UC Santa Barbara

UC Santa Barbara Previously Published Works

Title

Uncovering social network Sybils in the wild

Permalink

https://escholarship.org/uc/item/8m72n8sz

Journal

ACM Transactions on Knowledge Discovery from Data, 8(1)

ISSN

1556-4681

Authors

Yang, Zhi
Wilson, Christo
Wang, Xiao
et al.

Publication Date

2014-02-01

DOI

10.1145/2556609

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Uncovering Social Network Sybils in the Wild

ZHI YANG, Peking University
CHRISTO WILSON, University of California, Santa Barbara
XIAO WANG, Peking University
TINGTING GAO, Renren Inc.
BEN Y. ZHAO, University of California, Santa Barbara
YAFEI DAI, Peking University

Sybil accounts are fake identities created to unfairly increase the power or resources of a single malicious user. Researchers have long known about the existence of Sybil accounts in online communities such as file-sharing systems, but they have not been able to perform large-scale measurements to detect them or measure their activities. In this article, we describe our efforts to detect, characterize, and understand Sybil account activity in the Renren Online Social Network (OSN). We use ground truth provided by Renren Inc. to build measurement-based Sybil detectors and deploy them on Renren to detect more than 100,000 Sybil accounts. Using our full dataset of 650,000 Sybils, we examine several aspects of Sybil behavior. First, we study their link creation behavior and find that contrary to prior conjecture, Sybils in OSNs do not form tight-knit communities. Next, we examine the fine-grained behaviors of Sybils on Renren using clickstream data. Third, we investigate behind-the-scenes collusion between large groups of Sybils. Our results reveal that Sybils with no explicit social ties still act in concert to launch attacks. Finally, we investigate enhanced techniques to identify stealthy Sybils. In summary, our study advances the understanding of Sybil behavior on OSNs and shows that Sybils can effectively avoid existing community-based Sybil detectors. We hope that our results will foster new research on Sybil detection that is based on novel types of Sybil features.

Categories and Subject Descriptors: C.2 [**General**]: Security and Protection (e.g., firewalls); J.4 [**Computer Applications**]: Social and Behavioral Sciences

General Terms: Measurement, Security

Additional Key Words and Phrases: Online social networks, spam, Sybil attacks, user behavior, measurement

ACM Reference Format:
Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y. Zhao, and Yafei Dai. 2014. Uncovering social network Sybils in the wild. ACM Trans. Knowl. Discov. Data 8, 1, Article 2 (February 2014), 29 pages.
DOI: http://dx.doi.org/10.1145/2556609

The first version of this article appeared in the proceedings of Internet Measurement Conference (IMC) 2011 [Yang et al. 2011]. This version extends the original by including new Sections 4, 5, and 6.
This work is supported in part by the National Basic Research Program of China (Grant No. 2011CB302305), the National High Technology Research and Development Program of China (Grant No. 2013AA013203), and the National Science Foundation for Young Scholars of China (Grant No. 61202423). It is also supported by the National Science Foundation under grants IIS-0916307 and IIS-847925. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.
Authors' addresses: Z. Yang, X. Wang, and Y. Dai, Department of Electrical and Computer Engineering, Peking University, Beijing 80071, P.R. China; email: {yangzhi, wangxiao, dyf}@net.pku.edu.cn; C. Wilson and B. Y. Zhao, Computer Science Department, UC Santa Barbara, Santa Barbara, CA; email: {bowlin, ravenben}@cs.ucsb.edu; T. Gao, Security Group, Renren Inc., Beijing, China; email: tingting.gao@renren-inc.com.
Permission to make digital or hard copies of part or all 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 show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2014 ACM 1556-4681/2014/02-ART2 $15.00
DOI: http://dx.doi.org/10.1145/2556609

1. INTRODUCTION

Sybil attacks [Douceur 2002] are one of the most prevalent and practical attacks against distributed systems. In this attack, a user creates multiple fake identities, known as Sybils, to unfairly increase their power and influence within a target community. Distributed systems are ill-equipped to defend against this attack, since determining a tight mapping between real users and online identities is an open problem. To date, researchers have demonstrated the efficacy of Sybil attacks against P2P systems [Lian et al. 2007], anonymous communication networks [Bauer et al. 2007], and sensor networks [Newsome et al. 2004].

Recently, Online Social Networks (OSNs) have also come under attack from Sybils. Researchers have observed Sybils forwarding spam and malware on Facebook [Gao et al. 2010] and Twitter [Grier et al. 2010; Thomas et al. 2011], as well as infiltrating social games [Nazir et al. 2010]. Looking forward, Sybil attacks on OSNs are poised to become increasingly widespread and dangerous as more people come to rely on OSNs for basic online communication [Murphy 2010; Lenhart et al. 2010] and as replacements for news outlets [Kwak et al. 2010].

To address the problem of Sybils on OSNs, researchers have developed algorithms such as SybilGuard [Yu et al. 2006], SybilLimit [Yu et al. 2008], SybilInfer [Danezis and Mittal 2009], and SumUp [Tran et al. 2009] to perform decentralized detection of Sybils on social graphs. These systems detect Sybils by identifying tightly connected communities of Sybil nodes [Viswanath et al. 2010].

However, recent work has shown that one of the key assumptions of community-based Sybil detectors—fast mixing time—does not hold on social graphs where edges correspond to strong real-world trust (e.g., DBLP, Physics co-authorship, Epinions) [Mohaisen et al. 2010]. Thus, community-based Sybil detectors do not perform well on "trusted" social graphs. To date, however, no large-scale studies have been performed to validate the assumptions of community-based Sybil detectors on *untrusted* social networks such as Facebook and Twitter.

In this article, we describe our efforts to detect and understand Sybil account activity in Renren, one of the largest OSNs in China. In Section 2, we use ground-truth data on Sybils provided by Renren Inc. to characterize Sybil behavior. We identify several behavioral attributes that are unique to Sybils and leverage them to build a measurement-based, real-time Sybil detector. Our detector is currently deployed on Renren's production systems; between August 2010 and February 2011, it led to the banning of more than 100,000 Sybil accounts.

In Section 3, we analyze the graph structural properties of Sybils on Renren, based on the 100,000 Sybils identified by our detector, as well as 560,000 more identified by Renren using prior techniques. Most interestingly, we find that contrary to prior conjecture, Sybil accounts in Renren do not form tight-knit communities: $>70\%$ of Sybils do not have *any* edges to other Sybils at all. Instead, attackers use biased random sampling to identify and send friend requests to popular users, since these users are more likely to accept requests from strangers. This strategy allows Sybil accounts to integrate seamlessly into the social graph.

We analyze the remaining 30% of Sybils that are friends with other Sybils and discover that 69% (65,000 accounts) form a single connected component. By analyzing the creation timestamps of these edges, we determine that this component formed accidentally, not due to coordinated efforts by attackers. We briefly survey several popular Sybil management tools and show that large Sybil components can form naturally due to the biased way in which these tools target friend requests. These results demonstrate that even when Sybils on Renren do friend each other, existing community-based Sybil detectors would not be able to locate these loosely connected Sybil groups. These results

indicate that relying on graph structure alone is not a tenable strategy for detecting Sybils on today's OSNs.

One potential way to improve Sybil defense is to incorporate additional information about Sybil behavior. Toward this end, we examine the fine-grained behaviors of Sybils on Renren in Section 4. Our study is based on detailed clickstream data summarizing 16,010,645 clicks over a 1-month period for 50,000 Sybils. These traces allow us to contrast the session-level differences between normal users and Sybils, as well as capture the exact behavior of Sybils as they crawl the OSN and generate spam. Our results show that Sybils abuse the friend recommendation features of OSNs to locate targets for friending.

In Section 5, we examine collusion between spamming Sybils on Renren. We show that although Sybils do not have explicit friendship links with each other, Sybils collude behind the scenes to promote spam blogs, revealing that they are under the control of a single attacker. We use content similarity and temporal correlation to quantify the *inferred* links between Sybils and quantify the extent of Sybil collusion on Renren.

Finally, in Section 6, we investigate enhanced techniques to identify stealthy Sybils. The motivating factor is that attackers may adapt and change the behavior of Sybils, especially since the details of our detector are now publicly available. We show that by combining community detection with friend request statistics, we can detect Sybils that collude to evade our currently deployed Sybil detector. These results demonstrate that although Sybil behavior may evolve in the future, we will still be able to accurately detect and ban these malicious accounts.

In summary, the dataset leveraged in this article opens a window of insight into malicious activity on OSNs that was previously only available to OSN providers. Our dataset includes key pieces of additional information (e.g., friend requests, edge creation timestamps, and clickstreams) that cannot be inferred from crawling publicly available data. Although other studies have examined fake and malicious accounts on OSNs [Gao et al. 2010; Grier et al. 2010; Thomas et al. 2011], these studies rely on a limited range of publicly crawlable information to identify Sybils (usually friend relationships and wall posts).

Unfortunately, the caveat of focusing on a single dataset is that the generality of results becomes an issue. We do not claim that Sybils on all OSNs exhibit the same behaviors observed on Renren. Instead, we view this work as a first step toward bringing practical understanding to the area of social Sybil research. Our hope is that our results will foster new research on Sybil detection that is based on novel types of Sybil features and grounded in more realistic assumptions about Sybil behavior.

2. DETECTING SYBILS

In this section, we set the backdrop for our data analysis. First, we briefly introduce the Renren OSN and describe the role of Sybil accounts in Renren. Second, we describe experiments characterizing Sybil accounts on a verified ground-truth dataset provided by Renren. Finally, we describe and build a real-time Sybil account detector deployed on Renren and show how it led to the large Sybil dataset that we analyze in the remainder of the article.

2.1. The Renren Network and Sybil Accounts

With 220 million users, Renren (http://www.renren.com) is one of the most popular OSNs in China and provides functionality and features similar to Facebook. Like Facebook, Renren started in 2005 as a social network for college students in China and then saw its user population grow exponentially once it opened its doors to nonstudents. Renren users maintain personal profiles, upload photos, write diary entries (blogs), and establish bidirectional social links with friends. The most popular type of

user activity is sharing blog entries, which can be forwarded across social hops like "retweets" on Twitter.

As its user population has grown, Renren has become an attractive venue for companies to disseminate information about their products. This has created opportunities for Sybil accounts to spam advertisements for companies, a growing trend observed by the analytics team at Renren. The increased prevalence of spam on Renren mirrors similar findings from Facebook [Gao et al. 2010] and Twitter [Grier et al. 2010; Thomas et al. 2011].

To effectively attract friends and disseminate advertisements, most Sybil accounts on Renren blend in extremely well with normal users. They tend to have completely filled user profiles with realistic background information, coupled with attractive profile photos of young women or men, making their detection quite challenging.

Prior to this project, Renren had already deployed a suite of orthogonal techniques to detect Sybil accounts, including using thresholds to detect spamming, scanning content for suspect keywords and blacklisted URLs, and providing Renren users with the ability to flag accounts and content as abusive. However, these techniques are generally ad hoc, require significant human effort, and are effective only after spam content has been posted. To improve security for their users, Renren began a collaboration with our research team in December 2010 to augment their detection systems with a systematic, real-time solution. To support the project, Renren provided full access to user data and operational logs on their servers, as well as allowed us to test and deploy research prototypes of Sybil detectors on their operational network.

Defining Sybils. In this study, as in prior work [Yu et al. 2006; Danezis and Mittal 2009; Yu et al. 2008; Tran et al. 2009], we are interested in detecting and deterring the use of mass Sybil identities by malicious users. We broadly define Sybils as fake accounts created for the purpose of performing spam or privacy attacks against normal users. In other words, we are interested in *malicious Sybils*.

Attempting to specifically define the boundary between what is a "real" and what is a "fake" account on an OSN is a task fraught with ambiguity. Next, we discuss several cases where the boundaries between real and fake and malicious and benign are questionable. In each case, we clarify what types of malicious Sybils fall under our definition. These are the types of accounts that we aim to detect, measure, and ultimately stop through the rest of the article.

Identifying Malicious Activity. A key component of our definition is that Sybils engage in malicious activity. This leads to the following question: *what is the definition of malicious activity?*

We define malicious activity to be actions taken by an attacker that directly or indirectly support a monetization strategy. Examples of monetization strategies include targeting users with spam and phishing attacks. Prior work has shown that these strategies are widely used by attackers against other OSNs [Gao et al. 2010; Grier et al. 2010; Thomas et al. 2011]. Note that our definition *does not* cover legitimate monetization strategies, such as keyword, banner, or news-feed advertising.

Rather than simply focusing on the outward signs of malicious activity (e.g., spam), our definition also covers indirect behavior that is a required precursor to attacks. In particular, in order for attackers to reach a user on OSNs like Renren and Facebook, the attacker must first be friends with that user. This is because, by default, only friends may create posts on a user's wall. Thus, creating friendship links to normal users is a key step for Sybil accounts on Renren; the Sybils *cannot be monetized* without first acquiring many friends. To gather many friends, Sybils must send out a large amount of unsolicited friend requests to normal users. It is the social network provider's (e.g., Renren's) responsibility to detect and ban these *harmful* Sybil accounts to protect users from malicious attacks.

Although friending normal users is a precursor for malicious activity, our work is agnostic to the specifics of these malicious activities (e.g., spam, phishing). Furthermore, we make no assumptions about the methods and tools used to create and control the Sybils.

Benign Fake Accounts. Our definition of Sybils *does not* include fake accounts generated by users for benign purposes. Examples of benign activity include using a pseudonym to preserve privacy and anonymity, acting on behalf of young children, and separating work and personal identities. As we discuss in Section 2.3, these benign Sybils are unlikely to be flagged by the techniques proposed in this work.

It is possible that an attacker could also create benign Sybils that behave identically to normal users and appear on the surface to be real. However, we are only interested in detecting Sybil accounts that perform attacks—that is, they deviate from normal user behavior. Thus, benign Sybils would not be targeted by our techniques, since they behave *exactly* like normal users. The goal of our system is to catch fake accounts if they ever start performing malicious actions.

Inactive Accounts. Inactive accounts *do not* fall under our definition of malicious Sybils. Determining whether an inactive account is a malicious Sybil is challenging because there is no behavioral data (e.g., friend requests, status updates) to leverage for classification. Renren uses heuristics to detect naïve attempts to create bulk accounts (e.g., many creations that originate from a single IP address). However, we do not have access to this data, and thus we do not consider the problem of inactive accounts in this study.

The Sybil detector that we develop in this article will not catch inactive Sybil accounts until after they become active (i.e., start generating friend requests and spam). In this case, the goal of our detector is to catch these accounts as quickly as possible once they become active to minimize the amount of damage they can do to normal users.

2.2. Characterizing Sybil Accounts

Our approach to building a real-time Sybil detector begins by first identifying features that distinguish Sybil accounts from normal users. To help, Renren provided us with two sets of user accounts, containing 1,000 Sybil accounts and 1,000 non-Sybil accounts, respectively. The Sybil accounts were previously identified using existing mechanisms. A volunteer team carefully scrutinized all accounts in both sets to confirm that they were correctly classified by looking over detailed profile data, including uploaded photos, messages sent and received, email addresses, and shared content (blogs and Web links).

Using this dataset as our ground truth, we searched for behavioral attributes that serve to identify Sybil accounts. After examining a wide range of attributes, we found four potential identifiers. We describe them each in turn and illustrate how they characterize Sybils in our dataset.

Invitation Frequency. Invitation frequency is the number of friend requests that a user has sent within a fixed time period (e.g., an hour). Figure 1 shows the friend invitation frequency of our dataset, averaged over long-term (400-hour) and short-term (1-hour) time scales. Since adding friends is a goal for all Sybil accounts, they are much more aggressive in sending requests than normal users. There is a clear separation: accounts sending more than 20 invites per time interval are Sybils. This result holds true at both long and short time scales, meaning that invitation frequency can be used to detect Sybils without significant delays. For example, a threshold of 40 requests/hour can identify $\approx$70% of Sybils with no false positives. Prior to our work, Renren deployed a CAPTCHA that users must solve if they send 50 or more requests in a day, which explains the apparent upper limit on friend requests.

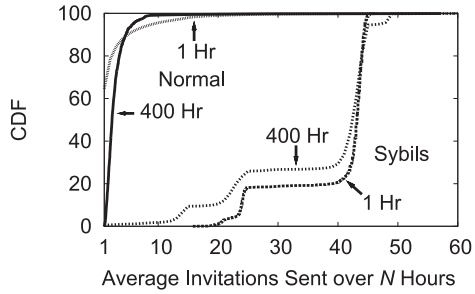


Fig. 1. Friend invitation frequency over two time scales.

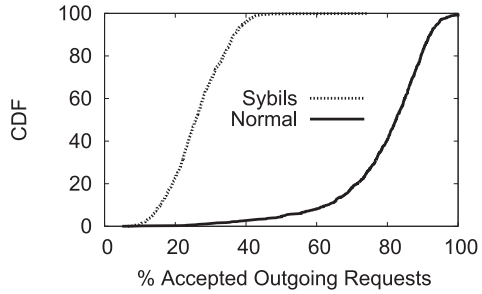


Fig. 2. Percentage of accepted *outgoing* friend requests.

Outgoing Requests Accepted. A second distinguishing feature is the fraction of outgoing friend requests confirmed by the recipient. The CDF shown in Figure 2 shows a distinct difference between Sybils and normal users. In general, non-Sybil users have high accepted percentages, with an average of 79%. On average, however, only 26% of all friend requests sent by Sybil accounts are accepted. This is unsurprising, since normal users typically send invites to people with whom they have prior relationships, whereas Sybils target strangers.

Despite prior studies that show users accept requests indiscriminately [Stringhini et al. 2010; Sophos 2007], our results show that most users can still effectively identify and decline invitations from Sybils. The fact that some users still accept requests from Sybils is explained by two factors. First, most Sybils target members of the opposite sex by using photos of attractive young men and women in their profiles. Although women make up 46.5% of the overall Renren user population, they make up 77.3% of the Sybils in our dataset. Second, Sybils typically target popular, high-degree users who are more likely to be careless about accepting friend requests from strangers. We further explore this point in Section 3.4.

Incoming Requests Accepted. Figure 3 plots a CDF of users by the fraction of incoming friend requests that they accept. The incoming requests accepted by non-Sybil users are spread across the board. In contrast, Sybil accounts are nearly uniform in that they accept all incoming friend requests (e.g., 80% of Sybils accepted all friend requests). In fact, many of the Sybils with $<100\%$ accept rate fall into this category because Renren banned them before they could respond to all outstanding requests. However, since Sybil accounts receive few friend requests, this detection mechanism can incur significant delay.

Clustering Coefficient. Clustering coefficient (cc) is a graph metric that measures the mutual connectivity of a user's friends. Since normal users tend to have a small number of well-connected social cliques, we expect them to have much higher cc values than Sybil accounts, which are likely to befriend users with no mutual friendships. Figure 4 plots the CDF of cc values for each user's first 50 friends (sorted by time). As expected, non-Sybil users have cc values orders of magnitude larger than Sybil users (average cc values of 0.0386 and 0.0006, respectively). Since cc can be computed based on invitations only (i.e., user responses are not required), it can potentially perform well as a real-time Sybil detection metric.

2.3. Building and Running a Sybil Detector

Our analysis results indicate that a threshold-based scheme can effectively detect most Sybil accounts. Our next step is to verify this assertion by comparing the efficacy of a simple threshold detector against a more complex learning algorithm.

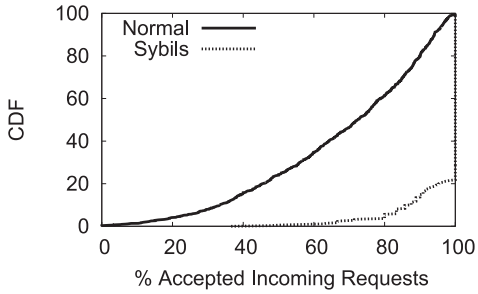


Fig. 3. Percentage of accepted *incoming* friend requests.

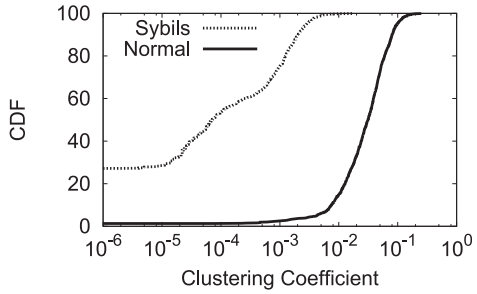


Fig. 4. Clustering coefficient for users' 50 first invitees.

Table I. Performance of SVM and Threshold Classifiers

		SVM Predicted		Threshold Predicted	
		Sybil	Non-Sybil	Sybil	Non-Sybil
True	Sybil	98.99%	1.01%	98.68%	1.32%
	Non-Sybil	0.66%	99.34%	0.5%	99.5%

We apply a Support Vector Machine (SVM) classifier to our ground-truth dataset of 1,000 normal users and 1,000 Sybils. We randomly partition the original sample into five subsamples, four of which are used for training the classifier and the last used to test the classifier. The results in Table I show that the classifier is very accurate, correctly identifying 99% of both Sybil and non-Sybil accounts. We compare these results to those of a threshold-based detector: *outgoing requests accepted* % $<$ 0.5 $\wedge$ *frequency* $>$ 20 $\wedge$ *cc*$<$ 0.01. Our results show that a properly tuned threshold-based detector can achieve performance similar to the computationally expensive SVM.

Real-Time Sybil Detection. Our analytical results using the ground-truth dataset led to the design of an adaptive, threshold-based Sybil detector that identifies Sybil accounts in near real time. The detector monitors all accounts using a combination of friend-request frequency, outgoing request acceptance rates, and clustering coefficient. The security team at Renren applied the analysis techniques from Section 2.2 to a large sample of Sybils (more than 100K Sybil accounts) to derive initial parameter values for the deployed detector (i.e., the parameter values given in this article are not the same values used by the deployed detector). After the detector has been bootstrapped, it uses an adaptive feedback scheme to dynamically tune the threshold parameters on the fly. The adaptive feedback is drawn from the customer complaint rate to Renren's support department, which we outline in more detail below.[1]

Tuning the thresholds minimizes the likelihood of false-positive classifications of normal accounts as Sybils. Because our system works by detecting abnormal behavior in friending or content dissemination, it is unlikely to detect benign Sybils that behave like normal users. However, as mentioned earlier, the drawback of this approach is that our detector will not catch inactive Sybils. Inactive Sybils will not be detected until after they begin friending normal users.

Our detector incorporates real-time changes in friendship links when calculating acceptance percentages. In some cases, normal users accept friend requests from Sybils only to later revoke the friendship. This causes the accept percentage for the Sybil to

[1]We omit details of the adaptive scheme and the specific parameter values to protect Renren's security and confidentiality.

drop. Similarly, when Renren bans Sybils, all of their edges are destroyed. This causes the acceptance percentages for other Sybils with which they are linked to drop. In both cases, the decrease in acceptance percentage helps our detector to more accurately detect Sybils.

After offline testing, Renren deployed our Sybil detection mechanism in late August 2010, and it has been in continuous operation ever since. From August 2010 to February 2011, Renren administrators used our system to detect and subsequently ban nearly 100,000 Sybil accounts in Renren. In addition to these accounts, Renren provided us with data on nearly 560,000 accounts that were detected and banned using prior techniques from 2008 to February 2011. For the remainder of this article, we will use all of these Sybil accounts (660,000 in all) to study the behavior of Sybil accounts.

Sybil Account Behavior. We confirmed that the Sybil accounts identified by our detector are actually malicious by analyzing the content generated by these accounts. Offline analysis confirmed that 67% of content generated by Sybils trips Renren's spam detectors (e.g., suspicious keyword filter, blacklisted URLs). Of the remaining accounts, the vast majority were banned before they had a chance to generate any content. Section 5 delves into the details of Sybil behavior and spam content on Renren.

False Positives. To assess false positives, we examine feedback to Renren's customer support department. Renren operates a telephone number and email address where customers can attempt to get banned accounts reinstated. Complaints are evaluated by a human operator, who determines if the account was banned erroneously.

We use the complaint rate, measured as the number of complaints per day divided by the number of accounts banned per day, as an upper bound on false positives. During the 2-week period between December 13 and 26, 2010, Renren received nearly 50 complaints per day, with the complaint rate being almost 0.015, which is extremely low. Of these complaints, manual inspection confirms that 48% of the accounts are Sybils, meaning that attackers attempted to recover Sybils by abusing the account recovery process. The majority of the remaining complaints can be attributed to compromised accounts. Thus, the true false-positive rate is even less than the daily complaint rate.

3. SYBIL TOPOLOGY ANALYSIS

We now begin our analysis of Sybil accounts with respect to graph topological characteristics (and user collusion, later in Section 5). The analysis in this section (and subsequent sections) is performed on a large, ground-truth dataset of more than 660,000 Sybil accounts gathered by Renren over more than 5 years. Roughly 15% of this dataset (100K) was gathered from the new detector described in the previous section. The remaining 560K Sybil accounts have been detected using a wide variety of orthogonal techniques including (but not limited to)[2]:

—**IP Address Tracking:** Used to detect bulk account creation (e.g., many creations that originate from a single IP address or subnet).
—**Content Analysis:** Scanning content for spam-related keywords and blacklisted URLs.
—**Account Activity Statistics:** Simple heuristics used to detect abnormal activity by monitoring user clicks. Examples of abnormal activity include sending large numbers of direct messages, browsing many profiles, or sending too many friend requests within short time frames.
—**User Complaints:** Renren users can flag accounts and content as abusive, which triggers a manual response from Renren's security team.

[2]To protect Renren's security and confidentiality, this article does not go into the detail of these detection techniques.

Table II. Overview of the Sybil Dataset Used in Section 3

Detection Technique	Duration of Collection	No. of Sybils Detected
Our real-time detector	Aug. 2010 ∼ Feb. 2011	100K
IP tracking, content and activity analysis, user complaints, etc.	Jan. 2007 ∼ Feb. 2011	560K

Table II summarizes our Sybil dataset. Given the size of the dataset, the time duration over which the collection spans, and the range of detection techniques used, we believe that this dataset is representative of Sybil activity on Renren.

In this section, we are interested in analyzing whether Sybils in the wild can be identified using the community-based Sybil detectors that have been proposed by researchers. These detectors assume that Sybils form tight-knit groups that can be detected by looking for small quotient cuts in the graph. Thus, the linkages between Sybils are critical for these detectors to function. However, if Sybils do not form tight-knit groups, these detectors will not function—that is, these detectors are not capable of detecting loosely connected or disconnected Sybils.

Our results show that Sybils on Renren do not conform to the assumptions of existing work. Analysis of the degree distribution of Sybil accounts demonstrates that contrary to expectations, the vast majority of Sybils do not form social links with other Sybils. Furthermore, temporal analysis of social links between Sybils indicates that these connections are often formed randomly by accident rather than intentionally by attacker.

3.1. Sybil Community Detectors

SybilGuard [Yu et al. 2006], SybilLimit [Yu et al. 2008], SybilInfer [Danezis and Mittal 2009], SybilRank [Cao et al. 2012], and SumUp [Tran et al. 2009] are all algorithms for performing decentralized detection of Sybil nodes on social graphs. At their core, all of these algorithms are based on two assumptions of Sybil and normal user behavior:

(1) Attackers can create unlimited Sybils and edges between them. Edges between Sybils are beneficial, because they make Sybils appear more legitimate to normal users.
(2) The number of edges between Sybils and normal users will be limited, because normal users are unlikely to accept friend requests from unknown strangers.

Under these assumptions, Sybils tend to form tight-knit clusters, since the number of edges between Sybils is greater than the number of edges connecting to normal users. We refer to edges between Sybils as *Sybil edges*, whereas edges connecting Sybils and normal users are called *attack edges*.

Sybil detection algorithms identify Sybil clusters by locating the small number of edge cuts that separate the Sybil region from the social graph. SybilGuard, SybilLimit, and SybilInfer all leverage specially engineered random walks for this purpose, whereas SumUp uses a max-flow approach. Although all of these algorithms are implemented differently, it has been shown that they all generalize to the problem of detecting communities of Sybil nodes [Viswanath et al. 2010].

Problem Statement. Although these five algorithms have been shown to work on synthetic graphs (i.e., real social graphs with Sybil communities artificially injected), to date no studies have demonstrated their efficacy at detecting Sybils in the wild. In other words, *all of these techniques rely on the assumption that Sybils exhibit strong clustering; if Sybils do not follow this behavior, these techniques will not be effective*.

In the following sections, we examine the characteristics of Sybils on Renren in order to ascertain whether they are amenable to identification by community-based Sybil detectors.

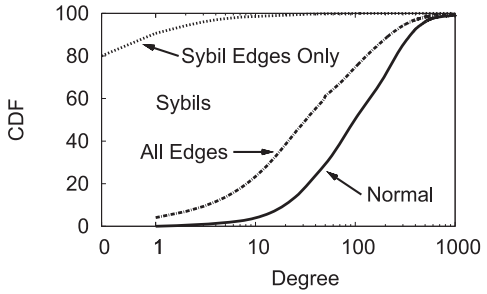


Fig. 5. The degree of Sybil accounts.

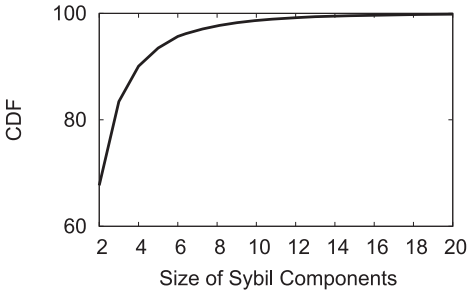


Fig. 6. The size of connected Sybil components.

3.2. Sybil Edges

We begin our analysis of Sybil topology by examining the degree distribution of Sybils on Renren. Our goal is to test the most basic assumption of community-based Sybil detectors: *do Sybils in the wild form tight-knit communities?* In order for Sybils to cluster, they must have at least one edge to another Sybil; otherwise, they will be disconnected.

Figure 5 shows the degree distribution of all 667,723 Sybil accounts compared to the degree distribution of a complete, 42 million node snapshot of the Renren social graph collected in 2008 [Jiang et al. 2010]. When all edges attached to Sybils are considered, the degree distribution is unremarkable. Sybils tend to have fewer friends than normal users, but both Sybils and the overall population follow the same general trend.

However, when we restrict the distribution to only edges between Sybils, we discover an unexpected result: only 20% of Sybils are friends with one or more other Sybils. This indicates that the vast majority of Sybils do not cluster with other Sybils. Instead, most Sybils only form attack edges and thus totally integrate into the normal social graph.

Sampling Bias. One potential issue when comparing degree distributions is whether the underlying samples are biased. In Figure 5, the degree distribution for the entire Renren population (the "Actual Degree Distribution" line) is not biased: the data comes from a complete snapshot of Renren gathered in 2008 [Jiang et al. 2010].

It is also unlikely that there is significant bias in the Sybil degree distribution. As mentioned earlier, it is possible that our detector did not locate some inactive Sybils. However, this is not an issue in practice: prior work on social Sybils found that 88% are dormant for 1 week or less [Thomas et al. 2011]. Given that our ground-truth Sybil data was collected (1) over the course of multiple years and (2) from multiple, complementary detection systems, it is likely that our dataset includes the vast majority of Sybils.

3.3. Sybil Communities

We now shift our focus to the minority of Sybils that do connect to other Sybils. Although we can conclude from Figure 5 that most Sybils in the wild do not obey the key assumption of community-based Sybil detectors, it is still possible that the connected minority are vulnerable to community detection. Thus, we now seek to answer the following questions: *what are the characteristics of Sybil communities on Renren*, and *would community-based Sybil detectors be able to identify them*?

To bootstrap our analysis, we construct a graph consisting solely of Sybils with at least one edge to another Sybil. The resulting graph is highly fragmented: it consists of 7,094 separate connected components. Figure 6 shows the size distribution of these

Table III. Statistics for the Five Largest Sybil Components

Sybils	Sybil Edges	Attack Edges	Audience
63,541	134,941	9,848,881	6,497,179
631	1,153	104,074	21,014
68	67	7,761	7,702
51	50	15,349	15,179
37	40	14,431	13,886

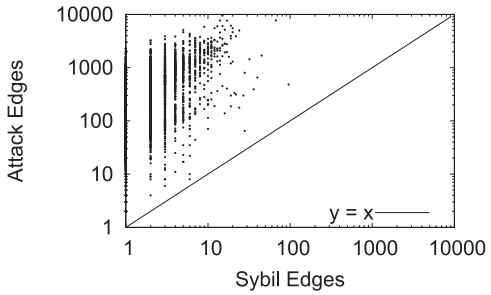


Fig. 7. Scatter plot of Sybil edges versus attack edges for Sybil components on Renren.

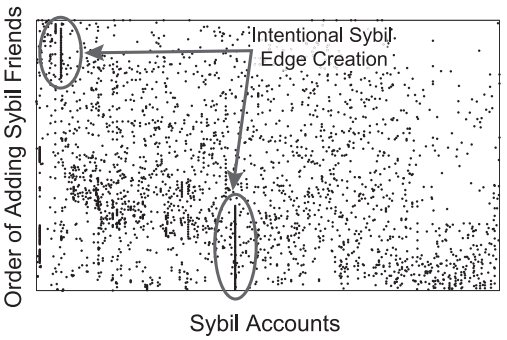


Fig. 8. The order of adding Sybil friends for 1,000 Sybils. Each column represents an individual Sybil.

Sybil components. The distribution is heavy tailed: although 98% of Sybil components have fewer than 10 members, the vast majority of Sybil accounts belong to a single, large connected component. Table III lists the details for the five largest Sybil components.

For connected components of Sybils to be identifiable by existing algorithms, they must form tight-knit communities. Put another way, the number of Sybil edges inside the community must be greater than the number of attack edges that connect to honest nodes. However, as shown in Table III, this assumption does not hold for the largest Sybil components on Renren.

Figure 7 shows a scatter plot comparing the number of Sybil edges and attack edges in each Sybil component on Renren. All components are above the 45-degree line, meaning that they have more attack edges than Sybil edges. Thus, no components meet the requirements for detection using existing community-based Sybil identification algorithms.

3.4. Sybil Edge Formation

We now examine the processes driving the formation of Sybil edges on Renren. In particular, we seek to determine if edges between Sybils are intentionally created by attackers. If so, then this means that community detection may still be a viable approach to detecting Sybils on OSNs. However, if Sybil edges are created unintentionally, then this raises a new question: *what process drives the accidental creation of Sybil edges?*

Temporal Characteristics. One simple litmus test for identifying intentional Sybil edge creation is examining the order in which edges were established. If Sybil edges are formed intentionally by attackers, then we would expect to see them created *before* friend requests are sent out to normal users. This behavior maximizes the utility of Sybil edges by giving Sybils the appearance of "normal" friend relations, thus (potentially) deceiving normal users into accepting friend requests from Sybils. In contrast, there is little utility for Sybils to gain by friending each other after friend requests have

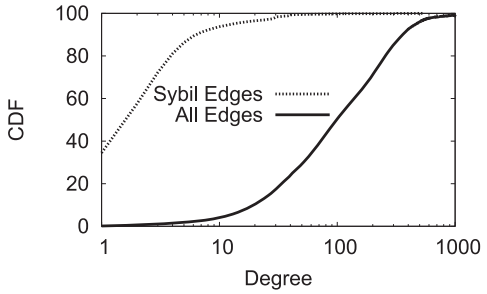


Fig. 9. Degree distribution of the largest Sybil component.

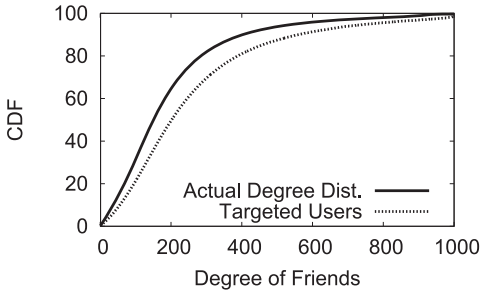


Fig. 10. Degree distribution of normal users (from Jiang et al. [2010]) and targets of friend requests from Sybils.

been sent to normal users. These edges will not be seen by normal users, and they are useless for carrying out attacks (e.g., sending out spam).

Figure 8 shows the order in which edges were created for 1,000 random Sybils drawn from the largest Sybil component on Renren (containing 63,541 Sybils). For each Sybil i with n edges, we construct the sequence of edges $\langle f_1, f_2, \ldots, f_n \rangle$, sorting the edges chronologically by creation time. Each column of the figure shows the sequence of edge creations for a particular Sybil, with black dots representing Sybil edges. Thus, if Sybils create edges to each other intentionally, we would expect to see unbroken lines of black dots on the figure, indicating rapid, sequential edge creation.

As shown in Figure 8, the order of Sybil edge creation is almost uniformly random. There are a few examples of solid vertical lines, indicating intentional edge creation. We highlight these examples in Figure 8. However, most Sybil edge creation is interspersed randomly with edges created to normal users. This indicates that the vast majority of Sybil edges in the large component were formed accidentally: attackers had nointention to link Sybils together and form a connected component. It is unclear why a tiny minority of Sybils exhibit correlated behavior.

Sybil Degree. In order to reinforce the idea that the vast majority of Sybil edges in the large component are not intentionally created, we plot the degree distribution of the large component in Figure 9. Here, 34.5% of Sybils only connect to one other Sybil, and 93.7% connect to 10 or fewer Sybils. It is unlikely that an attacker would expend the effort to link Sybils in such a loose way, since these edge counts are not high enough to make Sybils appear legitimate to normal users.

Biased Random Sampling. At this point, we have established that attackers do not create the vast majority of Sybil edges intentionally; instead, they appear to occur randomly by accident. To understand how this happens, we conducted a brief survey of three software tools used to manage Sybil accounts on Renren. The details for each tool are given in Table IV. The purpose of these tools is to automate the process of creating Renren accounts, forming edges between the Sybils and other users, and posting content en mass. The documentation for the tools in Table IV state that they select targets for friending by performing biased random sampling on the social graph to locate popular users. In practice, the tools simplify this process by leveraging Renren's friend recommendation features. These recommendation systems are designed to highlight popular, well-connected users. Thus, attackers are abusing Renren's features to locate a biased random sample of targets for friending.

Although we cannot be certain whether the Sybils in our dataset were created using the tools in Table IV, we can show that Sybils on Renren do bias friend requests

Table IV. Popular Sybil Creation and Management Tools

Tool Name & URL	Platform	Cost
Renren Marketing Assistant V1.0 http://www.duote.com/soft/30348.html	Windows	$37
Renren Super Node Collector V1.0 http://www.snstools.com/snstool/86.html	Windows	Contact Author
Renren Almighty Assistant V5.8 http://www.sns78.com/	Windows	Contact Author

toward users with high degrees. Figure 10 shows the degree distribution for all users who received friend requests from Sybils (we refer to these users as *targeted users*). We compare the degrees of the targeted users to the degree distribution of the same complete, 42 million node Renren snapshot (from Jiang et al. [2010]) used in Figure 5. Figure 10 illustrates that the degrees of targeted users are skewed toward high degrees when compared to the actual degree distribution of the Renren population.

Based on the advertised functionality of these tools, and the results in Figure 10, we can surmise that Sybil edges are created accidentally due to two factors. First, the goal of Sybils is to accrue many friends by sending out numerous friend requests. If a Sybil is successful, it becomes popular by virtue of its large social degree. Second, the biased random sampling performed by Sybil management tools is intentionally geared toward locating popular users. Thus, it is likely that these tools will, unbeknownst to the attacker, occasionally select Sybil nodes to send friend requests to. As shown in Figure 3, Sybils almost always accept incoming friend requests; hence, when this situation occurs, a Sybil edge is likely to be created.

4. SYBIL CLICKSTREAMS

To better understand the behavior of Sybil accounts, we now analyze the clickstream data of Sybils on Renren. We start by comparing the session-level characteristics of Sybils and normal users and then examine the types of activities in which Sybils engage within each session. Finally, we construct state-based models of Sybil and normal user behavior. These models help to underscore differences between normal and abnormal behavior on Renren.

Dataset and Methodology. Our study is based on detailed clickstream data for 50K Sybils and 50K normal users. These users were selected uniformly at random from Sybils and normal accounts that registered on Renren after June 1, 2009 (clickstream data was unavailable for older accounts). Our overall goal is to investigate whether Sybils can be detected early in their lifetime, so we focus on the first 30-day clickstream for each Sybil and normal user. In total, our month-long dataset includes 16,010,645 and 21,345,882 clicks for Sybils and normal users, respectively. Each click is characterized by an anonymized user ID, a timestamp, and a URL from which the click type can be ascertained.

Clicks from each user in the trace are grouped into sessions, where a session represents the sequence of a user's clicks during a single visit to an OSN. As in prior work, we define a session as over when a user does not click any links for 20 minutes [Benevenuto et al. 2009]. Session duration is calculated as the time interval between the first and last action within a session. Overall, the trace includes 3,182,481 and 1,596,531 sessions for Sybils and normal users, respectively.

4.1. Session-Level Characteristics

In this section, we characterize Sybils' activity at the session level. We seek to determine *how often and how long do Sybils connect to OSN sites, as compared with normal users?*

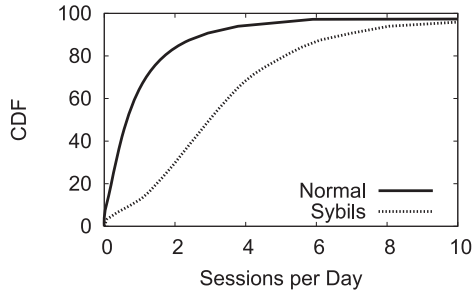


Fig. 11. Sessions per day for Sybil and normal users.

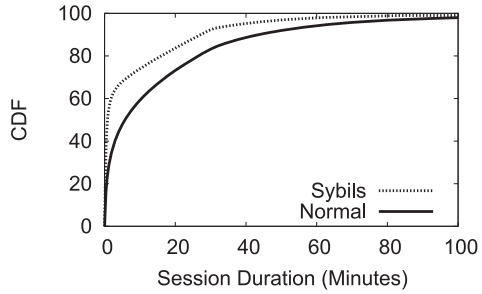


Fig. 12. Overall distribution of session durations.

Table V. Clicks from Normal Users and Sybils on Various Renren Activities
Activities with <1% of clicks are omitted for brevity.

Category	Description of Activity	No. of Sybil Clicks	%	No. of Normal Clicks	%
Photo	Visit photo	370,359	2.3	11,410,972	**52.9**
	Visit album	100,915	0.63	1,182,921	**5.5**
Message	Instant message a friend	137,420	0.86	2,085,133	**9.7**
Share	Share content	344,085	2.2	776,461	**3.6**
Friending	Invite from guide page	6,924,738	**43.1**	810,873	3.8
	Invite from recommendation	2,865,203	**17.8**	828,573	3.9
	Invite from searching	255,341	1.6	364,696	1.7
	Accept invitation	135,032	0.84	617,217	**2.9**
Profile	Visit profiles	4,381,877	**27.3**	2,059,880	9.6

Figure 11 shows the number of sessions that Sybil and normal users initiate per day. Sixty-four percent of normal users access Renren no more than once per day, whereas only 8% Sybils fall in this low-frequency range. Sybils average 3.9 sessions per day versus 1.5 for normal users. However, the session duration of Sybils is much shorter than for normal users. Figure 12 shows the distribution of session durations for normal users and Sybils. The median session duration for normal users is 6 minutes, whereas the median for Sybils is 48 *seconds*. In contrast, <25% of normal sessions are 48 seconds long. A very small percentage of Sybils exhibit sessions that are hours long. These Sybils rate limit themselves by only sending friend requests and spam every few minutes and are hence active for a long period of time.

The observed session-level Sybil characteristics are driven by attacker's attempts to circumvent Renren's security features. Renren limits each account to sending 50 friend requests per day. Thus, in order to collect many friends and increase the reach of spam, attackers create many Sybils, quickly log in to each one and perform malicious activities (e.g., sending unsolicited friend requests and spam), and then log out and move on to the next account. Furthermore, Renren only allows accounts to browse 100 profiles per hour. As shown in Table V, Sybils spend a great deal of clicks browsing profiles. Hence, attackers maximize the number of profiles that they can browse by logging in to each Sybil once per hour and maxing out its allotment of profile views.

4.2. Clicks and Activities

Having characterized Sybils at the session level, we now analyze the type and frequency of Sybil clicks within each session. As shown in Table V, we organize clicks into logical *categories* that correspond to high-level OSN features. Within each category

are *activities* that correspond to particular Renren features. The primary categories on Renren are:

—**Photo:** The photo category is the most popular activity in Renren. This includes uploading photos, organizing albums, tagging friends, and browsing friends' photos.
—**Message:** The message category includes textual interactions between users, such as status updates, wall posts, comments, and real-time instant messages.
—**Share:** The share category refers to users posting hyperlinks on their wall. Common examples include links to videos and news stories on external Web sites or links to blog posts on Renren's internal blogging service.
—**Friending:** This category encapsulates friend relationship management. This includes sending friend requests, accepting or denying those requests, and unfriending. Renren provides users with several ways to locate new friends. There is a "guide" that lists popular, high-degree users, as well a recommendation feature that takes graph connectivity and profile similarity into account. Users can also use keyword search to locate friends, or they can send invitations directly from a user's profile page.
—**Profile:** This category includes browsing user profiles. Much like Facebook, all accounts on Renren can be browsed by anyone, but the amount of information that is displayed is restricted by the profile owner's privacy settings. Unlike Facebook, Renren profiles display a list of the nine most recent visitors [Jiang et al. 2010].

Table V displays the most popular activities on Renren. The number of clicks on each activity is shown, as well as the percentage of clicks. Percentages are calculated for Sybils and normal users separately—that is, each "%" column sums to 100%. For the sake of brevity, only activities with $\geq$1% of clicks for either Sybils or normal users are shown.

Table V reveals contrasting behaviors between Sybils and normal users. Unsurprisingly, normal users' clicks are heavily skewed toward photos (58.4%) and instant messaging (9.7%). Surprisingly, attackers are not currently abusing these features by sending spam instant messages or by posting advertisements embedded in images. It is unclear why attackers are not leveraging these channels yet, although their strategies may shift in the future.

Normal users and Sybils share content at similar rates (3.6% and 2.2%, respectively). This is an important observation, because (as we discuss in Section 5) sharing is the primary channel for spam dissemination on Renren. The similar rates of legitimate and illegitimate sharing indicate that spam detection systems cannot simply leverage numeric thresholds to detect spam sharing. Instead, we examine correlative methods to identify spam sharing in Section 5.

Sybils' clicks are heavily skewed toward friending (63.3% for Sybils, 12.3% for normal users). This facilitates the primary goal of Sybil accounts on Renren: to accumulate friends. Sybils send friend requests using Renren's guide and recommendation features eight times more frequently than normal users. As discussed in Section 3.4, this indicates that attackers are abusing Renren's features in order to locate popular users to target with friend requests. Although Sybils accept close to 100% of the friend requests that they receive (see Figure 3), in absolute terms, Sybils receive fewer friend requests than normal users. Hence, normal users click the "accept friend request" link more often than Sybils.

Strangely, Sybils on Renren spend 27.3% of their clicks browsing profiles, whereas normal users only spend 9.6%. The reason for this behavior is not clear. The vast majority of friend invitations from Sybils are sent using the guide and recommendation features, so Sybils are not crawling in order to locate targets for friending.

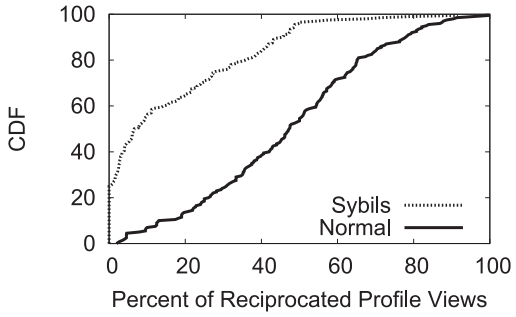
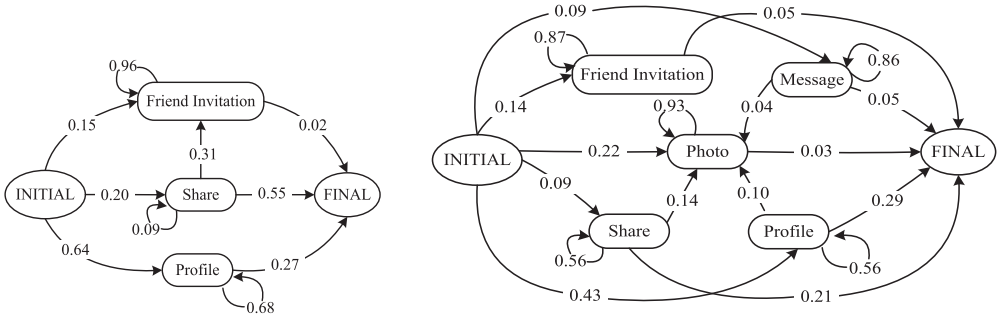


Fig. 13. Reciprocation of profile views.



(a) Sybil behavior model. (b) Normal behavior model.

Fig. 14. Categories and transition probabilities in the clickstream models of Sybils and normal users.

One possible explanation has to do with the recent visitor list that appears on each Renren profile. A Sybil can browse other user profiles in the hope that the user will see the visit in the log and reciprocate. The user will then see (and hopefully click) spam links in the Sybil account's profile. Figure 13 shows the percentage of profile visits that are reciprocated for a sample of 200 normal users and Sybils. Normal users exhibit a near linear relationship, with 50% of visits being reciprocated for 50% of users. Conversely, visits from Sybils are rarely reciprocated: the median reciprocation percentage for Sybils is just 7%. However, attackers are used to dealing with high attrition rates for spam [Kanich et al. 2008], so even these small numbers of reciprocated views may be worth it.

An alternative explanation for the profile browsing behavior of Sybils is to facilitate theft of personal information. Attackers can use information from personal profiles to launch targeted spear-phishing attacks. OSN profiles can also be commoditized by "cloning" them onto other Web sites [Wang et al. 2012]. Unfortunately, we cannot test conclusively for information theft by Sybils.

4.3. Clickstream Modeling

We now analyze the sequence of clicks from Sybils and normal users using a Markov chain model. In this model, each state is a category, and edges represent transitions between categories. We add two abstract states—initial and final—which we insert into the beginning and end of each click sequence. Figure 14 shows the category transition probabilities for both Sybils and normal users. The sum of all outgoing transitions from each category is 1.0. To reduce the complexity of the figure, edges with probability

<4% have been pruned (except for transitions to the final state). Categories with no incoming edges after this pruning process are also omitted.

Figure 14 demonstrates that Sybils follow a very regimented set of behaviors. After logging in, Sybils immediately begin one of three malicious activities: friend invitation spamming, spam sharing, or profile browsing. The profile browsing path represents crawling behavior: the Sybil repeatedly views user profiles until the daily allotment of views is exhausted. The share/friend invitation path represents spamming behavior. Renren's security systems place tight limits on the number of shares per account per day, hence the self-transition on the share category is very small. However, attackers do try and maximize the utility of each Sybil log in: 31% of spam shares are followed by friend spam generation.

Compared to Sybils, normal users engage in a wider range of activities, and the transitions between categories are much more diverse. The highest centrality category is photos, although it is not the most likely initial category. Intuitively, the most common path for normal users is to first browse to a friend's profile, then to start browsing their photos. Sharing and message generation are relatively low probability categories. As prior studies of interactions on OSNs have shown, many users generate new content less than once per day [Wilson et al. 2009]. The message category has a high probability self-transition due to the presence of real-time instant-message conversations.

4.4. Discussion

Our analysis shows that Sybils have very different clickstream characteristics compared to normal users. In this section, we briefly discuss how these differences can be leveraged to algorithmically distinguish normal users from Sybils. A more complete treatment of this subject (including implementation details) can be found in Wang et al. [2013].

At a high level, the goal is to use ground-truth clickstream data to train a classification algorithm, which can then be used to label users as normal or Sybil. We now discuss two potential classification algorithms that match these criteria.

The first is to use an SVM, a tool that we have already leveraged in our analysis in Section 2.3. Based on the results in this section, we could train an SVM on the following clickstream features: (1) session-level features, including average clicks per session, average session length, average interarrival time between clicks, and average sessions per day, and (2) features from click activities. As mentioned in Section 4.2, there are five categories of clicking activities on Renren. The percentage of clicks in each category could be used as features for an SVM, as well as the transition probabilities between categories.

A second possible classification method is to use maximum-likelihood estimation (MLE). Given the click sequence of a user, we can use MLE to examine which *clickstream* model (described in Figure 14) better explains the observed sequence. If the sequence is more likely to be explained by the Sybil clickstream model, the corresponding user would be classified as a Sybil or otherwise as a normal user. In our case, estimating the likelihood of a model M involves considering each transition $\{s_i, s_{i+1}\}$ during the user click sequence $\{s_1, s_2, \ldots, s_n\}$ and computing the likelihood $P_M(s_i, s_{i+1})$ that the user transits from category s_i to category s_{i+1} according to the model M. Thus, the likelihood P_M that model M reproduces the sequence is given by the product of the individual likelihoods according to model M.

Unfortunately, the limitation of these classification algorithms (SVM and MLE) is that they require training on large samples of labeled ground-truth data. For a practical Sybil detection system, it would be better to develop clickstream analysis techniques that leverage unsupervised learning on real-time data samples—that is, require zero or little ground truth. We leave the development of these techniques to future work.

Table VI. Average Actions per Sybil on Renren After June 1, 2009

Sybil Action	Avg. per Sybil
Write a blog post	0.11
Update status	2.13
Upload photo	1.81
Post to a friend's wall	0.73
Share a link	6.88

5. SPAM STRATEGIES AND COLLUSION

In Section 3, we revealed that the vast majority of Sybils on Renren do not form explicit friend connections with each other. However, this does not imply that each Sybil is independently controlled by a different attacker. In this section, we uncover collusion between Sybils by leveraging spam content similarity and temporal correlation to locate *inferred* relationships.

We begin by presenting an overview of the strategies used by Sybils to disseminate spam on Renren. The most common strategy is "sharing" links to spam blogs, and we present a detailed case study on this phenomenon. The results of the case study indicate that Sybils share links to the same content at about the same time. These observations motivate us to cluster Sybils based on content similarity and temporal correlation. Our results reveal that even under strict correlation thresholds, Sybils form large connected components. This indicates that Sybils are being controlled collectively by colluding attackers behind the scenes.

5.1. Share Spam on Renren

We begin our analysis by characterizing the overall spamming behavior of Sybils on Renren. Prior studies of OSN spam have identified different spam dissemination strategies used by attackers. On Facebook, attackers post wall messages laden with spam links [Gao et al. 2010]. On Twitter, attackers attempt to hijack "trending" topics [Stringhini et al. 2010] or maliciously create new ones [Grier et al. 2010].

On Renren, the dominant method for Sybils to disseminate spam is to "share" links to spam content. As shown in Table VI, shares per Sybil far outnumber status updates and wall posts. Sharing is an advantageous strategy because the attacker only needs to generate one unique piece of spam—that is, the original piece of content. Sybils simply forward links to this content to all of their friends. Users on all OSNs engage in link sharing (e.g., "liking" on Facebook), so this behavior is not overtly suspicious. However, as we show later, there are quantifiable differences between Sybil and normal sharing.

Out of our complete dataset of 660K Sybils, about 64% have not shared any content due to being banned by Renren before beginning to spam. Hence, in this section, we focus on the remaining 237,205 Sybils that have shared content. In total, these Sybils shared 3,491,988 links. Figure 15 shows the number of shares per Sybils. Here, 25% of Sybils only share a single piece of content before they are caught and banned, and $<$1% of Sybils go uncaught long enough to share 100 or more links.

To investigate the purpose of spam on Renren, we manually examined the shares of a random sample of 1,000 Sybils. We found that Sybils on Renren only share two types of links:

(1) **Blogs:** Of shares from Sybils, 62.5% link to spam blog posts. These blogs engage in typical spam activities, such as promoting shady online merchants and selling pharmaceuticals. Many of these blogs attempt to obfuscate themselves by copying content from popular, legitimate blogs and then slightly modifying the content to include spam text and links.

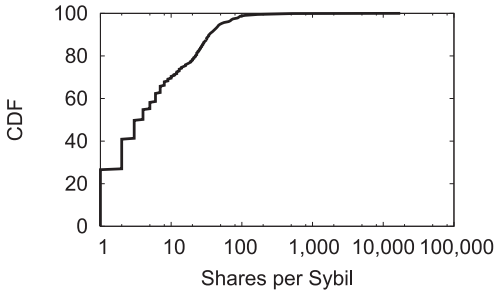


Fig. 15. CDF of shares per Sybil.

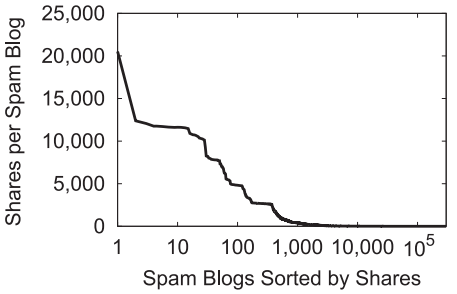


Fig. 16. Distribution of shares across spam blogs.

(2) **Videos:** The remaining Sybil shares link to bogus online videos. These videos use provocative titles and thumbnail images to entice users to click on them. Users are then redirected to sites that include spam content.

5.2. Case Study: Spam Blogs

We now turn our attention to spam blogs and the Sybils that promote them. We focus on shares of spam blogs because they are the most prolific type of spam on Renren.

Classifying Spam Blogs. Before we begin analyzing our data, we discuss our methodology for verifying that blogs shared by Sybils are truly spam. In the previous section, we manually verified that a subset of blogs shared by Sybils are spam—that is, they include links to phishing sites, Web sites selling contraband goods, sites that attempt drive-by-download exploits, and so forth. Furthermore, the security team at Renren confirmed that the majority of blogs shared by Sybils in our dataset were also banned by Renren's security systems. Reasons for banning these blogs included complaints from users, blogs containing spam keywords, and blogs containing blacklisted hyperlinks. Thus, in the remainder of this section, we conservatively assume that all blogs shared by Sybils are spam blogs.

Identifying Collusion. We start by addressing the fundamental question: *are Sybils colluding to promote spam blogs, or is each Sybil operating independently?* To answer this question, we calculated the amount of duplication among the spam blogs promoted by Sybils. Among the 3 million individual spam shares in our dataset, only 302,333 unique spam blogs are promoted. Figure 16 shows the number of shares for each spam blog, sorted from most to least promoted. The top 30 spam blogs are highly promoted, each garnering more than 10,000 shares; 25% of spam blogs receive 2 or more shares from Sybils, which is the bare minimum evidence for collusion. However, this result is a lower bound, since we only compared the precise links shared by Sybils on Renren. It is possible that these links eventually redirect to the same external Web sites, which would indicate additional levels of collusion among attackers.

Information Dissemination. One reason that Sybils on Renren collude to promote spam blogs has to do with the trending content section on the Renren homepage. Each day, the 100 most popular blog posts, images, and videos on Renren get featured on the site's homepage, which receives millions of hits per day. Attackers can use Sybils to inflate the popularity of spam blogs and try to make them artificially trend. Other researchers have observed similar attacks against Digg [Tran et al. 2009] and Twitter [Grier et al. 2010]. Currently, Renren relies on manual inspection by humans to filter spam out of the trending content section.

This raises our next question: *does Sybil collusion create quantifiable differences in the dissemination pattern of spam content?* Intuitively, content on OSNs should

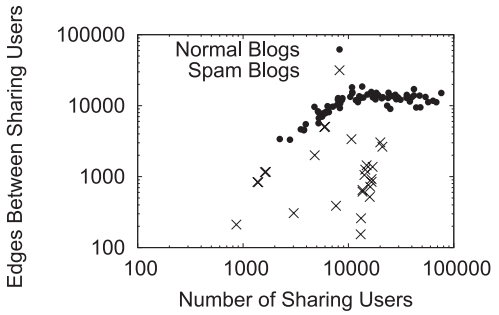


Fig. 17. Connectivity between users sharing normal and spam blogs.

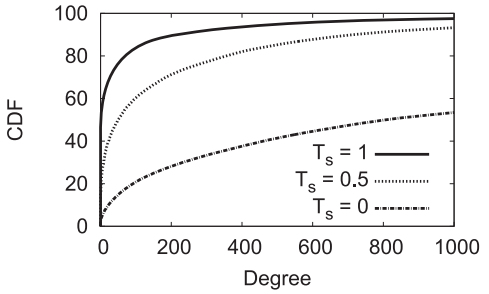


Fig. 18. The degree distribution for content similarity graphs with varying thresholds.

spread organically along social links. A user posts a link, then friends and friends of friends share, "like," and "retweet" it to an ever-expanding audience. Researchers have identified information dissemination trees matching this pattern on many OSNs [Kwak et al. 2010]. However, as we showed in Section 3, Sybils are not friends with each other. Thus, each Sybil that shares a spam link creates an independent information dissemination tree.

To test whether this intuition is true, we took a snapshot of the 100 most popular blogs on Renren on February 2, 2011. We manually inspected each blog and determined that 26 blogs were spam—that is, they included overt spam content and/or links to malicious Web sites. Figure 17 shows the number of edges that connect users who share links to the blogs in our sample. There is a clear distinction between normal blogs and spam blogs. Legitimate blogs exhibit an order of magnitude more shared edges, indicating that their popularity has grown organically along friendship links.

In contrast, spam blogs promoted by Sybils have far fewer shared edges. This result makes sense, given that we have shown that Sybils have few edges connecting to each other. Normal users are unlikely to reshare links to spam blogs, so their popularity growth is almost entirely driven by disconnected Sybils. Only two spam blogs from our sample approach the cluster of normal blogs in Figure 17.

5.3. Content-Based Sybil Components

At this point we have demonstrated that Sybils on Renren collude to disseminate spam content, despite having few social links between them. We now examine whether content similarity can be used to group Sybils into connected components. Intuitively, strongly connected components are likely to be under the control of a single attacker. Understanding these components allows us to estimate the number of attackers that are behind threats to Renren, as well as the relative strength of each attacker (as measured by the number of Sybils that they control).

We model collusion between Sybils as a *content similarity graph*. In a content similarity graph, Sybils are nodes and two Sybils are connected if they share similar content. We consider the following similarity criteria: let s_i and s_j be the sets of content that two Sybils i, j share. We define the *content similarity* between them as $s_{ij} = s_i \cap s_j / s_i \cup s_j$. Content similarity can range from 0 to 1, where 0 means that there is no duplication of content and 1 means the two Sybils share exactly same content. We say that two Sybils i, j share similar content if s_{ij} is larger than some threshold T_s (or equal to T_s in the special case of $T_s = 1$).

To understand collusion between Sybils, we built content similarity graphs using three values for T_s: 0, 0.5, and 1. $T_s = 0$ is the most lax threshold: pairs of Sybils sharing

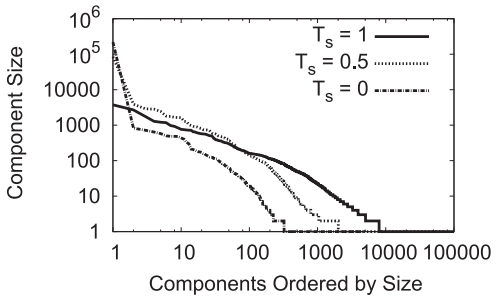


Fig. 19. Sybil-connected component sizes under different thresholds.

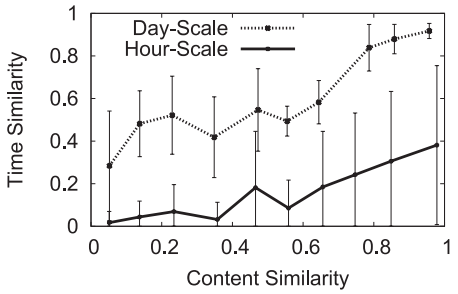


Fig. 20. Time similarity versus content similarity.

1 or more identical pieces of content will be connected. This is the same similarity threshold used in prior studies of OSN spam [Gao et al. 2010; Grier et al. 2010]. $T_s = 1$ is the strictest threshold: all content shared by two Sybils must be identical in order for them to be connected.

Figure 18 shows the degree distribution of our three content similarity graphs. Compared to the original social graph (see Figure 5), Sybils have many more edges connecting to each other in the content similarity graphs. Even under the tightest threshold $T_s = 1$, >50% of Sybils have at least one Sybil partner forwarding exactly the same content. These Sybils pairs are obviously under the control of a single attacker.

In the case of moderate and low thresholds, the number of edges significantly increases. Under the most permissive threshold $T_s = 0$, 97% of Sybils have edges to other Sybils. Large connected components that emerge under more permissive thresholds may represent Sybils that are "rented" by spammers. Studies have shown that black-market crowd-sourcing services allow spammers to cheaply hire workers to spam on OSNs for them [Wang et al. 2012; Motoyama et al. 2011]. These workers (and the Sybils they control) end up being involved in multiple spam campaigns, which helps explain why different groups of Sybils exhibit varying amounts of overlapping content.

Next, we study the component structure of the content similarity graph. Figure 19 shows the quantity and sizes of connected components under different thresholds, ordered from largest to smallest. As expected, tighter thresholds fragment the graph into a larger number of distinct components and reduce the size of the largest (giant) component. Under the lax $T_s = 0$ threshold, the 237K Sybils form 4.9K connected components. The giant component contains >219K (90%) Sybils. In contrast, there are 76K connected components under the moderate $T_s = 0.5$ threshold. The giant component only contains 84K (35%) Sybils, but the next 100 largest components contain an additional 60% of the Sybils. Finally, under the tight $T_s = 1$ threshold, there are 114K components, with the largest only containing 3,700 Sybils. In this scenario, the first 1,000 components only contain 40% of the overall Sybils.

The results in Figure 19 confirm our intuition that content similarity can be used to identify groups of colluding Sybils. Tight thresholds pinpoint Sybils that were created and managed by a single attacker. The maximum size of these components demonstrates the upper bound on the number of Sybil accounts that a single attacker can reasonably generate. More relaxed thresholds identify Sybils that belong to several overlapping spam campaigns, possibly under the control of different attackers at different times. These components can grow to huge sizes, as attrition forces attackers to gradually replace old Sybil accounts with new ones.

5.4. Temporal Correlation Between Sybils

We now investigate whether there are temporal correlations between Sybils that exhibit content similarity. We suspect that Sybils under the control of a single attacker will be active at similar times engaging in coordinated spam campaigns.

Let t_i and t_j be the sets of links that two Sybils i, j share during time interval S. We define the *temporal similarity* between the Sybils as $t_{ij} = t_i \cap t_j / t_i \cup t_j$. Like content similarity, temporal similarity can range from 0 to 1, with 0 meaning no overlap and 1 meaning exact overlap. The size of time interval S can be varied to control the granularity of comparisons. In our experiments, we evaluate time similarity over two time intervals: 1 hour and 1 day.

To test our hypothesis, Figure 20 shows content similarity versus time similarity for Sybils in our dataset under two time intervals. Each line plots the average time similarity for discreet sets of Sybil pairs with close content similarity. For example, the first point of the hour-scale line represents the average time similarity for all pairs of Sybils with content similarity in the range 0 to 0.1. The error bars show the standard deviation for each data point.

Figure 20 reveals that time similarity is roughly proportional to content similarity. As expected, Sybils that share similar content tend to do so at similar times, reinforcing our conclusion that Sybils are colluding. Setting the time interval to 1 day significantly increases the amount of collusion that we can identify. Sybils that share near-identical content (content similarity $\approx$0.95) also exhibit nearly 0.92 time similarity under the 1-day threshold. This shows that attackers move rapidly to complete social spam campaigns using many Sybils within short time spans.

6. MOVING SYBIL DEFENSE FORWARD

In Section 2.3, we present an accurate, scalable system that has been very effective at catching Sybils on the Renren social network. However, security is a constant cat and mouse game between attackers and defenders: now that Renren has deployed countermeasures against Sybils, attackers may try to adapt in order to circumvent our system.

In this section, we investigate the limits of our detection system. First, we propose a new, hypothetical strategy that attackers could use to obfuscate their Sybils from our detector. This new strategy is born from necessity: in order to avoid our deployed detector, Sybils must collude to inflate key metrics that are leveraged by our detector. We describe the new attack model in detail and formally define its parameters.

Next, we conduct simulated attacks against a real social graph: the Peking University (PKU) regional network of Renren. We observe that under the new attack model, colluding Sybils inevitably form tight-knit communities. We leverage this fact to strengthen our detector against the new attack model. Results from simulated attacks demonstrate that the improved detector is highly accurate at detecting Sybils operating under the new attack model. We conclude with a summary of results and a discussion of deployment scenarios.

6.1. New Attack Model

Our existing Sybil detector relies on four key metrics to identify Sybils:

(1) Invitation frequency: how frequently are friend requests sent by the user?
(2) Outgoing requests accepted: what percentage of friend requests sent by the user are accepted?
(3) Incoming requests accepted: what percentage of friend requests received by the user are accepted?
(4) Clustering coefficient: what is the user's clustering coefficient?

Of these four features, the two most important are *outgoing requests accepted* and *clustering coefficient*. *Invitation frequency* and *incoming requests accepted* may be manipulated by a dedicated attacker—that is, by slowing down the rate at which friend requests are sent and by randomly denying some fraction of incoming friend requests. However, the outgoing acceptance rate and clustering coefficient are more difficult to manipulate, since they depend on the behavior of the users who receive friend requests from the Sybil.

The only way for a Sybil to influence its outgoing acceptance rate and clustering coefficient is for it to send friend requests to other colluding Sybils. These Sybils are guaranteed to accept the requests, thus inflating the outgoing acceptance rate of the sender. Furthermore, if several colluding Sybils follow the same strategy, they will naturally form a well-connected cluster, and thus their clustering coefficient scores will go up.

These observations form the basis of our new, hypothetical attack model. *In order to avoid our deployed detector, Sybils must collude in order to inflate their outgoing request acceptance rate and their clustering coefficient*. Intuitively, this causes Sybils to form a community that links together the colluding Sybils. Although we do not observe such Sybil communities in our measured data (see Section 3.2), the existence of our detector forces attackers to adapt. Under this new regime, Sybils are forced to form communities in order to avoid detection.

Formal Model. We now formally define the parameters of the new attack model. Consider an attacker that controls N total Sybils. In order to avoid detection, each Sybil must maintain a friend request acceptance percentage of at least β. Let α be the probability that normal users accept friend requests from Sybils. In order to keep a high accept percentage, let each Sybil send friend requests to other Sybils with probability p and to normal users with probability $1 - p$. Thus, in order to avoid detection, each Sybil must send friend requests that obey the following inequality: $\beta \leq p + \alpha(1 - p)$. If each Sybil sends n total friend requests, then each Sybil will create $n * p$ Sybil edges and $n * \alpha(1 - p)$ attack edges.

6.2. Simulated Attacks and New Defense Strategies

We investigate the graph structure of Sybils that follow the new attack model by simulating attacks against a real social graph. We use the PKU regional network on Renren as our target social graph, since its size is reasonable (170,000 nodes) and its properties have been studied by prior work [Jiang et al. 2010]. In our simulation, we create N Sybils, each of which sends n friend requests divided between normal users and Sybils according to the inequality $\beta \leq p + \alpha(1 - p)$. We fix $\alpha = 0.26$, which is the median acceptance percentage for Sybil friend requests in our dataset (see Figure 2). Similarly, we set $\beta = 0.5$, which is the detection threshold used by our deployed detector.

We experiment with many combinations of attack parameters in order to observe their effects on Sybil graph structure. As shown in Table VII, we test small and large groups of Sybils (column N). Given $\alpha = 0.26$ and $\beta = 0.5$, p can be calculated using the inequality $\beta \leq p + \alpha(1 - p)$ ($p = 0.33$ in our simulations). For each value of N, we vary n, the number of friend requests sent per Sybil. Table VII shows the number of Sybil edges and attack edges per Sybil as n varies.

In order to avoid creating Sybil clusters that obviously deviate from normal graph structure, the attacker can target Sybil edges in such a way as to create "natural"-looking clusters. In our simulations, we use two common models to direct the creation of Sybil edges: (1) Erdős-Rényi, where the attacker links randomly chosen Sybils, and (2) Preferential Attachment, where the destination of each Sybil edge is chosen proportionally to the destination Sybil's degree. Normal users are selected uniformly at random as targets for attack edges.

Table VII. Simulated Sybil Attacks Against the PKU Graph

N	n	Edges per Sybil		Erdős-Rényi		Pref. Attach.	
		Sybil	Attack	Prec.	Rec.	Prec.	Rec.
1K	100	33	17	99.4%	1	99.8%	1
	300	99	52	98.3%	1	97.8%	1
	500	165	87	89.4%	1	91.3%	1
	1,000	330	174	79.6%	1	79.5%	1
	2,000	660	348	64.3%	1	63.4%	1
10K	100	33	17	98.9%	1	98.5%	1
	1,000	330	174	82.9%	1	82.7%	1
	2,000	660	348	72.9%	1	72.5%	1

Evaluating Existing Sybil Community Detectors. Although multiple Sybil detection algorithms have been proposed, such as SybilGuard [Yu et al. 2006], SybilLimit [Yu et al. 2008], SybilInfer [Danezis and Mittal 2009], and SumUp [Tran et al. 2009], prior work demonstrates that these Sybil detectors generalize to community detection [Viswanath et al. 2010]. In essence, these detectors assume that Sybils form tight-knit groups that can be detected by looking for small quotient cuts in the social graph.

Therefore, we now seek to answer the following question: *do Sybils form strong communities when attackers follow the new attack model?* If Sybils do form tight-knit communities under the new attack model, then this leads to a second question: *what is the accuracy of Sybil community detectors under these conditions?* If their accuracy is high, then we could use them to complement our existing, feature-based Sybil detector.

To answer these questions, we use the community detection algorithm developed by Blondel et al. [2008] to locate communities in the graphs listed in Table V. For each simulated attack, we locate the community with the largest number of Sybils and evaluate its *precision* (the ratio of Sybils to total number of nodes within the community) and *recall* (the ratio of Sybils within the community to total number of Sybils). Intuitively, these metrics tell us the false-positive (precision) and false-negative (recall) rates of the community detector when we use it to isolate Sybils.

The results shown in Table VII are mixed. On one hand, when $n \leq 300$, the community detector is able to identify Sybils with high precision. This proves that Sybils do form tight-knit communities when they follow the new attack model. However, as n grows, so does the false-positive rate. Thus, although recall is consistently 1 (i.e., all Sybils are caught), many normal users are mistakenly grouped into the Sybil community as well.

The results in Table VII indicate that by themselves, Sybil community detectors are not practical for locating Sybils that follow the new attack model. A stealthy attacker could create many Sybils, add Sybil and attack edges over a period of many days until n is large, and then begin using the Sybils to send spam. Thus, by the time the Sybils become actively malicious, a Sybil community detector will generate many false positives when attempting to isolate the Sybil community.

External Acceptance Percentage. In order for Sybil community detectors to be practical (i.e., not generate false positives), they must be able to identify Sybil communities early on in their existence. Unfortunately, graph structure alone is not enough to disambiguate malicious and benign communities. Thus, in order to identify Sybil communities early in their existence, we must leverage additional features beyond the graph topology.

Toward this end, we introduce a new feature: the *external acceptance percentage*. The external acceptance percentage is the fraction of friend requests sent by members of a community to users outside the community that are accepted. Intuitively, this feature

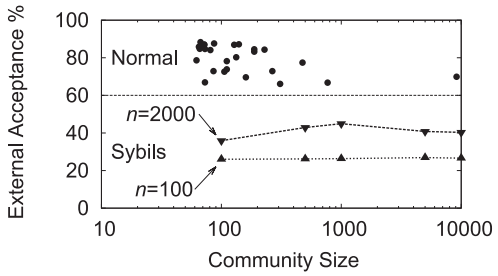


Fig. 21. The external acceptance percentage of normal and Sybil communities.

is useful, because for Sybils the vast majority of accepted friend requests are from other Sybils *inside* the local community. Conversely, rejections are from normal users *outside* the local community. Thus, Sybil communities will have a low external acceptance percentage, which can be used to quickly separate them from benign communities.

To confirm our intuition, we plot Figure 21, which shows the external acceptance percentage for benign and Sybil communities from the PKU network during our simulated attacks. There is a clear distinction between normal communities and Sybil communities, which holds even as attackers vary N and n. The Sybils in Figure 21 follow the Erdős-Rényi attachment model, but the results are the same for the Preferential Attachment model.

Figure 21 demonstrates that the external acceptance ratio is an effective feature for distinguishing between Sybil and benign communities. *Without the external acceptance ratio, Sybil community detectors generate too many false positives to be practically deployed.* However, by coupling latent information about rejected friend requests with graph structural information, the accuracy of Sybil community detectors can be dramatically improved.

6.3. Summary and Discussion

In this section, we have investigated the limits of our deployed, feature-based Sybil detector. We observe that in order for Sybils to avoid our detector, they must form tight-knit communities. Although Sybil community detectors are able to identify these clusters, they incur a large number of false positives in the process. We demonstrate that by incorporating additional information (the external acceptance percentage), Sybil and benign communities can be disambiguated, thus mitigating the inaccuracy of community detection.

These results lay out a road map that can be used by OSN providers to build comprehensive Sybil detection systems. In particular, comprehensive Sybil detection is composed of two complementary parts: the feature-based detector presented in Section 2.3, plus the enhanced community detection approach motivated in this section. The former technology catches Sybils that attempt to completely blend into the social graph—that is, that do not form edges to other Sybils. The latter technique (community detection plus external acceptance ratio) identifies Sybils that collude (and thus form communities).

In practice, the OSN operator would need to run offline community detection on the social graph once per day. Any community with an external acceptance percentage below a certain threshold (e.g., 0.6, as seen in Figure 21) would be flagged as suspicious. By running community detection every day, the OSN operator would catch Sybil communities while the number of edges per Sybil is low. Most OSNs limit the number of friend requests that each account can send per day (on Renren, the limit is 50). This slows the growth of n in Sybil communities and gives the OSN operator a window

of several days to catch the Sybil community before the precision of the community detector falls below 99% (see Table VII).

7. RELATED WORK

OSN Spam. Recent studies have characterized the growing OSN spam problem on Facebook [Gao et al. 2010] and Twitter [Grier et al. 2010]. These studies rely on offline heuristics to identify spam content in status updates/tweets, as well as aberrant behavior that is indicative of spamming. More recent work relies on Twitter's in-house spam detection systems to identify a ground-truth set of spamming accounts [Thomas et al. 2011]. These studies locate millions of spam messages on each OSN and use them to analyze large-scale, coordinated spam campaigns. In Section 5, we also identify large scale, coordinated spam campaigns on Renren that are promulgated by thousands of colluding Sybils. Similar to prior work, we leverage content and temporal correlation techniques to quantify the extent of Sybil collusion on Renren.

OSN Spam Detection. Various techniques borrowed from email spam detection have been applied to OSN spam. Two studies leverage honeypot accounts on MySpace and Twitter, respectively, to trap spammers who attempt to friend them [Webb et al. 2008; Lee et al. 2011]. Our results indicate that unless social honeypots are engineered to appear popular, they will only be targeted by a small subset of spammers.

Other studies use Bayesian filters and SVMs to identify spammers on Twitter [Yardi et al. 2010; Benevenuto et al. 2010; Wang 2010] and Facebook [Stringhini et al. 2010]. These techniques work well on Twitter, since following and followed information is public. However, detection on Facebook and Renren is less successful, since only existing friendships are publicly viewable, whereas invitation frequency is hidden. Our Sybil detector overcomes these issues by leveraging friend invitation information that is only accessible from within Renren. Moreover, we demonstrate in Section 6 that although Sybil behavior may evolve in the future, it is possible to detect these malicious accounts by combining network structural features with friend request statistics.

Internally, Facebook uses an elaborate security system called Facebook Immune System (FIS), which leverages several types of machine learning algorithms [Stein et al. 2011]. Although this article provides information about the high-level design of FIS, it does not reveal the specific features leverages by the machine learning algorithms, the accuracy of the detector, or the performance characteristics of the system. Thus, it is impossible to objectively evaluate the differences between FIS and our proposed detection system.

8. CONCLUSION AND DISCUSSION

In this article, we make several contributions to the area of Sybil detection on OSNs. First, we use ground-truth data about the behavior of Sybils in the wild to create a measurement-based, real-time Sybil detector. We show that a computationally efficient, threshold-based classifier is sufficient to catch 99% of Sybils, with low false-positive and false-negative rates. We have deployed our detector on Renren's production systems, and in the first 6 months of operation, it identified and banned more than 100,000 Sybil accounts.

Our second contribution is a characterization of Sybil graph topology on a major OSN. This characterization was the first of its kind when the conference paper was originally published [Yang et al. 2011]. Using edge creation information for more than 660,000 Sybil accounts on Renren, we show that Sybils on Renren do not obey behavioral assumptions that underlie previous work on decentralized Sybil detectors. Eighty percent of Sybils do not connect socially to other Sybils but instead focus on building

friendships with normal users. Even in rare cases where Sybils do form connected components, these clusters are loose rather than tightly knit. Temporal analysis indicates that these Sybil edges are formed accidentally by attackers rather than intentionally.

The third contribution is analysis of Sybil clickstream behavior on Renren. Our data captures the exact session-level sequences of actions that Sybils use to send spam and generate friend requests. The results show that Sybil accounts do not engage in time-consuming behaviors like photo browsing, but they do spend significant amounts of time crawling the OSN. We also find that Sybils abuse the friend recommendation features of OSNs in order to locate targets for friending. We provide suggestions for how these features can be leveraged to further improve the accuracy of behavior-based Sybil detection systems.

The final contribution of this article is a deeper understanding of collusion between Sybils. Based on 3,491,988 pieces of spam content shared by 237,000 Sybils, we show that social links between Sybils are inadequate for identifying colluding behavior. Sybils with no explicit social ties still act in concert to spread spam. For example, 50% of Sybils have at least one partner sharing exactly the same spam content. We capture these inferred relationships between Sybils using content similarity graphs and show that colluding Sybils exhibit high levels of temporal correlation.

Generality of Our Results. We view the results of our work as a case study on the behavior of Sybils in the real world. Prior to our work, there were no results in the literature that reported the behavior of social Sybils based on ground-truth data. However, given that all of our results come from Renren, this raises the following question: *do our results generalize to other social networks?*

Answering this question is challenging for two reasons. First, there are extremely few ground-truth datasets about social Sybils. This prevents us from evaluating Sybil behavior, or our techniques, in other contexts. Thomas et al. [2011] collected a subset of ground-truth Sybils from Twitter, but their dataset is (1) not complete and (2) not publicly available. No large-scale, ground-truth datasets of Sybils are available for Facebook, LinkedIn, Sina Weibo, or any other well-known social networking Web sites.

The second challenge is that each social network offers different features and capabilities to users, which means that the attack surface of each Web site is different. In other words, the behavior of Sybils on a given OSN is intimately related to the functionality offered by that OSN. We hypothesize that Sybils on Facebook may act similarly to Sybils on Renren, since these two OSNs offer almost identical features. Conversely, Sybils on Twitter and Sina Weibo may behave differently because they exploit features unique to microblogs. For example, Sybils on Twitter can use hashtags to spread spam, which obviates the need to send friend requests [Thomas et al. 2011].

Although we cannot be sure that our results generalize to all OSNs, our findings for a traditional—that is, untrusted—OSN, coupled with results from prior work on trusted OSNs [Mohaisen et al. 2010], suggest that we should explore new approaches to perform decentralized detection of Sybil accounts on OSNs.

Future and Ongoing Work. Several of the findings from our measurement study (in particular, the results in Section 5) indicate that there are additional features that can be used to detect social Sybils. To leverage these additional features, we have developed an enhanced Sybil detection system that uses a clickstream analysis abstraction. This abstraction is a generalization of the approach used in this article and can therefore incorporate additional behavioral features. The design and evaluation of this system can be found in Wang et al. [2013]. This system is currently being evaluated by Renren and LinkedIn, and preliminary results (as reported by security teams at both networks) have been quite promising.

REFERENCES

Kevin Bauer, Damon McCoy, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. 2007. Low-resource routing attacks against Tor. In *Proc. of Workshop on Privacy in Electronic Society*.

Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. 2010. Detecting spammers on Twitter. In *Proc. of CEAS*.

Fabricio Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgilio Almeida. 2009. Characterizing user behavior in online social networks. In *Proc. of IMC*.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 10.

Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In *Proc. of NSDI*.

George Danezis and Prateek Mittal. 2009. SybilInfer: Detecting Sybil nodes using social networks. In *Proc of NDSS*.

John R. Douceur. 2002. The Sybil attack. In *Proc. of IPTPS*.

H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. 2010. Detecting and characterizing social spam campaigns. In *Proc. of IMC*.

C. Grier, K. Thomas, V. Paxson, and M. Zhang. 2010. @spam: The underground on 140 characters or less. In *Proc. of CCS*.

J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, and B. Y. Zhao. 2010. Understanding latent interactions in online social networks. In *Proc. of IMC*.

C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. 2008. Spamalytics: An empirical analysis of spam marketing conversion. In *Proc. of CCS*.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue B. Moon. 2010. What is Twitter, a social network or a news media? In *Proc. of WWW*.

Kyumin Lee, Brian David Eoff, and James Caverlee. 2011. Seven months with the devils: A long-term study of content polluters on Twitter. In *Proc. of ICWSM*.

Amanda Lenhart, Kristen Purcell, Aaron Smith, and Kathryn Zickuhr. 2010. Social media and young adults. Pew Research Center.

Qiao Lian, Zheng Zhang, Mao Yang, Ben Y. Zhao, Yafei Dai, and Xiaoming Li. 2007. An empirical study of collusion behavior in the Maze P2P file-sharing system. In *Proc. of ICDCS*.

Abedelaziz Mohaisen, Aaram Yun, and Yongdae Kim. 2010. Measuring the mixing time of social graphs. In *Proc. of IMC*.

M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker. 2011. Dirty jobs: The role of freelance labor in Web service abuse. In *Proc. of Usenix Security*.

Samantha Murphy. 2010. Teens ditch e-mail for texting and Facebook. MSNBC.com.

Atif Nazir, Saqib Raza, Chen-Nee Chuah, and Burkhard Schipper. 2010. Ghostbusting Facebook: Detecting and characterizing phantom profiles in online social gaming applications. In *Proc. of WOSN*.

James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. 2004. The Sybil attack in sensor networks: Analysis and defenses. In *Proc. of IPSN*.

Sophos. 2007. Sophos Facebook ID probe shows 41% of users happy to reveal all to potential identity thieves.

Tao Stein, Erdong Chen, and Karan Mangla. 2011. Facebook immune system. In *Proc. of EuroSys Social Network Systems (SNS)*.

Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2010. Detecting spammers on social networks. In *Proc. of ACSAC*.

K. Thomas, C. Grier, V. Paxson, and D. Song. 2011. Suspended accounts in retrospect: An analysis of Twitter spam. In *Proc. of IMC*.

Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. 2009. Sybil-resilient online content voting. In *Proc. of NSDI*.

B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. 2010. An analysis of social network-based Sybil defenses. In *Proc. of SIGCOMM*.

Alex Hai Wang. 2010. Don't follow me: Spam detection on Twitter. In *Proc. of SECRYPT*.

Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Heather Zheng, and Ben Zhao. 2013. You are how you click: Clickstream analysis for Sybil detection. In *Proc. of Usenix Security*.

G. Wang, C. Wilson, X. Zhao, Y. Zhu, M. Mohanlal, H. Zheng, and B. Y. Zhao. 2012. Serf and turf: Crowdturfing for fun and profit. In *Proc. of WWW*.

Steve Webb, James Caverlee, and Calton Pu. 2008. Social honeypots: Making friends with a spammer near you. In *Proc. of CEAS*.

Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P. N. Puttaswamy, and Ben Y. Zhao. 2009. User interactions in social networks and their implications. In *Proc. of EuroSys*.

Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y. Zhao, and Yafei Dai. 2011. Uncovering social network Sybils in the wild. In *Proc. of IMC*.

Sarita Yardi, Daniel Romero, Grant Schoenebeck, and Danah Boyd. 2010. Detecting spam in a Twitter network. *First Monday* 15, 1.

Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. 2008. SybilLimit: A near-optimal social network defense against Sybil attacks. In *Proc. of IEEE S&P*.

Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. 2006. SybilGuard: Defending against Sybil attacks via social networks. In *Proc. of SIGCOMM*.

Received September 2012; revised September 2012; accepted September 2012