

# NaiveDox: A Tool for and Analysis of De-Anonymizing Internet Users by Cross-Referencing Sites

Kade Keith<sup>1</sup>, Zach Brown<sup>2</sup>, Matt Saari<sup>3</sup>,

<sup>1,2,3</sup> Department of Computer Science Texas A&M, College Station, TX 77840 USA

Oftentimes, internet users inadvertently over expose themselves on the web by using the same username across multiple sites. We develop a novel tool for automated de-anonymization that exploits this, so that internet users can be made aware of how anonymous they actually are. We then use this tool to show that a significant number of users of the site Github can have their profile de-anonymized. Additionally, we show that the content of social media posts can be used to answer personal security questions.

**Index Terms**—Doxing, Security Questions, Privacy, Anonymity

## I. INTRODUCTION

A COMMON perception among users on the Internet is that when they are not using their real name and instead use a handle or username that they made up, they are anonymous. This sentiment is conveyed by the adage “On the Internet, Nobody Knows You’re a Dog” [1]. We show that this mindset is dangerous, as many internet users inadvertently reveal much more about themselves than they realize, leaving themselves open to social engineering attacks. There are a number of ways that internet users can be de-anonymized. We focus in particular on the weakness that people use the same handle across different sites. By querying multiple sites, a unified profile can be built for a certain handle that can be used to identify and potentially impersonate or release incriminating information about the owner of that handle. When information that was thought to be private is revealed about a person, the process is often called doxing, and has had disastrous ramifications for many web denizens.

We develop a novel tool for automated de-anonymization, so that internet users can be made aware of how anonymous they are online. Previous software has conducted de-anonymization, but did not allow users to check their own anonymity. This tool is a web application where users can enter either a handle they use or their email and see how much of a profile can be constructed based on that handle/email. This tool makes users aware of how much information they are revealing online, and shows them how they can protect their anonymity. One particular area we focus on is information that could be used to answer security questions, which is a common method that websites use to authenticate users. If an attacker has enough information to answer a user’s security question, they could use this to take control of that user’s account. We also use this tool to conduct analysis of how anonymous the users of the website GitHub are.

The primary contribution of our project is a tool that people can use for real-time de-anonymization. To our knowledge no such tool exists. Our approach also utilizes more websites than any previous work. Additionally, our project is the first to automate the social engineering of answers to security questions based on online posting history (from sites like Reddit and Twitter). Once users begin to utilize our tool,

we will also be able to study how people react to learning how anonymous they are on the web. If someone looks themselves up and realizes they can be de-anonymized, we can later check their username again to see if they have successfully protected themselves.

May 6, 2015

## II. RELATED WORK

One question that arises is just how harmful de-anonymizing someone is. A large-scale analysis of Twitter users by Peddinti, Ross, and Cappos found that users behave significantly differently when they are not using their real name [2]. In their study, anonymous users were more likely to offer opinions on controversial topics than users that used their real names. Given this, it can be inferred that de-anonymizing an individual can have negative impacts for that person, because they are more likely to post inflammatory content if they believe they are anonymous.

Prior research has used a number of techniques to autonomously de-anonymize social networks. Narayanan and Shmatikov as well as Chen, Hu, and Xie found that a portion of social media accounts have their profiles matched across multiple social networks using just the structure of the graph [3], [4]. This approach, while novel and effective, has performance shortcomings. The subgraph-isomorphism problem is NP hard, and thus this approach is not viable in real time. Bilge et al. found that cloning a victim’s profile was an effective technique for performing identity theft attacks [5]. Balduzzi et al. found that the search by email feature of many social networks was a potential security risk [6]. We adopt this technique in our own application. Additionally, Rabkin showed that many of the security questions used on popular sites are weak [7].

## III. IMPLEMENTATION

We have created a web service and front end client that allows users to check how anonymous they are online.

The front end takes as input either a handle or email. It then validates that the user is human by generating a javascript checkbox. That information is sent to the service in a GET request. It then takes and queries various sites with user profile capabilities looking for that user. Sites queried include: Github,

Reddit, Twitter, Steam, Youtube, Spotify, Instagram, Flickr, Google Plus, LinkedIn, and Aggie Network

Our service takes an iterative approach to de-anonymizing users. Given a handle, it will first find as much information about that handle as possible. If it finds an email address, it will take that email and query sites that support search by email such as Facebook. If it finds a real name it will query sites that support search by real name such as Aggie Network.

One challenge of this approach is that sites attempt to block non-humans by giving them a captcha. We use pauses between requests to get around this. As a result, the service can take a minute or two to perform the querying and extraction. Another solution we developed to get around captcha is discussed in the results section.

Data extraction is done using a combination of apis, xpath, regex, and the BigSemantics web service. Sites like Facebook provide an api. This api provides basic information, but is designed so that scraping large quantities of data is difficult. The version of facebook that humans use also works to make it difficult for bots to mine information. The html is obfuscated. However with xpath it is still possible to get that information, and since information on Facebook is often much more public than its users believe it is a great source of data.

Once as many attributes as possible about a user have been gathered, those are packed into a json of the following format and sent back to the client.

```

1 {
2   "email": {
3     "github" : {
4       "link": "https://github.com/
5           keithkade/",
6       "source": "github",
7       "data": "keithkade@gmail.com"
8     }
9   },
10  "name": {
11    "github" : {
12      "link": "https://github.com/
13          keithkade/",
14      "source": "github",
15      "data": "Kade Keith"
16    },
17    "google plus" : {
18      "link": "https://plus.google.com/
19          115562781747668285547/",
20      "source": "google plus",
21      "data": "Kade Keith",
22      "from_field": "email"
23    }
24 }
```

The data is organized by field, such as name, email, address, etc. Each field contains a dictionary of entries for that field. The source website is the key. Each of those entries contains the data, the source site, and a link to the user profile. If one of those fields was used to query other sites, then the results of those queries will have a from\_field attribute.

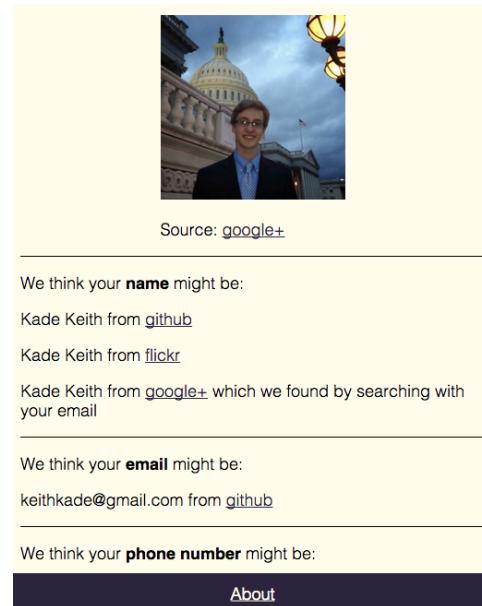


Fig. 1. A screen shot of the application results.

The client then visualizes that data for the user. The option to view the raw data is also provided.

The other chunk of data sent to the client are our guesses at answers to security questions. It is in a format similar to the one used for user attributes. The purpose of a security question is to provide a shared secret between two people, typically a website and a user, that the user can prove their identity with. The ideal questions is easy for an individual to recall when asked, but difficult for someone else to figure out. Our goal in providing guesses to these common security questions is not to debunk security questions as a form of knowledge-based authentication, but to show that people should be wary of questions that someone else could get the answer to.

Security questions were first used by banks to verify someone who wanted to perform an act but forgot the password to their account. The most common question was: "What is your mother's maiden name?" The question was difficult for anyone outside immediate family to answer, making it a good verification tool of a person's identity; however in today's social media age, info to questions like this one is easy sometimes easy to find.

We acquired a list of common security questions from sites the following sites: USAA, RBFCU, Facebook, and MySpace. While this is certainly not a comprehensive list of all security questions websites use, it provides both good and bad examples of questions based on our research into answering them.

TABLE I  
SECURITY QUESTIONS

Question	Difficulty	Plan of Attack
Who is your favorite actor, musician, artist, movie, or book?	Easy	Facebook - "Likes" are public by default and accessible without even logging in.  Reddit - We scan comments for keywords that signify the post may contain answer. Twitter - We scan tweets for keywords that signify a tweet may hold the information.
What was your high school mascot	Easy	Facebook - One of the most common pieces of public information on a person's page is high school name and city. With this information a Google search could provide a school's mascot.
Mothers Maiden Name	Easy	Facebook - Many people's parents are on Facebook and their relationship is listed as family.
In what city or town was your mother born?	Easy	Facebook - As mentioned above, once mother is found, we check for birthplace.
Fathers Middle Name	Medium	Facebook - Not too uncommon for people to list middle name or initial on Facebook. If it is just the initial that is likely enough to get a good guess when consulting common names.
Name of favorite pet	Medium	Reddit/Twitter - We scan posts for information pertaining to pets. With these a name may be found.
In what city were you born?	Medium	Facebook - We guess that someone's hometown is their birthplace. This may not be true for some, but many stay put in developing years so this is a good guess.
What street did you live on when you were 8 years old?	Medium	Facebook - Past addresses are a field that is public on some pages. Not entirely common but many use to be found by past friends.
Which phone number do you remember most from your childhood	Difficult	Facebook - Few people change their number so their current number may be the answer. However, this is not always listed as public makes this difficult.
When is your anniversary?	Difficult	Twitter - We searches posts to look for something about an anniversary. If a post is found the date may be mentioned. Additionally the timestamps of the post may be used to give a date.  Facebook - A visit to time line could provide answer, but we do not extract time line data. Many people post pictures or mention activities of their anniversary. Date check would provide the answer.
What was the last name of your first grade teacher?	Difficult	A paradigm example of a good security question. Most people remember their teachers. It is unlikely to be known by others though. Even friends from early schooling wouldn't remember unless in same class. Additionally it is rare for someone to post about this kind of info online.
Favorite Web Browser	Easy*	This question is odd in that no real information needs to be pulled. Looking up browser trends gave some conflicting info, but the general consensus is that roughly 90% of users use either Chrome, IE, Firefox, or Safari. Chrome is generally shown highest, IE second, and Firefox and Safari fairly close around third and fourth. With so few choices simple guesses would do.

From this table you can get an grasp on what makes a good question and what kinds of questions should be considered insecure. Questions about interests and family are the weakest. The popularity of social networking sites have led people to put out previously unprecedeted information about themselves into a publicly search-able database.

An example of a question that is still strong is: "What was the last name of your first grade teacher?" As mentioned in the plan of attack, there is little strategy we can hope to find this answer. The information is trivial, so it unlikely that someone share it publicly on websites, but the answer is very memorable. Additionally people wouldn't mind giving this answer out like they would things like social security numbers, which, while known by the individual and difficult to guess, are too valuable to give out. If these criteria could be met instead of going for the less than useful questions then security questions can still be a viable option for user identification as tools such as ours would have few available options to find the answers.

An interesting realization is that GitHub can make developers more vulnerable than the average internet user. Most

accounts link a username to an email and even to a real name or at least a portion of it. There is a real need for developers to allow their GitHub username to be used on the site. Developers want to be associated for the work they do. However many make the mistake of using a username that they have used on other sites, such as Reddit. For the analysis portion of our project, we want to crawl and create a database on every GitHub user. Our goal will be to use these crawlers to determine how many users have the links between accounts we would require.

#### IV. RESULTS

Our tool is capable of de-anonymizing a significant number of users. Some particularly successful cases include the following Reddit usernames: "drnic", "halfbyte", "plainprogrammer". Screenshots can be found in the appendix. Some of the information has been blurred out to protect privacy.

We wrote crawlers for Facebook, GitHub, and Reddit, and conducted analysis of the GitHub user base. We discovered that a significant portion of Github accounts can be linked to Reddit and Facebook. Despite being a site primarily used

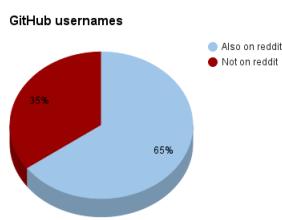


Fig. 2. Confidence: 99% Interval: .97

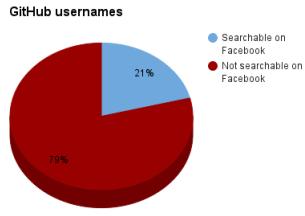


Fig. 3. Confidence: 99% Interval: 2.1

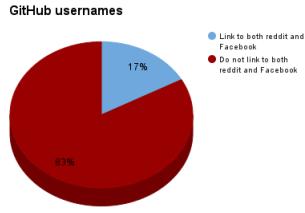


Fig. 4. Confidence: 99% Interval: 2.49

by programmers and developers, people who arguably should understand the risks, many use the same username on GitHub as well as Reddit. They also publish their real names and the email they used on Facebook. This suggests that a huge portion of the online population is de-anonymizable by naive techniques such as the ones we used. A much more feature complete tool could achieve this. It is possible that such tools have already been created and are being used by government agencies and malicious individuals. We hope that making tools such as ours will help individuals realize how vulnerable their information is and help them protect themselves.

We encountered a couple of issues with scale-ability. The first being that sites such as Facebook use captcha to prevent too many automated requests. We developed a workaround for this, but did not implement it fully as it would be unethical. Our service recognizes when Facebook sends us a captcha, downloads the image, and then hosts it on our own server. We then send out an email alert to ourselves so that one of us can go answer the captcha. Once one of us answers, the server simulates a response Facebook as though the captcha had been filled in there. A malicious agent could do this

on a much larger scale by impersonating an OCR researcher on Mechanical Turk and getting human beings to solve the captcha problems that their crawling and scrapping bot would run into. This by no means would allow the bot to perform its tasks at a pace equivalent to a bot that could solve captchas on its own, or one that simply didn't have to contend with captchas, but it would allow it to move at a reasonable pace. A pace fast enough to make de-anonymizing individuals on the web automatically plausible and affordable.

The second issue is that, sites like the Aggie Network are not always relevant, since they can only provide information for a limited user base (Texas A&M Students and Alumni). Sites from other colleges could be included, but to have a separate check for each college and university would not be reasonable. The use of Aggie Network however, does demonstrate that user have a tendency to reveal lots of their private information online, especially to semi-private networks, such as Aggie Network. This information is not as secure as users might believe.

## V. FUTURE WORK

The most promising future work is a study of the behaviors of people who use the tool we created. If someone uses our tool and finds that they are less anonymous than they thought, they will hopefully take action to protect themselves. We can analyze how effective those actions are by recording how much data is found about a user the first time they use the tool, and then checking at a later date to see if they have protected themselves by removing some of that data. The primary challenge of this is that we want to avoid storing data about our users if possible.

Additionally, just like GitHub has the tendency to de-anonymize developers there are possibly other sites that do the same thing for other professions for similar reasons. A future project could identify these sites and use them de-anonymize more individuals.

One of the main limitations of this kind of work is staying within legal and ethical bounds. Using our ability to de-anonymize people and possibly discover security questions is dangerous and it is important to remember that when developing these kinds of tools. There are a number of improvements that a malicious developer could make to the system, such as storing user profile data locally instead of requesting it real time, especially for sites like Aggie Network. This violates the Terms of Service of most sites though. Hopefully, we have shown that this information could be used to do nasty things, without actually doing them.

## VI. CONCLUSION

A strength of the internet is that it allows users to be anonymous if they so desire. This allows people to express themselves however they like without fear of giving up any of their privacy. Oftentimes though the user is in fact the greatest threat to his or her own privacy. Our tool provides value in that it educates users in real time on how anonymous they are online, and allows them to protect their anonymity if they have inadvertently revealed more about themselves than they would like.

## REFERENCES

- [1] "On the internet, nobody knows you're a dog," 2014. [Online]. Available: <http://knowyourmeme.com/memes/on-the-internet-nobody-knows-youre-a-dog>
- [2] S. T. Peddinti and K. W. Ross, ""on the internet , nobody knows you're a dog": A twitter case study of anonymity in social networks categories and subject descriptors."
- [3] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," 2009.
- [4] "De-anonymizing social networks," 2012.
- [5] D. B. Leyla Bilge, Thorsten Strufe and E. Kirda, "All your contacts are belong to us," *Proc. 18th Int. Conf. World wide web - WWW 09* (2009), p. 551, 2009. [Online]. Available: DOI:<http://dx.doi.org/10.1145/1526709.1526784>
- [6] T. H. E. K. D. B. Marco Balduzzi, Christian Platzer and C. Kruegel, "Abusing social networks for automated user profiling.pdf," pp. 1–21, 2010.
- [7] A. Rabkin, "Personal knowledge questions for fallback authentication: security questions in the era of facebook," *Proc. 4th Symp. Usable Priv. Secur*, p. 1323, 2008. [Online]. Available: DOI:<http://dx.doi.org/10.1145/1408664.1408667>

## APPENDIX

On the following pages are the outputs of our service successfully de-anonymizing three Reddit usernames.



Source: [spotify](#)      Source: [github](#)      Source: [instagram](#)      Source: [twitter](#)      Source: [google+](#)

---

We think your **name** might be:

[REDACTED] from [spotify](#)  
[REDACTED] from [github](#)  
[REDACTED] from [instagram](#)  
[REDACTED] from [twitter](#)  
[REDACTED] from [google+](#) which we found by searching with your email  
[REDACTED] from [facebook](#) which we found by searching with your email  
[REDACTED] from [flickr](#) which we found by searching with your email

---

We think your **email** might be:

[REDACTED]@[gmail.com](#) from [github](#)

---

We think your **phone number** might be:

[REDACTED] from [aggieNetwork](#) which we found by searching with your real name

---

We think your **geographic area** might be:

[REDACTED] from [twitter](#)

---

We think your **city** might be:

[REDACTED] from [aggieNetwork](#) which we found by searching with your real name

---

We think your **blog** might be:

[REDACTED] from [twitter](#)  
[REDACTED] from [github](#)  
[REDACTED] from [instagram](#)

---

We think the following text might contain an answer to the question "**What was your high school mascot?**":

[REDACTED] from [facebook](#)

Fig. 5. “drnic” dox report

Fig. 5. “drnic” dox report

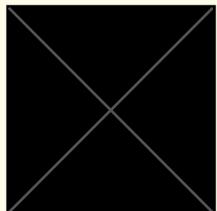
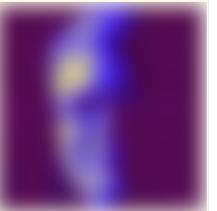
				
Source: <a href="#">spotify</a>	Source: <a href="#">github</a>	Source: <a href="#">instagram</a>	Source: <a href="#">twitter</a>	Source: <a href="#">youtube</a>
<hr/>				
We think your <b>name</b> might be:				
 from <a href="#">spotify</a>				
 from <a href="#">github</a>				
 from <a href="#">instagram</a>				
 from <a href="#">twitter</a>				
 from <a href="#">youtube</a>				
 from <a href="#">google+</a> which we found by searching with your email				
 from <a href="#">facebook</a> which we found by searching with your email				
 from <a href="#">flickr</a> which we found by searching with your email				
<hr/>				
We think your <b>email</b> might be:				
 from <a href="#">github</a>				
<hr/>				
We think your <b>geographic area</b> might be:				
 from <a href="#">twitter</a>				
<hr/>				
We think your <b>hometown</b> might be:				
 from <a href="#">flickr</a> which we found by searching with your email				
<hr/>				
We think your <b>blog</b> might be:				
 from <a href="#">twitter</a>				
 from <a href="#">github</a>				
<hr/>				
We think the following text might contain an answer to the question " <b>Who is your favorite musician, or artist?</b> ":				
 from <a href="#">facebook</a>				
 from <a href="#">facebook</a>				
 from <a href="#">facebook</a>				
<hr/>				
We think the following text might contain an answer to the question " <b>What is your favorite book?</b> ":				
 from <a href="#">facebook</a>				
<hr/>				
We think the following text might contain an answer to the question " <b>What is the name of your first school?</b> ":				
 from <a href="#">facebook</a>				
<hr/>				
We think the following text might contain an answer to the question " <b>What was your high school mascot?</b> ":				
 from <a href="#">facebook</a>				

Fig. 6. "halfbyte" dox report

Fig. 6. "halfbyte" dox report

Hello James Thompson



Source: [twitter](#)      Source: [github](#)      Source: [facebook](#)      Source: [instagram](#)      Source: [google+](#)

---

We think your **name** might be:

[REDACTED] from [twitter](#)  
[REDACTED] from [github](#)  
[REDACTED] from [facebook](#) which we found by searching with your email  
[REDACTED] from [instagram](#)  
[REDACTED] from [google+](#) which we found by searching with your email

---

We think your **email** might be:

[REDACTED] from [github](#)

---

We think your **phone number** might be:

[REDACTED] from [aggieNetwork](#) which we found by searching with your real name

---

We think your **geographic area** might be:

[REDACTED] from [twitter](#)

---

We think your **city** might be:

[REDACTED] from [aggieNetwork](#) which we found by searching with your real name

---

We think your **blog** might be:

[REDACTED] from [twitter](#)  
[REDACTED] from [github](#)

---

We think the following text might contain an answer to the question "Who is your favorite musician, or artist?":

[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)

---

We think the following text might contain an answer to the question "What is your favorite book?":

[REDACTED] from [facebook](#)

Fig. 7. “plainprogrammer” dox report

Fig. 7. “plainprogrammer” dox report

We think your city might be:  
[REDACTED] from [aggieNetwork](#) which we found by searching with your real name

We think your blog might be:  
[REDACTED] from [twitter](#)  
[REDACTED] from [github](#)

We think the following text might contain an answer to the question "Who is your favorite musician, or artist?":  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)

We think the following text might contain an answer to the question "What is your favorite book?":  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)

We think the following text might contain an answer to the question "What is the name of your first school?":  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)

We think the following text might contain an answer to the question "What is your favorite movie?":  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)

We think the following text might contain an answer to the question "What is your mother's maiden name?":  
[REDACTED] from [facebook](#)

We think the following text might contain an answer to the question "What was your high school mascot?":  
[REDACTED] from [facebook](#)

We think the following text might contain an answer to the question "What is your favorite sports team?":  
[REDACTED] from [facebook](#)

We think the following text might contain an answer to the question "What is your favorite tv show?":  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)  
[REDACTED] from [facebook](#)

[Show raw data](#)

Fig. 8. "plainprogrammer" dox report continued

Fig. 8. "plainprogrammer" dox report continued