A Taste of Tweets: Reverse Engineering Twitter Spammers

Chao Yang
SUCCESS Lab
Texas A&M University
yangchao@cse.tamu.edu

Jialong Zhang
SUCCESS Lab
Texas A&M University
jialong@cse.tamu.edu

Guofei Gu
SUCCESS Lab
Texas A&M University
guofei@cse.tamu.edu

ABSTRACT

In this paper, through reverse engineering Twitter spammers' tastes (their preferred targets to spam), we aim at providing guidelines for building more effective social honeypots, and generating new insights to defend against social spammers. Specifically, we first perform a measurement study by deploying "benchmark" social honeypots on Twitter with diverse and fine-grained social behavior patterns to trap spammers. After five months' data collection, we make a deep analysis on how Twitter spammers find their targets. Based on the analysis, we evaluate our new guidelines for building effective social honeypots by implementing "advanced" honeypots. Particularly, within the same time period, using those advanced honeypots can trap spammers around 26 times faster than using "traditional" honeypots.

In the second part of our study, we investigate new *active* collection approaches to complement the fundamentally *passive* procedure of using honeypots to slowly attract spammers. Our goal is that, given limited resources/time, instead of blindly crawling all possible (or randomly sampling) Twitter accounts at the first place (for later spammer analysis), we need a lightweight strategy to prioritize the *active* crawling/sampling of *more likely* spam accounts from the huge Twittersphere. Applying what we have learned about the tastes of spammers, we design two new, *active and guided* sampling approaches for collecting most likely spammer accounts during the crawling. According to our evaluation, our strategies could efficiently crawl/sample over 17,000 spam accounts within a short time with a considerably high "Hit Ratio", i.e., collecting 6 correct spam accounts in every 10 sampled accounts.

Keywords

Online Social Network Websites, Spam, Twitter

1. INTRODUCTION

Online Social Networks (OSNs) such as Twitter and Facebook have been utilized by social spammers to garner victims. Particularly, social spammers could actively garner their victims through creating spam accounts to initialize unsolicited social relationships or send unsolicited messages, rather than merely passively waiting for victims' visits as in the traditional email spam or web spam cases.

In fact, restricted by OSNs' anti-spam measures, many OSN spammers have evolved to launch *Targeted Social-Media Spamming* (i.e., spammers selectively choose their spamming targets by analyzing those targets' behaviors [34]). Many Twitter users have undergone the following experience: once they write some big brand names such as "Ipad" or "Best Buy" in their tweets, they may receive a slew of @mentions[1] offering "free" products or gift cards related to the brands [6, 5]. Note that before 2012, @mentions will be directly shown on the users' public timeline; since then, users will receive senders' @mentions in the users' "Notifications[2]", no matter the users follow the senders or not. Such observations indeed imply an obvious interaction between users' social behaviors and spammers' actions (as illustrated in Figure 1).

Figure 1: Illustration of interactions between users' social behaviors and spammers' actions.

The benefit for spammers to use this strategy to find targets is straightforward. Through selectively choosing targets to initialize unsolicited friend requests or send unsolicited messages, spammers could significantly decrease the risks of being detected under current OSNs' policies. (According to our observation, a Twitter account that constantly follows more than 50 accounts per day will highly possibly be suspended by Twitter within a week.) Furthermore, after knowing targets' tastes or social friend-circles, spammers could significantly increase their chances of successful spamming, either by actively pushing spam messages related to targets' tastes (e.g., on Twitter) or pretending to be in the same social friend-circle (e.g., on Facebook). In this way, social spammers could garner victims more effectively by launching customized actions based on their targets' social behavior characteristics. Thus, this is different from the scenario for traditional email spam or web spam, in which attackers usually know nothing about their targets and can merely blindly send spam.

[1] @mentions are tweets that are sent to specific users by using the tag of "@".

[2] A tab in the users' homepage shows users' social interactions with others.

ACSAC '14,December 08 - 12 2014, New Orleans, LA, USA

http://dx.doi.org/10.1145/2664243.2664258.

However, so far we know little about basic insights of the interactions between users' behaviors and spammers' actions. Such insights can facilitate us to understand common questions such as "Why do I get spam friends? [13]", "Why do I receive spam messages? [2]" and "How do spammers find their targets?". The desire of addressing such questions, and thus *obtaining insights for defending against social spammers*, form one important motivation of this work.

In addition, among many existing research efforts in fighting against spam/spammers [23, 41, 28, 15, 38, 36, 33], social honeypot techniques are quite promising, and have been widely deployed to collect spammers [28, 29, 36]. A social honeypot is essentially a specially created fake account with the intent to capture spammers' social interactions. However, current social honeypots are designed to be either *too static* (few behaviors performed by honeypots) or *too uniform* (few variations among honeypots' behaviors). As a result, those honeypots are not used in an optimal or effective way to trap as many spammers as they can. The fundamental reason is that we lack basic insights of the strategies utilized by spammers to select spam targets. Thus, a good understanding of spammers' tastes is pressing and we seriously need *systematic guidelines for building more effective (attractive) social honeypots*.

Furthermore, although many existing studies rely on social honeypots (or even manual identification) to collect likely spam accounts (aiming at further analyzing them to generate defense insights), such strategies are still not very efficient in terms of collecting a large number of spam accounts from the huge Twittersphere. In particular, the technique of social honeypots is relatively passive and typically requires a long time to attract many spam accounts. The strategy of manually labeling spam accounts is tedious, time-consuming, and very difficult to scale. Thus, given limited resources/time, a light-weight strategy to selectively sample *more likely* spam accounts (instead of blindly collecting all or randomly sampling accounts) during the crawling from the huge Twittersphere is strongly desired.

In this paper, through reverse engineering spammers' tastes (their preferred targets to spam), we aim at providing guidelines for designing more effective social honeypots, and designing lightweight and guided strategies to actively sample more likely social spam accounts. To achieve this goal, we use Twitter as a case study due to its great popularity and publicity. Specifically, to reveal which behaviors tend to incur spammers' contact, we implement 96 "benchmark" Twitter social honeypots with 24 diverse fine-grained social behavior patterns to trap spam accounts. After launching our social honeypots for five months, we successfully garner around 600 spam accounts. Using these data, we analyze spammers' tastes (how spammers find their targets), through comparing the effectiveness of social honeypots with different behavior patterns. Based on the analysis, we design and implement 10 more effective ("advanced") honeypots to trap Twitter spammers. Within the same time period, using those advanced honeypots can trap spammers around 26 times faster than using "traditional" honeypots. To further understand spammers' tastes, we also design an algorithm to extract semantic topic terms, which may highly attract spammers' attentions. In addition, with the concern of limited time/resources, through reverse engineering spammers' strategies of selecting targets, we gain the insights to design two guided approaches to prioritize the *active* sampling of more likely spam accounts from Twittersphere, which is an effective complement to existing *passive* social honeypots.

In summary, the main contributions of our study are:

- We present a deep analysis of spammers' tastes: spammers tend to contact with accounts that tweet messages and follow accounts related to specific topics.
- We deploy "advanced" (more effective) honeypots based on our provided guidelines, which can trap spammers around 26 times faster than using "traditional" honeypots.
- We design two lightweight, guided approaches to prioritize the sampling of more likely Twitter spam accounts in the huge Twittersphere. According to our evaluation, our designed two samplers can efficiently collect over 17,000 Twitter spam accounts in a short time with a considerably high "Hit Rate" (correctly collect around 6 spam accounts in every sampled 10 accounts).

2. RELATED WORK

Detection of spam accounts. The task of detection is to answer the question: given an OSN account, how can we tell whether it is a spam account or not? Most existing OSN spam account detection solutions can be mainly classified into two categories. The first category of work, such as [41, 28, 15, 38, 36, 33, 19], utilizes machine-learning techniques to classify spam accounts according to their collected training data and selections of classification features (e.g., the ratio of an account's following number to its follower number). While successful, some recent research has reported that spammers have begun to utilize multiple tricks to evade existing detection features by pretending to be legitimate ones [41, 33]. Thus, this type of approach may fail to detect those evasive spam accounts. The second category of research detects spam accounts by examining whether the URLs or domains of the URLs in the tweets exist in the public URL blacklists or domain blacklists [23]. One limitation of such approaches is that, current URL (and domain) blacklists usually have a big lag to identify malicious links in the spam tweets [23]. Our work is not a detection solution. However, our proposed sampling strategies (Section 5) can provide a guided approach to prioritize the sampling of more likely spam accounts (instead of blindly/randomly crawling) in the huge Twittersphere, thus providing a good first-layer filter for existing detection approaches.

Utilization of honeypots. A honeypot is a decoy (e.g., a computer, data, or a network site) mainly set up to attract attackers. Traditionally, the honeypot techniques have been widely used for capturing malware and related malicious activities. Server-side honeypots are mainly implemented by emulating vulnerable services or software to trap attacks, aiming at collecting malware and malicious requests [44], understanding network and web attacks [26], building network intrusion detection systems [27], or preventing the spread of spam email [20]. Client-side honeypots are mainly used to detect compromised (web) servers [32, 1, 39, 31]. In [14], Antonatos *et al.* proposed an approach to detect instant messaging (IM) threats using IM honeypots.

In the context of OSN, social honeypots are defined as OSN accounts that appear to belong to real users, but are actually fake accounts used for attracting spammers. Due to its simplicity and low false positives, social honeypots are a great way to collect spammers for further study, e.g., understanding their characteristics and then further building effective machine-learning features to detect them. Many existing studies [36, 28, 28] use this social honeypot technique. However, an important missing component in this line of research is that, we still know little about the interactions between users' behaviors and spammers' actions, e.g., why this social honeypot can attract few/many spammers. Essentially, we need *a systematic analysis on how to build more effective social honeypots*, which is an important goal of this work. Thus, this paper bridges the gap in existing research using social honeypots.

Measurement of spam campaigns and networks. In [43], Yardi *et al.* analyzed Twitter spam accounts' social behaviors and network structures by investigating a specific spam campaign [43]. In [21], Gao *et al.* conducted a study on detecting and characterizing social spam campaigns on Facebook, based on the observation that spam accounts in the same spam campaign tend to send similar spam messages simultaneously. In [37], Thomas *et al.* analyzed tools, techniques, and support infrastructure utilized by spam accounts through retrospecting suspended accounts. In [42], Yang *et al.* presented a deep empirical analysis on spammers' social network. All these studies deepen our understanding of spam campaigns and networks, thus providing valuable insights for defenders. In this paper, we perform a deep social honeypot measurement study to understand spammers' tastes, thus help to design new guidelines for building better social honeypots and guided strategies to prioritize the sampling of more likely spam accounts. Thus, our work is a new supplement to existing work.

Recently, Ghosh *et al.* investigated link farming activities in Twitter and and found that a small number of legitimate Twitter users account for the majority of link farming activities [22]. Irani *et al.* presented a new social engineering attack to trick benign users into contacting spammers, instead of using spam accounts to initiate contact with benign accounts [25]. The results show that recommendation systems, demographics and visitor tracking could affect normal users' choices in making new friends. Our study is essentially a reverse side story: focusing on revealing *spammers'* (instead of normal users') tastes, and design more effective social honeypots and new sampling strategies to actively collect likely spam accounts.

3. PROBLEM STATEMENT

We next introduce the research scope of this work. As discussed in Section 1, our research goal is to understand the characteristics of targeted spamming in Twitter, and further to gain new defense insights against them by reverse engineering the spamming tastes of those spammers. Particularly, we use a relatively strict/conservative view (similar to existing work [19, 41] and Twitter rules [7]) to consider an account to be a spam account, if it meets one of the following criteria: (1) tend to post spam or malicious URLs in the tweets; (2) tend to post scam words in the tweets; (3) repeatedly post duplicate tweets; (4) repeatedly send "@mention" messages to other accounts with few useful content.

To achieve our research goal, we first design 96 social honeypots with diverse social behavior patterns to garner spammers (see Figure 2). Based on the functions provided by Twitter, these social behavior patterns mainly vary in terms of tweeting behaviors, following behaviors, and application usage. Particularly, the content posted by users, the famous accounts followed by users and the applications used by users may display users' tastes, incurring spammers' contact. Then, these social honeypots could trap spammers by receiving spammers' unsolicited messages and obtaining spam followers. All of social honeypots' behaviors and their trapped spammers' actions (sending unsolicited messages or building unsolicited friendships) will be saved in a local database. Next, after deeply analyzing spammers' tastes by comparing the effectiveness of honeypots with different social behavior patterns, we can provide guidelines to build effective honeypots. Finally, through reverse engineering spammers' strategies of selecting targets, we design two lightweight, guided strategies (Hashtag Sampler and Friend Sampler) to prioritize the sampling of more likely Twitter spam accounts from the huge Twittersphere. More specially, Hashtag Sampler is designed to catch spammers that target on specific accounts if they tweet specific hashtags. Friend Sampler is designed to catch spammers that target on specific famous accounts' followers.

4. REVERSE ENGINEERING SPAMMERS

In this section, we describe our methodologies of extracting and analyzing social spammers' tastes. Specifically, we design and launch multiple social honeypots with diverse fine-grained behavior patterns to garner spammers. Next, through analyzing intrinsic properties of the interactions between users' social behaviors and spammers' actions, we could better understand the following questions: Who do spammers spam? How do spammers find their victims? Through these analysis, we further provide guidelines of building more attractive social honeypots.

4.1 Collecting Spammers' Tastes

4.1.1 Design of Social Honeypots

To analyze the interactions between users' behaviors and spammers' actions, we endow social honeypots with diverse fine-grained social behavior patterns to show diverse users' tastes. Since a Twitter account mainly has three categories of social behaviors (Tweet, Follow and install applications), we design social honeypots based on the variations of these three categories: Tweet Behavior (Tweet), Follow Behavior (Follow) and Application Usage (App) (see Table 1).

Tweet Behavior. The content tweeted by users (and tweeting frequency) may directly expose users' interests. Particularly, the keywords/topics posted by users may reveal their tastes, which could be utilized by spammers to find targets. In fact, many users' real experience has shown that different tweet keywords may behave very differently in terms of incurring spammers [5]. Accordingly, we divide our social honeypots' tweet behaviors into three sub-categories: Tweet Frequency, Tweet Keywords and Tweet Topics.(To reduce possible effects to other users, our social honeypots will not post any links and "@ mentions".)

Tweet Frequency: refers to how often to post one tweet. We divide tweeting frequency into the following three patterns: 1 tweet per hour, 2 tweets per day, and 1 tweet per day. For each pattern, we use 5 honeypot accounts to post tweets according to the specific tweet frequency. Those tweets are randomly selected from the dataset containing around half million Twitter accounts and 14 million tweets, which was collected from Apr. 2010 to Aug. 2010 by using Twitter Stream APIs.

Tweet Keywords: refer to special words or terms in the tweets, which may represent specific semantic topics. We divide tweet keywords into several patterns: popular trending topics, arbitrary hashtags, current affairs, bait words, and no hashtag tweets.

- "Popular trending topics" refer to those hot Twitter trending topics [8], which are widely used by Twitter users to express their opinions or experience on specific topics or events. For each day, we collect top (the most widely used) 10 trending topics. Then, we use 5 honeypot accounts to post these 10 trending topics. Each of them will post 2 trending topics.

- "Arbitrary hashtags" refer to those tweet terms with the tag of "#". The tweets containing the same hashtag will be grouped together by Twitter and can be searched out by users from Twitter Search [11]. These hashtags are also randomly selected from the pre-collected dataset. For each day, we use 5 honeypot accounts to post 10 tweets with hashtags. Each of them posts 2 tweets, which are randomly selected from our collected dataset.

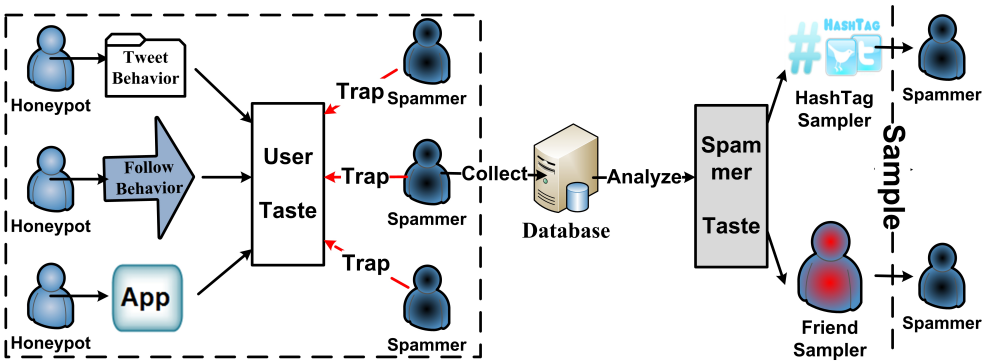


Figure 2: Illustration of the analysis flow.

Table 1: 96 "benchmark" social honeypots with 24 fine-grained social behavior patterns

Index	Category	Sub-Category	Pattern	Index	Category	Sub-Category	Pattern
1-5	Tweet	Frequency	Once per day	6-10	Tweet	Frequency	Twice per day
11-15	Tweet	Frequency	Once per hour	16-20	Tweet	Keywords	Trending Topics
21-25	Tweet	Keywords	Arbitrary Hashtags	26-30	Tweet	Keywords	Current Affairs
31-35	Tweet	Keywords	Bait Words	36-40	Tweet	Keywords	No Hashtags
41-45	Tweet	Topic (Twice per day)	Entertainment	46-50	Tweet	Topic (Twice per day)	Expertise
51-55	Tweet	Topic (Twice per day)	Sports	56-60	Tweet	Topic (Twice per day)	Economics
61-62	Tweet	Topic (Once per hour)	Entertainment	63-64	Tweet	Topic (Once per hour)	Expertise
65-66	Tweet	Topic (Once per hour)	Sports	67-68	Tweet	Topic (Once per hour)	Economics
69-70	Follow	Two accounts per day	Entertainment	71-72	Follow	Two accounts per day	Expertise
73-74	Follow	Two accounts per day	Sports	75-76	Follow	Two accounts per day	Economics
77-81	App	NA	Twitpic	82-86	App	NA	Instagr
87-91	App	NA	Twiends	92-96	Default	NA	NA

- "Current affairs" refer to important social events happened each day. To extract such events, we crawl the top 10 headlines from CNN.com, and use 5 honeypots to post those headlines (each posts two headlines).

- "Bait words" refer to those keywords that are mainly used by spammers in their scam webpages or messages to trap victims (e.g., "giftcard"). We use a list of 200 bait words, mainly obtained through feeding queries such as "scam word lists" to Google.com. Then, we also use 5 honeypots to post 2 messages containing bait words per account per day.

- "No hashtags" refer to tweets without any hashtags, which are used to compare with other social patterns. We use 5 honeypots to post 10 tweets, randomly selected from the database without any hashtags on each day.

Tweet Topics: refer to specific semantic topics in the tweets, which may explicitly reveal users' tastes. Particularly, we focus on the following four topics: *Entertainment, Expertise, Sports, and Economics*. Entertainment contains topics related to TV media, music, books and arts; Expertise contains topics related to IT technology, Science, Fashion and Household; Sports are related to golf, NBA, NCAA, NFL and NHL; Economics are related to business, finance and charity. To use our honeypots to tweet those semantic topics, we first collect tweets related with those topics by searching topic terms (e.g., "NBA") on Twitter. Then, for each topic, we use 5 honeypots to post one tweet per day. To compare, we use 2 honeypots to post 1 tweet per hour.

Follow Behavior. Besides the content posted by users, users' followings (especially those famous people or companies' official accounts) may also reveal their tastes. For example, if an account follows "Lady Gaga", the owner of the account may like music or live concert. Thus, this kind of following tastes might be utilized by spammers.

To extract spammers' such tastes, we use our honeypots to follow "verified accounts", whose tweets are related to four major topics mentioned above: Entertainment, Expertise, Sports and Economics. Specifically, for each topic, we manually collect 400 verified accounts from Twitter. These verified accounts are typically owned by famous people or organizations with high reputation, such as sports stars and official business accounts. Thus, through following those verified accounts, our honeypots explicitly show their interests to those topics. Particularly, for each topic, we use 2 social honeypots to follow 2 verified accounts per day. (To reduce possible effects, each account will follow 30 verified accounts at most.)

Application Usage. Users who install specific Twitter applications (e.g., multimedia sharing tools and online games) may also reveal their specific tastes and thus become spammers' targets. In our test, we choose three very popular Twitter social applications: Twitpic [10], Instagram [4] and Twiends [9]. (Twitpic and Instagram are popular photo and video sharing tools, and Twiends is an online Twitter friend-making tool.) For each application, we use five honeypots to install and use it.

Default. As a comparison, we also use five honeypots with default account registration configuration, which neither post any tweets nor follow any accounts.

In summary, as seen in Table 1, we design 96 honeypots with 24 diverse fine-grained behavior patterns to garner spammers. Since the aim of designing these social honeypots is to understand which

specific social behaviors tend to incur spammers rather than to trap more spammers, we refer these 96 honeypots as "benchmark" honeypots.

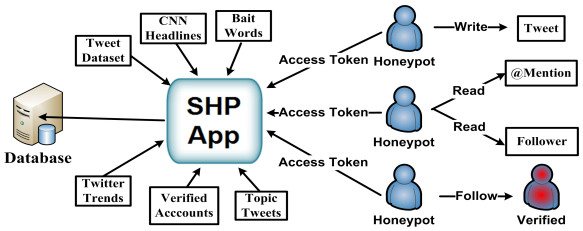


Figure 3: The implementation of social honeypots.

4.1.2 *Implementation of Social Honeypots*

To implement those "benchmark" honeypots, we develop a real-time Twitter application, named social honeypot app (SHP), which has three major operations: write, follow, and read. As illustrated in Figure 3, write operation implements tweet-behaviors by posting tweets on honeypots' timelines; follow operation implements follow-behaviors by following other accounts; read operation collects spam accounts and spam tweets trapped by our social honeypots, by reading honeypots' followers and "@mentions".

More specifically, the application obtains each honeypot's access token to automatically make the corresponding operations (write, follow and read) on the account to perform its designed social behaviors according to the protocol of OAuth 2.0. All the auxiliary data such as our collected tweet dataset, popular trending topics and bait words are loaded into the app to implement corresponding operations. Finally, the app will record each honeypot's social behaviors, and its received "@mentions" and followers into a local database everyday. In this way, we can collect the interactions between honeypots' behavior patterns and their trapped spammers' actions.

To make our honeypots more likely to be real accounts (i.e., to decrease the chance of being identified as honeypots by spammers), we register our honeypot accounts by using real names (e.g., *Tracy Thompson*) and valid email addresses. Also, we initialize the friendships among those honeypots. We admit that smart spammers might still recognize our social honeypots by deeply analyzing those honeypots' behaviors, because these honeypots are designed with a set of scheduled tasks. However, many normal accounts (e.g., official company accounts) are also customized to post particular messages/notifications in a scheduled way. Thus, it is not that trivial for spammers to distinguish honeypots from normal accounts. Also, this limitation is common for all this line of studies, which rely on deploying automated honeypots.

4.2 Analyzing Spammers' Tastes

We next show our data collection results and analysis of trapped spammers' tastes.

4.2.1 *Data Collection Result*

We implemented those 96 "benchmark" honeypots and ran them for five months. We collected $1,077$ unique accounts that at least follow one of our social honeypots, and 440 unique accounts that at least post one "@mention" to one honeypot. In total, there are $1,512$ unique accounts.

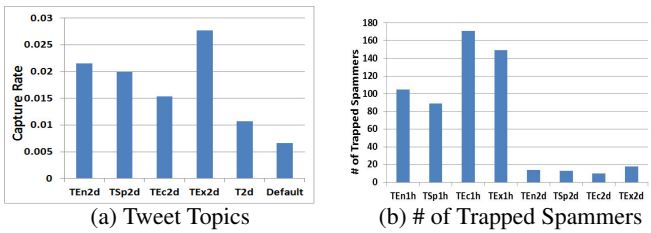
To extract spammers' tastes, we need to identify spam accounts from those $1,512$ accounts. We first found out 303 accounts that have been suspended by Twitter due to their violations to the Twitter Rule. Furthermore, following the definition of our target spam accounts' described in Section 3, we identified additional 278 spam accounts by manually examining accounts timeline and checking those accounts' posted URLs. In total, we obtain 578 spam accounts.

Note that the number of spam accounts trapped by our "benchmark" honeypots seems a little smaller than other earlier social honeypot studies (e.g., [28]). We believe this is due to the following reasons. First, those studies were conducted in early days when Twitter has relatively loose policies to identify/mitigate spammers. However, since 2009, Twitter has taken significant anti-spam efforts to actively filter/mitigate a lot of spam accounts [12]. In addition, in this work, to guarantee the correctness to analyze spammers' interests, we use a relatively strict way to consider an account to be spam. While there could be a few spam accounts missed in our data collection with this relatively strict spammer identification strategy, we believe that our major findings/conclusions in this paper will still hold.

4.2.2 *Analysis of Spammers' Tastes*

We next provide our analysis of spammers' tastes based on 578 trapped spam accounts. To better measure the effectiveness of social honeypots with different behavior patterns, we define a metric named *Capture Rate (CR)*, which is the average number of spam accounts trapped by a honeypot per day. Thus, a higher value of *CR* of honeypots with a specific pattern implies this pattern is more effective to trap spammers. Then, our analysis and measurement results are presented in the question-answer format.

Q1: If an account posts more tweets related with specific semantic topics, does it tend to attract more spammers' attention? Empirical Answer: Yes. One possible way to find targets for spammers is to analyze targets' tweet content, which may show targets' interests on specific topics. Then, through actively pushing spam related to those topics to those targets, attackers may achieve a better chance of success. As seen in Figure 4(a), posting messages related with specific topics (Entertainment – `TEn`, Sports – `TSp`, Economics – `TEc`, Expertise – `TEx`), will incur more spammers' contact than posting arbitrary messages even with the same tweeting frequency (twice per day). More specifically, `TEx2d`'s CR (the highest for tweeting topic twice per day) is around 3 times as that of `T2d`, and `TEc2d`'s CR (the lowest) is around 1.5 times as that of `T2d`. In addition, when we increase the frequency from twice per day to once per hour (e.g., from `TEn2d` to `TEn1h`), honeypots can trap more spammers (See Figure 4(b)). And the average values of CR for these four topics can be increased around 22.35 times (from 0.021 to 0.494). Thus, we can find that honeypots can trap more spammers through frequently tweeting messages related with certain semantic topics.

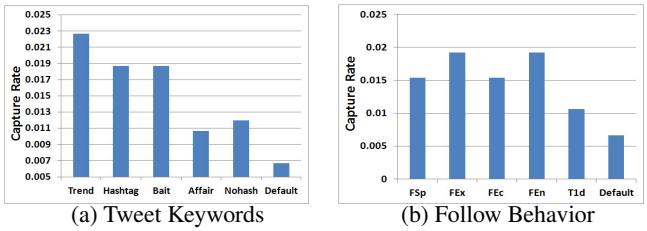


(a) Tweet Topics (b) # of Trapped Spammers

Figure 4: The effectiveness of tweet topics.

Q2: Do accounts that tweet more special terms (e.g., "Trending topics") tend to attract more spammers' contact? Empirical Answer: Yes. As seen in Figure 5(a), the values of CR for

tweeting trending topics (Trend), arbitrary hashtags (Hashtag), and bait words (Bait), are all higher than that of Nohash (arbitrary tweets without hashtags) and Default. This observation indicates that posting special key terms may also incur spammers' contact. This is because these key terms usually represent semantic topic meanings, which could be utilized by spammers to find targets (similar to tweeting topics). In addition, we can find that Trend is more effective than Hashtag. This might because trending topics are more timely and popular than arbitrary hashtags.



(a) Tweet Keywords (b) Follow Behavior

Figure 5: The effectiveness of tweet keywords and follow behavior.

Q3: Do users' following behaviors tend to expose them to spammers? Empirical Answer: Yes. As seen in Figure 5(b), similar to tweet topics, the values of CR for following verified accounts related with the topics of Entertainment (FEn), Sports (FSp), Economics (FEc) and Expertise (FEx) are all higher than that of T1d and Default. This observation implies that the behavior of following those famous ("verified") accounts could be utilized by spammers to find their targets. As a case study, we find one spam account, which mainly posts spam about TV media (the URLs in the tweets have been identified as suspicious by the URL shortening service), shares 19 followings (most of them are related the topic of art or TV media) with one honeypot.

Q4: Do accounts with the usage of social apps tend to be contacted by more spammers? Empirical Answer: No. According to our data, the capture rates of honeypots with the usage of Instagram, Twitpic and Twiends are 0.008, 0.008, and 0.009, respectively. These values are lower than that of most of other social patterns and similar to *Default*. Thus, using social apps do not help much in terms of attracting spammers. This might either because spammers have not use this strategy to find targets or the selections of applications used by our honeypots are not representative.

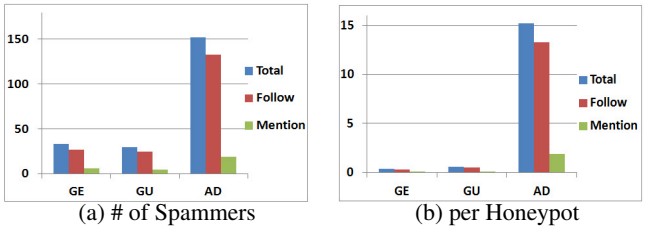
4.2.3 *Guidelines for Designing Effective Honeypots*

According to the above analysis, we could summarize the following guidelines for designing more effective social honeypots to trap Twitter spammers: **(1) post tweets related with specific topics; (2) post tweets containing special keywords such as Trending topics; (3) follow famous accounts related with specific areas.**

To evaluate the effectiveness of those guidelines, we denote *96 "benchmark" honeypots* as GE, and 51 honeypots of them[3] that *meet at least one guideline* as GU. We find that GU's capture rate (0.083) is over two times as that of GE (0.040). This observation indicates that GU (that meet guidelines) is more effective to attract spammers than GE.

To further evaluate the effectiveness of our guidelines, we deploy another 10 "advanced" honeypots (AD) with more guided social behaviors for a week right after finishing the 5-month running of "benchmark" honeypots. Specifically, in each day, each of them

[3]51 accounts are 16-25, 31-35, 41-76 as labeled in Table 1.



(a) # of Spammers (b) per Honeypot

Figure 6: The effectiveness of advanced honeypots.

will behave the following social patterns[4]: (1) post one topic tweet per hour related with each of those four topics mentioned in Section 4.1.1; (2) post one tweet containing one trending topic per hour; (3) post one tweet containing one arbitrary hashtag per hour; (4) post one tweet containing one bait word per hour; (5) Follow 5 experts related with each of four topics per day. Then, as seen in Figure 6, we compare the performance of AD with GE and GU by collecting data in the same week. We can find that AD is much more effective than GE and GU in trapping more spammers. Particularly, AD's capture rate (2.17) is 25.5 times as that of GU (0.085), and 45.2 times as that of GE (0.048). Although this comparison result may contain some bias due to a relatively short period time of data collection, such a huge difference could still validate the effectiveness of our guidelines for designing better social honeypots.

4.2.4 *Extracting Spammers' Interested Topic Terms.*

To better understand spammers' tastes, it is meaningful to extract specific semantic terms, which tend to be used by spammers to find targets. Although this could be partially achieved by analyzing hashtags in the tweets, a more generic and automated approach (not only limited to hashtags) is still needed. Due to the page limit, we mainly introduce the intuition of our way of extracting spammers' interesting topic terms.

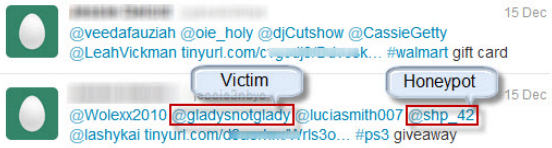


Figure 7: One real case study of potential victims.

As shown in Figure 7, we find that many spammers send illicit "@mentions" not only to our honeypots but also to other users (e.g., "gladynotglady" in this example), which are denoted as "potential victims" in this work. We denote each pair of potential victims and honeypots in one illicit "@mention" as a "victim relationship". Thus, we believe that there should be some common social behavior patterns between our honeypots and those potential victims in each victim relationship, which essentially incur spammers' contacts. An intuitive method is to extract the common terms used by both our honeypots and those victims, which may represent spammers' tastes. However, this approach will extract many widely-used common words, which are not representative for spammers' tastes. Although we use a big stop-word list to filter some common words, it could not help much, because many words tweeted by users are not even spelled completely or in a standard form. In-

[4]To prevent spammers identifying our accounts as honeypots based on the temporal patterns, some random delays are inserted before posting each tweet.

stead, we first use the Latent Dirichlet Allocation (LDA)[5] [17] algorithm to extract topic terms, which are better to represent semantic topics, from the tweets posted by our honeypots and potential victims. Then, we output those topic terms that are highly/frequently shared by the pairs of victims and honeypots. Accordingly, such topic terms may highly attract spammers' contact.

Specifically, our honeypots receive 449 "@mention" messages from spammers and form 5,716 victim relationships with 275 unique potential victims. Then, we extract $1,500$ and 600 topic terms for each honeypot and potential victim, respectively. We finally output the top 500 as semantic topic terms. (Due to the page limit, we skip to show those topic terms.) Furthermore, through extracting semantic topic terms, we could examine whether there is tweet similarities between spammers and their targets. Particularly, we extract semantic topic terms for 278 (manually) identified spam accounts by using LDA. Then, we extract all pairs of honeypots and spam accounts, if the spam accounts either follow or "@mention" the honeypot. Then, we find 81.69% of 360 pairs of two accounts share at least one semantic term. This observation indicates a relatively strong semantic similarity between spammers and their targets.

4.2.5 Ethical Considerations

The technology of deploying social honeypots on real OSNs may raise ethical considerations: whether such social honeypots will impact the normal operation of OSNs. In our work, both "benchmark" and "advanced" honeypots are designed to neither send any @mentions nor post any URLs/spam in the tweets. Also, those accounts only follow a relatively small number (several hundreds) of "verified" accounts. Thus, we believe our designed honeypots will have very limited impacts to other normal users. In addition, the technique of social honeypots has been commonly used to capture spammers [36, 28] or to understand the security vulnerability [18] on OSNs. Furthermore, as advocated in a recent study on the ethics of security vulnerability [30], such studies served as social functions are neither unethical nor illegal.

5. PRIORITIZING THE SAMPLING OF LIKELY SPAMMERS

In this section, we design two guided approaches to actively sample more likely Twitter spam accounts from Twittersphere, based on the observation that many spammers find their targets based on targets' social behaviors.

5.1 Motivation

The collection of spam accounts is usually the first step to analyze spammers' behaviors and to further generate defense insights. However, given the limited time/resources (especially for academic researchers), it is not trivial to collect a large-scale of spam accounts in the huge Twittersphere. Existing studies mainly rely on the following three strategies to collect (likely) spam accounts: implementing social honeypots[28, 29, 36], collecting suspended accounts [37, 25], and manual identification [28, 38, 42]. However, all these three strategies have certain limitations. The honeypot approach is a *passive* one, requiring time (and luck) to wait for spammers' contacts. Collecting suspended accounts requires to develop a robust crawler and takes a considerable long time (typically several months) to crawl Twitter and to wait for collected accounts to be suspended by Twitter. Manual identification could achieve a high accuracy, which requires tedious human work and is not scalable.

Motivated by the limitations of existing strategies to collect (likely) spam accounts, we design two lightweight, guided strategies (called *Samplers* in this paper) to prioritize the active sampling of more likely spam accounts: *Hashtag Sampler and Friend Sampler*. These two samplers are designed to be able to *efficiently* collect/sample a considerable number of targeted social-media spam accounts (a specific type of spam accounts that attract victims by socially interacting with others) in a short time period with a relatively high hit rate without any training process. We clearly admit that these two samplers are not designed to detect all types of spam accounts (e.g., those spam accounts that just tweet malicious content without interacting with any other users.).

5.2 Hashtag Sampler

Basic intuition: Spammers tend to follow those accounts that post spammers' interested keywords (hashtags). According to this intuition, an account might be suspicious if it follows many accounts that share some spammers' interested hashtags in their tweets. At the high level, Hashtag sampler is designed to preferentially sample likely spam accounts through checking *common followers* of multiple accounts that share/post *similar, multiple* hashtags as spammers do.

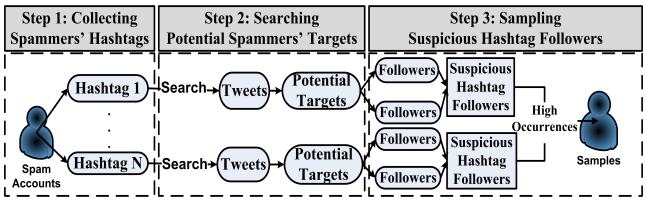


Figure 8: Illustration of Hashtag Sampler.

Detailed Strategy: As illustrated in Figure 8, Hashtag Sampler has three steps to sample likely spam accounts from Twitter: (1) collecting spammers' hashtags; (2) searching potential spammers' targets; (3) sampling suspicious hashtag followers.

Particularly, in Step 1, Hashtag Sampler collects keywords/hashtags that spammers are potentially interested in (i.e., hashtags in spam accounts' tweets) through identifying hashtags ("#") from tweets posted by our trapped spam accounts. In Step 2, for each hashtag, Hashtag Sampler searches the recent M tweets[6] that contain hashtags, through exactly querying the hashtag from Twitter Search. Then, we consider an account to be a potential spammers' *target*, if they send tweets containing that particular hashtag. Accordingly, by extracting the senders of those tweets, Hashtag Sampler searches out all potential targets. In Step 3, for each potential spammers' target, we obtain its followers by using Twitter API. After extracting all targets' followers, we denote those followers with high occurrences as *suspicious hashtag followers*. These hashtag followers essentially follow many other accounts that post that spammers' hashtag. Finally, Hashtag Sampler outputs those accounts as spam accounts, if they are sampled as suspicious hashtag followers with the usage of multiple different hashtags, i.e., they are considered as suspicious hashtag followers with a high occurrence by using different hashtags.

5.3 Friend Sampler

[5]LDA is a generative probabilistic topic modeling (clustering) algorithm, which could cluster terms in the large volumes of unlabeled text into several semantic topics by identifying the latent topics words in the text.

[6]For each query term, Twitter limits to return 1,500 tweets as maximal. Thus, in our experiment, we set $M = 1,500$.

Basic Intuition: Spammers tend to select famous accounts' followers as their targets. In fact, those Twitter accounts (especially famous accounts) followed by a user could also reveal this user's taste, which could be utilized by spammers to find their potential spamming targets. According to this intuition, Friend Sampler is designed to preferentially sample likely spam accounts through checking those accounts that excessively follow multiple famous accounts' followers, i.e., examining *common followers of the followers* of some famous accounts.

Detailed Strategy: As illustrated in Figure 9, Friend Sampler first randomly selects M verified (famous) accounts from those 400 verified accounts used in Section 4. For each account, Friend Sampler collects its N followers (if available), which could be considered as spammers' potential targets. Then, we examine extracted followers of those potential targets, and save them in a dataset with their numbers of occurrences (e.g., if an account follows two potential targets, its number of occurrences is 2), denoted as suspicious account set. Finally, Friend Sampler outputs N_{fd} accounts in the suspicious account set as spammers with the top numbers of occurrences.

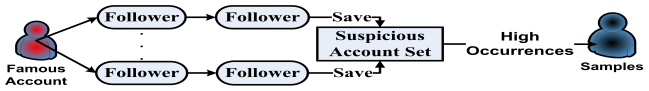


Figure 9: Illustration of Friends Sampler.

6. EVALUATION OF SAMPLERS

In this section, we mainly describe our evaluation methodologies and evaluation results for two samplers (Hashtag Sampler and Friend Sampler).

6.1 Ground Truth and Evaluation Metrics

Ground Truth: To evaluate the effectiveness of two samplers, we require some ground truth. However, as a common challenge for all OSN data analysis work, it is difficult to obtain perfect ground truth for a large-scale dataset. It is straightforward that an account can be considered as spam if it is suspended by Twitter. However, only considering suspend accounts as spam accounts will miss many other spam accounts, which have not been identified/suspended by Twitter. Thus, for the rest *unsuspended* accounts output by our samplers, we rely on a state-of-the-art machine-learning classifier to further examine whether they are spam accounts. This classifier is implemented based on `Random Forest` and uses the same feature set designed in [28]. Then, the classifier is trained by using 2,000 suspended accounts and 20,000 normal accounts (none of them post malicious URLs). The accuracy and false positive rate of this classifier is 99.2% and 0.97% respectively based on the training datasets through *10-fold cross validation* tests.

Note that the usage of the machine-learning technique here is to *estimate* the ground truth rather than to detect spam accounts. Also, we acknowledge that any machine learning classifier may not be absolutely accurate. However in our evaluation, we are mainly interested in getting the *estimation* of the accuracy, instead of absolute values. Furthermore, such a strategy is a common practice for similar studies on accuracy estimation of large scale unlabeled datasets [40].

Evaluation metrics: To measure the effectiveness of sampling strategies with the goal of collecting more likely spam accounts, two metrics are typically considered. The number of collected spam accounts, denoted in our work as "Hit Count (N_{hit})"; and the ratio of Hit Count to the total number of sampled accounts (N_{sample}), denoted as "Hit Ratio (H_r)". Thus, a higher value of Hit Count and Hit Ratio indicates that we can catch more spam accounts and more accurately, respectively. Motivated by the limitations of traditional ways of collecting spam accounts as described in Section 5, our two samplers are designed as lightweight, guided strategies to efficiently and effectively prioritize the sampling of more likely spam accounts instead of (otherwise) crawling/analyzing all accounts in the huge Twittersphere. Our two samplers are *not* designed to find/uncover all types of spam accounts, and they are *not* considered as spammer detectors. Accordingly, we use those two evaluation metrics (N_{hit}, H_r) instead of false positives/negatives in our evaluation. Particularly, many existing studies [45, 42, 24] similarly use these two metrics to measure the effectiveness by outputting the number of hits in a top list. With such notions, if we denote the number of suspended accounts as N_{sus} and the number of spam accounts output by the machine-learning classifier as N_{mal}, we could calculate Hit Count and Hit Ratio as follows[7]: $N_{hit} = N_{sus} + N_{mal}$; $H_r = N_{hit}/N_{sample}$.

6.2 Implementation

To implement Hashtag Sampler, we use 3,246 unique hashtags/keywords posted by 278 identified spammers. For each hashtag, Hashtag Sampler outputs $SF = 500$ (if available) suspicious hashtag followers. By using each spam account's hashtags, Hashtag Sampler samples $M = 500$ suspicious hashtag followers (if available) with the top occurrences as spammers. To implement Hashtag Sampler, we randomly select $M = 40$ verified (famous) accounts (10 accounts for each of four topics). For each verified account, we examine its $N = 5,000$ followers, which are retrieved by sending one "get-follower" request to Twitter. Then, for each follower, Friend Sampler continues to examine its followers, and samples $N_{fd} = 1,000$ top ranked accounts as spam accounts. Using these implementation parameters, we run our two samplers for four days to sample more likely spam accounts. After one month, we further examine whether those sampled accounts are suspended by Twitter.

6.3 Effectiveness of Hashtag Sampler and Friend Sampler

As seen in Table 2, Hashtag Sampler outputs 8,983 unique accounts to be likely spam accounts. Among them, 262 accounts have been suspended, and 4,665 others are output as spam accounts by the classifier. Thus, the hit count is 4,927 and the hit ratio is 0.5489, which implies that Hashtag Sampler could correctly collect one spam account by sampling less than two accounts.

Table 2: The effectiveness of Hashtag Sampler.

Item	N_{sus}	N_{mal}	N_{hit}	N_{sample}	H_r
Value	262	4,665	4,927	8,983	0.5489

Also, we further examine hit count and hit ratio by using each spammer's hashtags. As seen in Figure 10(a), over 40% spam accounts' hashtags can be used to collect over 100 spam accounts by sampling 500 accounts. This observation shows that Hashtag Sampler can effectively collect spam accounts by focusing on spammers' tastes. Also, we find that around 30% spammers' hashtags can not be used to correctly collect spam accounts. The reason is mainly because Twitter Search does not index every tweet due

[7]Since we could not obtain ground truth for those protected and nonexistent accounts output by our samplers, we do not count such accounts in N_{sample}.

to its resource constraints [3]. According to our observation, we could crawl very few (or even no) tweets by using those spammers' hashtags.

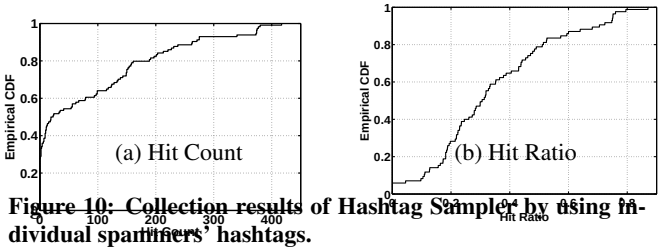


Figure 10: Collection results of Hashtag Sampler by using individual spammers' hashtags.

As seen in Figure 10(b), Hashtag Sampler could obtain reasonable hit ratios by using around 60% spammers' hashtags, which are higher than 0.3 (sampling 3 accounts will correctly collect 1 spam account).

We next show the evaluation results of the Friend Sampler. As seen in Table 3, Friend Sampler outputs 21,686 unique accounts as spammers. Among these accounts, 4,000 have been suspended, and 9,781 others are output as spam accounts by the classifier. Thus, the hit count is 13,781 and hit ratio is 0.6355.

Table 3: The effectiveness of Friend Sampler

Item	N_{sus}	N_{mal}	N_{hit}	N_{sample}	H_r
Value	4,000	9,781	13,781	21,686	0.6355

6.4 Diversity and Complementarity

Next, we analyze the diversity and complementarity of these two samplers. Essentially, we examine the number of spam accounts correctly sampled only by one algorithm, which can not be found by using the other one. If this number of each algorithm is high, it implies that these two samplers are very complementary. Thus they could be combined together to find more spam accounts. Specifically, to measure the diversity, we design a metric, named "Exclusive Ratio (E_r)", which is the ratio of the number of spam accounts that are exclusively sampled by one sampler (not sampled by the other one) to the total number of spam accounts sampled by this sampler.

As seen in Table 4, we can find that both two samplers can obtain relatively high exclusive ratios (over 77%). The ratio for Friend Sampler is even higher than 90%.

Table 4: Exclusive ratios between two samplers.

Algorithm	Hashtag Sampler	Friend Sampler
Exclusive Ratio	77.69%	90.57%

This observation shows that these two samplers are indeed complementary. And thus, they can be used together to collect more (likely) spam accounts.

As shown in Table 5, according to our collected dataset, the combination usage of these two algorithms could correctly collect 18,185 spam accounts. Among them, 4,249 accounts have been suspended by Twitter, and 13,936 other accounts are classified as spam accounts by the classifier. Thus, the hit ratio of combining these two algorithms is 0.6219. Compared with the dataset used for the purpose of building an effective machine learning classifier [16], which contains 355 manually identified spam accounts from 8,207 randomly crawled accounts (i.e., a hit ratio of only 0.04), this value of hit ratio is considerably high in terms of effectively crawling likely spam accounts in the huge Twittersphere.

Table 5: Result of combining two algorithms.

Item	N_{sus}	N_{mal}	N_{hit}	N_{sample}	H_r
Value	4,249	13,936	18,185	29,239	0.6219

7. LIMITATIONS AND FUTURE WORK

We acknowledge our manually identified spam accounts may contain some bias, and the machine-learning classifier we use to estimate the accuracy may not be absolutely accurate. However, it is challenging to obtain a perfect ground truth, and our strategies have been widely used in this line of studies [35, 19, 28, 40]. In addition, even though some values may vary according to different datasets, we believe that our major findings and insights are likely still valid independent of the datasets. It is possible that our advanced honeypots may also attract a few benign accounts' contacts. However, this highly depends on the goal of honeypots – trapping more spam accounts, or obtaining spam accounts only, for which we believe the former is more important. According to our data collection results, our advanced honeypots could trap *significantly more* spam accounts.

We note that our samplers are not designed to collect/cover all (types of) spammers in Twittersphere. In addition, we note that the number of collected spam accounts by our samplers is restricted by the number of inputs, e.g., hashtags and famous accounts. Our result is also limited by the Twitter Search API: one request could only obtain the recent 1,500 search results, and not even to mention not all tweets are indexed by Twitter Search. Thus, if our samplers are implemented by Twitter (without many restrictions), they could find more spam accounts.

8. CONCLUSION

In this paper, we perform a deep measurement study on how some Twitter spammers choose their spamming targets, through building social honeypots with diverse social behavioral patterns. Based on the analysis of spammers' tastes, we provide principled guidelines for building more effective (attractive) social honeypots. Furthermore, we design two new lightweight and effective samplers to guide the active sampling of more likely Twitter spam accounts, which we believe is a great complement to the passive social honeypot approach.

9. ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation (NSF) under Grant No. CNS-1218929. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

References

[1] Capture HPC. https://projects.honeynet.org/capture-hpc.

[2] Get rid of DM spam on Twitter. http://seanmalarkey.com/rid-dm-inbox-spam-auto-dms-mafia-invites-twitter.

[3] I'm Missing from Search. http://support.twitter.com/groups/32-something-s-not-working/topics/118-search/articles/66018-my-tweets-or-hashtags/-are-missing-from-search-known-issue.

[4] Instagr – Photo sharing for your iphone. http://instagr.am/.

[5] Seven Things I Learned to Bait Twitter Spammers. http://www.experiencetheblog.com/2011_07_01_archive.html.

[6] So Much Twitter Spam. http://blog.sysomos.com/2011/04/07/so-much-twitter-spam/.

[7] The Twitter Rules. http://help.twitter.com/entries/18311-the-twitter-rules.

[8] Trending topics. http://support.twitter.com/entries/101125-about-trending-topics.

[9] Twiends. http://twiends.com/.

[10] Twitpic. http://twitpic.com/.

[11] Twitter Search. https://twitter.com/#!/search-home.

[12] Twitter spam reduction. http://blog.twitter.com/2010/03/state-of-twitter-spam.html.

[13] Why do I get spam followers? http://www.quora.com/Why-do-I-get-so-many-spam-followers-on-Twitter.

[14] S. Antonatos, I. Polakis, T. Petsas, and E. Markatos. A Systematic Characterization of IM Threats Using Honeypots. In *Proceedings of Proceedings of the Network and Distributed System Security Symposium (NDSS'09)*.

[15] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting Spammers on Twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conffference (CEAS)*, 2010.

[16] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Goncalvess. Detecting Spammers and Content Promoters in Online Video Social Networks. In *ACM SIGIR Conference (SIGIR'09)*, 2009.

[17] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation., 2003.

[18] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The Socialbot Network: When Bots Socialize for Fame and Money. In *Proceedings of 2011 Annual Computer Security Applications Conference (ACSAC)*, 2011.

[19] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *Annual Computer Security Applications Conference (ACSAC'10)*, 2010.

[20] G. Dunlap, S. King, S. Cinar, M. Basrai, and P. Chen. Revirt: Enabling intrusion analysis through virtual-machine logging and replay. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI'02)*, 2002.

[21] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and Characterizing Social Spam Campaigns. In *Proc. of ACM SIGCOMM IMC*, 2010.

[22] S. Ghosh, B. Viswanath, F. Kooti, N.K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K.P. Gummadi. Understanding and combating link farming in the twitter social network. In *Proceeding of WWW*, 2012.

[23] C. Grier, K. Thomas, V. Paxson, and M. Zhangy. @spam: The Underground on 140 Characters or Less. In *ACM Conference on Computer and Communications Security (CCS'10)*.

[24] L. Invernizzi, P. Comparetti, S. Benvenuti, C. Kruegel, M. Cova, and G. Vigna. EVILSEED: A Guided Approach to Finding Malicious Web Pages. In *IEEE Symposium on Security and Privacy (Oakland)*, 2012.

[25] D. Irani, M. Balduzzi, D. Balzarotti, E. Kirda, and C. Pu. Reverse Social Engineering Attacks in Online Social Networks. In *Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, 2011.

[26] J. John, F. Yu, Y. Xie, A. Krishnamurthy, and M. Abadi. Heat-seeking Honeypots: Design and Experience. In *Proceedings of the 20st WWW*, 2011.

[27] C. Kreibich and J. Crowcroft. Honeycomb: Creating intrusion detection signatures using honeypots. In *Proceedings of the 2nd Workshop on Hot Topics in Networks (HotNets'03)*, 2003.

[28] K. Lee, J. Caverlee, and S. Webb. Uncovering Social Spammers: Social Honeypots + Machine Learning. In *ACM SIGIR Conference (SIGIR)*, 2010.

[29] K. Lee, B. Eoff, and J. Caverlee. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. In *5th International AAAI Conference on Weblogs and Social Media (ICWSM).*, 2012.

[30] A. Matwyshyn, A. Keromytis, A. Cui, and S. Stolfo. Ethics in security vulnerability research. In *IEEE Security and Privacy, Vol. 8, No. 2*, 2010.

[31] A. Moshchuk, T. Bragin, S. Gribble, and H. Levy. A crawler-based study of spyware on the web. In *Proceedings of the 13th Annual Symposium on Network and Distributed System Security (NDSS'06)*, 2006.

[32] N. Provos. A virtual honeypot framework. In *Proceedings of the 13th USENIX Security Symposium, (USENIX'04)*, 2004.

[33] J. Song, S. Lee, and J. Kim. Spam Filtering in Twitter using Sender-Receiver Relationship. In *Proceedings of the 14th RAID*, 2011.

[34] V. Sridharan, V. Shankar, and M. Gupta. Twitter Games: How Successful Spammers Pick Targets. In *Proceedings of 28th ACSAC*, 2012.

[35] M. Steyvers and T. Griffiths. Probabilistic topic models. In *Handbook of Latent Semantic Analysis*, 2007.

[36] G. Stringhini, S. Barbara, C. Kruegel, and G. Vigna. Detecting Spammers On Social Networks. In *Annual Computer Security Applications Conference (ACSAC'10)*.

[37] K. Thomas, C. Grier, V. Paxson, and D. Song. Suspended Accounts in Retrospect:An Analysis of Twitter Spam. In *Internet Measurement Conference (IMC'11)*.

[38] A. Wang. Don't follow me: spam detecting in Twitter. In *Int'l Conferene on Security and Cryptography (SECRYPT)*, 2010.

[39] Yi. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated Web Patrol with Strider HoneyMonkeys. In *Proceedings of the 13th Annual Symposium on Network and Distributed System Security (NDSS'06)*, 2006.

[40] C. Whittaker, B. Ryner, and M. Nazif. Large-Scale Automatic Classification of Phishing Pages. In *Proceeding of 17th NDSS*, 2010.

[41] C. Yang, R. Harkreader, and G. Gu. Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. In *Proceedings of the 14th RAID*, 2011.

[42] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing Spammers' Social Networks For Fun and Profit – A Case Study of Cyber Criminal Ecosystem on Twitter. In *Proceedings of the 21st International World Wide Web Conference (WWW)*, 2012.

[43] S. Yardi, D. Romero, G. Schoenebeck, and D. Boyd. Detecting spam in a Twitter network. In *First Monday, 15(1)*, 2010.

[44] V. Yegneswaran, J. Giffin, P. Barford, and S. Jha. An architecture for generating semantics-awarex signatures. In *Proceedings of the 14th USENIX Security Symposium (USENIX'05)*, 2005.

[45] J. Zhang, P. Porras, and J. Ullrich. Highly predictive blacklisting. In *The 17th USENIX Security Symposium (USENIX Security)*, 2008.