Privacy Concerns of FOAF-Based Linked Data

Peyman Nasirifard, Michael Hausenblas and Stefan Decker

Digital Enterprise Research Institute
National University of Ireland, Galway
IDA Business Park, Lower Dangan, Galway, Ireland
firstname.lastname@deri.org

Abstract. In this position paper, we introduce a potential problem that arises with the emergence of publicly-available, FOAF-based linked data. The problem allows a spammer to send context-aware spam, which has a high click-through rate. Unlike online profiles within social networks, FOAF-based structured data provides a more reliable and accessible "food" for spammers and attackers. Current solutions (e.g. Digital Signatures) and proposed methods to restrict unauthorized accesses to FOAF files can prevent a subset of such activities; however we show that they are not widely used. Moreover, some of these solutions may be contrary to the mission of the Semantic Web and open data initiative.

1 Introduction

"Congratulations! You have won the National Lottery!" is a common subject line in unwanted emails (i.e. spam), which we receive in our inboxes or junk folders. Although we delete them or mark them as spam, further spam emails sometimes arrive. Key industry players (e.g. Microsoft) invest a huge amount of money in fighting spam and spammers[1], but it seems that the latter is the winner.

Various techniques have been initiated and developed for spam fighting. Labels that are identifiable by humans (i.e. CAPTCHA[2]) are currently used by major email providers to restrict the sending of automated spam. Services like tinymail[3], which aims to hide email addresses, or Email Icon Generator[4], which creates an image out of an email address, are samples of such efforts for fighting spam. Some spammers hire people to circumvent these techniques[5]. The unwritten rule of the game between spammer and "spammee[6]" is the less information we provide on the Web regarding our contact information and personalities, the lower the probability that we will receive spam.

[1] http://www.cbsnews.com/stories/2004/01/24/tech/main595595.shtml

[2] http://www.captcha.net/

[3] http://tinymail.me/

[4] http://services.nexodyne.com/email/

[5] http://www.ibm.com/developerworks/web/library/wa-realweb10/

[6] We define "spammee" as a person, who receives spam

Context-aware spam, unlike common spam, has a high click-through rate, as it contains more relevant information. This issue can be easily (ab)used by a spammer or an attacker for sharing phishing links and/or other malware. Brown et al. [1] studied the vulnerabilities of major social networks (e.g. Facebook) against context-aware spam. They estimated that around 85% of Facebook users could be accurately targeted with context-aware spam.

FOAF[7] profiles are used by many people, including Semantic Web researchers and professionals, as a means to structure contact information and social networks. FOAF profiles are considered to be "machine-interpretable", as they are based on a formal model. FOAF-based linked data helps towards enabling the mission of the Semantic Web and/or Global Giant Graph[8].

In this position paper, we describe a potential (privacy) problem of publicly-available, FOAF-based linked data. We argue that a spammer can potentially benefit from FOAF profiles by sending context-aware spam and we support this argument by presenting an example of such an attack. We believe that FOAF profiles provide a ready input for spammers, who may utilize them to personalize the spam message and increase the click-through rate of the emails. We also discuss the possible *partial* solutions that currently exist, but are not widely used.

2 FOAF: Structured Data for Spammers

FOAF profiles are structured, decentralized and extensible. They are probably the most explicit and true representation of our social networks, as people we know are clearly listed, and our contact information and interests are disclosed. FOAF profiles are an honest representation of a person's attributes, as the user takes into account the fact that they will probably only be used by machines (psychological effect). Unlike automatically-generated FOAF files[9], manually-generated FOAF files are usually hosted on personal homepages. Manually updating FOAF profiles is a time-consuming and error-prone task. Therefore, some tools have been developed to help users to create / update their profiles (e.g. FOAF-a-Matic[10]).

This structured and honest knowledge about a person's life is what spammers and/or attackers are looking for. In the following section, we demonstrate how a context-aware spam message can be constructed and sent, using a FOAF profile plus publicly available search engines (e.g. Google). Although we performed the process manually, it can be automated using available APIs and packages and/or simple text parsing.

As FOAF profiles are linked data, we need to choose some seeds which are well-connected. In our example, we used the FOAF file of "Axel Polleres[11]", as it is

[7] http://www.foaf-project.org/

[8] http://dig.csail.mit.edu/breadcrumbs/node/215

[9] As an example, FOAF profiles on http://www.deri.ie/about/team/ were generated automatically using a script

[10] http://www.ldodds.com/foaf/foaf-a-matic

[11] We had his permissions to use his FOAF profile

complete and error-free. Our goal is to send a context-aware spam message to Axel using his FOAF profile.

Figure 1 demonstrates the overall view of the process. First step is to use Google to find his FOAF file to use as a seed. Using Google API and/or parsing the HTML results can automate this process. Moreover, using some available techniques [2], such as utilizing the filetype option, can help us to reach the desired FOAF profile at first or higher ranks (i.e. *filetype:rdf*). Our experiment with the query "Axel Polleres FOAF" returned the requested FOAF file as the first retrieved link. We followed the link and accessed Axel's FOAF profile. As it is well-structured linked data, it can be parsed using common RDF processing libraries, like Sesame or Jena. Parsing Axel's FOAF profile gave us valuable information about his friends and contact information.

The next step is to find the seed's email address. FOAF profiles include SHA1 hash code of email addresses. As SHA1 collisions are far from current power of computers[12], they are good sources for "verification" purposes. Therefore, for finding email addresses, we follow a similar approach to that for finding FOAF profiles. Our experiment with Google using query "Axel Polleres Email" returned the requested email as the first retrieved link. However, the result is a HTML page and needs further processing, but having the SHA1 hash code of the email, is used as a help. Many people, including our seed, try to mask their email addresses using various text-based techniques such as replacing "@" with "[at]" etc. After gaining access to a HTML page that contains the seed's email address, we first search for the keyword "email" or "e-mail" on that page. We then apply the following common patterns to retrieve the email:

- Remove spaces
- Replace 'at' or '[at]' with '@', and 'dot' or '[dot]' with a period
- Replace 'firstname' with the user's actual first name, and 'lastname' with the user's actual last name (gleaned from FOAF profile)
- Remove 'removeme', '[removeme]'
- other possible rules

As generating a SHA1 hash code is fast, we may continuously check if we have a valid email address. If we did not succeed at the first retrieved link, we may refer to the second or third results of search engine. Note that some people generate SHA1 hash code of their emails without the "mailto:" prefix.

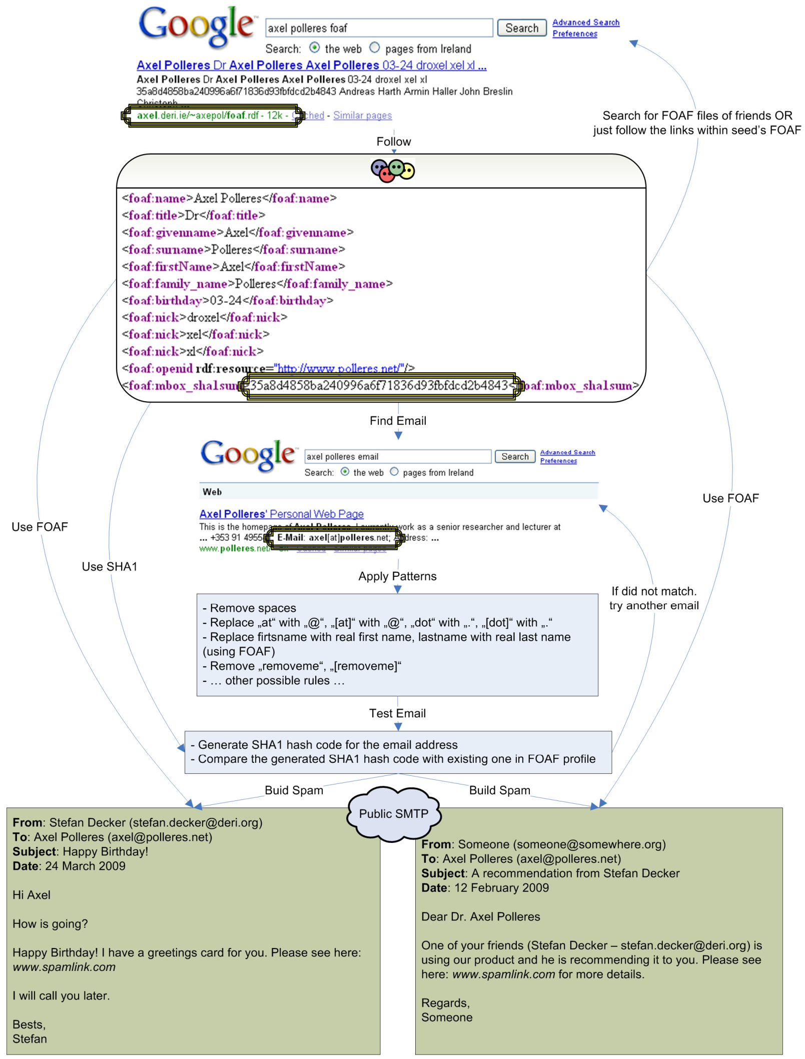
Using these techniques, we can successfully identify Axel's email address. In order to send a context-aware spam message, we need to identify Axel's friends and their contact information as well. Finding Axel's friends from his FOAF profile is straightforward, as they are clearly listed in the profile. In order to find their emails, we repeat the previous method.

One may claim that a possible problem that arises from this approach is resolving name ambiguity. As emails are unique and we have access to the SHA1 of the email, we always ensure the hash code of a person's email matches the SHA1. This will automatically resolve name ambiguity issues. Obviously, this may require parsing more results from the search results.

Returning to the scenario, we now have the email address of the seed plus the email addresses of his friends. In order to send a context-aware spam, we may use

[12] http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html

some pre-defined templates. Based on the granularity of information that people provide in their FOAF profiles, we select the appropriate template.



axel polleres foaf

Axel Polleres Dr Axel Polleres Axel Polleres 03-24 droxel xel xl ...
Axel Polleres Dr Axel Polleres Axel Polleres 03-24 droxel xel xl
35a8d4858ba240996a6f71836d93fbfdcd2b4843 Andreas Harth Armin Haller John Breslin
Christoph ...
axel.deri.ie/~axepol/foaf.rdf - 12k - Cached - Similar pages

```
<foaf:name>Axel Polleres</foaf:name>
<foaf:title>Dr</foaf:title>
<foaf:givenname>Axel</foaf:givenname>
<foaf:surname>Polleres</foaf:surname>
<foaf:firstName>Axel</foaf:firstName>
<foaf:family_name>Polleres</foaf:family_name>
<foaf:birthday>03-24</foaf:birthday>
<foaf:nick>droxel</foaf:nick>
<foaf:nick>xel</foaf:nick>
<foaf:nick>xl</foaf:nick>
<foaf:openid rdf:resource="http://www.polleres.net/"/>
<foaf:mbox_sha1sum>35a8d4858ba240996a6f71836d93fbfdcd2b4843</foaf:mbox_sha1sum>
```

axel polleres email

Web

Axel Polleres' Personal Web Page
This is the homepage of Axel Polleres. I currently work as a senior researcher and lecturer at
... +353 91 4955... E-Mail: axel[at]polleres.net; Address: ...
www.polleres.net/

- Remove spaces
- Replace „at“ with „@“, „[at]“ with „@“, „dot“ with „.“, „[dot]“ with „.“
- Replace firtsname with real first name, lastname with real last name (using FOAF)
- Remove „removeme“, „[removeme]“
- … other possible rules …

- Generate SHA1 hash code for the email address
- Compare the generated SHA1 hash code with existing one in FOAF profile

From: Stefan Decker (stefan.decker@deri.org)
To: Axel Polleres (axel@polleres.net)
Subject: Happy Birthday!
Date: 24 March 2009

Hi Axel

How is going?

Happy Birthday! I have a greetings card for you. Please see here: *www.spamlink.com*

I will call you later.

Bests,
Stefan

From: Someone (someone@somewhere.org)
To: Axel Polleres (axel@polleres.net)
Subject: A recommendation from Stefan Decker
Date: 12 February 2009

Dear Dr. Axel Polleres

One of your friends (Stefan Decker – stefan.decker@deri.org) is using our product and he is recommending it to you. Please see here: *www.spamlink.com* for more details.

Regards,
Someone

Google
Search
Advanced Search
Preferences
Search: ● the web ○ pages from Ireland
Search for FOAF files of friends OR just follow the links within seed's FOAF
Follow
Find Email
Use FOAF
Use SHA1
Use FOAF
Apply Patterns
If did not match. try another email
Test Email
Buid Spam
Build Spam
Public SMTP

Fig. 1. Overall view of sending spam using FOAF

For example, Axel put his birth date in his FOAF profile. A context-aware spam may be sent on his birth date from one (or more) of his friends, which encourages him to visit an online greetings card, that contains some harmful/harmless links. Another template may use Axel's friends for advertisement purposes. Figure 1 demonstrates two sample spam messages that have been created using some templates: one for a birthday and the other for sending a product recommendation. Using techniques and tools like URL shortening[13] can potentially be used to further hide the content and target of the links.

For our experiment, we did not use a template. We created a simple email with an embedded link and sent it via a customizable SMTP server to our victim (seed). He clicked that link, as it seemed to be from one of his friends.

This approach is recursive. The seed's friends may be used as new seeds and the process continued. Taking into account the small world phenomenon (i.e. Human Web), we can expect to reach all people in the FOAF-o-sphere. If not, common (Semantic) search engines can be used for identifying new seeds. We can also consider mutual links between people in FOAF profiles may represent a stronger / more reliable relationship, which can in turn be used for increasing spam's click-through rate.

3 Discussions and Potential Solutions

Privacy of FOAF profiles has been discussed by some researchers. Reagle [3] proposed to encrypt parts of the FOAF file to restrict unauthorized accesses. Frivolt and Bieliková [4] proposed to partition FOAF files, where each partition has a specific visibility. These solutions are not widely used and FOAF files are accessible to all and are indexed by major (Semantic) search engines. Moreover, it seems that hiding FOAF profiles can be contradictory to the open data initiative.

In comparison to common social networking sites, using FOAF for sending context-aware spam can be considered as a more reliable and accessible approach for spammers, due to several reasons:

- Finding users' email from online social networks could be very difficult, as most social networking sites hide the email addresses of the users. SHA1 hash code of the emails within FOAF profiles can be used as a means for verifying the valid email address.
- Crawling heterogeneous and highly customizable social networks (e.g. MySpace) offers a huge overhead for spammers, whereas FOAF is unique structured data.
- Someone may generate fake user profiles with incomplete names within online social networks, whereas FOAF is considered to be "reliable", as they are hosted on personal homepages and/or automatically generated from reliable data. Although, we are not aware of any study on the degree of "honesty" of FOAF profiles.

Generally, generated context-aware spam using FOAF can be categorized into two main groups: the first is where the sender of the email is supposedly a friend of the seed, then second is where the sender of the email is unknown to the seed, however the email may refer to friend(s) of the seed. Figure 1 illustrates examples of both categories.

[13] http://tinyurl.com/

Digital Signatures can potentially obstruct spam of the first type, but they are not widely used. The result of a survey that we did within our institute showed that just one person out of one hundred and twenty researchers and staff is using digital signatures permanently. Three researchers claimed that they use it, whenever they want to look serious or they want to contact somebody officially. One participant claimed that he used to have digital signature, but as his friends do not use it, he found it useless and he stopped using it. One participant claimed that he always wanted to have one, but due to time limitations, he did not investigate how he can install it on his email client. One participant claimed that managing private key / public key is very time-consuming. He said, however, there exist some tools that can help towards key management, but they require user verification (i.e. human-in-the-loop) at the end which brings lots of overhead for users. One participant with sufficient technical background tried to install a certificate on his email client, but after half an hour struggling, he stopped and complained about its complexity and argued that the process can be very time-consuming for a common user with no or little technical background. The result of the survey showed that 114 out of 120 participants never used digital signature within their email clients. The fact that lots of people are not really using digital signatures undermines the usefulness of digital signatures for those, who are using them permanently.

The other possible solution for the first group of spam is looking at detailed headers of the emails. This will show the "traversed" path of an email through its journey to reach the target person. This is probably a technical point which many users are not aware of and even for technical users, it is a time-consuming issue. Moreover, there exist some RFCs (e.g. RFC 4408 – Sender Policy Framework (SPF)) that help towards this direction, but they are not widely used.

For the second group of spam, spammers may always use major and free email providers for sending spam. This is still an open problem. Thanks to publicly available FOAF profiles, spammers can increase click-through rate of the spam by making them context-aware.

Generally, some partial solutions can be also considered to increase privacy of FOAF profiles:

- Remove SHA1 hash code from FOAF
- Use various hashing functions within FOAF, and not only SHA1
- Mask person's name and/or friends' name within FOAF

4 Conclusion

We all receive spam. Although FOAF-based linked data can bring lots of advantages for the community, it is necessary to be aware of malicious usage of such data. In this paper, we presented a potential privacy leak of publicly available FOAF profiles, demonstrated a context-aware spam that was sent just by using FOAF files, search engines and a customizable SMTP server and our victim clicked on the embedded link. We also argued that current *partial* solutions (e.g. digital signatures) are not widely used and if we put our FOAF profiles online, we may expect context-aware spam.

Acknowledgement. We thank anonymous reviewers for their feedbacks. We also thank all people, who participated in our survey. The work presented in this paper has been funded in part by SFI under Grant No. SFI/08/CE/I1380 (Lion-2) and the EU under Grant No. FP6-IST-5-35208 (Ecospace).

References

1. Brown, G., Howe, T., Ihbe, M., Prakash, A., Borders, K.: Social networks and context-aware spam, in Proceedings of the ACM conference on Computer supported cooperative work. ACM, San Diego, CA, USA (2008)
2. Calishain, T., Dornfest, R.: Google Hacks: 100 Industrial-Strength Tips and Tools. O'Reilly and Associates, Inc. 352 (2003)
3. Reagle, J.: FOAF Spheres of Privacy. Available online at http://reagle.org/joseph/2003/09/foaf-spheres.html
4. Frivolt, G., Bieliková, M.: Ensuring privacy in FOAF profiles. in Znalosti (2007)