An Empirical Study of Clustering Behavior of Spammers and Group-based Anti-Spam Strategies

Fulu Li

The Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA USA 02139
Email: fulu@mit.edu

Mo-Han Hsieh

Engineering Systems Division
Massachusetts Institute of Technology
Cambridge, MA USA 02139
Email: mohan76@mit.edu

ABSTRCT

We conducted an empirical study of the clustering behavior of spammers and explored the group-based anti-spam strategies. We propose to block spammers as groups instead of dealing with each spam individually. We empirically observe that, with a certain grouping criteria such as having the same URL in the spam mail, the relationship among the spammers has demonstrated highly clustering structures. By examining the spam mails gathered in a seven-day period, we found that if a spammer is associated with multiple groups, it has a higher probability of sending more spam mails in the near future. We also observed that the spam mails from the same group of spammers often arrive in burst and a very small fraction of the active spammers actually accounted for a large portion of the total spam mails.

Based on our findings, we proposed a group-based anti-spam framework. The preliminary results show that our approach can be used as a complementary tool for existing anti-spam systems to more efficiently block organized spammers.

1. INTRODUCTION

With the popularity of the Internet, Email has become an indispensable part of people's daily life. According to IDC (a leading market intelligence firm), the global *daily* email traffic will reach 35 billion in 2005, up from 9.7 billion in 2000 [7]. However, the increase in the worldwide use of email comes with an overwhelming increase in spam mails. It is hard to give a precise definition on what is a spam mail. In short, a spam mail is an *unsolicited*, unwanted bulk/commercial email that endangers the very existence of the email system with massive and uncontrollable amounts of messages [10].

Different studies have shown that spam mails account for more than 50% of all Internet emails. The cost of spam mails consists of several components: the loss of productivity (as people have to spend time on spam), the cost of bandwidth wasted by spam, the cost of storage and network infrastructures, etc. It is no surprise that the projected worldwide spam cost will reach almost 200 billion US dollars in 2007 with roughly 50 billion daily spams according to Radicati Group (a leading market research firm).

To find better anti-spam strategies, we have to better understand the motives of the spammers. Most spams take the form of advertising or promotional materials, among which roughly half of all spam mails are related to money, debt reduction plans, getting-rich-quick schemes, gambling opportunities, one third of spam mails are porn-based, 10% of spam mails are health-related, and the rest of them cover a variety of topics [7,10].

With the staggering amounts of daily spams, it is hard to imagine that those spammers acted *individually*. In fact, it is widely believed that most of the spam mails are *directly* sent from a collection of compromised machines controlled by some spammers (the spammers may purchase the right to use compromised machines from worm writers/attackers) [11]. Due to the widespread of computer worms (e.g. Trojan horses etc.), a worm writer/attacker can crack a large collection of computers. The compromised computers are often called bots. The worm writer/attacker can sell those bots to some spammers for financial benefits. Some bots offer the possibility to open a SOCKS proxy on a compromised machine, which can then be used for spamming. With the help of thousands of bots, spammers can send massive volume of spams within a short period of time [3,4,5].

The motivation of this work is to understand and analyze the community behavior of spammers through a large collection of spam mails. The findings in this study may help to lay the foundation for group-based anti-spam strategies in the sense that if we can find some common behavior patterns for a group of bots, e.g., a botnet, we may be able to effectively block those spammers as groups instead of blocking them individually. To the best of our knowledge, this is the first time that spammers have been classified and categorized

CEAS 2006 – Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA

by their communities and it is the first time that group-based anti-spam strategies have been explored.

The rest of the paper is organized as follows: we discuss the related work on spam traffic analysis and anti-spam strategies in Section 2; we give a comprehensive analysis on the community behavior of spammers through a large collection of spam mails in Section 3; the group-based anti-spam strategies are presented in Section 4; we discuss some challenges and open problems for anti-spam techniques in Section 5; the conclusions and future directions are presented in Section 6.

2. THE RELATED WORK

In this section, we give a brief overview of the related work on spam traffic analysis and the state-of-the-art of the anti-spam approaches.

In [2], Gomes and Cazita gave a comprehensive study on the characterization of spam traffic in terms of workload variation, density, inter-arrival time distribution, email size distribution, temporal locality, etc., compared with non-spam emails. Their characterization reveals significant differences in the spam and non-spam traffic patterns. The interesting observation is that non-spam email transmissions are typically driven by *bilateral* social relationship while spam transmissions are usually a unilateral action, solely based on the spammer's will to reach as many recipients as possible.

In [1], Jung and Sit examines the use of DNS black lists for address-based filtering of spams. The basic idea is that once the IP address of a host engaged in spam delivery is identified, it will be registered in a centrally maintained database. The database is made available via the Internet DNS and the mail recipients can query this database using standard DNS lookup and refuse to accept mail from hosts that are listed in the DNS black list database. Their studies found that around 80% of spam sources that they identified are listed in some DNS black list and some DNS black lists appear to be well-correlated with others.

In general, the anti-spam strategies can be classified into four major categories [6]: mail server blacklists, filtering-based approaches, networking-based schemes, and computation-based methods.

In mail server blacklists, a database of the IP addresses of mail servers used by spammers is maintained to block spam mails. In filtering-based approaches, filers are installed according to a set of policies/rules/patterns to block spam mails. The major difference between mail server blacklists and filtering-based approaches is that the operation unit for mail server blacklists is the IP address of the mail server while the operation unit for the filtering-based approach is the spam mail.

There are several variants of filtering-based approaches such as signature-based filtering, Bayesian filtering, rule-based filtering, challenge-response filtering, etc. We refer interested readers to [6] for details of each of the filtering-based techniques. The well-known anti-spam tool SpamAssassin uses sophisticated rule-based filters to mark and block spam mails.

In networking-based anti-spam schemes, the basic idea is to slow down the spam sender once it is identified as a spam source. In one networking-based approach, the spam source is slowed down by modified TCP protocol to intentionally reduce its transmission window size, e.g., the technique used in anti-spam tools by Turntide Inc., which was recently acquired by Symantec Inc.

As the name suggests, the computation-based method is to force the spam sender to perform time-consuming *computations* before the receiver would accept it, such as the technique used by Hashcash Inc. (http://www.hashcash.org). Hashcash technique is supported in SpamAssasin as of version 2.70.

According to [3,5], about 30,000 new machines are compromised *daily*, and become bots. One of the most common usages of botnets is to launch massive spams. The recent study in [11] suggests that spammers are believed to use compromised machines, e.g., bots, to *directly* send massive spam mails. The work in this study is to understand and identify the community structure of those spammers and explore group-based anti-spam strategies to effectively block organized spammers.

3. COMMUNITY BEHAVIOR ANALYSIS

In this section we analyze the community behavior of spammers though a large collection of spam mails. Section 3.1 describes the spam source data. An overview of the spam traffic is presented in Section 3.2. Section 3.3 shows the clustering structures of the spammer communities based on different grouping criteria.

3.1 Spam Data Source

We obtained spam data from Jaeyeon Jung at CSAIL MIT. The spam mails are collected over almost one year period at a domain mail server (due to privacy concern we are not able to disclose the real domain name) in such a way that the *IP*

addresses of the spam sources are recorded when the spammer tries to establish the TCP connection with the domain mail server to transmit the spam mail. The IP address of the spammer recorded during the 3-way handshake should be the real IP address of the spammer *in most cases* even though there are rare scenarios in which the spammer could still use spoofed IP address during the 3-way handshake, e.g., using *unused* IP addresses within the same LAN where the spammer is located or using BGP hijacking to propagate some fake route entries with some *unused* IP addresses to the nearby ISPs [12].

As we discussed in Section 1 that spamming becomes more and more *distributed* and there is *less* spoofing of IP addresses due to the fact the attacker already uses several levels of *indirection* to hide his identity and controls thousands of compromised machines, e.g., the bots, to *directly* send massive spam mails. In the rest of the paper, we ignore the rare cases in which the IP addresses could still be spoofed during the 3-way handshake in that blocking potential spam mails from *unused* IP addresses will *never* cause false alarms.

The spam mail data contains the full mail header information and the full mail contents including the attachment files. The mail header information contains the *real IP address* of the spam source, the route information, the TCP SYN fingerprint, which can be used to identify the OS information of the spam source. In the following empirical study, we use one-week spam data collected from Sept. 9, 2005 to Sept. 16, 2005. We will expand our study to use the one-year spam dataset in the next stage. There are totally 86819 spam mails sent to the given domain mail server during the one-week period.

3.2 Overview of the Spam Traffic

Those 86819 spam mails collected from Sept. 9, 2005 to Sept. 16, 2005 are originated from 41874 distinct spam hosts, e.g., spammers (each spammer has a distinct IP address).

We use TCP SYN fingerprint information to identify the OS information of the spam host machine. Among the total spam mails that we examined, 74% of them are sent from Windows machines, around 10% of the spams are from Linux host, about 5% of the spams are from BSD and Solaris machines, and about 11% of the spams are not accounted for due to the lack of the OS information in the spam data. We observe that very few spams are from Macintosh machines (we only identified 5 spam mails from Mac machines) and the majority of spams are sent from Windows machines. This may be due to the fact that Windows machines are more vulnerable to virus attacks and they are more prone to be victims of worms, e.g., the bots, which can be used by the spammers to launch massive spam mails. To the best of our knowledge, till today there has been no widespread virus to affect Macintosh machines.

Figure 1 shows the number of spam mails per hour in the one-week time span. The x-axis is the time in hour and the y-axis is the number of spams. Roughly, the spam mail arrival rate ranges from 0 to 1300 spams per hour, with an average rate about 520 spams per hour. An interesting observation is that there is an *idle* period of roughly 20 hours without a single spam mail in the middle of the week. This may be due to the fact that the bots are rented by hour and the attacker may divert those bots to launch other attacks such as distributed denial of service (DDoS) attack, etc. If each spammer acts *individually*, it is *unlikely* that all of the spammers stopped sending spam mails during the same period of roughly 20 hours. Another possibility could be the domain mail server was down for that 20-hour period.

Figure 2 illustrates the complementary cumulative probability (CCDF) of the number of appearances of any specific spam source IP address. The x-axis is the number of appearances of any specific spam source IP address and the y-axis denotes the CCDF. Both x-axis and y-axis are plotted in log scale. As we can see that the majority of the spam source IP addresses appeared once or twice. The number of spam mails that a spammer sent during the given week ranges from 1 to as many as 446. More precisely, 68% of the spammers sent only one spam, 15% of them sent two spams, less than 2% of them send more than 10 spams for the given week. However, those less than 2% of the spammers accounted for 20% of the total spam mails during this 7-day period.

We use *traceroute* to find out that two of the most active spammers, 65.54.195.210 and 65.54.195.215 are Microsoft group server. We also observe that 216.37.127.157, 216.37.127.117, 216.127.157.97, 216.127.157.37 are possible spoofed source IP addresses even though we use 3-way handshake to record the source IP address. This is due to the fact that an attacker can use *unused IP addresses on the same LAN* to spoof its source IP address. We cannot *ping* any of the four IP addresses on the same LAN.

According to [6], over 95% of spam mails have URLs. We show the complementary cumulative probability (CCDF) of the number of appearances of the same URL in Figure 3. The y-axis is the complementary cumulative probability and the x-axis is the number of appearances of the same URL. As shown in Figure 3 that some URLs appeared only once, some URLs appeared between 10 and 100 times, and very few URLs appeared close to 1000 times among all the spam mails.

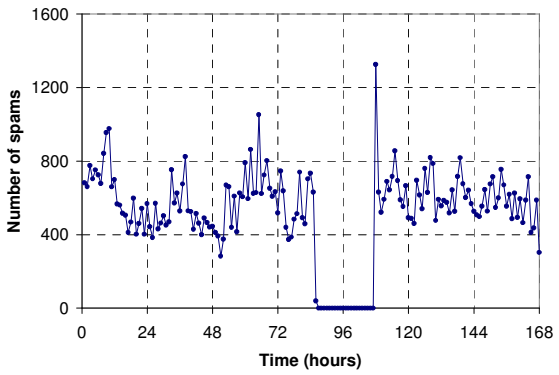

Figure 1: The number of spam mails per hour in the one-week time span.

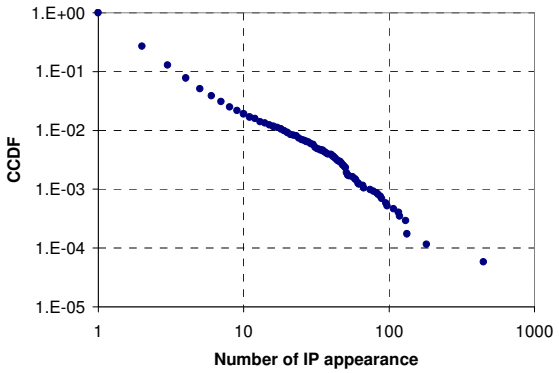

Figure 2: The CCDF of the number of appearances of the same IP address.

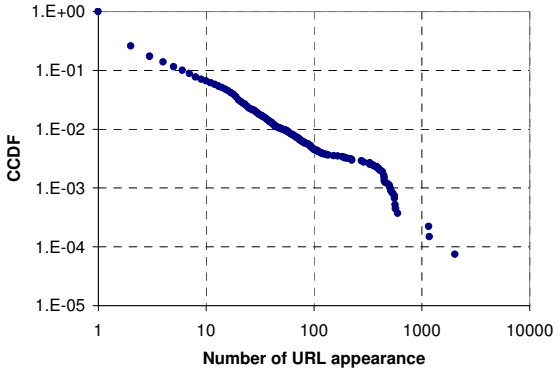

Figure 3: The CCDF of the number of appearances of the same URL.

3.3 Clustering Structures

In this section, we group the spammers in terms of the URLs, money amount, which appear in most of the spam mails.

Figure 4 shows the clustering structures of the spammers based on the URLs in the spam mails of Day 1. If the same URL appears in the spam mails from both source A and source B, then an edge is plotted to connect node A and node B, each of which is identified with a unique IP address. Clearly, the collected spammers *demonstrate highly clustering structures* based on the URL grouping. The number of members in each cluster ranges from 1 to 716.

According to [3], a typical botnet consists of several hundred compromised machines, which is in line with some cluster sizes observed in Figure 4. The major component with 716 spammers is further illustrated in Figure 5. An interesting observation is that the spam mail from the spammer at the pivoting (conjunction) point often comes earlier than spams from the spammers at those homogenous points further away from the center point in the cluster. Another key observation is that the more groups a spammer's IP address is associated with (due to multiple distinct URLs appeared in the spam mail from this spammer), the higher probability that more spam mails from this IP address will come in the near future. We hypothesize that those spammers at the pivoting points play a more important role in the botnet.

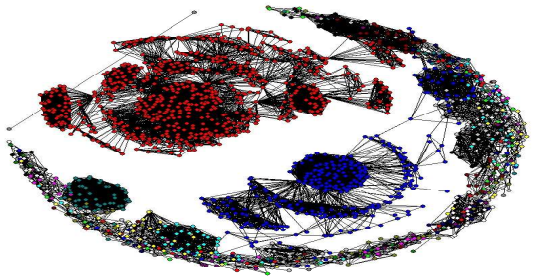

Figure 4: The clustering structure of the spammers based on the URLs in spam mails of Day 1.

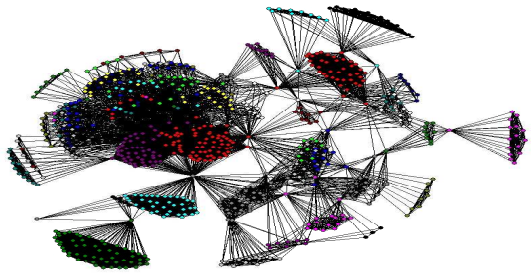

Figure 5: The major component of the clustering structure in Figure 4.

We calculate the correlation co-efficiency of the inter-arrival time of the spam mails from the spammers belonging to the same cluster according to the following equation:

$$\rho_k = \frac{1}{N-k}\sum_{i=1}^{N-k}\frac{(x_i - \bar{x})(x_{(i+k)} - \bar{x})}{\delta_x^2} \qquad (1)$$

where N is the number of spams, k is the lag index (the index lag between two spam inter-arrivals), x_i is the i^{th} spam inter-arrival time and $\bar{x}$ is the average spam inter-arrival time, δ_x^2 is the variance of the spam inter-arrival time, ρ_k is the correlation co-efficiency of the spam inter-arrival time with lag index of k. The intuition is to show that even some spam arrivals within the same group of spammers are far apart, e.g., the i^{th} spam and the $(i+k)^{th}$ spam with a lag index of large k, they may still be well correlated, meaning *the spams from the same group of spammers often arrive in burst*.

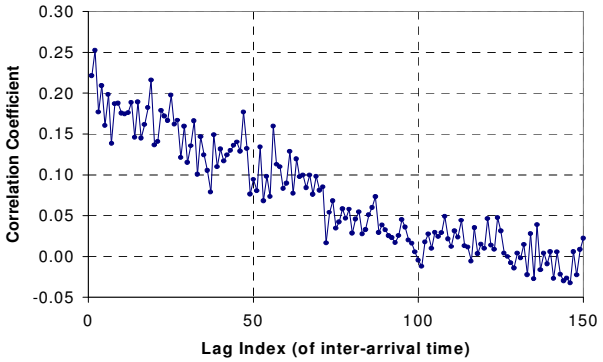

Figure 6: The correlation structure of the spam inter-arrival time of the spams from the spammers belonging to the largest cluster in Day 1.

Figure 6 shows the correlation structure of the spam inter-arrival time of the spam mails from the spammers belonging to the largest cluster in Day 1 based on Eq. (1). The x-axis denotes the lag index and the y-axis is the correlation co-efficiency. As we can see that the spam arrival within the same cluster of spammers demonstrates strong long-range dependency as the lag index approaches 75, the correlation co-efficiency is still around 0.1. The curve is a little bit noisy but the overall trend is clear. The correlation co-efficiency oscillates along the "trend line" due to the granularity of the timestamp in that at a given moment, e.g., a given second, there are often multiple spam arrivals at the domain mail server, where the spam mails are collected.

We observe similar clustering patterns from Day 1 to Day 7. Figure 7 shows the clustering structures of the spammers based on the URLs in the spam mails of Day 6 and Figure 8 illustrates the correlation structure of the spam inter-arrival time of the spam mails from the spammers belonging to the largest cluster in Day 6 based on Eq. (1). As we can see that even the lag index reaches 150, the correlation co-efficiency is still around 0.1, which demonstrates strong long-range dependency in the sense that spam mails from the same cluster often arrive in bursts.

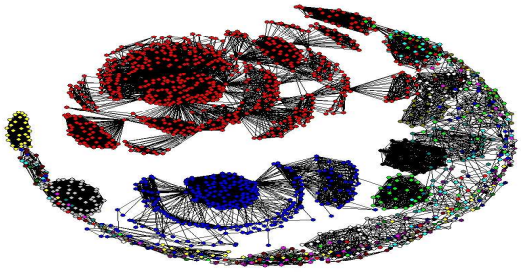

Figure 7: The clustering structure of the spammers based on spam mails of Day 6.

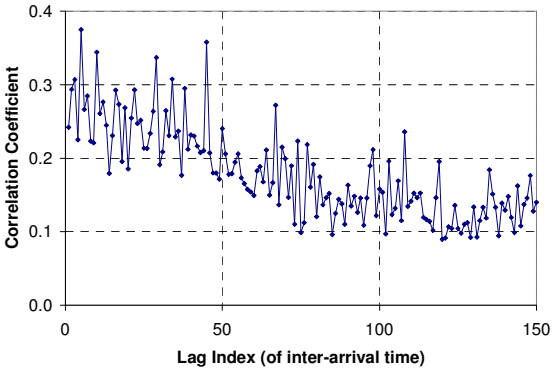

Figure 8: The correlation structure of the spam inter-arrival time of the spams from the spammers belonging to the largest cluster in Day 6.

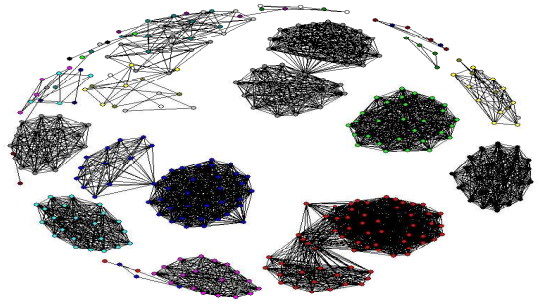

Figure 9: The clustering structure of the spammers based on the money amounts in spams of Day 1.

Figure 9 shows the clustering structure of the spammers based on the money amounts in spam mails of Day 1. As a majority of the spam mails are related to money, the clustering structure in Fig. 9 seems pretty interesting. Normally, the ad materials in the spam mail come with unit prices of the commodities or stocks, so it is unlikely that the spammers could intentionally spoof a random money amount. Compared with Fig. 4 and Fig.

7, we observe that the cluster sizes are relatively small for money-amount-based clustering structures, which may not be an effective one for group-based anti-spam strategies. Nevertheless, the money-amount-based criteria could be used as a complementary one together with URL-based criteria.

4. GROUP-BASED ANTI-SPAM STRATEGIES

In this section we present group-based anti-spam strategies based on our empirical study on the community behavior of spammers in Section 3. In the following discussion, we only consider the grouping structure of spammers based on URLs in the spam mail. The scenarios based on other criteria such as email attachment, money amount or stock symbol can be easily incorporated, following the same framework.

4.1 Design Objectives

In this section, we list several ideal properties of an anti-spam software system should possess. First, let us define the notations of false positive and false negative. We call it a *false positive* when a non-spam email is detected as spam. Accordingly, we call it a *false negative* when a spam is not detected with respect to its nature. We have the following design objectives for the anti-spam software system:

The minimization of both false positive and false negative; the false positive and false negative ratios should be kept as low as possible, in particular for the false positive as false positive detections may lead to the deletion of important legitimate emails.

Easy customization for individual users based on their own filtering criteria; for example, some users may want to do group-based anti-spam based on URL in the email, some users may want to do group-based anti-spam based on other criteria such as email attachment, money amount, stock symbol, etc.

Adaptation to the email traffic dynamics; namely, the anti-spam software system should be able to adjust the system parameters and states based on the dynamics of the email traffic in order to be effective.

The minimization of mail server resources: the related mail server resources include CPU time, network bandwidth, computer storage, etc.

Recoverability for false positive; in the sense that the anti-spam system should provide a backup mechanism for possible false positives so that the user can recover the legitimate email later.

4.2 The Framework

The presented group-based anti-spam framework can be used as a *complementary* component for other existing anti-spam system, e.g., SpamAssasin, to efficiently block spams from organized spammers. The basic idea is that if we can observe some group-based behavior/patterns of spam senders based upon some common signatures from the email content and/or headers, e.g., URL or some other criteria, we can assign a high spam score to the emails from this group. The more members for the given spammer group, the higher spam scores for the emails from the given group of senders. The intuition behind this is that it is *highly unlikely* for a large group of legitimate senders to send emails with exactly the same type of signatures, e.g., the same URL. Notably, in the selection process of the URLs, we only extract those with .com and/or .net domains, for example, it is unlikely that spammers send URLs with .edu domain as most of the URLs are for advertisement/commercial purposes.

From our empirical studies in Section 3, we observe that some spammers are associated with multiple groups based on our classification criteria, e.g., the URL in the email, and those "cross-group" spammers typically send more spams in the near future. It is critical to block those "highly active spammers" as we observe that the top 2% active spammers accounted for near 20% of the total spams. The basic idea here is to assign higher spam scores to those emails from the senders that associated multiple groups in order to block potential "highly active" spammers.

When an email comes, we first extract all the potential advertisement/commercial URLs from the mail content, then calculate the hash value for each URL in the given email and update the number of members counter based on distinct IP addresses for the corresponding associated groups. Finally, we update the number of associated groups counter for the given IP address of the mail sender. We use a time sliding window exponentially weighted moving average to calculate the average number of members for a given group. The detailed algorithm is described in Section 4.3. We say a new group is terminated if and only if the average number of members for a given group is below a given threshold, say $G_{\exp}$. Once a group is terminated, the original group has to be dismantled and the original members of this group have to update their state accordingly. Based on the number of members of each group, we assign a spam score for the given email. In this preliminary study, we assign a blocking probability for the given email based on the number of members in that group and the number of associated groups for a given spammer.

Till now, it is clear that we use the source IP address as a unique identifier of the email sender for group-based

classification, based on which we assign a spam score or a blocking probability for the given email. We do not use the IP address of the email sender for blacklist blocking. So, even for rare cases of spoofed source IP address, it will have little impact on the effectiveness of our group-based anti-spam approach. As mentioned in Section 2, there is little chance for spoofed source IP address to be successful because of 3-way handshake approach to record the IP address of the email sender. We discussed possible scenarios in which the 3-way handshake approach may not be able to detect spoofed source IP address in Section 2.

4.3 TSW-EWMA Algorithm

We use a time sliding window exponentially weighted moving average (TSW-EWMA) algorithm [15] to *dynamically* calculate the state of each URL-based group and determine if an old group should be terminated based on the spam arrivals. Let W be the window size in terms of some time unit, say hours. Let $I_{URLi-pre}$ and $I_{URLi-cur}$ denote the number of members, e.g., the number of distinct IP address, associated with this URL-based group, in the previous time window and the current time window respectively. Let I_{URLi} be the average number of members for the given URL. We have

$$I_{URLi} = \alpha \times I_{URLi-pre} + (1-\alpha) \times I_{URLi-cur} \qquad (2)$$

where α is the weight, e.g., the filter constant. This type of moving average filter places more importance to more recent email data by discounting older mail data in an exponential manner. At the same time, the filter smoothes transient behavior of email traffic.

4.4 A Simple Hash Lookup

First, we convert the ASCII characters of a given URL into binary data format and let x_i denote the number represented by the bits in the i^{th} character of the URL. Notably, each ASCII character is represented with a distinct 7-bit binary data [13]. Let m be a large number, say, $m > 2^7 \times L_{\max}$, where $L_{\max}$ is the maximum length with respect to the number of characters for a given URL, say, 80. We define a simple hash function as follows:

$$H(URL) = \sum_{i=1}^{n} x_i \bmod m, \qquad (3)$$

where n is the number of characters in the given URL. We use chaining, e.g., a link bucket, for hash function collision resolution.

Alternatively, we can use a SHA-1-based hash function [16], which takes an arbitrary length of URL (less than 2^{64} bits in length) as input and produces a 160-bit number as the corresponding URL digest. The SHA-1-based hash function has the well-known property of collision resistance. The drawback is that SHA-1-based hash function is more *computationally* expensive than the one described in Eq. (3).

4.5 Preliminary Results

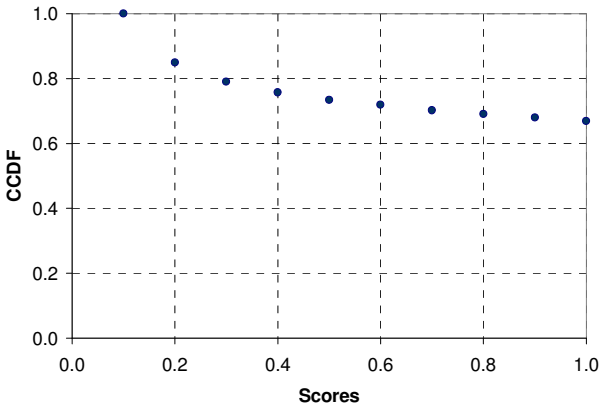

Figure 10: The Spam Score Distribution Based on Group-Based Approach.

As shown in Fig. 10, the x-axis indicates the spam scores and the y-axis denotes the CCDF (complementary cumulative probability) of the spam score. Clearly, our preliminary results show that the group-based approach can block 70% to 90% of the spams, depending on the implementation parameters. To the best of our knowledge, it is the first time that group-based anti-spam method has been explored.

5. ISSUES AND OPEN CHALLENGES

There are several challenges for group-based anti-spam strategies based on URL-grouping. For example, if a host is running DHCP, the host IP address could change from several hours to several days. The change of the IP addresses could change the clustering structure of the spammers.

Another issue is that the spammers could use various HTTP formats to intentionally hide the URL information. For example, the following links all indicate the same link as http://www.yahoo.com: http://3631052355 (a single decimal number of the IP address), http://0xD86D7643 (a single

hexadecimal number of the IP address) http://0330.0155.0166.0103 (the dotted form in octal) [14].

The spammers can also use some steganography techniques such as html color, graph, etc. to camouflage URL or other information used to group the spammers.

6. CONCLUSION

With the popularity of the Internet, Email has become a wonderful communication tool in people's daily life, with which people can reach friends, colleagues and family at *virtually* every corner of the world *instantaneously*. However, the flip side of the coin is the bulk, massive unsolicited commercial/advertisement spams, which have seriously threatened the very existence of Email.

In this paper we investigate the clustering structures of spammers based on spam traffic collected at a domain mail server. Our study show that the relationship among spammers demonstrates *highly clustering structures* based on URL-grouping. The inter-arrival time of spams from the same group of spammers exhibits long-range dependence in the sense that the spams from the same group of spammers often arrive *in burst*. We also observe that spammers associated with *multiple* groups tend to send more spams in the near future.

We present group-based anti-spam method based on the *number of members* in the group and *the number of groups* that a spammer is associated with. Our preliminary results show that group-based method can block 70% to 90% of the spams, depending on the implementation parameters. We need to emphasize that group-based anti-spam method may *not* be highly effective as a *stand-alone* approach as some groups may have only one member, but it can be used as a *complementary* tool for other existing anti-spam tools, such as SpamAssasin. We will continue to explore interesting properties of the clustering structures of spammers as our future directions.

7. ACKNOWLEDGEMENTS

We would like to thank Jaeyeon Jung, our shepherd, for the spam sources and her valuable suggestions that have inspired our interests in this project. We thank Mythili Vutukuru, Jeremy Stribling, Prof. Hari Balakrishnan and anonymous reviewers for their insightful feedback and comments that have greatly improved the quality of this paper.

8. REFERENCES

[1] J. Jung and E. Sit, "An Empirical Study of Spam Traffic and the Use of DNS Black Lists", *in the proceeding of ACM IMC' 04*, Oct. 2004.
[2] L. H. Gomes, C. Cazita, "Characterizing a Spam Traffic", *in the proceeding of IMC' 04*, Oct. 2004.
[3] "Know Your Enemy: Tracking Botnets", the Honeynet Project and Research Alliance, http://www.honeynet.org, March 2005.
[4] S. kandula, D. Katabi, M. Jacob, A. Berger, "Botz-4-Scale: Surviving Organized DDOS Attacks That Mimic Flash Crowds", *in the proceeding of USENIX NSDI*, May 2005.
[5] S. Katti, B. Krishnamurthy, D. Katabi, "Collaborating Against Common Enemies", *in the proceeding of ACM IMC'05*, Oct. 2005.
[6] P. Graham, "Different Methods of Stopping Spam", http://www.secinf.net/anti_spam/Stopping_Spam.html, Oct. 2003.
[7] "Dealing Effectively with Spam", GFI Software, http://www.secinf.net/anti_spam/Dealing_Effectively_with_Spam.html, May 2003.
[8] "Blocking over 98% of Spam using Bayesian Filtering Technology", GFI Software, http://www.secinf.net/anti_spam/Blocking_Spam_Bayesian_Filtering.html, Oct. 2003.
[9] J. Synoradzki, P. Wawrzyniak, M. Zmudzinski, "Four Popular Anti-Spam Filters for Exchange Reviewed",http://www.secinf.net/anti_spam/Preventing_Spam_Antispam_Filters_MS_Exchange.html, May 2004.
[10] "Spamfighting Overview FAQ", spamfaq.net, http://www.secinf.net/anti_spam/Spamfighting_Overview_FAQ.html, May 2003.
[11] B. Laurie, R. Clayton, "Proof-of-Work Proves Not To Work", *in the proceeding of the Workshop on Economics and Information Security*, MN, May 2004.
[12] H. Balakrishnan, "Computer Networks", Lecture Notes, class 6.829, MIT, Fall 2005.
[13] "ASCII Table and Description", http://www.lookuptables.com/.
[14] J. Graham-Cumming, "Tricks of the Spammer's Trade", in Hakin9 magazine, issue 3, 2004.
[15] S. Biswas, R. Morris, "ExOR: Opportunistic Multi-Hop Routing for Wireless Networks", in the proceeding of ACM SIGCOMM '05, Philadephia, USA, Aug. 2005.
[16] M.J.B. Robshaw, "MD2, MD4, MD5, SHA and Other Hash Functions", Technical Report TR-101, version 4.0, RSA Laboratories, July 1995.