaoTalk users (72.8%) have used real or real-like names in their profiles, which means that our discovered enumeration attacks seem to be practically dangerous. To mitigate these attacks, we present three countermeasures including a misuse detection system that can discover abnormal application activities within a certain time-window.

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 friend search (or recommendation) feature with phone numbers, which are used in many IM services by default. This feature provides a sufficiently convenient way for managing IM friends but according to our research it can introduce new and significant privacy risks; an attacker might collect IM users' personal data such as their accounts, names, phone numbers and even photos. Those personal data can be potentially misused for various cyber criminal activities such as spam, phishing and rogue accounts — it can be beneficial for spammers to collect real phone numbers used by someone together with associated users' real name. We note that user

* *Corresponding author*. Tel.: +82 31 299 4324.
E-mail address: hyoung@skku.edu (H. Kim).
http://dx.doi.org/10.1016/j.cose.2015.04.008
0167-4048/

accounts can be collected with only the phone numbers associated them.

As a case study, we analyze the security of this feature in KakaoTalk (http://www.kakao.com/talk/en) which is the most widely used IM service in Korea. Once this friend search 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.

Schrittwieser et al. (2012) reported a similar security flaw named *enumeration* attack in several smartphone messaging applications (e.g., WhatsApp, Viber and Tango). In this paper, we extend their work by presenting new *enumeration* attacks targeted the KakaoTalk service which already have several countermeasures unlike the other applications such as WhatsApp.

In this paper, we show that users' names and phone numbers can be obtained by automatically generating a specific sequence of user activities and examining the heap memory that is used for the KakaoTalk process. We reported the discovered attacks to the KakaoTalk developers, so the related software vulnerabilities have been confirmed and patched. We also suggest three countermeasures to prevent such *enumeration* attacks including a misuse detection system. Our prototype implementation of the detection system shows the feasibility of a server-side defense solution — abnormal behaviors from attackers (i.e., malicious programs) can effectively be detected by monitoring the attacker's activities in KakaoTalk. 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 s 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%) have used real name or real-like name in their profiles.
- Third, we suggest three countermeasures to mitigate such *enumeration* attacks. Here, we particularly implement a misuse detection technique to detect such automatic attacks based on their signatures. Our defense model would incur a high cost to attackers while achieving a high accuracy at the same time.

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 mitigate *enumeration* attacks in Section 5. Next, we discuss ethical issues in Section 6. Related work is discussed in Section 7. Finally, we conclude in Section 8.

2. Automated friends registration in KakaoTalk

KakaoTalk is the most widely used free IM in Korea — it currently has over 145 million registered users worldwide, including 93% of smartphone users in South Korea (Khan, 2014). 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 Fig. 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 (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 holders. We

Please cite this article in press as: Kim E, et al., Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.008

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

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, phishing or profile cloning attacks (Bilge et al., 2009). 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.

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. Fig. 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. Fig. 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:

1) A range of phone numbers in a valid format is generated and added into the address book.
2) By using the export feature, the list of KakaoTalk friends' phone numbers is exported (see Fig. 2(a)).
3) The added phone numbers are removed from the address book.
4) The registered friends' display names on the KakaoTalk application are changed to their original names via contacts sync.

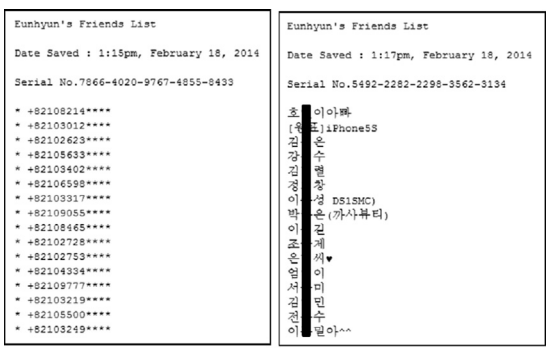

Fig. 2 – Examples of exported contacts information by using the export feature.

Please cite this article in press as: Kim E, et al., Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.008

5) By using the export feature, the list of KakaoTalk friends' original display names is exported (see Fig. 2(b)).
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 Fig. 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. Since the last four-digits in the generated phone numbers are already known to the attacker, the attacker can obtain the phone numbers even when those digits are hidden. 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. Fig. 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:

1) A phone number in a valid format is generated and added into the address book.
2) The list of friends is synchronized with the added phone number for the automated friends registration.
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.
4) If there exists a new friend, the friend is blocked in sequence to find his/her original display name.
5) The screen of blocked friends is captured.
6) The victim's display name is extracted from the captured image by using OCR software.
7) The combination of the entered phone number and the recognized display name is stored as the output of the attack.
8) This process is repeated with another phone number over and over, until there is no new phone number.

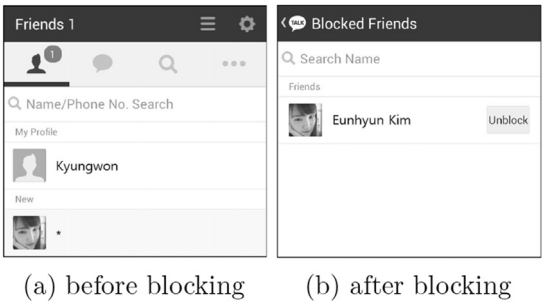

Fig. 3 – An example of the displayed user name in the list of blocked friends.

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.

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. Fig. 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

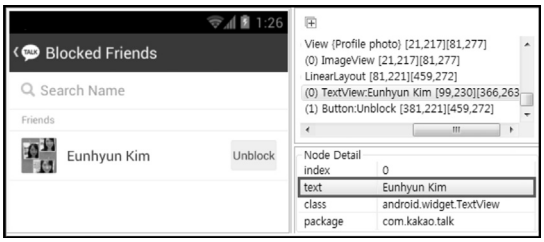

Fig. 4 – An example of the victim's name in a memory dump.

Please cite this article in press as: Kim E, et al., Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.008

is used to directly retrieve the victim's display name from the memory dump instead of using OCR software (see the 5th and 6th steps in Section 3.2).

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 512 MB 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 4 GB 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. To improve the accuracy of text recognition in the OCR 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 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. Our observations about the tested are summarized in Table 1. This table shows the mean time spent in each step, the mean total time (per phone number) to complete an attack attempt, and the accuracy of guessed profile name, respectively, for enumeration attacks. *Accuracy* is defined as "the number of profile names correctly guessed" divided by "the number of obtained user profiles".

As apparent in Table 1, the 'use of the export function' (**Export**) outperformed the other *enumeration* attacks in terms of speed. In this attack, the mean time required for a user's data is about 0.26 s. This is because 5000 phone numbers can be processed in a single batch using the contact export functions. Unlike **Export**, the other enumeration attacks were performed with an individual phone number. However, KakaoTalk recently removed the export feature for security reasons. Hence, this attack is no longer available.

Table 1 – Performance of *enumeration* attacks.

	Spent time (s)		
	Export	OCR	Debugging
Step 1	0.26	0.19	0.24
Step 2	<0.01	9.12	9.05
Step 3	<0.01	3.92	4.22
Step 4	<0.01	7.11	7.21
Step 5	<0.01	1.93	7.51
Step 6	<0.01	24.69	4.33
Step 7	–	4.11	–
Total	**0.26**	**51.07**	**32.56**
Accuracy	1.00	0.32	1.00

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 **Debugging**, the mean time required for a user's data is about 32.56 s — this enable us to collect around 2654 KakaoTalk users' personal data within a day — while the mean time required for a user's data is about 51.07 s when we use **OCR**. In **OCR**, the 6th step requires much computational and communication time for using the OCR web service at Naver. In **Debugging**, that step can be performed by extracting the display name from the memory area of the layout without any communication cost. Although the whole process of **OCR** and **Debugging** attacks in our implementation were sequentially performed, those attacks can be completely distributed over n parallel processors by partitioning the phone number space. Also, some operations can be pipelined so that an attacker can concurrently process more than one attempt.

Finally, we analyze how many KakaoTalk users set real or real-like names as their display names in user profiles. The collected users' profiles were carefully examined (with manual checking) with commonly used name formats in Korea. We found 36,817 out of 50,567 (72.8%) users have used real or real-like names in their profiles. 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, spear phishing, or profile cloning attacks (Bilge et al., 2009).

5. Countermeasures

In this section, we suggest three potential countermeasures to mitigate *enumeration* attacks in KakaoTalk. Since these methods are tackling different aspects of the system design, instead of recommending a single approach, we discuss those countermeasures without any preferences.

Please cite this article in press as: Kim E, et al., Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.008

Table 2 – Activities and their corresponding symbols for an *enumeration* attack scenario.

# of steps	Attacker's activity	Place	Symbol
1	Adding a target phone number into contacts	Smartphone	–
2	Synchronizing the list of friends with the updated contacts	KakaoTalk	A
3	Blocking the newly added friend	KakaoTalk	B
4	Viewing the list of blocked friends	KakaoTalk	C
5	Unblocking the friend from the list of blocked friends	KakaoTalk	D
6	Re-adding the friend into the list of friends	KakaoTalk	E

5.1. *Detecting attack patterns*

There are two representative schemes to detect attack patterns by monitoring events generated by a KakaoTalk application, respectively: (1) misuse detection and (2) anomaly detection. We discuss how to apply these schemes in practice.

5.1.1. *Misuse detection*

If a traffic pattern for an *enumeration* attack is already known, we can develop a system to efficiently detect that pattern by monitoring messages delivered from a KakaoTalk application.

To show the feasibility of this idea, we here design a prototype detector with a known signature for the discovered *enumeration* attack. We note that KakaoTalk already applied a patch to fix our discovered attacks in Section 3 after we reported them (see Section 6). However, we found another new *enumeration* attack scenario using "unblocking a friend" after the patch (see Table 2).[1]

In our prototype implementation, we used the signature for this new attack scenario. That is, when a sequence of those activities (A–B–C–D–E) performed by a KakaoTalk application is observed within a certain time-window, we assume that the corresponding attack was attempted.

To check whether the attack sequence occurs, we used a sequence alignment algorithm (Needleman and Wunsch, March 1970) which is frequently used in Bioinformatics to analyze biological sequence data – sequence alignment is a process of arranging two or more sequences placed one below each other with a scoring scheme which rewards with a positive score those positions at which the sequences agree and with a negative score (a penalty) those positions where there is a disagreement ('mutation') and insertion of a blank ('gap'). Here, we use a simple scoring scheme (+1 for matched events; otherwise, 0). Therefore, we can detect an attack sequence if the sequence has a score which is equal to 5. Surely, this technique can later be refined to improve the attack detection rate.

To avoid this detection strategy, an attacker's best response is to perform the sequence of activities for an *enumeration* attack across multiple time-windows rather than within a single time-window. That is, the attack efficiency depends on time-window size (the time for performing an attack is greater than the time-window size to avoid detection by the proposed system). As the window size increases, the attack efficiency decreases, but the false alarm rate might increase because user's normal activities such as "blocking a friend" and " viewing the list of friends" could be misclassified as an *enumeration* attack. Therefore it is challenging to set a reasonable time-window size to make *enumeration* attacks more expensive (i.e., slowing down the attack speed) while guaranteeing a low false alarm rate for each normal user.

We conducted experiments to show that the time-window size can be set adaptively based on how many the related user activities are likely to be generated. For realistic experiments, we generated synthetic datasets having the statistical properties of real users' normal activities. For doing this, we collected user activities from 7 real KakaoTalk users and analyzed the statistical characteristics (i.e., the average number of user activities per day and the average time interval between two consecutive activities) of the collected activities. Since we cannot directly access the underlying database at server-side, we monitored user activities at client-side by capturing their network traffic (with Wireshark, https://www.wireshark.org/). We found that the KakaoTalk activities used for the tested enumeration attack (i.e., A–B–C–D–E) can be uniquely identified with their payload sizes. With the collected dataset, we can see that 323 activities on average were generated per day and the mean time interval between those activities is about 38 s. Therefore, in our experiments, 323 user activities were artificially generated from a Poisson distribution (Haight, 1967) with the mean time interval of 38 s.

In the sequence of 323 activities, we randomly replaced $5 * K$ activities with K attack sequences (A–B–C–D–E) by assuming the worst case scenario — $5 * K$ normal activities, which are identical to those in *enumeration* attacks, can be sequentially undertaken in a day for a normal user although we did not find a signature of *enumeration* attack from real KakaoTalk users' activity records that we looked at. We evaluated how many false alarms could be generated with a specific time-window size and K. In each sequence representing a normal user, a false alarm occurs when an attack signature (A-B-C-D-E) is observed within the time-window. We repeated the simulations 2000 times. Fig. 5 shows the *attack detection rate* (ADR) results of the proposed detection scheme (for $K = 1$, 4, 7 and 10) with varying time-window size from 1 s to 4 h. *Attack detection rate* is defined as one minus the fraction of false alarms.

In Fig. 5, we can see that all ADR values decrease dramatically when the time-window size is greater than a specific threshold. For example, when $K = 1$, the curve has a gentle slope until the time-window size is around 1 h then plunges toward 0 when the time-windows size is greater than 1 h. We can see the similar trend from the other curves with $K = 4$, 7 and 10 although their scales are significantly different.

In Table 3, we can also see the exact time-window size for achieving the ADR of 0.90, 0.95, and 0.99, respectively, when

[1] That is, the attack sequence here is not consistent with that in Section 3.

Please cite this article in press as: Kim E, et al., Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.008

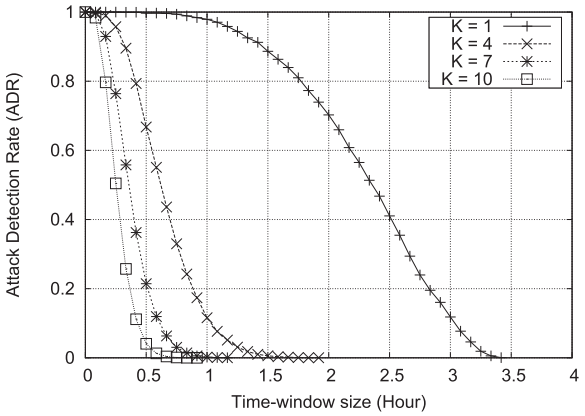

Fig. 5 – Attack detection rate results with varying time-window size and *K*.

$K = 1$, 4, 7 and 10. For example, when $K = 1$, the time-window size achieving the ADR of 0.95 is about 73 min while that size is about 16 min only when $K = 4$. That is, if we use 73 min as the time-window size for detecting *enumeration* attacks, in theory, the proposed misuse detection might slow down the performance of those attacks at least about 135 times compared with the case (about 32.56 s) without considering any defense mechanisms.

We also discuss how the detection accuracy may change with *K*. As *K* increases, a proper time-window size should be smaller to effectively reduce the number of false alarms. Probably, we can expect that misuse detection is not effective if normal users performed activities similar to attack signatures. However, we believe that those user activities (e.g., "blocking a friend", "unblocking a friend" or "re-adding the previously blocked friend") are not common in practice.

5.1.2. *Anomaly detection*

Another possible approach is to use *anomaly detection* since misuse detection systems can detect only known attacks for which they have a defined signature. Anomaly detection is used to recognize the presence of an unusual and potentially hazardous state through analyzing the characteristics of activities.

In general, 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 — a sequence of specific activities generated by an *enumeration* attack would be periodically repeated since an *enumeration* attack attempt should be performed within a very short time period. For example, if we find a series of "blocking a friend" and "unblocking a friend" operations from the collected users' activities, we can consider such activities as (unknown) attack attempts because those are different from the normal activity patterns. Therefore, we suggest designing an anomaly detection system to find a significant attack pattern from such repeated queries used in *enumeration* attacks. Unlike the misuse detection approach, an anomaly detection technique might be effective to detect novel *enumeration* attack sequences.

Table 3 – Time-window size with *K* and attack detection rate (m: minutes).

ADR / K	0.99	0.95	0.90
$K = 1$	40 m	73 m	85 m
$K = 4$	10 m	16 m	19 m
$K = 7$	6 m	9 m	11 m
$K = 10$	4 m	6 m	8 m

5.2. *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).

5.3. *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.

6. Ethical issue

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 profile name is not revealed anymore by the proposed attacks in this paper. However, we have found a side-effect

Please cite this article in press as: Kim E, et al., Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.008

from the patch that still revealing the user's 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.

7. Related work

7.1. Enumeration attack in IM application

Schrittwieser et al. (2012) 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 h. We extend this work by introducing several *enumeration* attacks targeting KakaoTalk, which is widely used in Korea.

A similar problem related to *enumeration* attack was already reported in social networks. Balduzzi et al. (2010) 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 and Acquisti (2005) 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. Luo et al. (2009) 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 and Greenberg (2005) 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%).

7.2. Detecting bots

Enumeration attacks are generally implemented through a bot that automatically performs some malicious actions by mimicking valid human activities (e.g., sending queries about e-mail addresses). Therefore we need to look at previous studies about detecting bots.

Detecting bots is still a challenging task. A commonly used technique is to apply conventional classification algorithms based on the training datasets of normal users and bot commands. That is, detectors are usually implemented based on the prior knowledge of the bot activity characteristics.

In many different applications, anomaly detection techniques were proposed to distinguish human users from bots. For example, Wang et al. (2013) introduced a system to detect automated activities in an online social network using server-side event models. Dave et al. (2012) developed an automated detection system for advertising networks to proactively detect click fraud attacks. Yu et al. (2010) also proposed a system for automatically identifying (search) bot traffic from query logs. For *enumeration* attacks, we can also build a similar system (at the server-side) to detect suspicious activities used in those attacks.

8. 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 Korea. 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%) have used real or real-like names in their profiles, serious invasions of privacy might be raised by the discovered *enumeration* attacks. To make matters worse, the leaked personal information could be misused for sophisticated spam, spear phishing, or profile cloning attacks (Bilge et al., 2009).

To mitigate such *enumeration* attacks, we suggest the three reasonable countermeasures. Among them, we particularly show the feasibility of the misuse detection technique through intensive simulation results on synthetic datasets generated with real KakaoTalk users' behaviors. Our results show the trade-off between the cost of attacks and the false alarm rate for normal user activities, and demonstrate that the proposed system might be capable of maximizing the attacker's cost with a low false alarm rate.

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.

Acknowledgments

This research was partly supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1008) supervised by the IITP (Institute for Information & communications Technology Promotion). This research was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2014R1A1A1003707), and 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

Balduzzi Marco, Platzer Christian, Holz Thorsten, Kirda Engin, Balzarotti Davide, Kruegel Christopher. Abusing social networks for automated user profiling. In: Proceedings of the

13th International Symposium on Recent Advances in Intrusion Detection. Springer; 2010.
Bilge Leyla, Strufe Thorsten, Balzarotti Davide, Kirda Engin. All your contacts are belong to us: automated identity theft attacks on social networks. In: Proceedings of the 18th International Conference on World Wide Web; 2009.
Dave Vacha, Guha Saikat, Zhang Yin. Measuring and fingerprinting click-spam in ad networks. In: Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication; 2012.
Gross Ralph, Acquisti Alessandro. Information revelation and privacy in online social networks. In: Proceedings of the ACM workshop on Privacy in the electronic society; 2005.
Haight Frank Avery. Handbook of the Poisson distribution. Wiley; 1967.
Khan Jeeshan. Kakaotalk launches official mac app. 2014. http://tropicalpost.com/kakaotalk-launches-official-mac-app/.
Luo Wanying, Xie Qi, Hengartner Urs. Facecloak: an architecture for user privacy on social networking sites. In: Proceedings of the IEEE International Conference on Computational Science and Engineering; 2009.
Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol March 1970;48(3):443–53.
Schrittwieser Sebastian, Frühwirt Peter, Kieseberg Peter, Leithner Manuel, Mulazzani Martin, Huber Markus, et al. Guess Who's texting you? Evaluating the security of smartphone messaging applications. In: Proceedings of the 19th Annual Network & Distributed System Security Symposium; 2012.
Smale Stephanie, Greenberg Saul. Broadcasting information via display names in instant messaging. In: Proceedings of the international ACM SIGGROUP conference on Supporting group work; 2005.
Wang Gang, Konolige Tristan, Wilson Christo, Wang Xiao, Zheng Haitao, Zhao Ben Y. You are how you click: Clickstream analysis for sybil detection. In: Proceedings of the 22nd USENIX Conference on Security; 2013.
Yu Fang, Xie Yinglian, Ke Qifa. SBotMiner: large scale search bot detection. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining; 2010.

Eunhyun Kim is a graduate student at Sungkyunkwan University, Suwon, Rep. of Korea. Her research interests include usable security, and user privacy. She received her BS in Computer Engineering from the Dongguk University, Gyeongju, Rep. of Korea, in 2011.

Kyungwon Park is a graduate student at Sungkyunkwan University, Suwon, Rep. of Korea. His research interests include system security, and network security. He received his BS in Computer Science from the Sungkyunkwan University, Suwon, Rep. of Korea, in 2015.

Hyoungshick Kim is an assistant professor at Sungkyunkwan University, Suwon, Rep. of Korea. His research interests include security engineering, usable security, and social computing. He received his PhD in Computer Science from the University of Cambridge, UK, in 2012. After completing his PhD, he worked as a post-doctoral fellow at the University of British Columbia, Canada. He also worked at Samsung Electronics Co., Ltd., Suwon, Rep. of Korea, researching and developing technologies for trustworthy home networks.

Jaeseung Song is an assistant professor in the Computer and Information Security Department at Sejong University, Seoul, Rep. of Korea. His research areas include IoT/M2M platforms, big data analytics, and the reliability and security of networked software systems. Prior to his current position, he worked for NEC Europe Ltd., Heidelberg, Germany, from 2012 to 2013, as a leading standard senior researcher working on R&D projects and IoT/M2M standardizations. He also worked for LG Electronics, Seoul, Rep. of Korea, from 2002 to 2008, as a senior researcher. He occupied leadership positions in 3GPP and oneM2M standard groups as a rapporteur and contributor. He received his PhD from Imperial College London, UK, in 2012. He received his BS and MS degrees in Computer Science from Sogang University, Seoul, Rep. of Korea, in 2000 and 2002, respectively.

Please cite this article in press as: Kim E, et al., Design and analysis of enumeration attacks on finding friends with phone numbers: A case study with KakaoTalk, Computers & Security (2015), http://dx.doi.org/10.1016/j.cose.2015.04.008