

# Why spammers should thank Google?

Mohamed Ali Kaafar, Pere Manils  
INRIA, France  
{kaafar, pere.manils}@inria.fr

## ABSTRACT

Buzz, the new online social networking (OSN) service from Google has been introduced a few weeks ago. Even though it raised big concerns (and even complaints) about several privacy issues, Buzz has been already launched inside millions of Gmail accounts. In this paper, we show that one of the major concerns Buzz might have to deal with is that it is integrated into the Google email service. In fact, to use Buzz one has to sign up for a Google profile that will primarily be seen by other Google users. However this profile, as shown in this paper reveals for the vast majority of Buzz users their Gmail usernames, and so their Google email addresses. We exploit the notion of Followers/Following in Buzz to crawl Google for Gmail accounts, demonstrating how it is easy and practical to collect millions of valid Gmail accounts from a single machine, in a very short period of time and without being noticed. The collected email addresses have many desirable properties from a spammer's perspective. They are valid email addresses, that refer to active and individual Buzz users that participate in online social activities, increasing then the efficiency of spam campaigns targeting these users. We then show how spammers can even use the Google infrastructure to categorize the email accounts they collected based on specific area of interest of users. As a conclusion, this paper demonstrate that integrating Buzz to email accounts, and hence to Google profiles offers spammers with a valuable, yet not risky, way to build a giant Google emails-made spammers database.

## Categories and Subject Descriptors

K.6 [Management of Computing and Information Systems]: Security and Protection; K.4.1 [Computers and Society]: Public Policy Issues

## General Terms

Security, Experimentation, Measurement

## Keywords

Online Social networks, privacy, spam, security, web crawling

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SNS'10, April 13, 2010, Paris, France.

Copyright 2010 ACM 978-1-4503-0080-3 ...\$10.00.

## 1. INTRODUCTION

Over the last few days of February 2010, Google has been deploying Buzz [6], its new online social network (OSN), to all Gmail accounts. Although many privacy issues in its original design have been already raised, Buzz will be without doubt one of the major OSN's actors. Since its official start, Buzz has already evolved and many of these identified privacy issues have been fixed. Google has for instance quickly addressed concerns about the feature giving users a ready-made circle of followings based on their email and chat frequency in Gmail, that may leak private information about Gmail users' conversations. Google has also changed many features in Buzz to give users the choice to display information publicly or not.

In this paper, we advocate that one of the major design issues in the Buzz network is its explicit integration into Google Email accounts. Indeed, Google Buzz is being advertised to users when logging into their Gmail accounts, and has been designed to be a transparent process to users that have already a Gmail account. Firstly, a user with a Buzz account means a user with a Gmail account, and more importantly that seemingly means an active Gmail user. Secondly, to use Buzz a user needs to sign up and hence create a Google profile. Even though Google profiles existed before the Buzz experience started, and are publicly searchable, the tiny frontier that exist between Buzz, public Google profiles and Gmail accounts may create a serious security risk for Google services. In particular, in this paper we show how it is now easy, quick and effective to crawl millions of Google email accounts, exploiting Buzz and related Google profiles without being noticed and from a single machine.

Indeed, to avoid automated searches, and so crawling, Google has limited the possibilities to search for its users' profiles by bounding the maximum number of profiles returned per search queries (a maximum of 1000 profiles per query). Google also limits the number of search queries a single machine (IP address) can perform on Google servers per day (limited to few hundreds queries, after which a verified captcha is needed). Crawling a significant number of profiles requires then important and expensive resources and lasts for very long periods. In essence, if an adversary would like to collect Google profiles, she/he would need to perform brute force searching by combination of names, in different languages, so that the Google search engine would reveal different public profiles of Google users.

This paper proposes a methodology that exploits Buzz design, and in particular the bind between Buzz and Gmail services, to not only retrieve in a very short period of time millions of Google profiles, but also to collect a significant number of valid Gmail addresses (our experiments allowed us to collect 4 million profiles and 1 million *active* Google email addresses in only 30 hours).

This personal information that many Google users would like to hide, is intrinsically embedded into a majority of Google profiles' URLs. Spammers have now easy ways to build a world-wide spam database, with very cheap resources. We then demonstrate how such data can even be processed by the Google search engine itself to profile users' behind these Gmail addresses, and build categorized emails addresses for spammers.

This paper makes the following contributions:

**Crawling Google profiles (section 3).** Exploiting the Followers/Following lists, enabled by default on the profiles of Buzz' users, we propose a method to efficiently collect a significant amount of Google profiles, that are linked to Active Email accounts. This method avoids brute force searches and hence circumvent the measures Google establishes to avoid active crawling.

**Building a spammers' effective database (section 4).** We observe that Google profiles, by default, include the users' Gmail username. We then show that for a large proportion of collected users' profiles, we can link both public profiles to active email addresses. This is a serious privacy risk for those users who use their email accounts as private addresses, and might not want to reveal this personal information. Although that information may be willingly made available on other online social networks, it is important to note that in Buzz, users have no choice but to reveal it or use a 21-digit URL. In the light of our statistics, we conclude that Buzz in its current design, can create an efficient and ready-to-use database for spammers.

**Profiling spammers' targets (section 5).** Using the collected users' accounts, we provide as a proof of concept ways that spammers can use to refine the data so that they can generate automated and large scale spam campaigns, yet targeting special community of users. We also show how spammers can exploit the Followers/Followings' lists of users to mount effective social phishing attacks.

## 2. BACKGROUND

In the following, we introduce different notions related to Google services. In particular we focus on Google profiles and Buzz basic design, being largely exploited in our crawling methodology and email addresses collection.

### 2.1 Google Profile

A Google profile is typically a collection of specific data associated to a user, as in many other OSNs (e.g. Facebook or twitter, etc.). Profiles have been introduced by Google in April 2009. They have then given Google users the ability to create a thumbnail of personal information, such as their name, photos, location, links to other web sites, etc.

Google profiles URLs are constructed simply by appending the *prefix* `www.google.com/profiles/` to one of the two options Google users might opt for. The first option, that we refer to by login name-based URL, incorporates user's Gmail address to the prefix, ending with URLs of the form `prefix/username`. The second option, we denote identifier-based URL, is to truncate the username by a suffix of 21-digits. The way Google is generating this suffix is not yet revealed<sup>1</sup>. This identifier-based URL ends up with a profile address that is complicated and quite difficult for users to remember, which might push many of them to not change the default setting of login-name URL.

It is worth noticing that the two options of the URL described above, are peculiar to users that have Gmail accounts. Others may

choose any available username to be part of the profile URL. However, since in this paper we focus on Buzz users, these have de facto Gmail accounts.

As a conclusion, if not identifier-based URLs, profiles could end up revealing users' email addresses. We will exploit this feature in section 4.

### 2.2 Buzz

Buzz is built into Gmail accounts as a service allowing users to post feeds updates, links, images, etc. and sharing them with their Gmail contacts. It has been introduced by Google as a competitor of other well-known OSNs like twitter or Facebook. Buzz's design has similarities with the twitter concept of Followers and Followings. Put simply, one user can opt to see updates from contacts she/he chooses to follow (Followings) and other contacts may choose to get the user's posts (Followers). We will not discuss the details of Buzz integration into Gmail, but it is worth noting that every Buzz user has a Gmail account. More interestingly, since Buzz's Followers/Followings (denoted  $F - F$  in the rest of this paper) lists are build based on Gmail contacts and more precisely on those to whom the user has presumably sent emails, or from whom she received emails, that means Buzz users are active email users with valid email addresses.

We also stress *the default setting* of Google Profiles is to display the  $F - F$  lists. These followers (resp. followings) lists however only show followers (resp. followings) that have public profiles, and display the number of others they do not. Finally, we report a statement from the Google accounts manager, when discussing ways to edit the profile: "The more information you add, the easier it will be for people to find you.". This may explain the high percentage of public profiles of Buzz users we observed during our measurement, as reported in the following section.

## 3. THE PROFILES' CRAWLER

The first step in building a spammers' Google emails database, is to collect Google profiles. This is processed by the crawler we design. We will show in section 4, how email addresses can be extracted from these profiles, and in section 5 we will refine our list of emails based on contents from Google public profiles.

### 3.1 Crawling methodology

In order to collect as many profiles as possible, we have exploited the  $F - F$  lists that appear by default in users' profiles. These lists provide links to the Followers and Following's profiles, if public<sup>2</sup>. Otherwise, the number of users (either Followers or Followings) maintaining non-public profiles is displayed. To be able to access these lists, a third entity must "just" be logged in with a Google account.

The design of our crawler is quite simple (refer to figure 1). It consists of a global queue shared by many threads that read and write content on it. This queue contains the numeric identifiers that refer to the users' profiles. In a first step (1 in the figure), the queue is filled with seed profiles resulting from few random searches on the Google profiles' search web site. This task is performed by the main thread. While the queue is being filled, each of the child threads picks an identifier from the queue (2) and starts performing the following recursive work.

Firstly, each thread *resolves* the identifier-based URL provided by Google profiles search engine, to obtain the profile's URL that is human readable URL (3), and that is often used by Google users

<sup>1</sup>All the Google identifiers we collected start with 11 or 10, which discards the hypothesis of a simple hash function.

<sup>2</sup>Users may also opt to not create a Google profile. For the sake of simplicity, we also consider non existent profiles as non public.

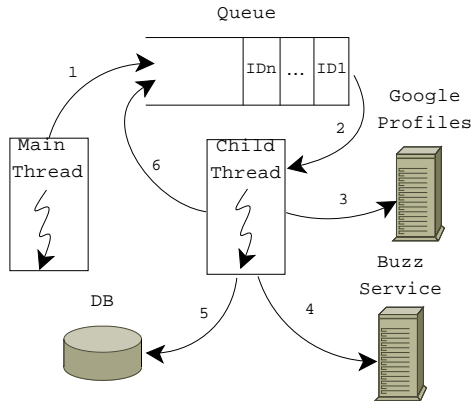


Figure 1: Crawler diagram.

to access their profile. Recall from section 2.1 that two options are offered to Google users: either displaying their profile URL as login name-based URL or as an identifier-based URL. Surprisingly, few days after Buzz has been launched, Google changed the behavior of its Google profile search engine to only display identifier-based URL as search results. However, these URLs, when users did not change the default setting, are redirected to login name-based URLs (this is performed with a regular Moved Permanently HTTP redirection). In this case, our crawl reveals both identifier-based URL and login name-URL, and more importantly links the identifier to the login name.

Having retrieved the users’ profiles, each thread performs two HTTP requests to the Buzz service to retrieve public profiles from the  $F - F$  lists (4), if displayed by the processed profile. Each time a new profile is found, it is inserted in the database (5) and appended in the queue (6).

Secondly, once the thread finishes parsing the whole list of publicly available profiles of  $F - F$ , and re-injected them in the profiles queue, it picks a new identifier and recursively reiterate the process.

The rationale of recursively parsing the Followers/Followers lists is to walk through as many profiles as possible in a short amount of time. Ideally, in order to ensure a sufficient number of profiles to be parsed in the queue, the main thread should periodically inject new seeds resulting from random profiles search. We discuss the impact of not feeding the profiles queue with fresh and random profiles in the following section. Nevertheless, the way we crawl profiles with a unique seed at the crawler’s start does provide a lower bound of the number of profiles that can be collected without the need to perform search queries (which we leave as future work).

During our crawl, we observed a surprising, yet risky, behavior of the Google profile search engine and of the Buzz service retrieving the  $F - F$  lists.

As described previously, the profiles’ search engine outputs only identifier-based URL in the search results. However, many of these URLs are redirected to login name-based URLs. In addition, when requested to provide the list of  $F - F$  of a single user, the Buzz service returns a structure that describes for all the Followers and Followings of this user, both their identifier and login names.

Our crawler allows us then to match the identifier to login names of users that did not yet choose to change the default setting of their profiles, so as to not display the login names. In other words, even

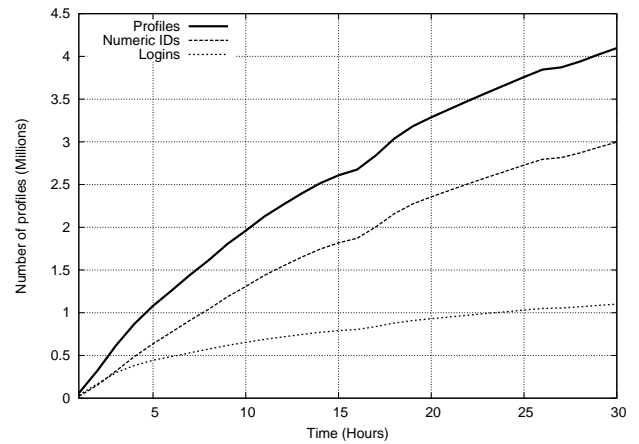


Figure 2: Cumulative number of gathered profiles in function of crawling time.

though it is unclear how Google is computing the identifier, allowing adversaries to collect both login names and their corresponding identifiers, might compromise the forward-security of these identifiers forever, i.e. even if users decide in the future to switch to the identifier-based URLs.

We stress our crawler does not need to brute force entries for the profiles search queries, and so does not need to perform several queries instantiating different languages so as to retrieve worldwide profile names. The main input being the list of  $F - F$ , our crawler walks in an automated way through as many profiles as social links would exist among Buzz users.

Finally, we note that our crawler run on a Dell PC with two quad-core CPUs Intel Xeon at 1.60GHz with 3GB of RAM, and a high-speed Internet connection. We point out that we assume a conservative stance, as we test our crawler using a single machine with a single IP address.

### 3.2 Crawling Results

We crawled Google profiles using the methodology described above for approximately 30 hours. Figure 2 shows the number of public profiles we collected in function of time (for the time being, ignore the curves entitled “Numeric IDs” and “Logins”). During the first 24 hours, we observe in figure 2 that we collected more than 3.6 million public profiles (solid line). This confirms the spider’s effect when crawling social links recursively reported in many previous researches on other OSNs (e.g. [3]), and that even if the slope of the increase flattens slightly during the last 15 hours of our crawl. This can be explained by two facts. First, the very conservative strategy we choose (i.e. profiles queue initially feed by only 5 random searches, and not incremented again). Second, Buzz is still in its infancy and seemingly not yet rolled out to *all* Gmail users.

Since the purpose of this crawl is to provide a proof of concept collecting valid Gmail addresses, we decided we gathered enough public profiles and stopped our crawler once 4 million profiles have been processed. The crawler quitted with approximately 5 million profiles that had not yet been harvested.

When querying  $F - F$  lists, Google answers with the URLs of the profiles but also with the number of  $F - F$  that do not have public profiles. Table 1 summarizes the results we obtained during our crawl.

First, we observe that among the 4 million profiles we retrieve, 72% display the list of their  $F - F$  lists publicly. The 28% remaining profiles’ users, for whom we do not observe  $F - F$  lists

**Table 1: Statistics of the collected profiles. The *traitors* are the users that have exposed their friends’ profiles from the  $F - F$  lists.**

Number of profiles	Profiles with Public F-F lists	Unique traitors	Avg. ratio Pub/Non_Pub
1M	636K (64%)	101K	1.8
2M	1.38M (69%)	332K	1.4
3M	2.14M (71%)	700K	1.2
4M	2.8M (72%)	1.22M	1.2

are then either non Buzz users, or have chosen to uncheck the “publicly display  $F - F$  lists” option, proposed when creating or editing the profiles. Note users’ profiles can be crawled without being part of Buzz, since other users may choose to follow them and so their profiles are retrieved from other users’  $F - F$  lists.

Second, we compute for each user the ratio between the number of public profiles and the number of non public profiles. On average the number of public profiles represent 1.2 times the number of non revealed profiles. This shows that *per user*, the number of other Google users that choose to reveal their profiles is slightly higher than the number of users that hide their profiles. However, this provides only a rough approximation of the number of non public profiles, as our crawler only collects unique public profiles, and is unable to reveal uniqueness of non public profiles. It then over estimates the number of non public profiles.

#### 4. BUILDING THE SPAMMERS’ DATABASE

In the previous section, we have shown how a simple crawl of Buzz social links allows us to collect a significant number of Google profiles in a short period of time. This without performing a large amount of search queries, and without the need to address brute force or dictionary-based search of profiles. In the following, we focus on extracting from the collected public profiles the Gmail usernames that spammers can exploit.

Recall from section 2.1 that the default URLs of the profiles of Gmail users are login name-based. Apart from a few randomly generated profiles (seeds of the crawler), almost all the profiles we collected are profiles of Gmail users, since they are Buzz users. However, users may choose to switch to identifier-based URLs hiding then their Gmail address. Figure 2 depicts the evolution of the cumulative number of collected profiles for both identifier-based (Numeric IDs) and login name-based URLs (Logins), during the period of crawl. It is interesting to note that a high percentage of Google users changed their profile URLs for numeric identifiers. In essence, the login name-based URLs represent 25% of the totally collected URLs. After a 30 hours-crawl, we retrieved more than 1.099.868 profiles with login name-based URLs. In the following, we validate the retrieved Gmail logins, and hence show how effective the spammer’s emails database would be.

We need then to validate that each login name-based URL with a non numeric ID corresponds to an actual Gmail address. Although this step might be useless for spammers as they can send spams indiscriminately to whatever the collected logins are numeric or not, it is clear that efficient spam campaigns would benefit from valid email addresses, as they would be less costly and more importantly, less detectable. Indeed, spammers’ databases guaranteeing valid email addresses are logically sold at higher prices.

To check if a given non-numeric ID is a Gmail address, we connect to one of the Gmail’s MX servers and, following the SMTP

protocol, we send the appropriate commands to emulate the sending of an email (without actually sending it) to the Gmail address to be validated. Depending on the configuration of the MX server receiving such commands, the server’s response may indicate whether the recipient address has a mailbox on the server or not, and hence, proving the validity of the email address. Gmail MX servers favor this verification.

Among the 4M of profiles we collected, we distinguished more than 1M login-name based URLs, based on a simple verification of non existence of a 21 digits in the URL’s suffix. Checking the collected login names, we end up with a total of 1.011.878 valid email addresses, demonstrating that spammers would hit in more than 96% of the cases. We do not claim our method distinguishing between login names and numeric (Google-made) identifiers is the most suitable to perform an optimal spammers’ hit rate, since this method yields few false negatives<sup>3</sup>. Our method also results in false positives. We did not succeed in verifying 4% of the login names. This is explained by profiles that are followed by Buzz users, and have only a Google profile without an associated Gmail account. That is to say, our crawler retrieves also in the list of users’ followings, Google accounts being tracked by Buzz users without having been proposed by Gmail based on Gmail conversation and chat. The small proportion of non valid emails (less than 4% of the collected emails) offers from this perspective, a high efficiency of the collected emails database.

### 5. PROFILING SPAMMERS’ TARGETS

So far, the spammers’ objective was to generate a database of valid Gmail addresses crawling the Google OSN service as quickly as possible. In this section, we show that Google services may even be exploited by spammers to go one step further and classify Gmail addresses according to users’ interests: This in order to perform targeted-audience spam campaigns. Moreover, we show how dangerous could be an OSN revealing the relationships between individuals to a potential adversary when this adversary knows about their email addresses.

#### 5.1 Exploiting the Google Search Engine

One first method to categorize profiles consists in exploiting the Google profiles search engine in order to collect profiles that Google has indexed by a particular keyword. We choose 5 keywords (as an illustrative example for potential spam campaigns) and perform a Google search for each keyword looking for profiles related to the selected keyword. For each login name-based URL’s profile returned by the search, we then retrieve the  $F - F$  lists when available. By grouping all the discovered non-numeric IDs for a keyword, we obtain a list of Gmail addresses that most likely are related or even interested in the searched keyword. The intuition behind this, is that if an individual is interested in a keyword, his followers and followings are likely interested in the same keyword.

Table 2 shows the results we obtain using the method explained above. We search for profiles related to 5 different categories, exploiting the keywords: books, games, car, health and technology. In the books’ category example, the search result contains 3141 profiles, but we note again that Google only returns 1000 of them. After retrieving the  $F - F$  lists from the accessed profiles, we end up with a total of 7663 profiles, 4145 of which have a login name-based URL. In essence, for this particular example, a spammer can as a worst case quadruple the efficiency of his spam campaign (compared to a naive profile search).

<sup>3</sup>“Logins” might be constructed as 21 digits identifier-like names.

**Table 2: Gmail addresses classification using the Google search engine.**

Category	Search results	Collected profiles	Collected logins
books	3141	7663	4145
games	3306	6005	3171
car	2741	6081	3931
health	2814	3264	2022
technology	3483	5891	3628

**Table 3: Gmail addresses classification using off-line parsing.**

Category	Gmail addresses
books	80018
games	87921
car	67364
health	65706
technology	138270

This method however is still limited by the number of Google profiles returned by the Google profiles' search (a maximum of 1000 results per search that are barely invariable with subsequent queries). One can also argue that  $F - F$  lists might not be a good indicator for users' interest. However, as long as the objective of the spammers is not to flood all emails recipients with spams, this still allows for targeted yet efficient spam campaigns.

As a conclusion, this method shows that mixing email addresses with personal information and offering facilities to correlate them is a bad practice, that spammers looking for very specific targets can benefit from.

## 5.2 Off-line Parsing of the Collected Profiles

The previous method queries the Google search engine to retrieve profiles related to one keyword. This method gets benefits from the accuracy of the Google search engine, but is however limited by the number of returned profiles (although we expand the results with profiles from the  $F - F$  lists). In section 3, we crawl the Google services in a way to bypass the limitations that Google may set to avoid automated searches. In the following, we exploit the database of profiles we already collected in the initial crawl, and categorize them off-line, so as to provide a significant collection of email addresses based on users' area of interests.

We download the content of each profile stored in the database. Then we (again) rely on the off-line Google indexing engine (we instrumented the Google desktop search application, i.e. Google Desktop) filled with keywords to classify them accordingly. We choose the same 5 keywords as in the previous section, so as to consider the same categories. Once categorized in one or more categories, each profile is added with the  $F - F$  lists of profiles that we again assume to be also interested in these areas. We stress that our categorization implies that the same profile may appear in several categories, increasing the scam vectors spammers would benefit from.

To illustrate our method, we consider as subset of 50 K profiles and present the results of Gmail addresses classification in table 3. It is worth noticing that using this technique, we do classify on average 20 times more profiles with email addresses than relying on results from Google profiles search. Regardless of the number

of profiles that Google would return, this method builds then on the  $F - F$  lists retrieval that the Buzz service provides. As an example, we consider the books category, where the off-line categorization gathered more than 80 K email addresses, while the Google search engine proposed only 1000 profiles (and 7663 profiles if the search results is swollen with  $F - F$  lists). Note finally that this method, although very effective (from the spammer's perspective), is more time and resources consuming, since the spammer needs to download each user's profile. However, this can be processed while the crawler performs the resolution step explained in section 3, even if it might impact the crawler's speed.

## 5.3 Exploiting Social Relationships

Social phishing has been reported in many previous researches (e.g. [2, 7]) as an effective way to mislead OSN users and to take advantage of social relationships that might exist between them. In this section, we stress that these exploitation may even be more damaging when adversaries know the personal email addresses of their targets. At a large scale, knowing the email addresses of targets' friends can be particularly harmful, as spammers can forge not only attractive, but also less suspicious emails, by spoofing a friend's email address.

Worms propagation already benefits from social relationships. Melissa or ILOVEYOU [4, 5] are typical examples of worms that once infecting a particular machine, try to exploit the user's email address lists stored in the system (contacts from Microsoft Outlook or from Windows Address Book) in a tentative to spread across other machines. More recently, other worms like Koobface [1] spread using the OSN capabilities by sending messages to the friends of the user whose machine has been infected. In all these cases, a user's machine must be already infected by the worm in order to take advantage of his contacts list.

We believe Google is now offering new capabilities to spammers and worms propagation. The crux of the problem is that besides social relationships being made publicly available for third parties, adversaries may now know about the personal email addresses of the individuals associated to the extracted social links. Users machines being infected is no more a constraint, as it has been released by the knowledge of the email addresses. Indeed, retrieving the  $F - F$  lists of a particular profile, say  $X$ , gives (as shown in the previous sections) a potential list of  $X$ 's friends email addresses. Spammers could then use this knowledge to automate attractive spam-sending processes, with very specific content to convince their victims. As an illustration, we provide the following example:

```
From: Y@domain.com
To: X@gmail.com
Subject: great photos!
Body: Hi X-name, checkout these photos of
X-friend-1 and X-friend-2 trip in Hawaii.
(embedded malicious link or attachment)
```

The sender address might even be spoofed and looks as coming presumably from another friend of  $X$ .  $X$ -friend-1 and  $X$ -friend-2's email addresses can be "CCed" so as to comfort the recipient with the idea that the email is sent from a legitimate friend. The hope, from the adversary's perspective, is that the contacted users trust the spoofed email and clicks on the malicious link. As future work, we will extend our analysis to study whether such technique might even bypass the Gmail spam filters. This section however already shows how that social phishing and efficient spams are now made easier and potentially more effective on Google domains.

## 6. CONCLUSIONS AND FUTURE WORK

This paper presents a first experience crawling the recently deployed Buzz service, and shows how an adversary can exploit the social links revealed by Buzz to efficiently retrieve a significant amount of Google email addresses. In practice, we validated the collected email addresses and showed how a potential spammer can even build a classification of these Gmail addresses, with a very high hit rate. We also introduce security concerns Google might deal with in the future as it is now revealing sensitive information that can be correlated: social links along with corresponding email addresses. Spammers and worms developers can easily benefit from such information leakage. Keeping in mind how damaging cross-sites attacks could be, we intend to propose a deeper analysis of Buzz, if adversaries would extend information about users exploiting other OSNs public information. As a first step, we will then enhance our crawler's processing with the objective of retrieving as many profiles as possible for a continuous crawl in time.

## 7. REFERENCES

- [1] D. Baltazar, J. Costoya, and R. Flores. The real face of koobface: The largest web 2.0 botnet explained. Technical report, Trend Micro Threat Research, July 2009.
- [2] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *18th International World Wide Web Conference*, April 2009.
- [3] J. Bonneau, J. Anderson, F. Stajano, and R. Anderson. Eight friends are enough: Social graph approximation via public listings. 2009.
- [4] CERT. Love letter worm. CERT Advisory CA-2000-04.
- [5] CERT. Melissa macro virus. CERT Advisory CA-1999-04.
- [6] Google. Buzz. <http://www.google.com/buzz>.
- [7] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, 2007.