32ND INTERNATIONAL CONFERENCE ON DISTRIBUTING COMPUTING SYSTEMS WORKSHOP YEAR 2012

Detecting and Validating Sybil Groups in the Wild

Jing Jiang, Zifei Shan, Wenpeng Sha, Xiao Wang, Yafei Dai
Department of Computer Science and Technology
Peking University, China
{*jiangjing, swp, wangxiao*}*@net.pku.edu.cn,* {*shanzifei, dyf*}*@pku.edu.cn*

***Abstract*—Sybil attacks are one of the well-known and powerful attacks against online social networks. Sybil users propagate spam or unfairly increase the influence of target users. Previous works focus on detecting sybil users. However, sybil users alone do not harm the system. What is really dangerous is that multiple sybil users collude together and form a sybil group. In this paper, we present the first attempt to identify and validate sybil groups in Renren online social network. We build sybil group detector based on multiple attributes. We apply the sybil group detector to Renren, and identify 2,653 sybil groups and 989,764 sybil users. We design automatic validation mechanisms of sybil groups, by analyzing action time similarity of users in a group. Overall, 2440 (91.9%) sybil groups and 985,797 (99.6%) sybil users are successfully validated. Our sybil group detection and validation mechanisms have important implications for system design to defend against sybil attacks in online social networks.**

***Keywords*-Sybil Groups; Detection; Validation; Online social networks;**

I. Introduction

In recent years, online social networks (OSNs) become huge and they are still growing throughout the world. Unfortunately, the relative openness and the tremendous growth of OSNs attracts the interest of malicious parties.

Sybil attacks are one of the well-known and powerful attacks against OSNs. The malicious attacker generates a sybil group, and pretend to be multiple, distinct users (called sybil users). Sybil attacks have several harms in OSNs. First of all, multiple sybil users are utilized to unfairly increase influence and power of target users [1], [2]. For example, commercial services use many sybil users and promote clients' content to the top position in Youtube[1]. In Twitter, some political campaigns disguise themselves as spontaneous grassroots behavior that are actually carried out by a single organization [2]. In Facebook, gamers control sybil users to achieve higher status in social games [1]. Secondly, spammers target OSNs as media to propagate spam [3], [4], [5], [6], [7]. In Facebook, compromised accounts send malicious wall posts with embedded URLs [3]. Malicious users rely on unsolicited mentions or embedding hashtags to send spam content in Twitter [7]. Sybil attacks become increasingly dangerous as more people use OSNs as primary interfaces to the Internet [8].

Successfully defending against sybil attacks is important to ensure fairness and credibility in the system, to reduce user burden of dealing with spam, and to positively impact the overall value of OSNs going forward. Previous works utilize friend relationships to detect sybil users, including SybilGuard [9], SybilLimit [10], SybilInfer [11], SumUp [12] and the Sybil detector [13].

Initial studies [9], [10], [11], [12], [13] focus on detecting sybil users. However, sybil users alone do not harm the system. What is really dangerous is that multiple sybil users collude together and form a sybil group. The attacker controls the sybil group to attack the system seriously. However, few studies have analyzed the relationship between sybil users and detected sybil groups in OSNs. Identifying sybil groups can be used to detect attackers, who create and control sybil users. A further step can be taken to study behaviors of attackers, and design new mechanism to prevent attacks.

In this paper, we detect sybil groups in Renren social network [2], the largest and most popular OSN in China. We observe that normal user's popularity is at least equal to the social degree in Renren. We identify suspicious users who have the popularity much smaller than the social degree. We further modify the set of suspicious users by their social relationships. Finally, we classify suspicious users by their IP addresses, and detect sybil groups. We apply the sybil group detector to Renren, and identify 2,653 sybil groups and 989,764 sybil users (Section II).

We design automatic validation mechanisms of sybil groups, by analyzing action time similarity of users in a group. We apply the validation methods to sybil groups. We observe that users in sybil groups show extremely high similarity of action time than that of users in normal groups. It indicates that sybil users are simultaneously controlled by the same attacker. Overall, 2440 (91.9%) sybil groups and 985,797 (99.6%) sybil users are successfully validated (Section III).

In summary, we present the first attempt to identify and validate sybil groups in online social networks. We utilize multiple attributes to detect sybil users and identify sybil groups in the real system. Our results are confirmed by automatic validation mechanisms, instead of simulation ex-

[1]

[2]

periments or manual inspections. Our sybil group detection and validation mechanisms have important implications for system design to defend against sybil attacks in OSNs.

II. Detecting Sybil Groups

Before diving into the detection of sybil groups, we begin by providing background information about the Renren social network. We then study characteristics of normal users in Renren. We further build the sybil group detector. Finally, we use our mechanism to detect sybil groups in Renren.

A. The Renren Social Network

Renren was set up in 2005, and it is the oldest and biggest OSN in China [14]. Renren has similar features and user interface, and it can be best characterized as Facebook's Chinese twin. Users maintain personal profiles, upload photos and write blogs. Users also establish bidirectional social relationships with friends, view friends' profiles and exchange comments.

As the growth of user population, Renren also becomes an attractive platform for companies to promote products. Some attackers create sybil groups to unfairly increase the power of target users in social games, or spread advertisements for companies. Renren has deployed several orthogonal techniques to detect sybil users. In order to further improve security and defend against sybil attacks, Renren has built a collaboration with our research team since December 2010 [13]. To support the project, Renren provides user data on their servers, which is preprocessed and anonymized.

B. Relationship between Popularity and Social Degree

We begin by giving the definition of some properties in Renren. People establish bidirectional social relationships with friends in OSNs. *Social degree* is defined as the number of friends. Standard user is limited to a maximum of 1,000 friends in Renren. Users may pay a subscription fee to increase this limit to 2,000. *Popularity* is defined as the number of visits a user's profile receives [14]. The popularity reflects how attractive the user's profile is.

The popularity always increases as the the social degree grows. For example, Alice hopes to make a friend with Bob, and sends a friend request to Bob. Bob receives the friend request, and agrees to build the social relationship with Alice. Their social degrees both grow by 1, and their popularity is likely to increase by at least 1. Alice may browse Bob's profile, find the profile interesting and send the friend request. Alice may also visit Bob's profile after the establishment of friend relationship. Alice's profile visiting increases Bob's popularity. Bob may also view Alice's profile and increase Alice's popularity: Bob views Alice's profile, and decides to accept the friend request. After they make friends, Bob may further visit Alice's profile.

The popularity sometimes grows as the social degree remains unchanged. The popularity is the number of profile browsing by all visitors. A significant of visitors are strangers [14]. Strangers increase the popularity but have no contribution to the social degree. Generally speaking, the user's popularity is at least equal to the social degree.

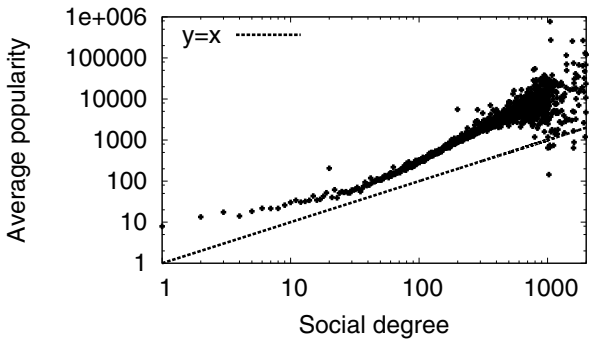

Figure 1. Social degree versus Average popularity

We measure the relationship between the popularity and the social degree in the real system. We obtain 1,000,000 random users' popularity and social degree. We compute the average popularity of users who have the same social degree. Then we plot the social degree versus average popularity in Figure 1. As the growth of social degree, the average popularity also rises up. Most of points are above the line $y = x$, and the average popularity is higher than the social degree. Only 8 points are below the line $y = x$. We manually check 8 points, and find them abnormal. Some of users never publish any content, but have more than 1,000 friends; others are already forbidden by Renren.

C. Sybil Group Detector

In this subsection, we design the sybil group detector based on several attributes, including popularity, social degree, friend relationship and IP address. We validate the sybil group detector in the section III.

First of all, we identify suspicious users and summarize the implementation in algorithm 1. The initial set of suspicious users consists of people who have the popularity smaller than 10, and the social degree bigger than 20. In the subsection II-B, we observe that normal users have the popularity at least equal to the social degree. However, the attacker easily controls some sybil users to send friend requests, and manipulate other sybil users to accept friend requests. Sybil users do not need to view others' profiles in the establishment of social relationships, and their popularity is smaller than the social degree. Then we filter out suspicious users with loose relationships with other suspicious users. For every suspicious user, we compute the number of friends who are also in the suspicious set. If the user has less than 5 suspicious friends, the user is deleted from the suspicious set. It ensures that the user and at least 5 friends are both abnormal. Some people have the popularity bigger than the social degree, but they have strong relationships

Algorithm 1 Identification of suspicious users.

Input:

U: the set of all users
$popularity_i$: the popularity of the user i ($i \in U$)
$socialDegree_i$: the social degree of the user i ($i \in U$)
F_i: the set of user i's friends ($i \in U$)
S: the set of suspicious users

Procedure:

```
S = φ
for i ∈ U do
    if popularity_i <= 10 then
        if socialDegree_i >= 20 then
            S = S ∪ {i}
        end if
    end if
end for
for i ∈ S do
    if |F_i ∩ S| < 5 then
        S = S − {i}
    end if
end for
for i ∈ U\S do
    if |F_i ∩ S| > 0.5 * |F_i| then
        S = S ∪ {i}
    end if
end for
return S;
```

with users in the suspicious set. These people blend in well with normal users, but they are still suspicious. We add users into the suspicious set, who have more than 50% of abnormal friends.

Secondly, we divide suspicious users into sybil groups. We utilize IP addresses when users register their accounts. If suspicious users are registered with similar IP addresses, they are likely to be controlled by the same attacker. We classify suspicious users based on prefixes of their IP addresses. For example, the suspicious user has the IP address as A.B.C.D. We extract the prefix as A.B, and put the user into the corresponding group. All suspicious users are divided into groups by their IP addresses. Attackers often create many sybil users and control them to collude together. Small groups are useless for attacks. So we filter out groups with less than 5 users. Remaining groups are identified as sybil groups, and people in these groups are considered as sybil users.

D. Detection Results

We apply our sybil group detector to Renren, and identify 2,653 sybil groups and 989,764 users. The largest sybil group has the size 38,994. We classify sybil groups by their sizes, and plot the distribution of groups and users in Table I. For groups with the size between 6 and 10, they cover 11.4% of groups and 0.2% of users. 39.3% of groups have the size between 11 to 100, and 41.3% of groups have the size between 101 to 1,000. The majority of groups have the medium size. 43.8% of users belong to large groups with the size between 1,001 and 10,000. Only 6 groups have the size bigger than 10,000, but these groups have 128,481 users in total. Large sybil groups are extremely dangerous in online social networks. If all users in a large sybil group collude together, they can harm the system seriously.

Size	# of Groups	# of Users
6-10	302 (11.4%)	2,306 (0.2%)
11-100	1,042 (39.3%)	41,475 (4.2%)
101-1,000	1,097 (41.3%)	384,415 (38.8%)
1,001-10,000	206 (7.8%)	433,087 (43.8%)
>10,000	6 (0.2%)	128,481 (13.0%)
All	2,653 (100%)	989,764 (100%)

Table I
DETECTION RESULTS. EACH ROW PROVIDES THE NUMBER OF DETECTED GROUPS AND USERS IN A GIVEN SIZE.

Our sybil group detector has several thresholds of parameters. Thresholds of popularity and social degree in algorithm 1 are predetermined by Renren. Due to the privacy protection, Renren provides us anonymous users, who have the popularity smaller than 10 and social degree bigger than 20. In future work, we will contact Renren and obtain more anonymous information. Then we will study how different thresholds influence detection results.

III. VALIDATING SYBIL GROUPS

Given the lack of publicly available datasets, previous works [9], [10], [11], [12] use simulation to evaluate their methods. However, simulation experiments consider several important factors, and ignore other factors. It is still unknown about the performance of these methods in real systems. In order to assess their methods, Tran et al. manually inspect suspicious articles [12], and Yang et al. examine feedback from customer support department [13]. These techniques require much human effort, and they are effective only after suspicious articles have been posted or abnormal users have been forbidden.

In this section, we design automatic mechanisms to validate our sybil group detector. We study action time similarity of users in sybil groups, in comparison with normal groups. In order to build normal groups, we randomly select 200,000 users from 200 universities. We use the same technique in subsection II-C, and divide users into groups based on prefixes of IP addresses. Then we filter out groups with less than 5 users, and identify remaining groups as normal groups. Since people in the same university are likely to have similar IP addresses, the majority of groups have more than 5 users. Overall, we obtain 1,480 normal groups and 179,319 normal users.

A. Validation Methodology

To validate sybil groups, we use the widely acknowledged distinguishing feature: the action time similarity. The action time similarity is based on the intuition that all users in a sybil group take coordinated actions within the similar time. For example, many users post spam in a short time in Facebook, and their posting time is similar [3]. In order to save the time cost, the attacker simultaneously controls all users in a sybil group to take actions.

Algorithm 2 Computation of median interval of action time.

Input:

G: the set of all users in a group
$actionTime_i$: the action time of the user i ($i \in G$)
m: the median interval of action time for a group.

Procedure:

1: $n = |G|$
2: sort $actionTime_i$ for all users in a group, and output results as $actionTime_{i'}$ ($actionTime_{1'} < actionTime_{2'} < ... < actionTime_{n'}$)
3: **for** each $i \in [1, n-1]$ **do**
4: $interval_i = actionTime_{(i+1)'} - actionTime_{i'}$
5: **end for**
6: sort $interval_i$ and output results as $interval_{i'}$ ($interval_{1'} < interval_{2'} < ...interval_{(n-1)'}$)
7: $m = interval_{[n/2]'}$
8: **return** m;

We utilize the median interval of action time to measure the similarity. We summarize the computation of median interval in algorithm 2. Firstly, we sort action time of all users in a group. Then we measure the absolute time interval between consecutive actions, and extract the median value of all such intervals. If users take actions within the short time, their action time is similar and the median interval is small; if users randomly take actions within the long time, their action time is dissimilar and the median interval is large.

The median interval characterizes the time similarity of actions taken by all users in a group. However, the median interval is negatively correlated with the group size. The bigger the group is, the more intensively user actions are distributed in the time period, which causes the smaller median interval. In contrast, the small group is likely to have the large median interval. In order to reduce the impact of the group size, we multiply the median interval by the group size.

We define three validation methods:

- **The validation based on last login time**: The last login time describes the time when the user logins into the OSN for the last time. For a group, we compute the median interval of last login time multiplied by the group size. If the value is much smaller than that of a normal group, users in the group login in group with the similar time, and the group is validated as the sybil group.
- **The validation based on register time**: The register time is defined as the time when the user registers a new account in the system. For a group, we compute the median interval of register time multiplied by the group size. If the value is much smaller than that of a normal group, the register time of users are distributed too intensively, and we validate the group as the sybil group.
- **The validation based on friend establishment time**: The friend establishment time describes the time when the user establishes the social relationship with a friend. For a group, we compute the median interval of friend establishment time multiplied by the number of friend relationships. Note that we mainly consider friend relationships within the group. The attacker simultaneously controls sybil users to make friends with others in the same group. Therefore, establishment time of social relationships within the group is likely to be similar. Sybil users also send friend requests to normal users outside of their group. The friend establishment time is determined by normal users, instead of attackers. If the result is extremely smaller than that of a normal group, social relationships within the group are established within the short time, and their group is validated as the sybil group.

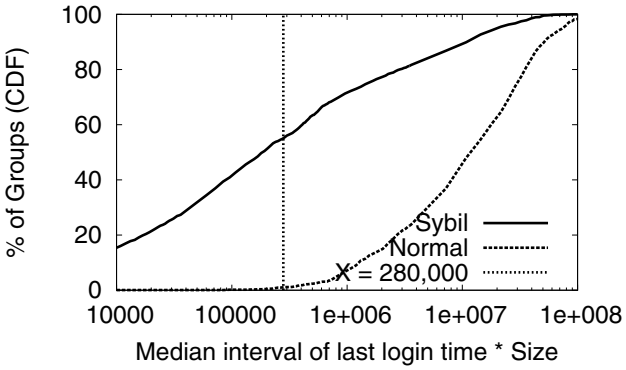

Figure 2. Group distribution of sybil groups and normal groups

B. Validation Results

First of all, we apply the validation based on last login time to verify sybil groups detected in the subsection II-D. For each group, we compute the value of the median interval of last login time multiplied by the group size. Figure 2 shows the group distribution of sybil groups and normal groups. 55.2% of sybil groups have the value smaller than 280,000, while only 1% of normal groups have the value smaller than 280,000. We further consider the number of users in groups and plot user distribution in Figure 3. 63.1% of sybil users are in groups with the value smaller than

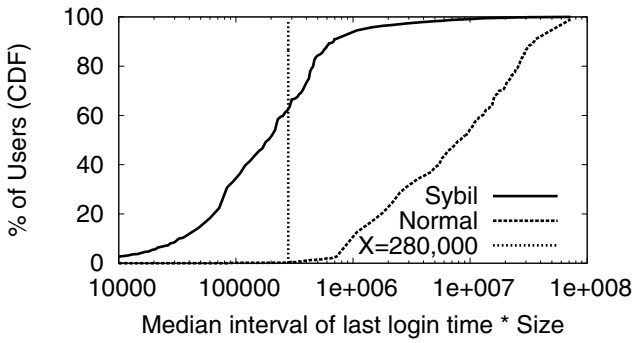

Figure 3. User distribution of sybil groups and normal groups

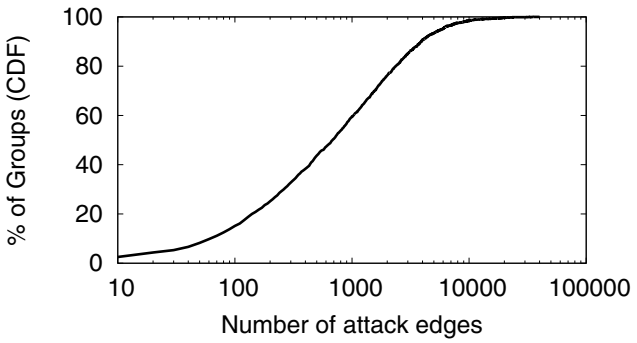

Figure 4. Number of edges with normal users

# of Validations	# of Groups	# of Users
1	640 (24.1%)	115,478 (11.7%)
2	956 (36.0%)	331,838 (33.5%)
3	844 (31.8%)	538,481 (54.4%)
All	2440 (91.9%)	985,797 (99.6%)

Table II
VALIDATION RESULTS.

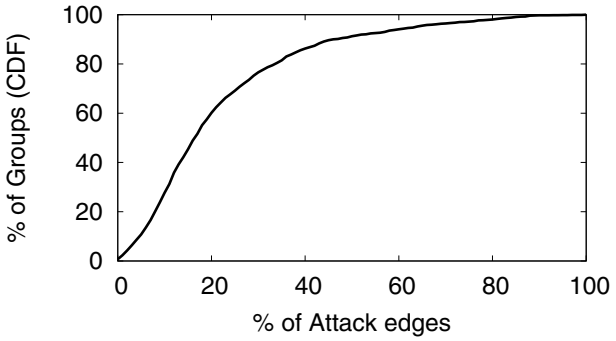

Figure 5. Percentage of edges with normal users

280,000, while 0.24% of normal users are in groups with the value smaller than 280,000. Consequently, 55.2% of sybil groups have much smaller value than that of normal groups. Users in these groups show obvious similarity of last login time, and they are simultaneously controlled by attackers. 1464 (55.2%) sybil groups are successfully validated.

We also apply validation methods based on register time and friend establishment time, respectively. If the group is validated by at least 1 validation method, the group is successfully verified. Table II show the number of sybil groups and sybil users, who are successfully verified by 1, 2 or 3 of validation methods, respectively. Overall, 2440 (91.9%) sybil groups and 985,797 (99.6%) sybil users are successfully validated. 31.8% of sybil groups and 54.4% of sybil users are even validated by 3 methods. Results show that our sybil group detector achieves good performance.

IV. MEASURING SYBIL GROUPS

We measure the topological characteristics of sybil groups validated in the section III. In particular, we analyze how sybil groups connect to normal users in the wild. Following the definitions proposed in previous works [11], [13], *attack edge* is defined as the social relationship between a sybil user and a normal user.

We begin measurement of sybil topology by examining the attack edges of sybil groups in Renren. For every sybil group, we compute the number of attack edges with normal users. Then we plot results of all sybil groups in Figure 4. Only 15% of sybil groups have less than 100 attack edges; 40% of sybil groups have more than 1,000 attack edges. Some of sybil groups build a large number of social relationships with normal users.

We take a further step, and compare attack edges and all edges. For each sybil group, we calculate the number of attack edges divided by the number of all edges. Figure 5 shows the percentage of attack edges. 28% of sybil groups have less than 10% of attack edges. Users in these sybil groups seldom build social relationships with normal users. These sybil users are still harmful to the system, because they can spread advertisements by leaving comments in target users' profiles. In contrast, 24% of sybil groups have more than 30% of attack edges, and 9% of sybil groups have more than 50% of attack edges. A large number of friend relationships are built between sybil users and normal users in these groups. Previous detection algorithms [9], [10], [11], [12] identify sybil users by locating the small number of edge cuts that separate the sybil users from normal users. Since some sybil groups have a large percentage of attack edges, previous detection algorithms are unlikely to succeed on social graphs. This opens the door for the improvement of detection algorithms to consider more trustful graphs.

V. RELATED WORK

Initial studies characterize the spam problem in online social networks. Previous approaches use blacklists to identify malicious URLs in Facebook [3] and Twitter [4].

Ratkiewicz et al. detect the early stages of viral spreading of political misinformation in Twitter [2]. These works mainly detect spam messages, and analyze malicious behaviors. In contrast, we focus on identifying and validating sybil groups, rather than spam content.

Various techniques are applied to study sybil users or spammers in OSNs. First of all, several sybil defense schemes [9], [10], [11], [12] are based on the assumption that sybil users can hardly make friends with normal users [15], [16]. Secondly, honeypots are deployed to trap spammers who attempt to make friends with them in Twitter [6], [17], [18] and Myspace [5], [17], [19]. Thirdly, researchers manually identify spam tweets in Twitter [20], phantom profiles in Facebook [1] and spammers in Youtube [21]. Finally, Thomas et al. identify accounts suspended by Twitter for disruptive activities [7]; Yang et al. analyze friend requests to detect sybil users [13]; Yardi et al. examine spam around the Twitter meme to detect spammers [22].

Our works are different from these studies in several fields: first of all, previous works identify sybil users or spammers, without detecting malicious groups. We analyze the relationship between sybil users, and further identify sybil groups. Secondly, some initial studies mainly use simulation experiments or manual inspections to verify their methods; other works verify sybil users only after spam content has been posted or abnormal users have been forbidden. We apply our sybil group detector to the large-scale datasets in Renren, and design automatic mechanisms to validate our detector in the real system. Malicious behaviors are not necessary in our validation methods, and potential attacks can be prevented beforehand.

VI. Conclusions

In this paper, we present the first attempt to identify and validate sybil groups in the real system. First of all, we build the sybil group detector based on multiple attributes, including popularity, social degree, friend relationship and IP address. We apply the sybil group detector to Renren, and identify 2,653 sybil groups and 989,764 sybil users. Secondly, we design automatic validation mechanisms of sybil groups, by analyzing action time similarity of users in a group. Overall, 2440 (91.9%) sybil groups and 985,797 (99.6%) sybil users are successfully validated. Our sybil group detection and validation mechanisms have important implications for system design to defend against sybil attacks in OSNs.

Acknowledgment

This work is supported by National Science Foundation of China under the National Basic Research Program of China under Grant No.2011CB302305.

References

[1] A. Nazir, S. Raza, C.-N. Chuah, and B. Schipper, "Ghostbusting facebook: Detecting and characterizing phantom profiles in online social gaming applications," in *Proc. of The 3rd Workshop on Online Social Networks*, June 2010.

[2] J. Ratkiewicz, M. D. Conover, B. M. Meiss, Goncalves, A. Flammini, and F. Menczer, "Detecting and tracking political abuse in social media," in *Proc. of ICWSM*, July 2011.

[3] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proc. of ACM Internet Measurement Conference*, November 2010, pp. 35–47.

[4] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: The underground on 140 characters or less," in *Proc. of ACM Conference on Computer and Communications Security*, October 2010, pp. 27–37.

[5] D. Irani, SteveWebb, and C. Pu, "Study of static classification of social spam profiles in myspace," in *Proc. of ICWSM*, May 2010.

[6] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. of Annual Computer Security Applications Conference*, December 2010, pp. 1–9.

[7] K. Thomas, C. Grier, V. Paxson, and D. Song, "Suspended accounts in retrospect: An analysis of twitter spam," in *Proc. of ACM Internet Measurement Conference*, November 2011.

[8] M. Kirkpatrick, "Social networking now more popular than email, report finds," ReadWriteWeb, March 2009.

[9] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman, "Sybilguard: Defending against sybil attacks via social networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 576–589, June 2008.

[10] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *Proc. of the IEEE Symposium on Security and Privacy*, May 2008, pp. 3–17.

[11] G. Danezis and P. Mittal, "Sybilinfer: Detecting sybil nodes using social networks," in *Proc. of NDSS*, February 2009.

[12] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content voting," in *Proc. of NSDI*, April 2009, pp. 15–28.

[13] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncovering social network sybils in the wild," in *Proc. of ACM Internet Measurement Conference*, November 2011.

[14] J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, and B. Y. Zhao, "Understanding latent interactions in online social networks," in *Proc. of ACM Internet Measurement Conference*, November 2010, pp. 369–382.

[15] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," in *Proc. of SIGCOMM*, August 2010, pp. 363–374.

[16] A. Mohaisen, N. Hopper, and Y. Kim, "Keep your friends close: Incorporating trust into social network-based sybil defenses," in *Proc. of the IEEE Infocom*, April 2011.

[17] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: Social honeypots + machine learning," in *Proc. of SIGIR*, July 2010, pp. 435–442.

[18] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on twitter: Human, bot, or cyborg?" in *Proc. of Annual Computer Security Applications Conference*, December 2010, pp. 21–30.

[19] S. Webb, J. Caverlee, and C. Pu, "Social honeypots: Making friends with a spammer near you," in *Proc. of CEAS*, Mountain View, USA, August 2008.

[20] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Proc. of CEAS*, Washington, USA, July 2010.

[21] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonglves, "Detecting spammers and content promoters in online video social networks," in *Proc. of SIGIR*, Boston, USA, July 2009.

[22] S. Yardi, D. Romero, G. Schoenebeck, and D. boyd, "Detecting spam in a twitter network," *First Monday*, vol. 15, no. 1-4, January 2010.