Characterizing Botnets from Email Spam Records

Li Zhuang
UC Berkeley

John Dunagan Daniel R. Simon Helen J. Wang
Ivan Osipkov Geoff Hulten
Microsoft Research

J. D. Tygar
UC Berkeley

Abstract

We develop new techniques to map botnet membership using traces of spam email. To group bots into botnets we look for multiple bots participating in the same spam email campaign. We have applied our technique against a trace of spam email from Hotmail Web mail services. In this trace, we have successfully identified hundreds of botnets. We present new findings about botnet sizes and behavior while also confirming other researcher's observations derived by different methods [1, 15].

1 Introduction

In recent years, malware has become a widespread problem. Compromised machines on the Internet are generally referred to as *bots*, and the set of bots controlled by a single entity is called a *botnet*. Botnet controllers use techniques such as IRC channels and customized peer-to-peer protocols to control and operate these bots.

Botnets have multiple nefarious uses: mounting DDoS attacks, stealing user passwords and identities, generating click fraud [9], and sending spam email [16]. There is anecdotal evidence that spam is a driving force in the economics of botnets: a common strategy for monetizing botnets is sending spam email, where spam is defined liberally to include traditional advertisement email messages, as well as phishing email messages, email messages with viruses, and other unwanted email messages.

In this paper, we develop new techniques to map botnet membership and other characteristics of botnets using spam traces. Our primary data source is a large trace of spam email from Hotmail Web mail service. Using this trace, we both identify individual bots and analyze botnet membership (which bots belong to the same botnet). The primary indicator we use to guide assigning multiple bots to membership in a single botnet is participation in spam campaigns, coordinated mass emailing of spam. The basic assumption is that spam email messages with similar content are often sent from the same controlling entity, because these email messages share a common economic interest. Therefore, the sending machines of these spam email messages are likely also controlled and operated by a single entity (though this may be a different entity than the first). By grouping similar email messages and related spam campaigns, we identify a set of botnets.

Our focus on spam is in contrast with much previous work studying botnets. Previous studies have used or proposed such techniques as monitoring remote compromises related to botnet propagation [6], actively deploying honeypots and intrusion detection systems [13], infiltrating and monitoring IRC channel communication [3, 6, 11, 14], redirecting DNS traffic [8] and using passive analysis of DNS lookup information [15, 17]. Focusing on spam instead has at least a couple of major benefits. First it supports a greatly simplified deployment story: the analysis can be done on an existing email trace from one of the small number of large Web mail providers (e.g., GMail, Hotmail, Yahoo Mail). Second, by focusing on spam, the factor directly related to the *economic motivation* behind many botnets, it is harder for botnet owners to evade detection compared to previous approaches – in particular, stopping sending spam email destroys the purpose of these botnets. Lastly, grouping bots into botnets by analyzing spam is potentially a less ad-hoc and easier task than analyzing IRC/DNS logs, because IRC messages or DNS queries vary greatly from one botnet implementation to another [3, 6, 8, 11, 14, 15, 17].

Our approach is not without caveats and challenges. One obvious caveat is that we are not able to uncover botnets not involved in email spamming. However, as we will show later, the number and sizes of botnets we discover are similar to previous findings with other methods, suggesting that our method covers a large portion of all botnets. To name a few challenges, first, it is not trivial to identify spam email messages from the same campaign as they are often slightly different. The presence of hosts with dynamic IP addresses makes counting number of machines in a botnet hard. Lastly, the logs we analyze is large in size (>1TB in our experiment). A useful method has to scale to datasets of this and potentially larger sizes. Our work answers all these challenges.

The primary contributions of our work are:

- We are the first to analyze entire botnets (in contrast to individual bot) behavior from spam email messages. We propose and evaluate methods to identify bots and cluster bots into botnets using spam email traces.
- Our work is the first to study botnet traces based on economic motivation and monetizing activities. Our approach analyzes botnets regardless of their internal organization and communication. Our approach is not thwarted by encrypted traffic or customized botnet protocols, unlike previous work using IRC trackers [6, 11] or DNS lookup [14, 15, 17].
- We report new findings about botnets involved in email spamming. For example, we report on the relationship between botnets usage and basic properties such as size. We also confirm previous reports on capabilities of botnet controllers and botnet usage patterns.

We successfully found hundreds of botnets by examining a subset of the spam email messages received by Hotmail Web mail service. The sizes of the botnets we found range from tens of hosts to more than ten thousand hosts. Our measurement results will be useful in several ways. First, knowing the size and membership gives us a better understanding on the threat posed by botnets. Second, the membership and geographic locations are useful information for deployment of countermeasurement infrastructures, such as firewall placement, traffic filtering policies, etc. Third, characterizing botnets behavior in monetizing activities may help in fighting against botnets in these businesses, perhaps reduce their profits in sending spam, generating click fraud, and other nefarious activities. Finally, such information about botnets may also give law enforcement help in combating illegal activities from botnets. We believe that the techniques presented here may also be applicable to related domains, such as identifying botnet membership through click fraud (analogous to spam) identified in search engine click logs (analogous to email traces).

The rest of the paper is organized as follows. We compare our work with related work in Section 2. We present our approach of extracting bots and botnets by mining spam emails in Section 3 and 4. We describe the results of our analysis in Section 5. Finally, we conclude in Section 6.

2 Related Work

Techniques to gather botnets for study fall mainly into two categories [15]. The first category of techniques collect botnets traffic from the "inside", using IRC channel infiltration[3, 6, 11] or traffic redirection [8]. The second category of techniques track botnets from external traces, for example, using DNS lookup information [14, 17], or flow data across a large Tier 1 ISP network [12]. Our work falls into the second category, using spam email messages as the external trace of botnets. This data source is interesting because it is relatively easy to collect and comprehensive in nature. In comparison, DNS probing [14, 15, 17] requires extra queries to DNS servers. The tracking capability could be limited by the querying rate to DNS servers.

While previous work focuses on traffic generated by botnets, our work is the first to study botnet traces based on economic motivation and monetizing activities. Along this direction, we expect a new category of traces can be used to characterize botnets from different perspectives (see Section 6). Our work takes activities from individual bots and aggregates them into botnets. The aggregation techniques proposed in this paper may generally benefit analysis of other traces in this category.

Several previous studies [2, 16] use spam email messages collected at a single or small number of points to gain insight into different aspects of the Internet. SpamScatter [2] clusters spam email based on the destination website linked to from the spam email messages, mainly for studying the machines hosting these *landing page*. In contrast, we cluster email based on content and study the source (i.e. sending) infrastructure. Ramachandran and Feamster [16] also studies the interaction between spam email messages and botnets. However, they do not infer botnet memberships from spam email data. Their work is more about characteristics of bots in general and studies network-level characteristics among all email messages and sender addresses (or bots).

3 Overview

Our technique takes as input a large dataset of spam email messages, collected at Hotmail over a period of days to weeks, and outputs a list of probable botnets involved in generating these spam messages and their corresponding statistics (such as sizes, activity over time and the geographic distribution of participating hosts).

The major steps involved in identifying the botnets are briefly described below. The next section presents them in detail.

1. **Cluster email messages into spam campaigns.** We assume that spam email messages with identical or similar content are sent from the same controlling entity. Our first step is to identify these groups of messages, which we will refer to as *spam campaigns*. A lot of spam messages from the same campaign are similar but not identical, to evade detection. We use shingling [4] to efficiently group them. The basic idea is to compute a number of fingerprints (e.g. 10) for each message, and messages sharing more than a few common fingerprints are those identical or very close in content.

2. **Assess IP dynamics.** Hosts with dynamic IP addresses will affect our results by raising the estimation of hosts involved over a period of time. We use a model to reverse this effect by computing parameters of IP dynamics for different parts of the IP address space. Concretely, for each C-subnet, we extract 1) the average time until an IP address gets reassigned; 2) the IP reassignment range. Using these parameters, we propose a way to estimate the probability whether two spam messages sent at different times are initiated from the same machine. This approach bears resemblance to [18].
3. **Merge spam campaigns into botnets.** Multiple spam campaigns can come from the same botnet. Based on the first two steps, we merge individual spam campaigns together into a set of spam campaigns initiated by the same *botnet* if the sending hosts significantly overlap. For each spam message in a spam campaign, we estimate the likelihood that the sending host also participates in another spam campaign, taking IP dynamics into account. Then, if a large number of senders participate in both spam campaigns, we merge the two together.

As we work with large datasets ($>$1TB), the steps above poses formidable computational challenges for a single computer. We design most of our algorithms to use the MapReduce [10] model and run them on a cluster of 120 computers, such that the experiments have acceptable turnaround times. Due to space limitation, however, we do not cover these implementation details in this paper.

4 Methodology

In this section, we discuss in detail our approach to extracting botnet membership by analyzing spam email data. We first define a set of terms used in the discussion below.

- A *spam* email message is an unsolicited bulk email message, often sent to many people with little or no change in content.
- A *spam campaign* is a set of email messages with the same or almost the same content, or content that is closely related–e.g. linking to the same target URL.
- A *botnet* is a set of machines that collaborate together to run one or more spam campaigns.

4.1 Datasets and Initial Processing

We work with an email dataset collected from the Hotmail Web mail service, referred to hereafter as the "Junk Mail Samples (JMS)" dataset. It is a randomly-sampled collection of messages reported by users or automatical identified as spam, containing about 5 million spam messages collected over a 9-day period from May 21, 2007 to May 29, 2007. The sample rate of JMS dataset is 0.001. The size of the dataset is about the same as the one used in [16] (collected over 1.5 years however), and one order of magnitude larger than that used in [2] (collected in 7 days). We think the 9-day duration is reasonable given the fact that spam campaigns change fast over time [2].

We do some initial processing of the raw-format messages before the next step. The first is to extract a reliable sender IP address heuristically for each message. Although the message format dictates a chain of relaying IP addresses in each message, a malicious relay can easily alter that. Therefore we cannot simply take the first IP in the chain. Instead, our method is as follows (similar to the one in [5]). First we trust the sender IP reported by Hotmail in the *Received* headers, and if the previous relay IP address (before any server from Hotmail) is on our trust list (e.g. other well-known mail services), we continue to follow the previous *Received* line, till we reach the first unrecognized IP address in the email header. This IP address is then taken as the email source. We also parse the body parts to get both HTML and text from each email message. In the end, we have for each message the sending time and content (HTML/plaintext) along with sender IP address.

4.2 Identifying Spam Campaigns

A spam campaign consists of many related email messages. The messages in a spam campaign share a set of common features, such as similar content, or links (with or without redirection) to the same target URL. By exploiting this feature, we can cluster spam email messages with same or near-duplicate content together as a single spam campaign.

Spammers often obfuscate the message content such that each email message in a spam campaign has slightly different text from the others. One common obfuscating technique is misspelling commonly filtered words or inserting extra characters. HTML-based email offers additional ways to obfuscate similarities in messages, such as inserting comments, including invisible text, using escape sequences to specify characters, and presenting text content in image form, with randomized image elements.

The algorithm to cluster together spam email messages with the same or near-duplicate content must be robust enough to overcome most of the obfuscation. Fortunately, most obfuscation does not significantly change the main content of the email message after being rendered, because it still needs to be readable and deliver the same information. Thus, we first use ad hoc approaches to pre-clean the raw content and get only the rendered content, and then use the shingling [4] algorithm to cluster near-duplicate content together. The basic idea is to generate a set of fingerprints that represent the pre-cleaned content of each message. If two messages share significant number of fingerprints, they will be marked as "connected" in content.

Now, we consider each email message as a node in a graph, and draw an edge between two nodes if the corresponding two messages are connected in content, or share the same embedded links. We then define each connected component in the graph as a *spam campaign*. Using the Union-Find algorithm [7], we can label all connected components on the graph, with each label representing a spam campaign. We can thus generate a list of detected spam campaigns. To assign labels, we associate each spam campaign with the list $\{(\text{IP}_i, t_i)\}$ of *IP events* consisting of the IP address IP_i and sending time t_i extracted from each email message in the campaign.

Text shingling is only one possible approach to group emails into spam campaigns. Other ways to do so is complementary to our overall approach. For example, one could use the target URL-based approach proposed in [2] to find spam campaigns. Different approaches have different pros and cons. For example, text shingling certainly cannot handle spam messages that are completely images, while the URL-based approach will miss spam campaigns that contain different URLs in messages and then redirect to the same website.

4.3 Skipping Spam from Non-bots

Many spam messages are not sent from botnets. We use a set of heuristics to filter out these messages.

- We build a list of known relaying IP addresses, which includes SMTP servers from email service providers, ISP MTA servers, popular proxies, open relays, etc. If the sender IP address of a message (extracted in Section 4.1) is on this list, we exclude that email from further analysis, as these servers are only relaying others' messages.
- We also remove campaigns whose senders are all within a single C-subnet, which is likely to be owned by the spammer himself instead of bot machines.
- Some more powerful spammers may employ multiple connections at the same physical location to directly send spam. Therefore we employ another rule that removes campaigns with senders from less than three geographic locations (cities).

Admittedly, the above list cannot remove all non-botnet spam campaigns. We try to strike a balance between letting too many non-botnet campaigns in and removing wrongly too many botnet-originating campaigns. Hotmail already blocks most spam messages from spammer servers and many open relays using volume-based and other policies. Moreover, we are confident that spam campaigns originating from hundreds or even thousands of geographic locations are operated by botnets. Finding ways to clearly characterize the nature of campaigns coming from smaller numbers of geographic locations is future work.

4.4 Assessing IP Dynamics

Many home computer users currently connect to the Internet through dial-up, ADSL, cable or other services that assign them new IP addresses constantly — anywhere from every couple of hours to every couple of days. This affects our estimation of number of hosts involved in each spam campaign. We correct this by estimating how "dynamic" each IP address is, and compensate by "merging" some dynamic IP addresses with other IP addresses in the same spam campaign.

The problem of IP dynamics was first presented and studied in [18]. However, we are not able to directly use their results because our application requires a different set of parameters. We design a similar but different approach of estimating IP dynamics:

We begin by assuming that within any particular C-subnet, the IP address reassignment strategy is uniform. We also assume that IP address reassignment is a Poisson process and measure two IP address reassignment parameters in each C-subnet: the average lifetime J_t of an IP address on a particular host, and the maximum distance J_r between IP addresses assigned to the same host.

The dataset from which J_t and J_r are measured is the log of 7 days' user login/logout events (June 6-12, 2007) from the MSN Messenger instant messaging service. For each login/logout event, we obtain an anonymized username and IP address for that session. We then associate login/logout events for the same username to construct a sequence:

$$\text{username}: \begin{array}{l} (\text{IP}_1, [\text{login-time}_1, \text{logout-time}_1]), \\ (\text{IP}_2, [\text{login-time}_2, \text{logout-time}_2]), \\ (\text{IP}_3, [\text{login-time}_3, \text{logout-time}_3]), \\ \ldots \end{array}$$

We assume that each user connects to the MSN Messenger service from a small, fixed set of machines (e.g. an office computer and a home computer), and detect cases where multiple IP addresses are associated with a particular username. We label each such change as an IP address reassignment if the IP addresses are sufficiently "close": we define "close" as within a couple of consecutive B-subnets; otherwise, we assume that two different machines are involved. We then aggregate our detection among all IP addresses in the same C-subnet and remove anomalous events. We then calculate, based on the Poisson process assumption, J_t and J_r for each individual C-subnet.

Thus, given two IPs at two different times, (IP_1, t_1) and (IP_2, t_2), if either IP_1 or IP_2 is out of the distance range (J_r) of another, we regard these two events as from two different machines. If both IP_1 and IP_2 are within the distance range (J_r) of each other, we make the computation below.

$$\mathbb{P}[\text{IP}_1 = \text{IP}_2 | \text{ actually the same machine}]$$

$$= \frac{J_r - 1}{J_r} \exp\left(\frac{-(t_2 - t_1)}{J_t}\right) + 1/J_r = w(t_1, t_2).$$

This is the probability that a machine has kept the same IP address after an interval of duration $t_2 - t_1$.

$$\mathbb{P}[\text{IP}_1 \neq \text{IP}_2 | \text{ actually the same machine}]$$
$$= \frac{J_r - 1}{J_r} \left[1 - \exp\left(\frac{-(t_2 - t_1)}{J_t}\right)\right] = 1 - w(t_1, t_2).$$

This is the probability that a machine changes its IP address – that is, that an IP reassignment happens – during an interval of duration $t_2 - t_1$.

Figure 1 shows the Probability Density Function (PDF) of IP reassignment time among all C-subnets (about 25% of C-subnets never see IP reassignment in the 7 day log). According to the figure, a large portion of IP addresses get reassigned almost every day.

4.5 Identifying Botnets

Each spam campaign is represented as a sequence of events (IP, t), where each event is a spam email message that belongs to the spam campaign. The question is, given two spam campaigns SC_1 and SC_2, how do we know whether they share the same controller (i.e. they are part of the same botnet)? We put two spam campaigns in the same botnet if their spam events are significantly connected. We now define the connection between two spam campaigns.

Given a event (IP_1, t_1) from spam campaign SC_1 and a event (IP_2, t_2) from spam campaign SC_2, we assign a connection weight between them. The connection weight is the probability that these two events would be seen if they were actually from the same machine. We have defined this probability in Section 4.4, i.e. $w(t_1, t_2)$ if two IP addresses are equal, or $1 - w(t_1, t_2)$ if two IP addresses are not equal but within distance range of each other, or 0 otherwise. For all events in a spam campaign SC_1, we use

$$W = \frac{\sum_i \max_j [w(t_i, t_j) \text{ or } (1 - w(t_i, t_j)) \text{ or } 0]}{|\text{SC}_1|}$$

to measure the fraction of events in SC_1 that are connected to some events in SC_2, where i and j represents IP events in SC_1 and SC_2. W, called as *connectivity degree*, ranges from 0 to 1. If this W is large, it means a significant portion of the events in SC_1 are connected to events in SC_2, and thus, we should merge SC_1 into SC_2.

We use the connectivity degree W to decide whether we should merge a spam campaign into another as they are in the same botnet. We expect a bimodal pattern in the distribution of W: a large portion of W values are small, which correspond to pairs of non-connected spam campaigns; while a small portion of W values are relatively large, which correspond to pairs of spam campaigns from the same botnet; there are few W values in the middle. The W value in the middle is a reasonable threshold to merge spam campaigns. The PDF of W in Figure 2 meets our expectation. Based on this, we select 0.2 as a reasonable threshold to decide whether a spam campaign should be merged to another. In our experiments, we also test thresholds from 0.05 to 0.35, and we found that this change had very little effects to the botnet detection results. Because the detection is not sensitive to the threshold, it gives us more confidence in the validity of the clustering.

The connectivity degree W is also related to the way that botnet controllers use their botnets. If a botnet controller always use all its bots to run each spam campaign, we will observe that each spam campaign has $W = 1$ to other spam campaigns from this botnet. However, as we will show in Section 5.2 botnet controllers use only a subset of available bots each time.

4.6 Estimating Botnet Size

Now, each botnet contains a sequence of events (IP, t) that correspond to all spam sent by this botnet. We want to identify distinct machines that generate these events. In Section 4.4, we have already defined the probability that two events are from the same machine. We use this definition to examine events in a botnet: when an event (IP_2, t_2) is from the same machine of a previous event (IP_1, t_1), IP_2 is a reoccurrence of IP_1. So, we can estimate the probability that an IP address is a reoccurrence of any previous IP address:

$$c = 1 - \prod_i \mathbb{P}[\text{IP is not a reoccurrence of IP}_i],$$

where i ranges over all events that happen before this IP event. The value of c equals 1 if the IP address is a reoccurrence, 0 if the IP address is not a reoccurrence. We can count the number of distinct machines appeared in the downsampled dataset (JMS) in this way.

Furthermore, we want to estimate the total size of botnets from the downsampled dataset (JMS). We assume bots in the same botnet behave similarly — each bot sends approximately equal number of spam messages.

We define the following quantities:

- r: downsample rate of the dataset
- N: number of spam email messages observed
- N_1: number of bots observed with only one spam email in the dataset

We want to measure botnets size and number of spam email messages sent per bot:

- s: the mean number of spam messages sent per bot
- b: number of bots (i.e. botnet size)

The estimated number of spam email messages from a botnet is $N/r = sb$. The expected number of bots observed with only one spam email message is

$$N_1 = b\left[r(1-r)^{s-1}\binom{s}{1}\right] = N(1-r)^{s-1}$$

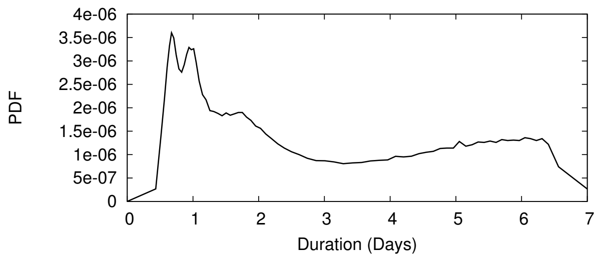

Figure 1: PDF of IP Reassign Duration

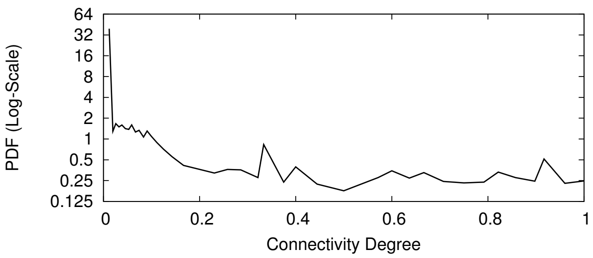

Figure 2: PDF of the Campaign Merge Weight

Thus, we get the average number of spam email messages sent per bot (s) and botnet size (b):

$$s = \frac{\log(N_1/N)}{\log(1-r)} + 1,\ b = \frac{N}{rs}$$

5 Metrics and Findings

In this section, we present results on metrics and characteristics of botnets, and their behavior in sending spam messages. These metrics are measured on spam campaigns and botnets detected as described in Section 4.

5.1 Spam Campaign Duration

The duration of spam campaigns, defined as the time between the first email and the last email seen from a campaign, is an important metric about behavior of botnets. Here we present measurement of this in the JMS dataset. Note that this is often *different* from the lifetime of the botnets themselves, as spammers often rent the same botnets to launch multiple spam campaigns over time.

We get our results using the following method. We look at those spam campaigns that happen to appear first on the second day in our dataset and count how many days they last. We do not look at those appearing on the first day because they may well be already running before that day. And as most campaigns run continuously, starting from the second day is likely enough to ensure that these campaigns do indeed start on that day. Additionally, we remove 7% of the spam email in the JMS dataset because there are not enough similar spam messages for these campaigns to give reliable results — these email messages might be user introduced or automatical detected false positives.

Figure 3 shows the Cumulative Distribution Function (CDF) of spam campaign durations. We can see that over 50% of spam campaigns actually finish within 12 hours. After that the durations distributed rather evenly between 12 hours to 8 days, and about 20% of campaigns persist more than 8 days.

Figure 4 shows the CDF of each spam campaign, weighted by email volume. Comparing this to Figure 3, we can see that short-lived spam campaigns actually have larger volume. In particular, more than 70% of spam messages are sent by spam campaigns lasting less than 8 hours.

5.2 Botnet Sizes

The capability of botnet controllers and level of activity of botnets are two important metrics for understanding botnets. To measure the capability, we need to estimate the total size of each botnet based on our 9 days of observation. To measure the level of activity, we estimate the active working set of each botnet in a short time window, such as one hour. As botnet population is dynamic over time, we use “botnet size” to refer to the estimated number of bots actually used for activities during our time window. This size is estimated as explained in Section 4.6. Infected machines are often not cleaned for several weeks. During the period of infection, machines have activities at least every few days. Thus, bots actually used during an observation window of nine days give a good approximation of the number of machines controlled by a botnet controller. If we do not consider the IP dynamics, the number of distinct IP addresses appeared in JMS dataset could be two times larger than the number of distinct machines estimated.

We detected 294 botnets in the JMS dataset and the following measurements are based on these 294 botnets. The estimated total sizes of botnets indicates of an upper bound on the capabilities of spammers or botnet controllers – they likely have only compromised this many machines total. Issues such as proxy and NAT could affect the accuracy of the botnet size estimation. This is a topic for future study.

Figure 5 shows the CDF of estimated botnet size[1]. In our dataset, the estimated total sizes of botnets ranges from a couple of machines to more than ten thousands machines; about 50% of botnets contain over 1000 bots each, which is consistent with a similar metric in [15]. The number of spam email messages sent per bot ranges from tens to a couple of thousands during the 9-day observation window (Figure 6). Some botnet controllers are conservative in limiting number of spam email messages sent per bot.

[1] This is the estimation of the number of bots actually used, not just those seen in our dataset.

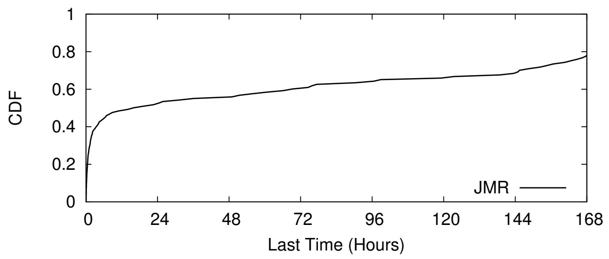

Figure 3: CDF of spam campaign duration

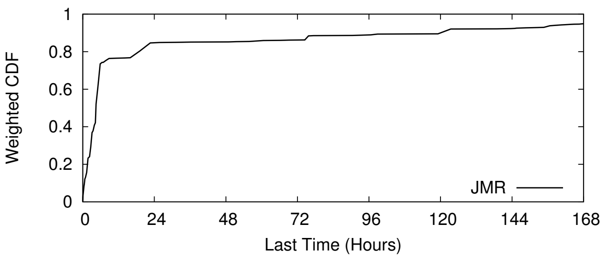

Figure 4: CDF of spam campaign duration weighted by email volume

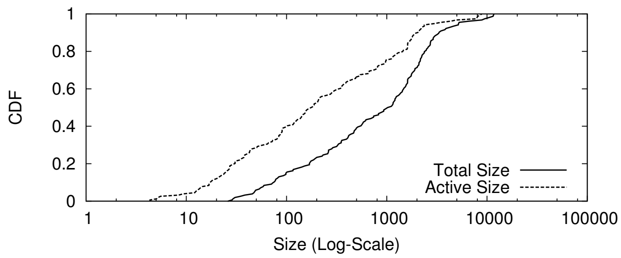

Figure 5: CDF of botnet size

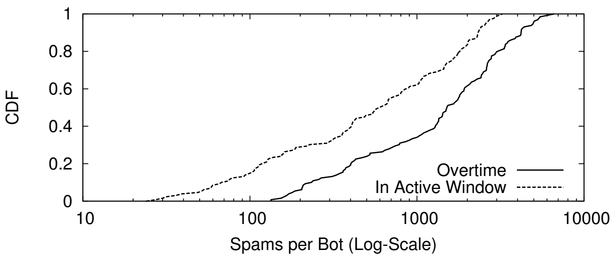

Figure 6: CDF of spam email messages sent per bot

5.2.1 Active Size vs. Level of Activity

In a time window t (1 hour in our experiments):

- "Active size" of a botnet is defined as the number of machines/IPs used for sending spam email messages by this botnet during this time window t.
- "Spam sent per active bot" in a botnet is defined as number of spam email messages sent from each bot in a botnet during this time window t.

In the experiment, we study events of a botnet in each time window t during the 9-day duration. Since we limit t to one or a couple of hours, we can reasonably assume that IP reassignment does not happen. To measure the active size and number of spam email sent per active bot during all time windows (1 hour each), we calculate characteristics in each time window and then average results over all time windows during the 9-day period.

The active size of a botnet and the number of spam email messages sent per active bot has strong impact on the efficiency and effectiveness of IP blacklisting or volume-based filters in filtering spam sent by botnets. Spammers generally use two method to evade IP based filtering: 1) they send fewer spam messages per bot (which looks like legitimate use); 2) they use a small portion of machines at one time and round-robin among all machines in their control.

Figure 7 shows the relationship between the average active size of a botnet and the number of spam email messages sent per active bot. We see that large-size botnets tend to send less spam per bot, small-size botnets tend to send more spam per bot, while mid-size botnets behave both ways. This suggests that spam controllers may have clear plans about the number of spam messages to be sent, and then stop after these goals are met. Alternatively, the number of email addresses that spammers possess may limit the total number of spam messages sent from their botnets. We also find that there is no significant relationship between active botnet duration and the number of spam messages sent per bot. Taken together with Figure 7, we conclude that botnet size is the primary factor that determines the number of spam messages sent per bot.

5.2.2 Activity Ratio

The *activity ratio* is defined as the ratio of active size to estimated total size of a botnet. The activity ratio in each time window (one hour) is calculated and then averaged over 9 days. The average activity ratio ranges in (0, 1]. The value of 1 means the botnet uses all machines it controls; while 0 means a botnet uses none of its machine. The average activity ratio indicates whether botnets controllers use all machines they have, or use a small fraction of machines and round robin among these machines.

Figure 8 is the CDF of activity ratio of botnets. About 80% of botnets use less than half of bots at a time in their network. We find that the activity ratio and the total size are not related. That is, in general, a botnet controller might use any portion of bots in his or her control regardless of the total number controlled.

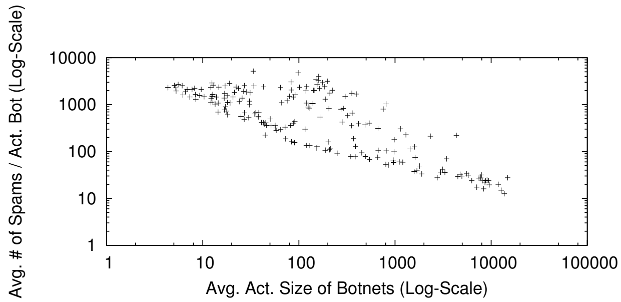

Figure 7: Average active size of botnets vs. average number of spam email messages sent per active bot

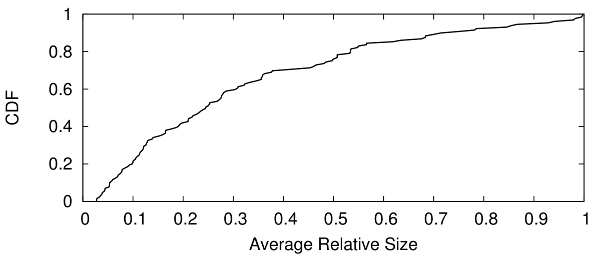

Figure 8: CDF of activity ratio of botnets

5.3 Per-day Aspect: Life Span of Botnets and Spam Campaigns

If we look at all spam email messages received in a day by an email server (or an end user), how much spam is from long-lived botnets or spam campaigns? If a new botnet is being used every day for a new spam campaign, monitoring botnets might not be helpful to anti-spam filters. However, if some botnets are devoted to the spamming business, identifying these botnets is more promising.

We study the duration of botnets and spam campaigns on a per-day basis. We look at spam email messages received on a particular day, identify botnets or spam campaigns these spam messages belong to, and compute the distribution of botnets and spam campaigns with activity on that particular day.

In our experiment, we study botnets with activity on the last day of our 9-day observation window, and then look backward to their first activities. Each botnet is at least active for x $(1 \leq x \leq 9)$ days. Figure 9 shows that about 60% of spam received from botnets each day are sent from long-lived botnets. This is a good indication that monitoring botnet behavior, membership, and other properties using the approaches proposed in this paper can help to reduce significantly the amount of spam received on a daily basis.

5.4 Geographic Distribution of Botnets

The geographic distribution of botnets is an important metric about the ability of botnet controllers in compromising and taking over machines. Figure 10 shows that about half of botnets detected from the JMS dataset control machines in over 30 countries. Some botnets even control machines in over 100 countries. This shows that currently botnets are very widely distributed, in part because of the wide distribution of malwares, viruses, etc. It could also because malicious people have developed more sophisticated means to control widely distributed machines efficiently. Others have observed that a botnet typically sends spam messages with the same topic from all over the world, especially from those IP ranges assigned to dial-up, ADSL or cable services [1]. The wide geographic distribution in our results is consistent with their observations. Using the estimation method proposed in Section 4.6, the total number of bots involved in sending spam email all over the world during the 9-day observation period of the JMS dataset is about 460,000 machines.

6 Conclusion and Future Work

Our work is a first step to study botnets from their economic motivations. By directly tracing the actual operation of bots using one of their primary revenue sources (spam email), we get a picture of bot activity: one that confirms and deepens the understanding suggested by previous work. By identifying common characteristics of spam email, we associated email messages with botnets. This allows us to make estimates about the size of a botnet, behavioral characteristics (such as the amount of spam sent per bot), and the geographical distribution of botnets.

We hope our work opens new directions in understanding botnet activities. We think there are at least a couple of interesting future directions. First, we want to validate the results detected from spam email by cross-referencing with results inferred using other techniques such as IRC infiltrating. Comparing with other detection results will also let us know the portion of botnets that do not spam at all, which are missed from our approach. Second, we want to use detection results in this paper as an extra source of information to filter spam email. For example, we assign different volume thresholds to senders belong to different botnets given their previous behavior. We may also check the existence of same botnets in query log or ad click log. Third, certain techniques such as image shingles can to be used together to cluster image-based spam email messages. Finally, we want to further study possible countermeasurements from botnet controllers in order avoid being detected by our approach.

7 Acknowledgements

The first author did this work while she was a summer intern at Microsoft Research. This work was also supported in part by TRUST (Team for Research in Ubiq-

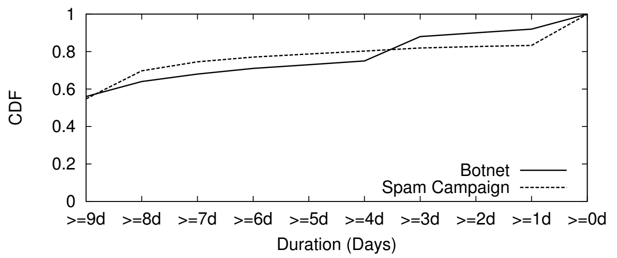

Figure 9: CDF of botnets and spam campaign duration from a per-day-activity aspect

uitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244), Cisco, British Telecom, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia, and United Technologies. The opinions in this paper are those of the authors and do not necessarily reflect the views of their employers or funding sponsors.

References

[1] Shadow server. `http://www.shadowserver.org/`.

[2] ANDERSON, D. S., FLEIZACH, C., SAVAGE, S., AND VOELKER, G. M. Spamscatter: Characterizing internet scam hosting infrastructure. In *USENIX Security'07*.

[3] BINKLEY, J. R., AND SINGH, S. An algorithm for anomaly-based botnet detection. In *SRUTI'06*.

[4] BRODER, A. Z., GLASSMAN, S. C., MANASSE, M. S., AND ZWEIG, G. Syntactic clustering of the web. In *WWW'97*.

[5] BRODSKY, A., AND BRODSKY, D. A distributed content independent method for spam detection. In *HotBots'07*.

[6] COOKE, E., JAHANIAN, F., AND MCPHERSON, D. The zombie roundup: understanding, detecting, and disrupting botnets. In *SRUTI'05*.

[7] CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L., AND STEIN, C. *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.

[8] DAGON, D., ZOU, C., AND LEE, W. Modeling botnet propagation using time zones. In *NDSS'06*.

[9] DASWANI, N., STOPPELMAN, M., AND THE GOOGLE CLICK QUALITY AND SECURITY TEAMS. The anatomy of clickbot.a. In *HotBots'07*.

[10] DEAN, J., AND GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. *Commun. ACM 51*, 1 (January 2008), 107–113.

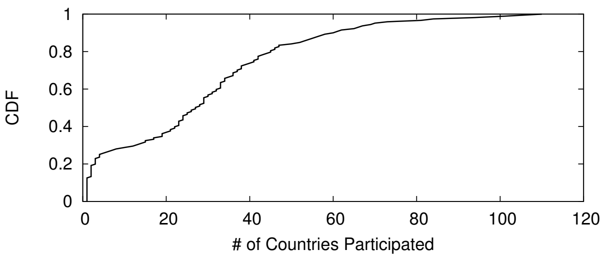

Figure 10: Number of countries in botnets

[11] FREILING, F. C., HOLZ, T., AND WICHERSKI, G. Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. In *ESORICS'05*.

[12] KARASARIDIS, A., REXROAD, B., AND HOEFLIN, D. Wide-scale botnet detection and characterization. In *HotBots'07*.

[13] KRASSER, S., CONTI, G., GRIZZARD, J., GRIBSCHAW, J., AND OWEN, H. Real-time and forensic network data analysis using animated and coordinated visualization. In *IAW'05*.

[14] RAJAB, M. A., ZARFOSS, J., MONROSE, F., AND TERZIS, A. A multifaceted approach to understanding the botnet phenomenon. In *IMC'06*.

[15] RAJAB, M. A., ZARFOSS, J., MONROSE, F., AND TERZIS, A. My botnet is bigger than yours (maybe, better than yours). In *HotBots'07*.

[16] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the network-level behavior of spammers. In *SIGCOMM'06*.

[17] RAMACHANDRAN, A., FEAMSTER, N., AND DAGON, D. Revealing botnet membership using dnsbl counter-intelligence. In *SRUTI'06*.

[18] XIE, Y., YU, F., ACHAN, K., GILLUM, E., GOLDSZMIDT, M., AND WOBBER, T. How dynamic are ip addresses? In *SIGCOMM'07*.