Abusing Social Networks for Automated User Profiling

Marco Balduzzi[1], Christian Platzer[2], Thorsten Holz[2],
Engin Kirda[1], Davide Balzarotti[1], and Christopher Kruegel[3]

[1] Institute Eurecom, Sophia Antipolis
[2] Secure Systems Lab, Technical University of Vienna
[3] University of California, Santa Barbara

Abstract. Recently, social networks such as Facebook have experienced a huge surge in popularity. The amount of personal information stored on these sites calls for appropriate security precautions to protect this data.
In this paper, we describe how we are able to take advantage of a common weakness, namely the fact that an attacker can query popular social networks for registered e-mail addresses on a large scale. Starting with a list of about 10.4 million email addresses, we were able to automatically identify more than 1.2 million user profiles associated with these addresses. By automatically crawling and correlating these profiles, we collect detailed personal information about each user, which we use for automated profiling (i.e., to enrich the information available from each user). Having access to such information would allow an attacker to launch sophisticated, targeted attacks, or to improve the efficiency of spam campaigns. We have contacted the most popular providers, who acknowledged the threat and are currently implementing our proposed countermeasures. Facebook and XING, in particular, have recently fixed the problem.

1 Introduction

With the introduction of social networks such as Facebook, the Internet community experienced a revolution in its communication habits. What initially began as a simple frame for social contacts quickly evolved into massively-used platforms where networking and messaging is only one of the multiple possibilities the users can call upon. While basic messaging is still one of the key features, it is clear that the participants see the main advantage in the well-organized representation of friends and acquaintances.

For such an organization to work properly, it is imperative to have certain knowledge about the participants. Suggesting users from the same area with the same age, for instance, can lead to a renewed childhood friendship, while a detailed work history might open unexpected business opportunities. On the other hand, this kind of information is also of great value to entities with potentially malicious intentions. Hence, it is the responsibility of the service provider to ensure that unauthorized access to sensitive profile information is properly restricted. In fact, various researchers (e.g., [1–3]) have shown that social networks can pose a significant threat to users' privacy. The main problem is twofold:

– Many users tend to be overly revealing when publishing personal information. Although it lies in the responsibility of each individual to assess the risk of publishing sensitive information, the provider can help by setting defaults that restrict

the access to this information to a limited number of individuals. A good example is Facebook, where detailed information is only exchanged between already connected users.
- Information exists in social networks that a user cannot directly control, and may not even be aware of. The best example is the use of the information provided during the registration phase (e.g., name, contact e-mail address, and birthday). Even though this data may never be shown in the public user profile, what most users do not realize is the fact that this information is still often used to provide other functionality within the social network (e.g., such as determining which users might know each other).

In this paper, we describe a novel, practical attack that impacts thousands of users. Moreover, we have shown that this attack is effective against eight of the most popular social networks: Facebook, MySpace, Twitter, LinkedIn, Friendster, Badoo, Netlog, and XING. We discovered that all of these social networks share a common weakness, which is inherent in a feature that is particularly useful for newly-registered users: *Finding friends*. With the functionality to search for friends, social networks need to walk the thin line between revealing only limited information about their users, and simplifying the process of finding existing friends by disclosing the personal details of registered users. A common functionality among these popular social networks is to let users search for friends by providing their e-mail addresses. For example, by entering "gerhard@gmail.com", a user can check if her friend Gerhard has an account on the social network so that she can contact and add him to her friend list. Note that an e-mail address, by default, is *considered to be private information*, and social networks take measures *not to reveal this information*. That is, one cannot typically access a user's profile and simply gain access to his personal e-mail address. One of the main purposes of protecting e-mail addresses is to prevent spammers from crawling the network and collecting e-mail to user mappings. With these mappings at hand, the attacker could easily construct targeted spam and phishing e-mails (e.g., using real names, names of friends, and other personal information [4]). This kind of profiling is also interesting for an attacker to perform a reconnaissance prior to attacking a company. By correlating mappings from different social networks, it is even possible to identify contradictions and untruthfully entered information among profiles.

In our experiments, we used about 10.4 million real-world e-mail addresses that were left by attackers on a dropzone on a compromised machine (which was taken down). We built a system to automatically query each social networking site with these addresses, just as an adversary would, and we were able to identify around 876,000 of these addresses on at least one of the investigated social networks. Furthermore, we implemented a simple guesser that we used to create new e-mail addresses (e.g., for John Doe, addresses such as *john.doe@gmail.com*, *john@gmail.com*, *jdoe@yahoo.com*, etc. would be created) and show that this is an effective and simple technique in practice to find thousands of more accounts.

In summary, we make the following three contributions:

- We describe a real-world, common weakness in eight popular social networks consisting of millions of users, and present a system that automatically takes advantage of this weakness on a large-scale.

– By using e-mail addresses as a unique identifier, we demonstrate that it is possible to correlate the information provided by thousands of users in different social networks. This is a significant privacy threat, because it allows to link profiles that otherwise have no common information. Furthermore, adversaries can leverage this information for sophisticated attacks.
– We present our findings and propose mitigation techniques to secure social networks against such attacks. Our findings were confirmed by all social network providers we contacted. Some of them have already addressed the problem.

The remainder of the paper is structured as follows: In Section 2, we briefly discuss ethical and legal considerations. In Section 3, we explain our attack and how we implemented it for the social networks under examination. In Section 4, we present our findings and assess the potential threat to social networking users. Section 5 discusses possible mitigation solutions. In Section 6, we present related work, with a special focus on privacy-related issues in social networks. We conclude our paper in Section 7.

2 Ethical and Legal Considerations

Crawling and correlating data in social networks is an ethically sensitive area. Similar to the experiments conducted by Jakobsson et al. in [5, 6], we believe that realistic experiments are the only way to reliably estimate success rates of attacks in the real-world. Nevertheless, our experiments were designed to protect the users' privacy.

First, for the crawling and correlation experiments we conducted, we only accessed user information that was publicly available within the social networks. Thus, we never broke into any accounts, passwords, or accessed any otherwise protected area or information. Second, the crawler that we developed was not powerful enough to influence the performance of any social network we investigated. Third, we used MD5 on the real names of users to anonymize them properly and handled this data carefully.

We also consulted the legal department of our university (comparable to the IRB in the US), and received a legal statement confirming that our privacy precautions were deemed appropriate and consistent with the European legal position.

3 Abusing E-Mail Querying

Many social network providers such as Facebook, MySpace, XING, or LinkedIn offer a feature that allows a user to search for her friends by providing a list of e-mail addresses. In return, she receives a list of accounts that are registered with these e-mail addresses. From a user's point of view, this feature is valuable: A user can simply upload her address book, and the social network tells her which of her friends are already registered on the site. The feature enables a user to quickly identify other users she knows, and with which she might be interested in establishing a connection.

While the e-mail search functionality commonly available in social networks is convenient, a closer examination reveals that it also has some security-relevant drawbacks. We show that an attacker can misuse this feature by repeatedly querying a large number of e-mail addresses using the search interface as an oracle to validate users on the social network. This information can then be abused in many ways, for example:

– A spammer can automatically validate his list of e-mail addresses (e.g., find out which addresses are most probably real and active) by querying a social network, and only send spam e-mails to those users [7].
– The previous attack can be combined with *social phishing*, i.e., the spammer crawls the profile of a user and uses this information to send targeted phishing e-mails (if the user has a public profile and a public friend list) [4].
– An attacker can generate detailed profiles of the employees of a company and use this information during the reconnaissance phase prior to the actual attack.

Note that it has been recently reported that spammers have started to shift their attention to social networking sites to collect information about users that they can then use for targeted e-mails [8]. The report states that spammers have been using bots to spy information from social networks that they can then use for launching attacks such as guessing passwords (i.e., using reminder hints such as "What is my favorite pet?"). The prerequisite for these current attacks, however, is that a bot is installed on the victim's machine. In comparison, we describe the exploitation of a common weakness in a social network functionality that allows us to retrieve information about users even if they are not infected by a bot.

In each of these cases, the attack is only feasible since the social network provider enables a large-scale query of e-mail addresses. Before going into details on how this feature can be abused in practice, we provide an overview of the context of this type of attacks and previous instances of similar problems.

3.1 Historical Context

Historically, a user search/verification feature was available in many different protocols and services, as we discuss in this section.

SMTP. The *Simple Mail Transfer Protocol* (SMTP) provides two commands, `VRFY` and `EXPN`, to verify a user name or to obtain the content of a mailing list, respectively [9]. A `VRFY` request asks the mail server to verify a given e-mail address, and if a normal response is returned, it must include the mailbox of the user. In addition, an `EXPN` request asks the server for the membership in a mailing list, and a successful response must return the mailboxes on the mailing list.

Spammers began to abuse these two commands to query mail servers for a list of valid e-mail addresses, and to verify if a given e-mail address was in use. Due to this abuse by spammers, SMTP servers now commonly do not provide these two commands anymore (at least not to unauthenticated users).

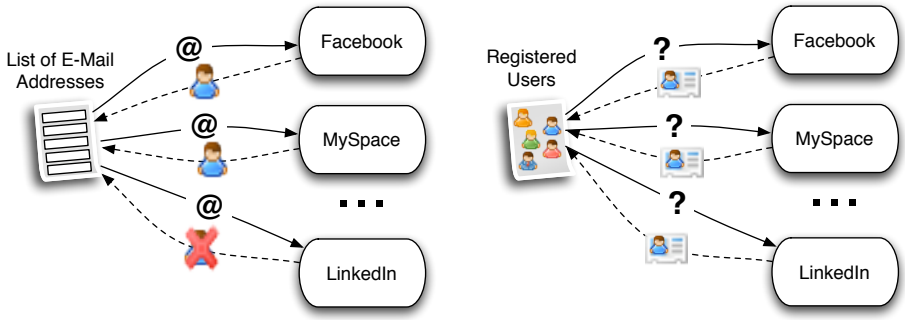
Finger User Information Protocol. This protocol is used to query a remote server for status and user information [10]. The *finger daemon* typically returns information such as the full name, whether a user is currently logged-on, e-mail address, or similar data. While providing this kind of information is useful in general, an attacker can collect information about a specific user based on the finger protocol, and then use this information for social engineering attacks. Furthermore, the public exposure of the information is questionable from a privacy and security point of view. For these reasons, the majority of Internet hosts does not offer the finger service anymore.

Secure Shell. Certain versions of the OpenSSH server allowed a remote attacker to identify valid users via a timing attack: By analyzing the response time during authentication, an attacker could determine whether or not the supplied username is valid [11]. By adjusting the timing for both successful and failed user verification, this flaw was fixed. A similar flaw can be used to reveal private information with the help of timing attacks against web applications [12].

Note that, as discussed above, the conceptual problem that we address in this paper is not necessarily new, but its implications are novel and are far greater because of the large amount of sensitive information contained in user profiles on social networks. We believe that history is repeating itself and that it is only a matter of time before attackers start making use of such features for malicious purposes.

3.2 Automated Profiling of Users

As explained previously, a user can typically send a list of e-mail addresses to a social network and, in return, she receives a mapping of which of these e-mail addresses have a corresponding account on the site. An attacker can abuse this and query for a large number of e-mail addresses on many different social networks (see Figure 1a). As a result, she learns on which social networks the specific address is registered.



(a) Querying social networks for registered e-mail addresses on a large scale.

(b) Crawling every profile found in the first step to collect personal information.

Fig. 1: Automated user profiling based on information collected on social networks.

In the second step, the attacker retrieves the user's profile from the different networks in an automated way (see Figure 1b). From each profile, she extracts the (publicly-accessible) information she is interested in, for example, age, location, job/company, list of friends, education, or any other information that is publicly available. This information can then be aggregated and correlated to build a rich user profile.

Throughout the rest of this paper, we show that the two steps can indeed be automated to a high degree. Furthermore, we demonstrate that this attack is possible with

only very limited resources. In fact, by using a single machine over a few weeks only, we collected hundreds of thousands of user profiles, and queried for millions of e-mail addresses (i.e., each social network was successfully queried for 10.4 million addresses, adding up to a total of about 82.3 million queries). This emphasizes the magnitude and the significance of the attack since a more powerful, sophisticated, and determined attacker could potentially extract even more information (e.g., by using a large botnet).

An attacker can also abuse the search feature in a completely different way, extending the attack presented in the previous section. During the profiling step, an attacker can learn the names of a user's friends. This information is often available publicly, including social networking sites such as Facebook and Twitter. An attacker can thus obtain the tuple (first name, last name) for each friend of a given user, but not the e-mail addresses for these friends: The e-mail address itself is considered private information and not directly revealed by the social networking sites. However, an attacker can automatically try to guess the e-mail addresses of the friends of a user by abusing the search feature. We implemented two different, straight-forward techniques for generating new e-mail addresses, based on user names.

For the first technique, for each friend, we build 24 addresses. Given a name in the form *"claudio bianchi"*, we generate six prefixes as *"claudio.bianchi," "claudiobianchi," "claudio_bianchi," "c.bianchi," "c_bianchi,"* and *"cbianchi"*. Then, we append the four most popular free e-mail domains *"gmail.com," "yahoo.com," "aol.com,"* and *"hotmail.com."*

For the second technique, we use context information for generating e-mail addresses: If a user has an e-mail address with a certain structure (e.g., automatically generated e-mail accounts often include the last name of the user and a static prefix), we try to detect this structure by searching the user's first and last name within the e-mail address. If we identify a pattern in the address, we use this match and generate two additional e-mail addresses that follow the same pattern (including both the first and last name) for each friend. If we do not detect a pattern, we generate e-mail addresses similar to the first algorithm. However, instead of appending common prefixes, we use the prefix of the user on the assumption that the friends of a user might be a member of the same e-mail provider.

3.3 Implementation of the Attack

Our prototype system has been implemented as a collection of several components. One component queries the social networks, one extracts and stores the identified information from user profiles, and one automatically correlates the information to discover as much information as possible about a user. An overview of the system and the relationship of the components is shown in Figure 2.

We designed our system to be efficient and stealthy at the same time. Therefore, we had to find a compromise between normal user behavior, which is stealthy, and brute-force crawling, which is efficient but bears the danger of frequently-suspended accounts. Our solution was tweaked for each social network, to find the right combination of timeouts and number of requests. Furthermore, our solutions was carefully designed not to overwhelm the tested networks.

In the following, we describe the system and its components in more detail.

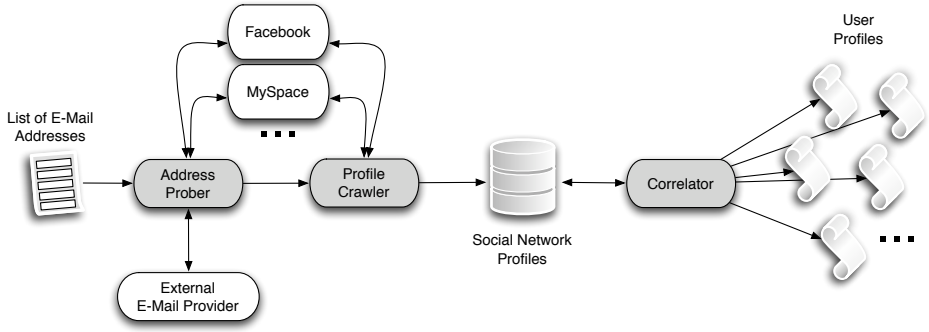


Fig. 2: Overview of system architecture.

Address Prober. The *Address Prober* is an HTTP client that is responsible for uploading the list of e-mail addresses to be queried to the social network. The social network, in return, sends back the list of accounts that are registered with those addresses. The data we are interested in is the profile ID and, if possible, the name, which is attached to the source e-mail address. At this point, some of the processed networks offer some additional data, such as the location or a brief job description.

The client itself is implemented in Python, and embeds an extension to the standard `urllib` library [13] that supports postings in the multipart/form-data format. We adopted such encoding to be able to send to the social networking site a file containing the list of addresses. Typically, this file is required to be formatted in the standard CSV format. On some other networks, for example in Badoo, the list of emails need to be passed as a string of comma-separated addresses.

The Address Prober also supports external e-mail providers such as, for example, Google's webmail service Gmail, and permits to upload lists of e-mail addresses to such accounts. The motivation behind this feature is that some social networks only support e-mail queries if the source is an external e-mail account with an attached address book. Hence, we automatically upload (and afterwards delete again) contacts from specific webmail accounts, before querying the social network.

With this technique, in the worst case (i.e., some sites such as Facebook allow lookups of up to 5,000 addresses), we are able to check sets of around 1,000 e-mail addresses at once. With a short delay, which we set to 30 seconds to ensure that all data is correctly processed and not to overwhelm the network, the prober is able to process data with an overall average speed of 500,000 e-mail addresses per day. A positive side-effect of this technique is that we can query social networks that support Gmail imports in parallel, resulting in a higher overall throughput.

Profile Crawler. The *Profile Crawler* is responsible for a deeper investigation of the user profiles discovered in the previous step. The goal is to gather as much information about a single user as possible. For this purpose, it is mandatory to implement tailored solutions for each supported social network. In the first round, the crawler visits iteratively the user's profile pages for all the social networks, and stores them in a

database. On average, we were able to visit 50,000 pages in a single day from a single machine with a single IP address. Some networking sites provided mechanisms to limit the number of profiles visited per day from a single account, while others did not have any limitation mechanism in place. Finally, the crawler automatically parses the data that has been retrieved and extracts the information of interest, such as sex, age, location, job, and sexual preferences. That is, the Profile Crawler enriches the user profiles discovered by the Address Prober with a set of general and sensitive information.

Correlator. After the crawling phase, the *Correlator* component combines and correlates the profiles that have been collected from the different social networks. The goal of the Correlator is to use the email address as a unique identifier to combine together different profiles and identify the ones that belong to the same person.

When it finds two profiles associated with the same e-mail address, the Correlator compares all the information in the two profiles to identify possible inconsistencies. In particular, it compares all the fields that can assume a small set of values, e.g., sex (either male or female), age (a positive integer number), and current relationship (married, single, or in a relationship).

Using the Correlator, it is possible to automatically infer information that the user might have wanted to keep private. In particular, the correlator has two main goals:

- *Identity Discovery* - If a person provides his full name in social network A, but registers a profile in the network B using a pseudonym, by cross-correlating the two profiles, we can automatically associate the real user's name also to the account B. We are even able to correlate the information about a given person that uses two different pseudonyms by linking the two accounts with the help of the provided e-mail address, which is not possible with the technique proposed by Irani et al. [14]. The combination of information from different sources can be even more worrisome if this information is privacy-relevant. For example, Alice could have a business profile on LinkedIn, and another profile on a dating site in which she does not reveal her real name, but she provides other private information such as her sexual preferences. It is very likely that Alice assumed that it was not possible to link the two "identities" together because there is no public information on the two networks that can be used to match the profiles.
- *Detection of Inconsistent Values* - Sometimes, the information extracted from different social networks is contradictory. For example, Bob can provide his real age on his profile on social network A, while pretending to be 10 years younger on social network B. Again, we can identify this kind of fraudulent (or embellished) profiles in an automated way by cross-correlating the information extracted during crawling the different networks.

4 Evaluation with Real-World Experiments

We performed several experiments on different social networks. As a starting point, we used a set of 10,427,982 e-mail addresses, which were left on a dropzone on a compromised machine that was taken down by law enforcement officials. Based on

the log files of this machines, we saw that these e-mail addresses had been used for spamming, and thus, they provided a real-world test case for our system.

4.1 Results for E-Mail Queries

We used our *Address Prober* component on eight social networks, namely Facebook, MySpace, Twitter, LinkedIn, Friendster, Badoo, Netlog, and XING. These networks were chosen because they provide a representative mix of different types of social networks (e.g., business, friendship, and dating). Furthermore, all of these sites have millions of registered users and operate in different countries. Of course, they also vary in their popularity. Facebook, for example, is the most popular social networking site and reports to have more than 400 million active users [15].

	Network	Query method	E-mail list length	# queried e-mails	# identified	Percentage
		method	*size efficiency*	*speed efficiency*	accounts	
1	Facebook	Direct	5000	10M/day	517,747	4.96%
2	MySpace	GMail	1000	500K/day	209,627	2.01%
3	Twitter	GMail	1000	500K/day	124,398	1.19%
4	LinkedIn	Direct	5000	9M/day	246,093	2.36%
5	Friendster	GMail	1000	400K/day	42,236	0.41%
6	Badoo	Direct	1000	5M/day	12,689	0.12%
7	Netlog	GMail	1000	800K/day	69,971	0.67%
8	XING	Direct	500	3.5M/day	5,883	0.06%
				Total of	1,228,644	11.78%

Table 1: Discovered profiles

Table 1 shows the profiles that have been discovered by the e-mail queries that we performed on these sites. Clearly, direct queries to the social networking sites yield faster results than those that are coupled with GMail accounts. Also, while we were able to query 5,000 e-mail addresses at once on Facebook, the best results for XING were 500 addresses per query. The scan method and e-mail list length directly affect the speed of the queries. In general, direct queries are about one order of magnitude faster, and we can check several million e-mail addresses per day. For social networks on which we need to use the GMail support, we can still probe several hundred thousand addresses per day. Also, note that we only adopted a single machine in our tests, while an attacker could perform such an attack in parallel using many machines. In total, we were able to identify 1,228,644 profiles that are linked to one of the e-mail addresses we probed. Most profiles were found on Facebook (4.96%), LinkedIn (2.36%), and MySpace (2.01%).

Table 2 shows the number of profiles that were created with the same e-mail address on different networks. For example, one can see that there are almost 200,000 users who were registered in at least two social networks. In sum, a total of 876,941 unique e-mail addresses we had access to were covered by one or more of the listed social networks.

Number of Social Networks	Number of Profiles
1	608,989
2	199,161
3	55,660
4	11,483
5	1,478
6	159
7	11
8	0
Total unique	876,941

Table 2: Overlap for profiles between different networks.

Combination	Occurrences
Facebook - MySpace	57,696
Facebook - LinkedIn	49,613
Facebook - Twitter	25,759
Facebook - MySpace - Twitter	13,754
Facebook - LinkedIn - Twitter	13,733
Facebook - NetLOG	12,600
Badoo - FriendSter	11,299
Facebook - MySpace - LinkedIn	9,720
LinkedIn - Twitter	8,802
MySpace - Twitter	7,593

Table 3: Top ten combinations.

Table 3 shows the top ten combinations among social networks. That is, the table shows which combinations of networks we typically encountered when we identified a user who is registered on different sites with the same e-mail address. The two most popular combinations are Facebook with MySpace, and Facebook with LinkedIn. Note that the more diverse information a social networking site offers about users as public information, the more significant our attack becomes. In the case of LinkedIn and Facebook, we have two social networking sites with different goals. Whereas Facebook aims to foster friendship, LinkedIn aims to foster business collaborations. Hence, we can combine the business information about a user with the more personal, private information they may provide on the friendship site (e.g., under a nickname).

These results of our experiment clearly demonstrates that a potential attacker can easily abuse social networks to enrich his spamming list with the information retrieved from different networks.

4.2 Extracted Information from Profiles

In this section, we provide statistics about the information collected when the *Profile Crawler* visited the user profiles. We present for each of the social networks an overview of what kind of information is available, and also for what percentage of users we were able to extract this information.

Network	Name Surname	Profiles are open	Photo	Location	Friends	Average friends	Last login	Profile visitors
Facebook	✓	99.89	76.40	0.48	81.98	142	n/a	n/a
MySpace	✓	96.26	55.29	63.59	76.50	137	94.87	n/a
Twitter	✓	99.97	47.59	32.84	78.22	65	n/a	n/a
LinkedIn	✓	96.79	11.80	96.79	96.75	37	n/a	n/a
Friendster	✓	99.72	47.76	99.51	50.23	37	8.79	n/a
Badoo	✓	98.61	70.86	95.23	n/a	n/a	92.01	n/a
Netlog	✓	99.98	43.40	77.54	64.87	31	n/a	73.33
XING	✓	99.88	57.20	96.04	47.25	3	n/a	96.83

Table 4: Crawling results (values are in percentage): general information

	Age	Sex	Spoken language	Job	Education	Current relation	Searched relation	Sexual preference
Facebook	0.35	0.50	n/a	0.23	0.23	0.44	0.31	0.22
MySpace	82.20	64.87	n/a	3.08	2.72	8.41	4.20	4.07
Twitter	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
LinkedIn	n/a	n/a	n/a	96.79	60.68	0.00	n/a	n/a
Friendster	82.97	87.45	n/a	30.88	2.72	64.59	77.76	n/a
Badoo	98.61	98.61	47.81	17.06	19.92	22.48	n/a	22.80
Netlog	97.66	99.99	44.56	43.40	1.64	25.73	23.14	29.30
XING	n/a	n/a	84.54	99.87	49.21	n/a	n/a	n/a

Table 5: Crawling results (values are in percentage): sensitive information

Table 4 provides an overview of *general information* such as profile photo, location, and friends available on the different networks. The column *profiles are open* shows the percentage of how many profiles the crawler was able to access, and validate against the name and surname already extracted from the Address Prober. Profiles that are closed include profiles that are configured to be completely private, or that are not accessible anymore. In Facebook, more than 99% of the profiles are open, but only little information is shown by default to anonymous users and persons that are not a friend of the user. On the contrary, the profile photo and the list of friends are usually accessible.

Typically, the different pieces of information can be either private or public, and the social network provider assigns a default value for the privacy setting of them. From our observations, it seems that many users do not change the default privacy settings for a specific type of information. Note that when some data is not accessible, it either means that the user has not set such a value (it is optional) or that is not accessible due to privacy reasons.

Table 5 shows the availability of *sensitive information* on the individual networks. Depending on the purpose of the social network, different types of information are made public by default. Dating platforms, for instance, focus on personal information such as age, sex, or current relationship status, while business networks emphasize educational and work-related aspects. However, all of these details are more sensitive and can be used for the accurate profiling of an account. Precise values such age and sex can easily

Network	Personal homepage	Phone	Birthday	IMs	Physical appearance	Income	Prof. skills	Interests Hobbies
Facebook	✓	✓	✓	✓			✓	✓
MySpace			✓		✓	✓		✓
Twitter	✓						✓	✓
LinkedIn	✓	✓	✓	✓			✓	✓
Friendster								✓
Badoo	✓			✓	✓			✓
Netlog						✓		✓
XING	✓	✓	✓	✓			✓	✓

Table 6: Crawling results: extra information

be correlated across different social networks, essentially forming richer sets of publicly available data than initially desired by the user. We provide a detailed overview of this aspect in Section 4.4.

Finally, Table 6 shows what kind of *additional information* each social network supports. We refrain from providing a percentage for these fields, because this type of information is only available for a minority of the sampled cases.

4.3 Automated Guessing of User Profiles

While it is useful for the attacker to have access to e-mail lists that she can readily query, it is also interesting for her to automatically generate new e-mail addresses that she could then re-validate against the social networks. Using the *e-mail guesser* as discussed earlier, we are able to generate addresses that we do not previously know, and verify their existence in social networks. By starting with 650 profiles and using straight-forward automated e-mail guessing techniques, we were able to identify the e-mails of about 20,000 users along with their associated profiles (a thirty-fold increase compared to the initial profile set). Hence, our experiment demonstrated that even if the attacker does not have access to a large e-mail database, by starting with a small set, she can still successfully guess addresses and identify thousands of new profiles.

4.4 Detecting Anomalous Profiles by Cross-Correlation

In the following, we present the output of the correlation phase, and we discuss several interesting examples of anomalous profiles we automatically discovered during our empirical experiments.

Discovering Mismatched Profiles. Based on the data provided by the different social networks, we configured the Correlator to analyze six information fields that are popular among the different social networks we examined: Name, location, age, sex, current relationship, and sexual preference.

Before proceeding to the comparison, we had to normalize the values provided in the different social networks to a common dictionary. For example, sex was translated

to either "male" or "female," while the current relationship and the sexual preference's values were translated into a set of common keywords built from an analysis of the entire dataset. For instance, values like "heterosexual," "straight," and "man looking for women" were all translated into the keyword "heterosexual." Likewise, we normalized the current relationship field to one of the four following values: "Single," "in a relationship," "married," and "complicated." Finally, we filtered the geographical location by comparing the field (or part of it) against a dictionary of more than 260,000 cities.

	# of Occurrences on X networks						
Information	2	3	4	5	6	7	Total
Name	199,161	55,660	11,483	1,478	159	11	267,952
Location	22,583	2,102	174	11	3		24,873
Age	19,135	887	36				20,085
Sex	17,282	854	34				18,170
Sexual preference	760	13					773
Current relation	1,652	38	1				1,691

Table 7: Information provided on multiple profiles belonging to the same user

Table 7 shows the number of users that provide a certain information on multiple social networks. For example, 22,583 users included their location on two networks, 2,102 on three networks, and 174 on four networks. Since the name is a mandatory field, the first line of the table matches the number of profiles reported in Table 2.

For each field, the Correlator computed the total number of distinct values provided by the same users across different networks. For example, if Alice is registered on three social networks where she provides as age 26, 26, and 22 the Correlator reports two mismatched values.

		% Total	% of mismatched values		
Information	Value	mismatches	2	3	4+
Name	*string*	72.65	62.70	35.37	17.66
Location	*city*	53.27	51.74	16.24	3.72
Age	$0 < n < 100$	34.49	33.58	17.84	30.56
Sex	*m, f*	12.18	12.18		
Sexual preference	*hetero, homo, bi*	7.63	7.63		
Current relation	*single, relationship, married, complicated*	35.54	35.42	5.13	

Table 8: Overview of profiles where a mismatch was detected - Data are normalized.

Table 8 summarizes the results. The first column shows the percentage of profiles, from the total shown in Table 7, for which the Correlator found mismatching values. About one-third of the people seems to misrepresent their current relationship status, declaring, for example, to be single on one network and to be married on a second one.

It is also interesting to note that 2,213 users (12% of the ones registered in more than one network) pretend to be male on a network and female on a different one. The very high percentage of people using different names is a consequence of various factors. First, the name comparison is more problematic because, as explained in Section 2, we only store the MD5 of the names to preserve the users privacy. This means that we lose the ability to distinguish between small differences and completely fake names. Second, in some social networks, it is very common to provide a nickname instead of the real user name. For example, John Doe on LinkedIn may appear simply as JDoe77 on MySpace.

The last three columns in Table 8 show how many unique values where provided by the same user (either two, three, or more) on different social networks. These percentages are normalized by the number of accounts owned by the user. That is, a value of 10% in Column 3 means that 10% of the people that own an account on at least three social networks provided three different values for that specific field.

Mismatches in Provided Age Information. Five of the eight social networks we examined either offer the possibility for a user to specify his age, or automatically compute this information from the user's birthday. During our experiments, the *Correlator* automatically found a total of more than 20,000 users for which we had at least two profiles on different networks which also included the person's age. Surprisingly, about one-third of these users (6,919) showed a mismatch for the age information provided in the different profiles (see Table 9 for details). This number only includes those profiles in which the difference of age is at least two years. In fact, a mismatch of only one year is likely to be the consequence of outdated profiles (i.e., if a user needs to manually specify his age, he might forget to update the value at each birthday).

Range	#	%
2 - 10	4,163	60.17
11 - 30	1,790	25.87
31 +	966	13.96
Profiles with Age	20,085	
Total mismatched	6,919	

Table 9: Overview of profiles where a mismatch was detected in the age.

Among the profiles with an age that differs more than two years, we were able to identify 712 users (10% of this set) who claim to be underage, even though they appear to be more than 18 years old in another networks (or vice versa). For example, we observed that many teenagers increase their age to register to Badoo, since the site restricts its access to adults only.

A Short Glimpse into Hidden Profiles. Probably the most serious consequence of the attack presented in this paper is the ability to uncover hidden relationships between different profiles, allowing an attacker to infer private information about a subject.

By looking at the results of our experiments, it is possible to find examples of possibly hidden profiles and online identities that users probably wish to keep secret. As a proof of concept of the impact that correlating profile information can have on a user's privacy, we picked some random profiles that showed mismatching values. In one case, a married person owned an account on a dating-related social network under a different name, with a list of friends who were much younger. While such information may be a complete misinterpretation, nevertheless, there may be many cases where an attacker may try to use the information to his advantage.

Because of the ethically sensitive aspects of analyzing this kind of interconnections, we did not perform an in-depth investigation of the problem, limiting the result of our analysis to aggregated figures.

5 Countermeasures

In this section, we present several mitigation strategies that can be used to limit the extent of our attack. Each approach has its own advantages and limitations, which we review in the rest of the section. We discussed the different countermeasures with several popular social network providers to incorporate also their view of the problem, especially considering the operational practicability of each proposed solution.

1) Raising Awareness: Mitigation From the User's Perspective. Clearly, if users were to use a different e-mail address on each social networking site, it would become more difficult for the attacker to automatically correlate the extracted information. Because the e-mail address is the unique ID that identifies a specific user, an effective defense technique would be to educate users to use a different e-mail address each time they register for and enter personal information into a social networking site. Unfortunately, educating users on security and privacy issues is not an easy task. Often, users may choose to ignore the risks and opt for the ease of use (e.g., analogous to users using the same password across many web sites – which has been reported to be quite common [16]).

2) Possible Solution: CAPTCHAs. When searching for e-mail addresses, a user could be forced to solve a CAPTCHA (i.e., a type of challenge-response test which is hard to solve for a computer [17]). This would prohibit automated, large-scale queries to a certain extent since CAPTCHAs cannot be (easily) solved by a computer.

However, introducing this kind of countermeasure has three main drawbacks. First, the user experience is reduced if a user needs to solve a CAPTCHA frequently, and this should be avoided by all means. Even if solving a CAPTCHA is only required for users that perform many queries, the network operators tend to dislike this mitigation strategy due to a potential negative user experience. Second, using this approach is not a real solution to the problem since an attacker can also hire *real* people to solve the challenge-response tests. This type of service is surprisingly cheap on the underground market, with 1,000 solved CAPTCHAs costing about $2 [18]. Third, different CAPTCHA systems are prone to attack such that a computer can solve the test with a reasonable success rate, completely defeating the countermeasure [19–21].

3) Possible Solution: Contextual Information. Another potential approach to mitigate the problem is to require *contextual information* for each query. If a user U wishes to search for his friends F_1, F_2, ... F_n, he has some context information for each of them that he should include in his query. For example, a user knows the full name of each friend, he can estimate their age, or knows their approximate location. It is probable that the attacker lacks this kind of information.

Unfortunately, it is inconvenient for a user to provide contextual information to perform a query. While a user can, for example, store the full name together with the e-mail address within the address book application, this information might no be available for all friends. Furthermore, additional contextual information such as age or location needs to be provided manually. As a result, this solution is likely not feasible from an operational point of view.

4) Possible Solution: Limiting Information Exposure. Our attack is possible since the search result contains a mapping between the queried e-mail address and the profile name (if an account with this e-mail address is registered). Thus, a viable option to prevent our attack is to not return a mapping between e-mail address and profile name in the search result. This could, for example, be implemented by simply returning a list of registered accounts in a random order, without revealing which e-mail address belongs to which account. Note that a user typically does not need the correct mapping, he is only interested in the fact that one of his friends is registered on the social network such that she can add him to his friends list.

5) Possible Solution: Incremental Updates. Another fact that enables our attack is the huge number of searches we can perform: We can query thousands of e-mail addresses at once, and also repeat this process as often as we wish. A natural approach for mitigation is, thus, to implement some kind of limitation for the queries a user is allowed to perform. For example, by enforcing *incremental updates*, a user is allowed to initially query many e-mail addresses, but this step can only be performed once. This enables a user to search for his current friends on the given social network in the beginning. Afterwards, the number of queries can be restricted to only a small number of e-mail addresses (for example only 50). This enables a user to incrementally extend his network, but also limits the number of e-mail addresses a user can search for.

6) Possible Solution: Rate-limiting Queries. Another viable option to limit our attack is *rate-limiting* the number of queries: That is, we restrict the (total) number of queries a user can send to the social network, therefore limiting the amount of e-mail addresses a given user can check. An option could be to either rate-limit the number of queries (e.g., only two large queries per week) or have a total upper bound of e-mail addresses a user can search for (e.g., a total of 10K e-mail addresses a user can check).

Most social network providers already have different kinds of rate-limiting in place. For example, rate-limiting is used to prohibit automated crawling of their site, or regulating how many messages a given user can send per day to stop spamming attacks. Therefore, rate-limiting the number of e-mail searches a user is allowed to perform fits into the operational framework of these sites. When we contacted the most popular social network providers, the majority of them preferred this solution. In the meantime,

Facebook and XING have already implemented this countermeasure and now limit the number of lookups that can be performed by a single source.

Limitations of the Countermeasures. Note that although there is much room for improvement in defending against e-mail-to-account mapping information leakage attacks, the attacker could still extract information from the social networking site for specific, targeted users (e.g., by only sending e-mail queries consisting of a single user). Hence, if social networking sites choose to provide e-mail searching functionality, there is always a potential for misuse and the privacy of the users may be at risk. However, the countermeasures we described in this section raise the difficulty bar for the attacker, mitigating the problem at least on a large scale.

6 Related Work

The large popularity of social networks and the availability of large amounts of personal information has been unprecedented on the Internet. As a result, this increasing popularity has lead to many recent studies that examine the security and privacy aspects of these networks (e.g., [3, 4, 7, 22–26]). As more and more Internet users are registering on social networking sites and are sharing private information, it is important to understand the significance of the risks that are involved.

The structure and topology of different social networks was examined by different research groups (e.g., [27–30]). The main focus of previous work was either on efficient crawling or on understanding the different aspects of the graph structure of social networks. We extend previous work by contributing a novel way to enumerate users on social networks with the help of e-mail lookups. Furthermore, we implemented several efficient crawlers for social networks and – to the best of our knowledge – we are the first to perform large-scale crawls of eight social networks.

Our attack is facilitated by the fact that an attacker can use an e-mail address to link profiles on different social networks to a single user. The idea of correlating data from different sources to build a user profile has been studied in different contexts before. For example, Griffith and Jakobsson showed that it is possible to correlate information from public records to better guess the mother's maiden name for a person [31]. Heatherly et al. [32], and Zheleva and Getoor [33] recently showed that hidden information on a user's profile can also be inferred with the help of contextual information (e.g., the political affiliation of a user can be predicted by examining political affiliation of friends).

Concurrently and independently of our work, Irani et al. [14] performed a similar study of social networks. They showed that it is straightforward to reconstruct the identify (what they call the *social footprint*) of a person by correlating social network profiles of different networks. The correlation is done either by using the user's pseudonym or by inferring it from the user's real name. In contrast, our work focuses on automated techniques to find profiles of the same person on different networks. In fact, due to the friend-finder weakness that we discovered on all tested networks, we are able to associate profiles by e-mail addresses. As a result, we produce a more precise correlation: On one hand, we can make sure that different profiles belong to the same individual

(Irani et al. have a positive score of only 40% for the pseudonym match and 10%-30% for the real name match). On the other hand, we can reveal the "'hidden profiles" of users that they may actually wish to hide. Indeed, this is a major advantage of our approach; we can link profiles that are registered using different pseudonyms or information, but based on the same e-mail address. Finally, we conducted our studies on a larger set of data by crawling 876,941 unique profiles (versus 21,764 profiles studied by Irani et al.) and extracting up to 15 information fields from each profile (versus 7).

Also, note that our work is also related to the area of *de-anonymization*, where an attacker tries to correlate information obtained in different contexts to learn more about the identity of a victim. Narayanan and Shmatikov showed that by combining data with background knowledge, an attacker is capable of identifying a user [34]. They applied their technique to the Internet movie database (IMDb) as background knowledge and the Netflix prize dataset as an anonymized dataset, and were indeed able to recognizes users. Furthermore, the two researchers applied a similar technique to social networks and showed that the network topology in these networks can be used to re-identify users [35]. Recently, Wondracek et al. [36] introduced a novel technique based on social network groups as well as some traditional browser history-stealing tactics to reveal the actual identity of users. They based their empirical measurements on the XING network, and their analysis suggested that about 42% of the users that use groups can be uniquely identified.

In this paper, we continue this line of work and show that an attacker can cross-correlate information between different social networking sites in an automated way. The collected information reveals the different online identities of a person, sometimes uncovering "secret" profiles.

7 Conclusion

In this paper, we presented a novel attack that automatically exploits a common weakness that is present in many popular social networking sites. We launched real-world experiments on eight distinct social networks that have user bases that consist of millions of users. We leverage the fact that an attacker can query the social network providers for registered e-mail addresses on a very large scale. Starting with a list of about 10.4 million e-mail addresses, we were able to automatically identify more than 1.2 million user profiles associated with these addresses.

We can automatically crawl the user profiles that we map to e-mail addresses, and collect personal information about each user. We then iterate through the extracted friend lists to generate an additional set of candidate email addresses that can then be used to discover new profiles. Our attack is significant because we are able to correlate information about users across many different social networks. That is, users that are registered on multiple social networking web sites with the same e-mail address are vulnerable. Our experiments demonstrate that we are able to automatically extract information about users that they may actually wish to hide certain online behavior. For example, we can identify users who are potentially using a different name on a dating web site, and are pretending to be younger than they really are. The correlation that we are able to do automatically has a significant privacy impact.

After we conducted our experiments and verified the feasibility of our attack, we contacted the most popular social network providers such as Facebook, MySpace, XING and Twitter, who all acknowledged the threat, and informed us that they are going to adopt some of our countermeasures. By now, Facebook and XING have already fixed the problem by limiting the number of requests that a single source can perform, and we expect that other social networks will also implement countermeasures.

Acknowledgments This work has been supported by Secure Business Austria, by the European Commission through project FP7-ICT-216026-WOMBAT, by the POLE de Competitivite SCS (France) through the MECANOS project and by the French National Research Agency through the VAMPIRE project.

References

1. Dwyer, C., Hiltz, S.: Trust and Privacy Concern Within Social Networking Sites: A Comparison of Facebook and MySpace. In: Proceedings of the Thirteenth Americas Conference on Information Systems (AMCIS). (2007)
2. Fogel, J., Nehmad, E.: Internet social network communities: Risk taking, trust, and privacy concerns. Comput. Hum. Behav. **25**(1) (2009) 153–160
3. Gross, R., Acquisti, A., Heinz, III, H.J.: Information revelation and privacy in online social networks. In: ACM Workshop on Privacy in the Electronic Society (WPES). (2005)
4. Jagatic, T.N., Johnson, N.A., Jakobsson, M., Menczer, F.: Social phishing. Commun. ACM **50**(10) (2007) 94–100
5. Jakobsson, M., Finn, P., Johnson, N.: Why and How to Perform Fraud Experiments. Security & Privacy, IEEE **6**(2) (March-April 2008) 66–68
6. Jakobsson, M., Ratkiewicz, J.: Designing ethical phishing experiments: a study of (ROT13) rOnl query features. In: 15th International Conference on World Wide Web (WWW). (2006)
7. Brown, G., Howe, T., Ihbe, M., Prakash, A., Borders, K.: Social networks and context-aware spam. In: ACM Conference on Computer Supported Cooperative Work (CSCW). (2008)
8. News, H.: Spam-Bots werten soziale Netze aus (September 2009) `http://www.heise.de/security/Spam-Bots-werten-soziale-Netze-aus--/news/meldung/145344`.
9. Klensin, J.: Simple Mail Transfer Protocol. RFC 5321 (Draft Standard) (October 2008)
10. Zimmerman, D.: The Finger User Information Protocol. RFC 1288 (Draft Standard) (December 1991)
11. Bugtraq: OpenSSH-portable Enabled PAM Delay Information Disclosure Vulnerability (April 2003) `http://www.securityfocus.com/bid/7467`.
12. Bortz, A., Boneh, D.: Exposing private information by timing web applications. In: 16th International Conference on World Wide Web. (2007)
13. Python Software Foundation: Python 2.6 urllib module. `http://docs.python.org/library/urllib.html`
14. Irani, D., Webb, S., Li, K., Pu, C.: Large online social footprints–an emerging threat. Computational Science and Engineering, IEEE International Conference on **3** (2009) 271–276
15. Facebook: Statistics. `http://www.facebook.com/press/info.php?statistics` (April 2010)
16. Florencio, D., Herley, C.: A large-scale study of web password habits. In: 16th International Conference on World Wide Web (WWW), New York, NY, USA (2007)

17. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: Using Hard AI Problems for Security. In: 22nd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). (2003)
18. Danchev, D.: Inside India's CAPTCHA solving economy (August 2008) http://blogs.zdnet.com/security/?p=1835.
19. Chellapilla, K., Simard, P.Y.: Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). In: Neural Information Processing Systems (NIPS). (2004)
20. Mori, G., Malik, J.: Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In: IEEE Conference on Computer Vision & Pattern Recognition (CVPR). (2003)
21. Yan, J., El Ahmad, A.S.: A low-cost attack on a Microsoft CAPTCHA. In: 15th ACM conference on Computer and Communications Security (CCS). (2008)
22. Bilge, L., Strufe, T., Balzarotti, D., Kirda, E.: All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. In: 18th International Conference on World Wide Web (WWW). (2009)
23. Bonneau, J., Preibusch, S.: The Privacy Jungle: On the Market for Privacy in Social Networks. In: Workshop on the Economics of Information Security (WEIS). (2009)
24. Chew, M., Balfanz, D., Laurie, B.: (Under)mining Privacy in Social Networks. In: Proceedings of Web 2.0 Security and Privacy Workshop (W2SP). (2008)
25. Jones, S., Millermaier, S., Goya-Martinez, M., Schuler, J.: Whose space is MySpace? A content analysis of MySpace profiles. First Monday **12**(9) (August 2008)
26. Krishnamurthy, B., Wills, C.E.: Characterizing Privacy in Online Social Networks. In: Workshop on Online Social Networks (WOSN). (2008)
27. Bonneau, J., Anderson, J., Danezis, G.: Prying Data out of a Social Network. In: First International Conference on Advances in Social Networks Analysis and Mining. (2009)
28. Chau, D.H., Pandit, S., Wang, S., Faloutsos, C.: Parallel Crawling for Online Social Networks. In: 16th International Conference on World Wide Web (WWW). (2007)
29. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and Analysis of Online Social Networks. In: ACM SIGCOMM Conference on Internet Measurement (IMC). (2007)
30. Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N., Zhao, B.Y.: User Interactions in Social Networks and their Implications. In: 4th ACM European Conference on Computer Systems (EuroSys), ACM (2009)
31. Griffith, V., Jakobsson, M.: Messin' with texas, deriving mother's maiden names using public records. In: Third Conference on Applied Cryptography and Network Security (ACNS). (June 2005)
32. Raymond Heatherly, M.K., Thuraisingham, B.: Preventing private information inference attacks on social networks. Technical Report UTDCS-03-09, University of Texas at Dallas (2009)
33. Zheleva, E., Getoor, L.: To Join or Not To Join: The Illusion of Privacy in Social Networks with Mixed Public and Private User Profiles. In: 18th International Conference on World Wide Web (WWW). (2009)
34. Narayanan, A., Shmatikov, V.: Robust De-anonymization of Large Sparse Datasets. In: IEEE Symposium on Security and Privacy. (2008)
35. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: IEEE Symposium on Security and Privacy. (2009)
36. Wondracek, G., Holz, T., Kirda, E., Kruegel, C.: A Practical Attack to De-Anonymize Social Network Users. In: IEEE Symposium on Security and Privacy. (2010)