Analyzing Spammers of Social Networks using Honeypot- A Case Study of Microblogging of China

Yi Zhou, Kai Chen, Li Song, Xiaokang Yang
Department of Electronic Engineering
Institute of Image communication and Information Processing
Shanghai Jiaotong University, Shanghai, China
E-mail: {zy_21th, kchen, song_li, xkyang}@sjtu.edu.cn

Abstract— **in this paper, we conducted a case study of spammers in two popular Chinese microblogging networks: Sina Weibo (weibo.com) and Tencent Weibo (t.QQ.com). To start with comparison of various design choices for microblogging bot, we implemented and built 50 honeypot which can post/comment/fans/follow/log automatically. Next we launched a 7 month experiment using honeypot to attract spammers and log their information. Then we analyzed the users captured by the honeypots and manually picked spammers out of legitimate users. Lastly, with these spammer samples, we investigated their features such as social information, activity, account age and spamming strategy, and figured out several distinguishing characteristics of spammers on these two social network communities, which may be helpful to the future study on automatic detection of social spammers.**

Keywords- Social Network; China Microblogging; Spammer; Honeypot.

I. INTRODUCTION

Microblogging, such as Twitter.com, Sina Weibo, is a new way of communication in which huge information are shared and discussed in short messages. Unfortunately, this wealth of information also attracted the interests of spammers to spread advertise to generate sales, disseminate pornography, viruses, phishing and so on.

In 2008, a Sophos probe showed that 41% of the Facebook users are willing to reveal all potential identity thieves, which highlights dangers of irresponsible behavior on social networking sites [1]. Jagatic et al. found that phishing attempts have larger rate of success if the spammer uses stolen information from victims' friends in social networks [2]. Baltazar et al. reported the existence of botnets targeting social networks [3]. Gao et al. confirmed the existence and studied the characteristics of large-scale social spam campaigns in Facebook [4]. Grier et al. presented a characterization of spam on Twitter [5]. They find that 8% of 25 million URLs posted to the site point to phishing, malware, and scams listed on popular blacklists. Twitter is a highly successful platform for coercing users to visit spam pages, with a click through rate of 0.13%.

There are basically two main focuses of related study: spam messages and spam accounts (spammers).

Spam detection has applied itself in many aspects on social media, such as emails, blog comments, product reviews, tagging systems, and social network websites. Heymann et al. did a survey showing that the three main anti-spam strategies commonly used in practice are: Identification-based, Rank-based and Interface or Limit-based [6]. Also the methods of analysis have divided into two main categories: statistical analysis relying on statistical features of spam messages, and content analysis focusing on content features of them. Huang et al. proposed some features and analyzed the features of comments with statistical methods [7]. Statistics show that comment spam would be filtered out effectively with high precision and recall. Yin et al. employ content features, sentiment features and contextual features of documents, using a supervised learning approach to identify online harassment [8]. While singular spam message is not easily detected by setting up classifier, major efforts of research have been spent on studying spam messages in group, or spam campaigns. Mukherjee et al. purposed to detect spam reviews in group [9]. They used frequent pattern mining to find candidate groups, computed spam indicator values, and ranked the results using SVM. Gao et al. also proposed to reconstruct spam messages on social network websites into campaigns for classification [10].

Another way of detecting spam is finding the spammers who are the source of the spam messages. This is especial effective in social network websites because their socially interactive nature. Webb et al. studied social network spam using social honeypots [11]. In their experiment, 51 profiles were created on MySpace, and these social honeypots were contacted by 1,570 spam bots over a five-month period. Stringhini et al. [12] and Lee et al. [13] also used honeypots and machine learning techniques to detect spammers on Facebook and Twitter.

Even though the problems of spammer in western social networks, such as Twitter, have been well studied, characteristics of Chinese microblogging networks have not been. To our best knowledge, this is the first spammer report of Chinese microblogging networks. The main contributions of this paper are the following:

- We compared various design choices for microblogging bot, and built 50 honeypots in two popular Chinese microblogging networks: Sina Weibo (weibo.com) and Tencent Weibo (t.QQ.com). These honeypot can not only receive but also post and follow automatically.
- We launched a 7 month experiment to attract spammers and log their information, and analyzed the spammers and investigated their features.

II. DESGIN OF HONEYPOT

Microblogging honeypot runs without human inspection and log information of its fans. Each honeypot has its own personal information different with others' which contains information such as name, birthday, and location. The goal of honeypot is to attract as many social spammers as possible. Our honeypot is not a "Silent" honeypot like [13] which can only read information. Our honeypots can automatically follow random users in order to join their friend network. Besides that, the honeypot profiles post microblogs with the help of the control program, just like an active user does, so that they could attract spammers who choose their targets from public timeline.

Microblogging honeypot is a kind of web bot. There are several ways to build web hot, however, most social networks sites have authentication to prevent unwelcomed robot searching behavior. Others have time and thread limitations to client users. These all have bring new challenges to the web bot technology.

There are two main type of ways to build microblogging honeypot, one is using API provide by social networks, the other is programming to simulate browser. Since API provide by even social networks themselves cannot search microblogs by keywords in Sina Weibo and Tencent Weibo, API way are not used in our work. As shown in Figure 1. There are following types of design choices to build microblogging honeypot using browser simulation:

- Programming from scratch to simulate browser
- JavaScript Engine + Browser Kernel
- Browser Plugin
- Browser Automaton Framework

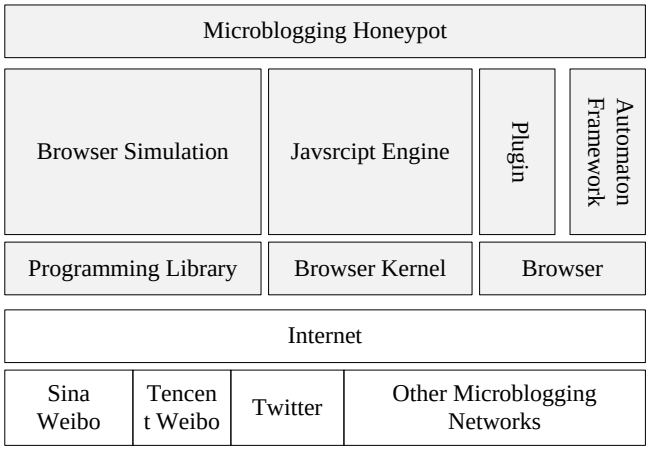

Figure 1, Architecture of microblogging honeypot using browser simulation

A. *Programming from scratch to simulate browser*

In this design, honeypot are implemented by sending request information and parsing response from server. Any programming language can be used, Python and other high-level language are good choice due to their many easy-to-use libraries. Although it is quiet convenient using Python to implement without pay attention to "low level" programing such as socket programming, there are still many challenges works to do:(1) Manually trace, analyze and parse the POST & GET message in "Browser-Server" Communication. (2) Manually handle JavaScript. Authentication is always difficult to simulate.

B. *JavaScript Engine + Browser Kernel*

Browser kernel (such as WebKit [14]) and JavaScript Engine (such as PhantomJS [15]) can be used to solve the problem of handle JavaScript manually. So we can program over WebKit and provide more complicated and controlled behavior such as login and search on higher-level. And WebKit will manage response & request itself. PhantomJS can to perform DOM handling, CSS selector, JSON.

C. *Browser Plugin*

It's still needs large effort to create a bot based on PhantomJS and WebKit. Lots of programming work should be done to ensure the performance of DOM handling and efficiency of JavaScript engine.

Chickenfoot [16, 17] programming system embedded in the Firefox web browser. Without examining source code of one page, end-users can automate, customize and integrate web applications. They don't need to know how to manage a suitable URL, how to set up the cookies. Once you've login on a web site (Usually by input user name, password and a button click), the browser will take care of them itself.

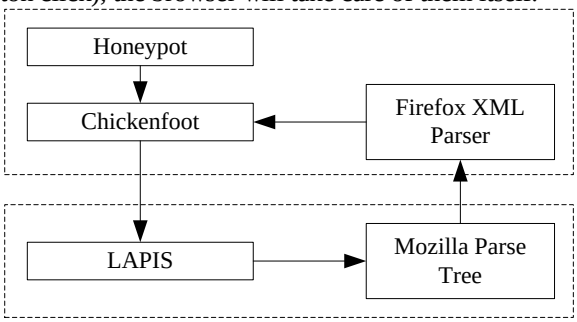

Figure 2, Architecture of chickenfoot.

D. *Browser Automaton Framework*

Generally, WebDriver Automation [18] is the enhance solution similar to Chickenfoot. In this design, Cookies, authentication, session identifiers, user agents, client-side scripting, and proxies are all looks different when you build a honeypot upon a web browser. Ajax applications such as Social Network & Google mail make situation in traditional approaches worse.

WebDriver Automation is a closely binding of browsers like IE, Firefox and Chrome which provided unified interface of WebDriver. Here, everything on web page is regarded as "Web Element" from end-user programmer. Crawler can simulate the human being to interact with browser, no longer to simulate the browser itself. So it will change the crawler behavior. It will no longer notice about the JavaScript behind the buttons. The authentication process will simply be filling in the input boxes on the login page then click a login button like a real human user.

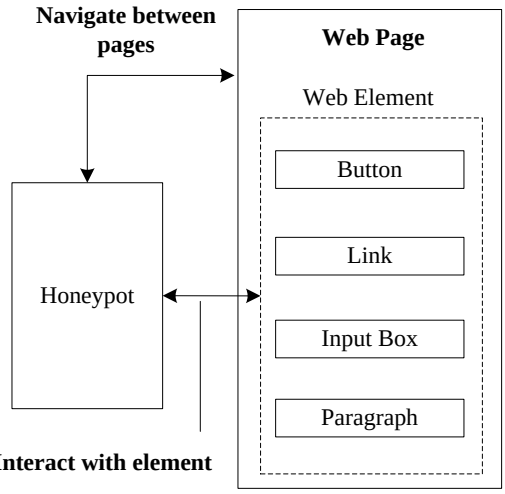

Figure 3, Architecture of WebDriver Automation.

E. *Comparsion and Summarization*

Table-1 is the comparison between above four ways in following metric.

- *Graphic View*: Executing process of honeypot can be visited or not.
- *Implement Level*: Low-User handles "Request-Response" in Http manually; Medium-User handles HTML & DOM manually; High-User controls abstract DOM element in Browser without looking into how it works.
- *Simulate Behavior*: Crawler is trying to simulate browser's behavior or higher level, user's behavior.
- *Remote/Local*: honeypot can run on remote server or only locally.
- *HTML Inspections*: End-user programmer is required to read HTML of web page in advanced or not.

Table-1, Comparison of four ways to build honeypot

	A	**B**	**C**	**D**
Graphic View	No	No	Yes	Yes
Implement Level	Low	Medium	High	High
Simulate Behavior	Browser	Browser	User	User
Remote/ Local	Local	Local	Local	Remote +Local
HTML Inspection	Yes	Yes	No	Yes
Performance	Fast	Fast	Slow	Slow
Cookies Handling	Manual	Manual	Auto	Auto+ Manual
Memory Consumption	Low	Low	High	High

In our work, the way of browser automaton framework is used to build honeypots, because it's easy for programmer to coding and easy to extend to distribute network environment.

III. EXPERIMENT

A. *Honeypot Profiles and Experiment Settings*

We build 50 honeypot profiles on Weibo.com and QQ.com, 25 for each platform. These profiles run 5 times a day. During every turn, honeypots post a fixed number of microblogs and follow a fixed number of users. In order to avoid IP verification, each profile runs at different time of the day.

The experiment last from September 2011 to April 2012. The honeypot profiles monitored their fan lists, logged all their fans homepage addresses and then stored these addresses into a database.

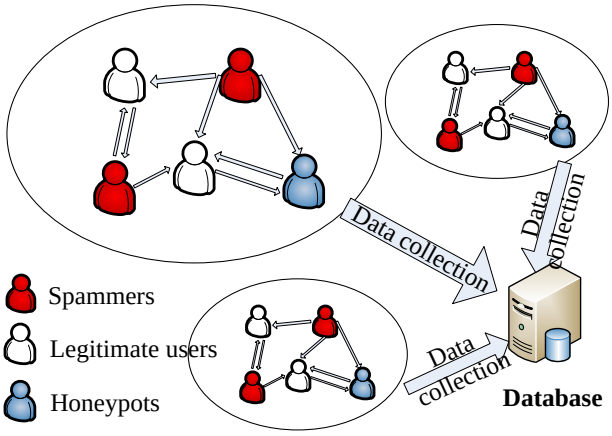

Figure 4, Overall Framework of Our Experiment

B. *Experiment Statistic*

Our honeypot profiles posted 23,666 microblogs, followed 9514 users and finally attracted 1,810 users by the end of the experiment. We collected some running features as shown in Table 2.

Table 2 Honeypot Running Record

	Weibo.com	QQ.com	Total
Microblogs	12,631	11035	23666
Followings	3552	5962	9514
Followers	517	1293	1810
Friend number	279	608	887
Microblogs number to gain a fan	24.43	8.53	13.08

- *Microblogs*: the number of microblogs posted by the honeypot profiles in the process of the experiment.
- *Followings*: Number of users the honeypot followed.
- *Followers*: Number of users following the honeypots, also known as "fans".
- *Friend number*: number of users maintaining a bi-direction following relationship with the honeypots.
- *Microblogs number to gain a fan*: average number of microblogs posted by the honeypots per fan.
- *UFBR (User follow back rate)*: the proportion of users who follow the honeypots back after being followed.

- *NFR (Non-friend follower ratio)*: the proportion of non-friend followers among the total followers.

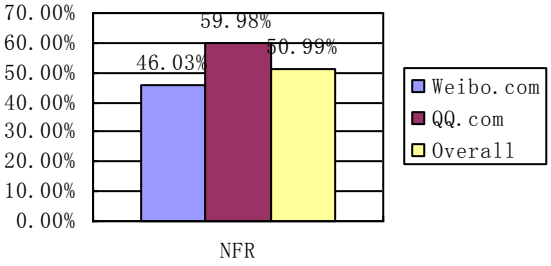

Figure 5, Non-friend Follower Ratio

As we can see from Table 2 and Figure 5, our honeypot profiles were efficient in attracting microblogs users. By following users and posting microblogs actively, these honeypot profiles managed to reach their friend network out to cover a quite large number of users. Comparing the two different social communities, we are convinced that it is easier for the honeypots profiles to gain popularity on QQ.com than on Weibo.com. Honeypots on Weibo.com have to post 24.43 microblogs to gain a single fan which is almost three times of QQ.com. Moreover, non-friend follower ratio of QQ.com (52.98%) is also higher than that of Weibo.com (46.03%), which means QQ.com users are more initiative in following strangers and establishing relationships.

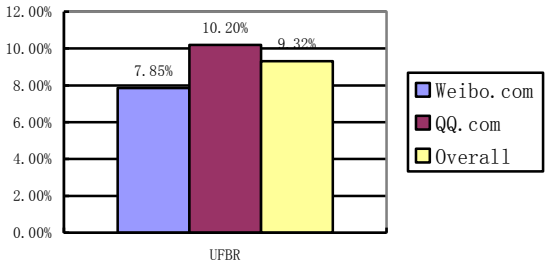

Figure 6, User Follow Back Rate

Figure 6 shows the UFBR (user follow back rate) of the two communities. The higher UFBR (10.20% compared with 7.85%) of QQ.com proves that users on QQ.com are more likely to establish friend relationships with strangers.

It should be mentioned that our honeypot profiles ran in a similar way with some spammers: We follow a large number of users in order to gain more fans so that we can be seen by more potential spammers. Some spammers also do this so as to gain more followers and expand their spamming scope. The overall UFBR of 9.32% shows the effectiveness of gaining popularity through this way. This can also, to some extent, reveal the vulnerability of social network communities in defending against spamming activities.

C. *Follower Analysis*

The experiment resulted in 1810 user being captured by our honeypot profiles, 517 on Weibo.com and 1,293 on QQ.com. As mentioned above, their homepage addresses have been collected and stored into a database. We scanned the database and *manually* checked all the users in order to put them into two categories: spammers and non-spammers.

We define users who post microblogs containing the following three kinds of URLs as *spammers*.

- URLs pointing to web pages containing advertisement or sales information;
- URLs whose destinations are phishing sites;
- URLs containing malwares.

We browsed through the users' homepages and investigated the URLs in their microblogs. Since both social network communities enables various kind of multi-media microblogs which represented by URLs, we have to manually check all the URLs so as to tell malicious URLs from legitimate ones.

The final result is shown in Table 3 as follows.

Table 3 Users captured by honeypots

	Weibo.com	QQ.com	Total
Spammers	114	567	681
Non-spammers	403	726	1,129
Total	517	1,293	1,810

Among the 1810 users attracted, 681 (37.62%) are labeled as spammers, which reveals that our honeypot profiles are not only effective in attracting microblogs users but also effective in capturing malicious users such as spammers.

Now let's look into the follower distribution on these two social network communities. As is shown in Figure 7, 22% of the total samples on Weibo.com are spammers while on QQ.com this number increases to 44%, which is double of the Weibo.com.

This fact has proved our previous guess that spammers on QQ.com are easier to survive and are more commonly seen than spammers on Weibo.com.

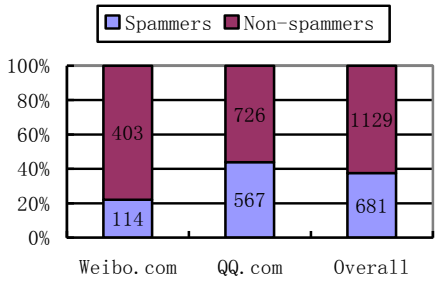

Figure 7, Follower Distribution

Moreover, we count the number of spammers among the honeypots' non-friend followers, in order to know how many

spammers followed our honeypots initiatively, since our honeypots don't follow back users that followed us.

The result is shown in Figure 8 Most of the spammers don't have a friend relationship with our honeypots, in other word, they follow the honeypots initiatively. Since our honeypots choose users randomly from public timeline, the less proportion of friend spammers on QQ.com may suggest that spammers on QQ.com are less active than those on Weibo.com so that they are less likely to be chosen by the honeypots.

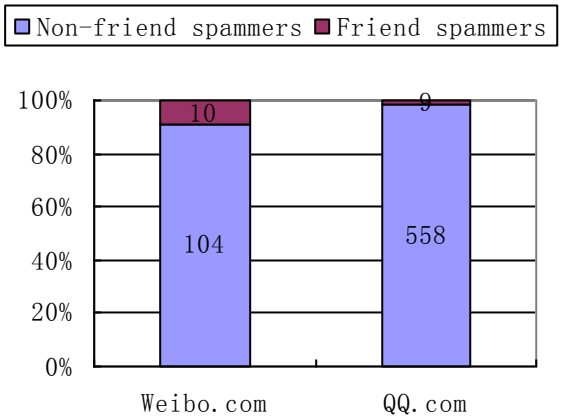

Figure 8, Spammer Distribution

IV. SPAMMER ANALYSIS

After picking out all the spammers in our samples collected by honeypot profiles, we made a deeper investigation on these social spammers.

For each spammer, we manually visited its homepage, scanned all its microblogs to make clear the spamming strategy, and logged its social information such as following and followers. We collected the following kinds of data of every spammer for the further analysis:

- *Followings*: number of users the spammer follow
- *Followers*: number of users following the spammer
- *Microblogs number*: number of microblogs posted by the spammer
- *Account age*: the time the spammer account has existed which is calculated since its first microblogs was posted
- *Spamming strategies*: whether the spammer acts aggressively or cautiously, specifically, whether the spammer post non-spam microblogs.

By analyzing the above information, we figured out some distinguishing characteristics of the spammers on Weibo.com and QQ.com. Besides that, we compared these characteristics of spammers to these of legitimate users to decide whether they really differ.

A. Dataset

In order to do the comparison we built two datasets for each social network communities: the spammer dataset and the legitimate user dataset.

The spammer dataset contains the spammers captured by our honeypot profiles. By the end of the experiment, some spammers on Weibo.com have been suspended by the service provider and their information is not available. We filtered out all such spammers and form the final spammer dataset.

For the legitimate user dataset, we randomly chose 50 legitimate users from the public timeline (manually checked non-spammers) for each community and logged their following and follower numbers, account age and microblogs number.

The detail of the datasets is shown in Table 4

Table 4 Spammers and legitimate users dataset

	Spammers	Legitimate users
Weibo.com	81	50
QQ.com	366	50

B. Follwoings/followers Ratio

As we mentioned above, one of the most common way for spammers to gain popularity and get more spam targets is to follow a huge number of users and wait for them to follow back. This kind of behavior is abnormal because most legitimate users don't act like this.

As a result, most spammers tend to follow a lot of users while having relatively few followers. Their ratio of following to followers should be quite large. On the contrary, this number should be quite low for legitimate users. We got this feature by calculating followings/followers of both spammers and legitimate users. The results are shown in Figure 9.

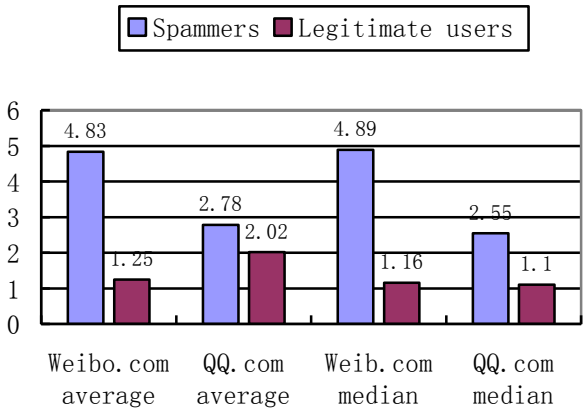

Figure 9, Followings/followers Ratio

It is clear that there exists a big gap between the spammers and legitimate users in followings/followers ratio. On Weibo.com the ratio of spammers is almost 4 times of the ratio of legitimate users, which means that the ratio is a distinguishing characteristic of social spammers. On the other hand, the gap between average ratios on QQ.com is not so distinguishing. It's because some of the legitimate users on QQ.com also acted in a "spammer" way for the sake of gaining popularity. As we refer the median, which is more reliable, the ratio of spammers doubles the ratio of legitimate

users. Thus, we conclude that the ratio of followings and followers is a distinguishing spammer characteristic.

C. User Activity

The user activity can be measured by the frequency he or she takes part in social network activities including posting, commenting and communicating with others using private messages.

In our case, we define user activity as the number of microblogs a user posts per day. By calculating *the ratio of total microblogs and account age*, we get the average number of microblogs posted by a user per day. For active users, this number would be bigger than the less active ones.

We assume that spammers would prefer to be more active than legitimate users since they want their posts to be seen more often by their targets.

Figure 10 shows the user activity of spammers and legitimate users on both social communities, represented by their numbers of microblogs posted per day.

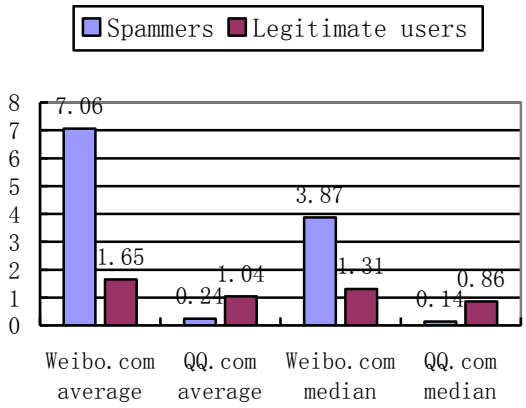

Figure 10, Microblogs Posted per Day

The result on Weibo.com proves our assumption that spammers are much more active than legitimate users.

However, spammers on QQ.com are much less active than legitimate users. They seem to be very cautious so as to avoid verification of the service provider. In spite of this fact, user activity can still be view as a distinguishing spammer characteristic of which Weibo.com spammers have high values and QQ.com spammers have low values.

D. Spamming Strategies

Different spammers will apply different spamming strategies. Some prefer to act more aggressively so as to influence more victims while others are more cautious in order to avoid being suspended.

We investigated how spammers publish spams. We define aggressive spammers as those who don't post any non-spamming microblogs and cautious spammers as those who mix spams with ordinary microblogs.

Figure 10 shows the percentage of two kinds of spammers on both social networks.

Cautious spammers are more than aggressive spammers on both social network communities which indicates that the aggressive ones are more likely to be suspended by the service provider.

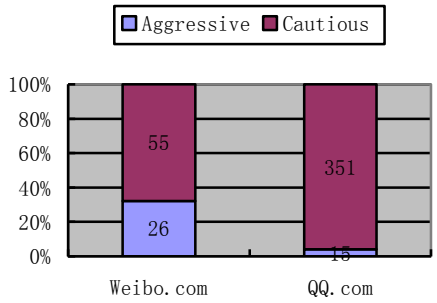

Figure 10, The Number of Two Kind of Spammers

We next investigated the account age of the two kind of spammers, which is shown in Figure 11.

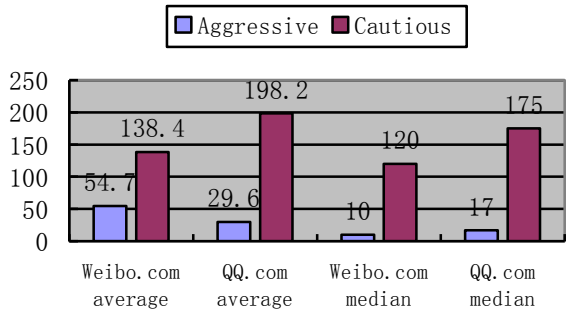

Figure 11, Spammers' Account Age

Account age of cautious spammers is much longer than that of aggressive spammers which indicates that cautious spammers can live and distribute spams for a longer time and thus make a bigger influence.

We are convinced that the aggressive spammers are weaker in front of the service provider's anti-spam verification. They are more likely to be suspended and thus have a relatively short account age. Spammer controllers have to register new accounts once the current ones are suspended. This may explain why the aggressive spammers have short account ages. On the other hand, cautious spammers are less likely to be found out and suspended. They mix spams with ordinary microblogs and sometimes have a relatively low activity. They may have a long time influence on its victims and raise challenge for the anti-spam mechanism.

E. Summary of Spammer Analysis

We set up 50 honeypot profiles on Weibo.com and QQ.com to attract users and recorded their information. We picked out spammers from legitimate users by manually checking every captured user's microblogs content. We built a spammer dataset for each social network community using these spammer accounts and a legitimate user dataset as well. After that, we manually analyzed several features of the two

user classes and made a comparison on these features. We found them useful in distinguishing spammers from legitimate users.

The followings are some conclusions about spammers on Weibo.com and QQ.com.

- The following/follower ratio of spammers is usually higher than legitimate users. They tend to follow a large amount of users in order to gain popularity but always have relatively few followers.
- There exists a big gap between the average numbers of microblogs posted per day of these two classes. On Weibo.com, spammers post quite a lot microblogs every day, much more than legitimate users while on QQ.com spammers post far less microblogs than legitimate users. This is mainly due to the different strategies taken by spammers on these two platforms.
- More spammers choose a cautious spam posting pattern. They mix spam microblogs with ordinary ones so that they can avoid the anti-spam verification taken by the service providers.
- Aggressive spammers are more likely to be detected so they tend to have a shorter life while cautious spammers can live much longer and have a deeper influence on the network. The latter kind of spammers may become the trend of social network spammer.

V. CONCLUSION

In this paper, we compared various design choices for microblogging bot, and built 50 honeypots in two popular Chinese microblogging networks: Sina Weibo (weibo.com) and Tencent Weibo (t.qq.com). And we launched a 7 month experiment to attract spammers and log their information, and analyzed the spammers and investigated their features.

ACKNOWLEDGMENT

The work is partially supported by the National Grand Fundamental Research 973 Program of China (Grant No.2010CB731406, 2010CB73140), National Natural Science Foundation of China (Grant No. 61129001), and Shanghai Science and Technology Committees of Scientific Research Project (Grant No. 11dz1505502).

REFERENCES

[1] Sophos Facebook Id probe. http://www.sophos.com/en-us/press-office/press-releases/2007/08/facebook.aspx.

[2] T.N. Jagatic, N.A. Johnson, M. Jakobsson, and T.N. Jagatif. Social phishing. Comm. ACM, 50(10):94–100,2007.

[3] J. Baltazar, J. Costoya, and R. Flores. Koobface: The largest web 2.0 botnet explained. 2009.

[4] H. Gao, J. Hu, C.Wilson, Z.Li, Y. Chen, and B.Y. Zhao. Detecting and characterizing social spam campaigns. In Proc. of the 10th annual conference on Internet measurement. 2010.

[5] C. Griery, K. Thomas, V. Paxsony, and M. Zhang. @spam: The Underground on 140 Characters or Less. In Proc. of the 17th ACM conference on Computer and communications security. 2010.

[6] P. Heymann, G. Koutrika, H. Garcia-Molina. Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges. IEEE Internet Computing 11(6), 36–45 (2007).

[7] C. Huang, Q. Jiang, and Y. Zhang. Detecting Comment Spam through Content Analysis. In Proc. of the 11th International Conference on Web-Age Information Management. 2010.

[8] D. Yin, Z. Xue, L. Hong, B.D. Davison, A. Kontostathis, and L.Edwards. Detection of harassment on web 2.0. In Proc. of Content Analysis in Web 2.0. 2009.

[9] A. Mukherjee, B. Liu, J. Wang, N. Glance, and N. Jindal. Detecting Group Review Spam. In Proc. of the 20th World Wide Web Conference. 2011.

[10] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary. In Proc. of the 18th ACM Conference on Computer and Communications Security. 2011.

[11] S. Webb, J. Caverlee, and C.Pu. Social honeypots: Making friends with a spammer near you. In Proc. of Conference on Email and Anti-Spam. 2008.

[12] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In Proc. of the 26th Annual Computer Security Applications Conference. 2010.

[13] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots + machine learning. In Proc. of the 33rd International ACM SIGIR Conference. 2010.

[14] WebKit Open Source Project. http://www.webkit.org

[15] PhantomJS Open Source Project. http://www.phantomjs.org

[16] Michael Bolin, Matthew Webber, Philip Rha, Tom Wilson, and Robert C. Miller. "Automation and Customization of Rendered Web Pages." ACM Conference on User Interface Software and Technology (UIST), 2005, pp 163-172.

[17] Michael Bolin. End-user Programming for the Web. MEng thesis, Massachusetts Institute of Technology, June 2005.

[18] WebDriver Selenium Open Source Project. http://code.google.com/p/selenium.