

Measuring the botnet using the second character of bots

Zhitang Li

School of computer science and technology, Huazhong University of Science and Technology, Wuhan, China

Email: junehu1210@gmail.com

Jun Hu, Zhengbing Hu, Bingbing Wang, Liang Tang and Xin Yi

School of computer science and technology, Huazhong University of Science and Technology, Wuhan, China

Email: {leeying, hzb}@mail.hust.edu.cn

Abstract—Botnets have become one of the most serious threats to the Internet. They are now the key platform for many Internet attacks, such as spam, distributed denial-of-service (DDoS), and we call these attacks “the second character of bots”. In this paper, we focus on characterizing spamming botnets by leveraging both spam payload and spam nodes traffic properties. Measurement of botnets is an important and challenging work. However, most existing approaches work only on specific botnet command and control (c&c) protocols (e.g., IRC) and structures (e.g., centralized). In this paper, we present two measurement frameworks (MFNL and MFAL) that based on the second character of bots to measure the size of the botnet. We have easily implemented our prototype system and evaluated it using many real network traces, and we also compare these two approaches from several points.

Index Terms—botnet, SMTP, spam, size, MFNL, MFAL

I. INTRODUCTION

It is widely accepted that botnet poses one of the most significant threats to the Internet [13]. But the size of the botnet continues to be an issue of debate among the research community [2, 3, 4, 5, 6, 7]. The recent measurement approaches can be divided into two categories: active measuring and passive measuring.

In the active measuring area, most researchers using honeynet and crawler to infiltrate the botnet getting some information of the peers and the state of the network, by running an instance of the botnet and developing a crawler using specific protocol queries, they can collect some information, such as node IDs, IPs, and port numbers. The advantage of this method is that it can directly get the topology of the network and has a more correct result, but the crawler is based on the protocol, so it need a priori knowledge of botnet (such as captured bot binaries and the protocol the botnet using), so it will be difficult to operate on a new botnet, especially on the encrypted botnet.

In the passive measurement area, the monitors are fixed on the edge of the backbone or the core routers, or the bound of the ISP, the monitors can get the node numbers and the flow character etc. Using these overview flow information, adopt suitable methods, they can measure the botnet. The advantage of this method is that it needs no a priori knowledge, but with a high false positive rate. We pick out some typical papers in the both areas and analyze them respectively below.

M.A. Rajab [1] pointed out that botnet size can be the size of footprint or the size of live population. According to different definitions, the size varies with different measurement methods. For example, earlier studies have proposed a number of techniques to measure the size of botnets which led to different results. Dagon [2] established that botnet size could reach 350,000, but Rajab [5] indicated that the effective size of botnets rarely exceeded a few thousand.

In this paper, we attempt to estimate the live population of the botnet, including IRC botnet and P2P botnet. Two new methods are proposed to measure the size of the botnet.

Early study primarily focused on IRC botnet. At present, there are a number of mechanisms for estimating botnet sizes, but there are some limitations in each of them. The most direct method is to infiltrate the botnet by joining the command and control channel [5]. Because of its simplicity, this technique suffers from bot cloning and temporary migration of bots. So it is difficult to provide an accurate bot count in these cases, because there is no significant difference between actual bots and temporary clones or migrants.

Dagon [2] developed a technique for counting infected bots by manipulating the DNS entry associated with a botnet’s IRC server and redirecting connections to a local sinkhole. This technique can only measure the botnet’s footprint, but knows nothing about whether the bots belong to the same botnet. Moheeb [5] explored the use of DNS cache snooping to uncover a botnet’s footprint. But this technique requires DNS servers that

probe from arbitrary clients and reply to queries for cached results. Ramachandran [6] suggested inferring bot counts by observing DNS lookups for hosts listed in DNS-based blackhole list. The approach can only provide an overall estimate of possible bots in DNS-based blackhole lists, but clearly it cannot estimate the size of specific botnet.

Zhuang [10] explored a technique to measure the botnet size by clustering the spam[9] on Hotmail server. Although we both measure the size of botnets from emails, our approaches are different from hers. Her approach has two limitations. First, her measurement results rely on the choice of the datasets and anti-spam technology. The integrity of the data sets directly impact on the accuracy of her results. Second, she did not know exactly what content the botnet sent. She clustered single content spam to judge whether the spam campaigns belonged to the same botnet.

In summary, this paper makes the following contributions. 1. We explore two new mechanisms (MFNL and MFAL) to measure the live population of the botnet. 2. Overlap across different botnet populations is a challenge of the measurement of the botnet, but our approach can strictly distinguish the bots from different zombie networks and can estimate the size of specific zombie networks. 3. We analyze and compare the two approaches and propose to use them together to get a better result.

II. BACKGROUND

A botnet is a collection of a large numbers of bots. symantec[14] define a bot as:

Bots are similar to worms and Trojans, but earn their unique name by performing a wide variety of automated tasks on behalf of their master.

A bot master can be a person or a group who control bots remotely without having to log on to their operating system. The bot master can give instructions or upload data to the botnet. In this paper, we are only looking at the malicious uses of botnets, especially spamming.

The essential properties of a botnet are that the bots communicate with some C&C servers/peers, perform malicious activities, and do so in a similar or correlated way. Fig. 1 shows the typical botnet architecture.

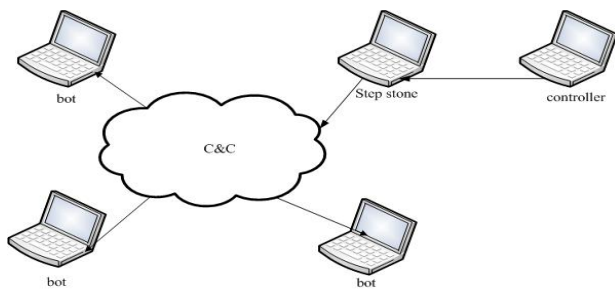


Figure 1 The typical botnet architecture

The botnet C&C differs from each other in how big they can get, how easy they can be detected and disrupted and how they exchange messages and commands. The oldest is the centralized form (e.g. IRC

botnet), through time, there comes decentralized and hybrid architecture (e.g. P2P botnet) to make it more difficult to track them or to take them down.

Botnet have multiple nefarious uses: mounting DDoS attacks, stealing user passwords and identities, generating click fraud [15], and sending spam email [16].

From a brief history of the botnet, we find that the C&C of the botnet changed a lot through time to make it more difficult to track them or to take them down, so most of the current botnet measurement approaches working only on specific botnet C&C and structures (IRC botnet) will become ineffective as botnet change their C&C protocols. But the malicious activities still keep the same through time, especially the spam, the cheapest way to get the potential profit. In this paper, we focus on the botnet which sending Spam as its main function.

Botnets have been widely used for sending spam emails at a large scale [6, 11]. Ramachandran [11] performed a large scale study on the network behavior of spammers, providing strong evidence that botnets are commonly used as platform for sending spam. Almost every bot in a botnet sends spam. Spam senders belong to different zombie networks, serve different controllers for different purposes and send spam with different content. But if they belong to the same botnet and serve the same controller, they may send spam with the same or similar content in the favor of the controller. Since most bots are family computers, they only run a few hours every day. However, they have to complete sending a certain amount of junk emails in a short period of time.

III. CASE STUDY :MYBOT

TABLE I. RESULT OF THE SPECIMEN

	Mybot A	Mybot B
IP	218.199.92.68	218.199.92.69
Time of the packets	99.13 minutes	91.2 minutes
Total packets num	45378	40000
SMTP packets num	22463	19600
Ratio of SMTP packets	0.495	0.49
Frequency of SMTP packets (num/minute)	226.59	214.91
Spam num	1144	1001
Spam sending frequency (num/minute)	11.54	10.98
URLs	http://fabslick.com http://lowlydrool.com http://rosywise.com http://slickslick.com http://treatlove.com	http://fabslick.com http://lowlydrool.com http://rosywise.com http://slickslick.com http://treatlove.com

Mybot is the most known IRC bot and is the one using a completely centralized IRC protocol. We monitored the spamming behavior of the bot in our test environment. Our test environment is setup in a secure environment to avoid infecting other machines or participating in

malicious attacks. We used a botnet specimen for our case study. Table I shows some observed results of the specimen. In our study, we take one SMTP session as a successful spam. Fig. 2 shows the whole SMTP session.

From Table I, we find that the bot has a high unusual SMTP frequency and sent a large number of spam than normal users. In about 90 minutes it has sent more than 1000 spam, but the URLs in all these spam is only five. We run the same specimen on Mybot A and Mybot B, so they belong to the same botnet, from the Table I, we find the spam has the same URLs, this proved our previous inference.

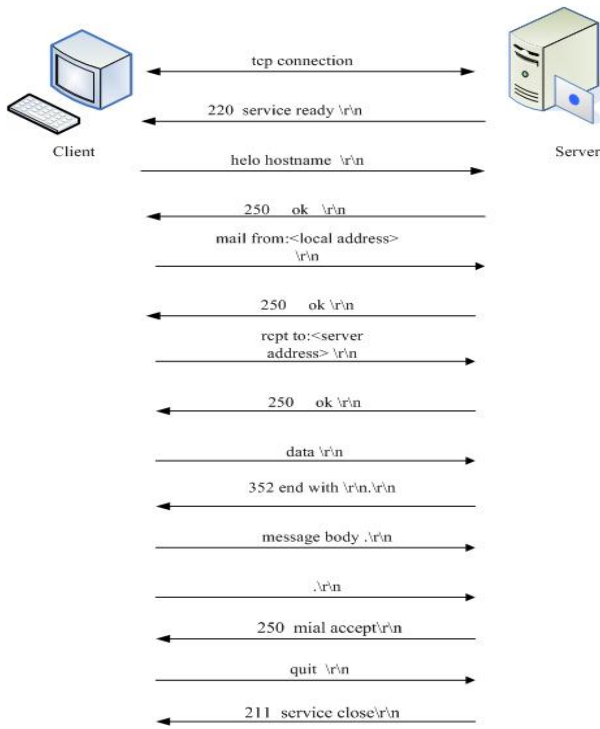


Figure 2 The whole SMTP session

IV. MEASURING FROM THE NETWORK LEVEL(MFNL)

---BY MONITORING GROUP ACTIVITIES IN SMTP TRAFFIC

A. SMTP Decode Module

In the last section, through the analysis of the botnet specimen, we find that in a short time the bot sent a lot of SMTP packets, and sent a large number of spam than normal users, and the spam sent by a bot specimen has the same or similar content. According to these characteristics, this paper presents a botnet measurement technology based on the analysis of the SMTP packets. The whole process can be described as Fig. 3.

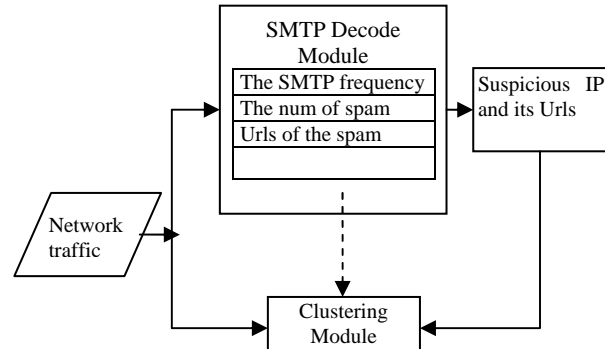


Figure 3 The process of FMNL

The whole system contains two modules: SMTP decode module and clustering module. In the SMTP decode module we extract all the SMTP packets from the network traffic, statistic the SMTP frequency, the number of spam each IP sent and extract the URLs of the spam. If the values exceed the threshold we will mark the IP as a suspicious bot.

Approximately 70 billion emails are sent each day, half of which are spam[17]. Supposing 10 billion of the spam is sent by 150 million bots, each bot must send at least 60 spam a day on average. Ordinary users only keep their computers on for a few hours each day, so such a bot need to send 50 or more spam per hour. By observing the activity of a sample of mybot has proved the inference, which sent 1144 spam in 90 minutes.

In the cluster module, we cluster them into different spam campaign based on the same or similar URLs, each campaign consider belonging to the same botnet.

Suspicious IP algorithm

```

1 for each packet
2 {
3   if (SMTP)
4     get the source IP;
5   if (is a new email session)
6   {
7     (the num of the spam sending by this source ip)++;
8   }
9 }
10 if(time out)
11 {
12   for each IP
13   {
14     stastic the num of the spam it has sent;
15     if(count>THRESHOLD)
16     {
17       the ip is a suspicious bot;
18     }
19   }
20 }

```

B. Cluster the spam campaigns

From the last section and the case study of mybot, we know that if the bots belong to the same botnet. They will send spam with the same or similar URLs, because of the bot controller's profit. So when we extract the URLs from the SMTP packets, we cluster the same and similar URLs to a spam campaign, all the IP in this campaign are considered in the same botnet, then we can easily get the size of this botnet.

If the URLs we extracted are short, we can just use string comparison algorithm to cluster them into a spam campaign. But for long and complicated URLs we defined entropy reduction, leverages information theory to quantify the probability of a coming URL matching a stored URL (signature). Given a coming URL (e), let $R_e(x)$ denote the expected number of bits used to encode a random string x with the signature. $R(x)$ Denote the expected number of bits used to encode a random string x without the signature.

The entropy reduction $d(e)$ is defined as the difference between $R_e(x)$ and $R(x)$, for example, $d_e = R_x - R_e(x)$. The similarity is defined as

$$P(e) = \frac{2^{R_e(x)}}{2^{R(x)}} = \frac{1}{2^{R(x)-R_e(x)}} = \frac{1}{2^{d(e)}} \quad (1)$$

Given a regular expression e , if $d(e)$ is small, P_e tends to be large, which means e is similar with the stored URL.

If the URLs in the spam sent by one IP is the same or similar with the URLs in the spam sent by another IP, we will cluster them into a spam campaign. All the IPs in the same campaign belongs to a botnet. The number of the IP is the size of the botnet.

C. Experimental evaluation

Fig. 4 shows our test environment which is setup in a secure environment to avoid infecting other machine or participating in malicious attacks. Our test environment setup consists of one software firewall installed with IPCop. Five computers, installed with Windows XP SP2 and the subnet E which used by the normal users are all connected through a switch to the WAN. One host will be logging all the traffic with Wireshark. The other four will server as bots in our botnet. The infected PC A and PC B have run the IRC specimen (mybot.exe), infected PC C and PC D have run the SDbot.exe.

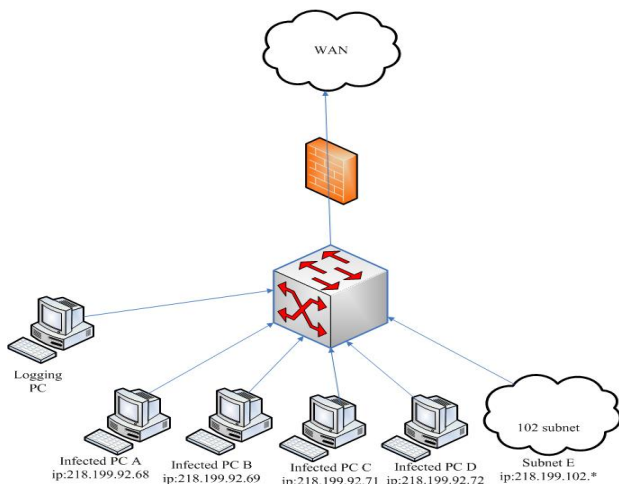


Figure 4 Our test environment

We captured half an hour data of A, B, C, D, E separately in the morning, noon, and evening for our experiments. Fig. 5 shows the spam num/half-hour of each. From Fig. 5 we find that the bots sent a sharp large num of spam than the normal users.

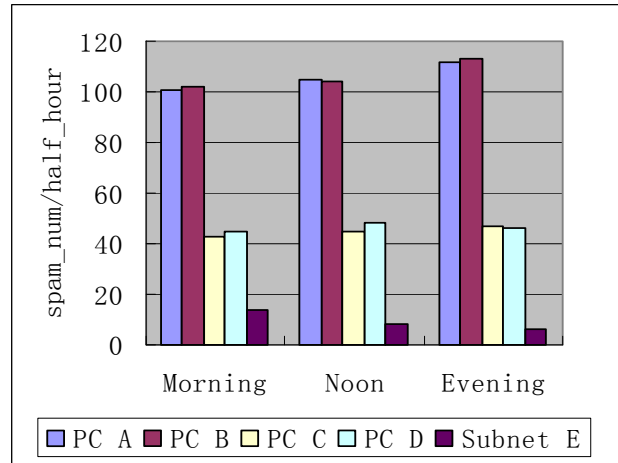


Figure 5 Spam num/half-hour of A, B, C, D, E

The Fig. 6 shows part of the measurement results in the test environment. The 218.199.92.69 and 218.199.92.68 were in the same botnet. It fits well with our test environment. Using our system, the 218.199.92.71 and 218.199.92.72 were clustered to another same botnet too.

URL	IP	SCALE
http://fabslick.com	218.199.92.68 218.199.92.69	2
http://lowlydrool.com	218.199.92.68 218.199.92.69	2
http://rosywise.com	218.199.92.68 218.199.92.69	2
http://slickslick.com	218.199.92.68 218.199.92.69	2
http://treatlove.com	218.199.92.68 218.199.92.69	2

Figure 6 Part of the result

For further study, we have some interesting findings; the most interesting finding is the fraction of the target IP of the bots. From Fig. 7 we find Mybot and SDbot has different target IPs. This is because different botnet was rented by different merchant for different purpose. But Mybot A and Mybot B almost have the same distribution; SDbot C and SDbot D have the same situation. This is because bots in the same botnet have the same instructions from their botmaster. They may get the spam and the sending list from the same place.

MFNL can be fixed on the edge of the backbone or the core routers, or the bound of the ISP. It will be very effective and sensitive to the nodes which sent a lot of spam in a short time. For most SMTP packets are not encrypted now, so it will work well to get the size of a botnet in a local area network.

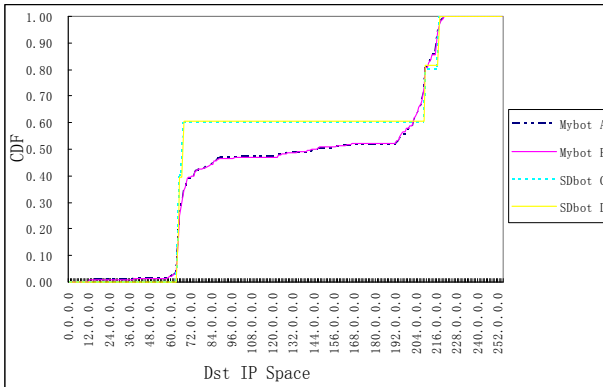


Figure 7 The fraction of the target IP

V. MEASURING FROM THE APPLICATION LEVEL(MFAL)

---BY USING URL AND COLLABORATIVE MAILSERVERS

In the last section we have proposed our approach from the network level to measure the botnet. This section we will present another approach to measure the botnet size by using URL and collaborative Mailservers.

A. Overview of the Systems

Botnets have been widely used for sending spam emails at a large scale [6, 11]. Ramachandran [11] performed a large scale study on the network behavior of spammers, providing strong evidence that botnets are commonly used as platform for sending spam. Almost every bot in a botnet sends spam. Spam senders belong to different zombie networks, serve different controllers for different purposes and send spam with different content. But if they belong to the same botnet and serve the same controller, they may send spam with the same or similar content in the favor of the controller. Since most bots are family computers, they only boot a few hours every day. However, they have to complete sending a certain amount of junk emails task, usually a large number of messages in a short period of time. From section III by observing the activity of a sample of Mybot proved the inference, which sent more than 1000 spam in about 90 minutes. So if we find an IP sending a large number of emails in a certain time, we think it is a suspected bot, and we will store the content and query the Mail Servers to find out all the IPs sending emails with the same content. Those IPs are considered to be the members of the same botnet. We can get the size of this botnet based on the number of IPs. In the same way we can roughly determine the size of different botnets. In our method, we first get the IP sending large num of spam. Then we compare the content of the spam and highlight the URLs embedded in email content, because URLs play an important role in directing users to phishing web pages or targeted product web sites. In this way, we can almost get all the members of a botnet [2].

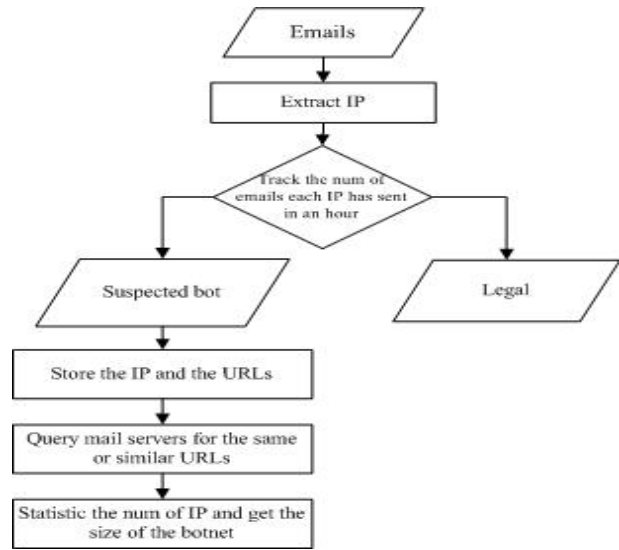


Figure 8 The MFAL system modules

Fig. 8 shows our general idea. The system is based on two elements. First, suspicious bots send a lot of emails (spam) in a short time; Second, the bots in the same botnet send the same or similar spam.

MS (Mail Server) will identify and track the source of the emails, which will be discussed in section 3. From the tracking, we will get the number of emails each IP sent. If the number exceeds a certain value, we will mark the IP as a suspected bot. Then we query all the other MSs for all the IPs sending the same content. The botnet size is roughly determined by the number of the IPs.

B. Architecture of MFAL

MFAL, is a distributed system that identifies the source of each e-mail and then stores this information in a distributed database that is used and updated by all peers. MFAL depends on the collaboration of peers. Fig 9 shows the architecture of the MFAL.

MFAL consists of several distinct parts: identifying the source of e-mails, keeping track of how many e-mails were recently sent by a source, and disseminating this information for the purposes of statisticing the size of the botnet. Additionally, these tasks must be coordinated for each e-mail as it is received by an MS. Thus, MFAL comprises four parts that perform the respective functions.

First when an e-mail arrives at a MS, the message is passed to the source identification module to determine the source of the e-mail.

Second, MFAL queries the distributed database about the number of e-mails that were recently sent, typically within an hour, from the same source as the current e-mail.

Third, if the num of spam sending by this soure cross the threshold, MFAL extract and store the URLs.

Fourth, MFAL queries the distributed database to get all the IP sent the same URLs. Then get the size of the botnet.

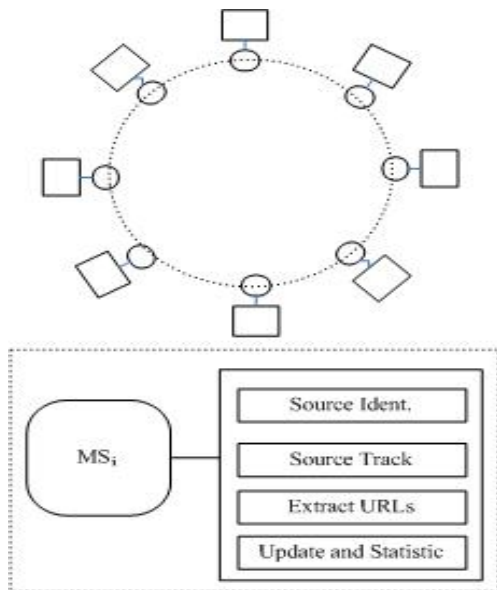


Figure 9 Architecture of MFAL

C. IP Source Identification and Tacking

When receiving the emails, MS will extract the following information: URL string, source IP address and mail-sending time. The most challenging thing is source identification since the email-senders can be faked.

We assume that once the email is received by a trusted relay, the received lines added by these relays are correct. That means the received line added by the first trusted relay, identifies the entrusted host that injected the email. The host is considered to be the source of the email.

The key challenge then is to identify this first trusted relay, and its corresponding received line, which contains the defacto IP address of the sender. But the bots typically do not have static IP address. Thus, we consider that a relay is trusted if it has a static IP address. As pointed out in [8], we can find the first trusted relay from the MX entry and several approaches. When we find the ingress relay, we check from the received line corresponding to the ingress relay, using several heuristics to get the source of the email.

Many spam messages are not sent from botnets, so we can not recognize all the IP sending large number of emails as suspicious bots. We need to remove the static IP which is notorious in blacklist, such as popular proxies, open relays, etc. If the sender IP address of a message is one of these, we exclude that IP from further analysis.

When we get the real IP, we need to track the number of the emails the IP sent. Since a bot does not necessarily target multiple recipients within a single domain, it will need a collaborative effort to determine whether an IP is a suspicious bot.

Assuming that MS_1 received an email, first we extract the IP of the email, then query the number of the emails sent by this IP on the MS_1 in an hour, and record the num as n_1 ;

Then, we query the number of the emails sent in the same hour by this IP on the MS_2 and record it as n_2 ;

We query the rest of the MSs, and record the number of emails as $n_3, n_4 \dots n_n$.

So we can get the total number of the emails sent by this IP in an hour, recorded as N .

$$N = n_1 + n_2 + \dots + n_n; \tag{2}$$

If N is large and crosses a certain threshold, which is 50 in the present study, we will mark this IP as a suspicious bot.

When we find the suspected bots, we store the following information on the MS for later query.

IP	TIME	CONTENT
----	------	---------

From section III we know that in a botnet each bot sends the same or similar content emails for controllers' benefit. If we record the content sent by this IP, we know exactly what is sent by this botnet. Then we can easily get the size of the botnet. We just need to query the MSs to get all the other ips which send the above content. We can surely say that the IP sending the same content is a member of the same botnet. When we get all the members of the botnet, we can know the exact size of the botnet.

D. IP SOURCE IDENTIFICATION AND TRACKING

Since it is believed that a fraction of bots send very few spam each day[11], it's not accurate to consider the number of IP which sent a large amount of email in a short period of time (in this paper we defined it as an hour) as the size of the botnet. However, from these statistics we can know exactly what the botnet is sending. The sending content of an IP may be more than one, so we need to query each MS to get all the IP sending the same content in a period. This period can be decided by the life-cycle of the botnet (several days).

Botnets are now used for a commercial purpose to attract users to buy the goods and services in the spam. So the same or similar content email will be sent to thousands or maybe millions of users by bots. Although the recent botnets are very smart, and they can be customized for different users with different mails, it is impossible for spammers to produce thousands of the same meaningful spam. Therefore, different versions of the spam should be very similar to each other with minor changes. In particular, different versions of the spam will contain many of the same sentences, especially the URLs.

Fig. 10 shows an example of three emails sharing the highlighted URL, but mixed with a number of other information. The example came from a sample of Peacomm (a P2P botnet), and we collected the SMTP (Simple Mail Transfer Protocol) data every 15 minutes using Wireshark.



Figure 10 The emails sent by a sample of peacomm

The URLs cannot be easily changed: users must be able to correctly contact the spammers if the spam is to be of any use to the spammers. So the URL can be used as a valuable resource to compare the content of the spam. URL is very important for a botnet to achieve the purpose, and all bots will send the spam containing the same URL.

In the section IV We defined entropy reduction, leverages information theory to quantify the probability of a coming URL matching a stored URL (signature) to cluster the spam with the same or similar URLs into a spam campaign. Here we can use the same method as section IV, then statistic the number of IPs in the campaign to get the size of a botnet.

E. IP DYNAMIC AND ESTIMATING BOTNET SIZE

When We get all the IP in a life cycle of the botnet, we still can not fully determine the size of the botnet because many home computers are connected to the internet through ADSL, cable or other devices which make the user's IP changes daily or even hourly [12]. Zhuang [10] has proposed an effective method to estimate IP dynamics. After IP dynamic treatment, we can count the number of distinct machines, and then obtain the size of the botnet.

Our study is based on the spam sent by a sample of Peacomm (one of the P2P botnet) and the emails collected from HUST mail server. In our dataset, a total of 103 emails containing the highlighted URL were sent on the same day from 70 different IP address. We suspect the corresponding hosts were from the same botnet.

In order to measure the whole size of the botnet, we defined α_i as the ratio of the number of the users of MSⁱ and HUST Mail Server. We defined k_i as the number of the users of MSⁱ, and k_h as the number of the users of HUST Mail Server. So α_i can be denoted as:

$$\alpha_i = \frac{k_i}{k_h} \quad (3)$$

Where k_i and k_h always be a constant, we can get them easily. So we can roughly get the size of the botnet when we get the distinct IP (B_h) on HUST Mail Server. We defined B as the total member of the botnet, then

$$B = \sum_{i=1}^n \alpha_i * B_h \quad (4)$$

By using this method in our experiment, most botnets have hundreds to thousands of IP addresses; this result is consistent with other's research. Since our experiment is simple and only uses sampled emails, the reported botnet sizes are expected to be much smaller than the actual sizes.

MFAL can be fixed as a plug in on the Mail servers to achieve all of its functions.

VI. COMPAREMENT OF MFNL AND MFAL

We have proposed two approaches to measure the size of the botnet. Each of them has their own advantages and disadvantages.

MFNL can be fixed on the edge of the backbone or the core routers, or the bound of the ISP. It will be very effective and sensitive to the nodes which sent a lot of spam in a short time. For most SMTP packets are not encrypted now, so it will work well to get the size of a botnet in a local area network. It is the MFNL's advantage that MFNL is easy to fix and have high accuracy. But it can only get parts of the results, because it is applicable to the LAN. However, the target of the botnet permeates many countries. Another disadvantage is that MFNL will be ineffective when the threshold is low, however today's botnet gets more 'cleverer' than before. For escaping the detection, botmaster use some tactics to design the botnet making them hard to track. One of these tactics is that make parts of the bots's spamming behavior not so obviously, for example, sending less spam in a short time to hide themselves. MFNL will not be so sensitive to these bots.

MFAL is a collaborative system, which has a global view, while MFNL just has a local view. And MFAL can get all the bots whatever they sent a lot of spam or less spam. But MFAL is hard to carry out in the real world and IP source identification is the most challenge thing.

We find that the advantage of MFNL is the disadvantage of the MFAL, while the disadvantage of MFNL is the advantage of MFAL. So maybe we use them together, we can get a better result.

VII CONCLUSION AND FUTURE WORK

We have proposed two measurement frameworks (MFNL and MFAL) that based on the second character of bots to measure the size of the botnet, We have easily implemented our prototype system and evaluated it using many real network traces. And we also compare these two approaches from several points. At last we conclude that if we want to get a much better result we need to use the two approaches together.

In the future we want to make the MFNL and MFAL work together to get a better result.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 60573120, and by the National High Technology Research and Development Program of China(863 Program) under Grant No. 2007AA01Z420, and by the Key project sponsored by Natural Science Foundation of Hubei Province under Grant No. 2008CDA021.

REFERENCES

- [1] M.A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "My Botnet Is Bigger Than Yours (Maybe, Better Than Yours): Why Size Estimates Remain Challenging," in First Workshop on Hot Topics in Understanding Botnets(HotBots'07), April 2007: http://www.usenix.org/events/hotbots07/tech/full_papers/rajab/rajab_html/.
- [2] David Dagon, Cliff Zou, and Wenke Lee. Modeling Botnet Propagation Using Time Zones. In Proceedings of the 13th Network and Distributed System Security Symposium NDSS, February 2006.
- [3] Felix Freiling, Thorsten Holz, and Georg Wicherski. Botnet Tracking: Exploring a root-cause methodology to prevent denial-of-service attacks. In Proceedings of 10th European Symposium on Research in Computer Security, ESORICS, September 2005
- [4] Bill McCarty. Botnets: Big and bigger. IEEE Security and Privacy Magazine, 1(4):87–90, 2003.
- [5] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. A Multifaceted Approach to Understanding the Botnet Phenomenon. In Proceedings of ACM SIGCOMM/USENIX Internet Measurement Conference (IMC), pages 41–52, Oct., 2006.
- [6] Anirudh Ramachandran, Nick Feamster, and David Dagon. Revealing Botnet Membership using DNSBL Counter-Intelligence. In Proceedings of the 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), July 2006.
- [7] L. Spitzner. The Honeynet Project: Trapping the Hackers. IEEE Security and Privacy Magazine, 1(2):15–23, 2003.
- [8] A. Brodsky and D. Brodsky. A distributed content independent methods for spam detection. In First Workshop on Hot Topics in Understanding Botnets, 2007.
- [9] SpamCop. <http://www.spamcop.net/>, 2007.
- [10] Li Zhuang, John Dunagan, Daniel R. Simon, Helen J. Wang. Characterizing Botnets from Email Spam Records. In LEET'08.
- [11] RAMACHANDRAN, A., FEAMSTER, N., AND DAGON, D. Revealing botnet membership using dnsbl counter-intelligence. In SRUTI'06.
- [12] XIE, Y., YU, F., ACHAN, K., GILLUM, E., GOLDSZMIDT, M., AND WOBBER, T. How dynamic are ip addresses? In SIGCOMM'07.
- [13] COOKE, E., JAHANIAN, F., AND MCPHERSON, D. The zombie roundup: understanding, detecting, and disrupting botnets. In SRUTI'05.
- [14] Symantec. Crimeware: Bots. http://www.symantec.com/avcenter/cybercrime/bots_page1.html, 2008.
- [15] DASWANI, N., STOPPELMAN, M., AND THE GOOGLE CLICK QUALITY AND SECURITY TEAMS. The anatomy of clickbot. In HotBots'07.
- [16] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the network-level behavior of spammers. In SIGCOMM'06.
- [17] SpamCop. <http://www.spamcop.net/>, 2007.