Spamming Botnets: Signatures and Characteristics

Yinglian Xie, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten[+], Ivan Osipkov[+]
Microsoft Research, Silicon Valley
[+]Microsoft Corporation
{yxie,fangyu,kachan,rina,ghulten,ivano}@microsoft.com

ABSTRACT

In this paper, we focus on characterizing spamming botnets by leveraging both spam payload and spam server traffic properties. Towards this goal, we developed a spam signature generation framework called *AutoRE* to detect botnet-based spam emails and botnet membership. AutoRE does not require pre-classified training data or white lists. Moreover, it outputs high quality regular expression signatures that can detect botnet spam with a low false positive rate. Using a three-month sample of emails from Hotmail, AutoRE successfully identified 7,721 botnet-based spam campaigns together with 340,050 unique botnet host IP addresses.

Our in-depth analysis of the identified botnets revealed several interesting findings regarding the degree of email obfuscation, properties of botnet IP addresses, sending patterns, and their correlation with network scanning traffic. We believe these observations are useful information in the design of botnet detection schemes.

Categories and Subject Descriptors

C.2.3 [**Computer Communication Networks**]: Network Operations—*network management*; C.2.0 [**Computer Communication Networks**]: General—*security and protection*

General Terms

Algorithms, Measurement, Security

Keywords

Spam, botnet, regular expression, signature generation

1. INTRODUCTION

Botnets have been widely used for sending spam emails at a large scale [14, 4, 19, 24]. By programming a large number of distributed bots, spammers can effectively transmit thousands of spam emails in a short duration. To date, detecting and blacklisting individual bots is commonly regarded as difficult, due to both the transient nature of the attack and the fact that each bot may send only a few spam emails. Furthermore, despite the increasing awareness of botnet infection and their associated control process [4, 17, 6], little effort has been devoted to understanding the *aggregate* behaviors of botnets from the perspective of large email servers that are popular targets of botnet spam attacks.

An important goal of this paper is to perform a large scale analysis of spamming botnet characteristics and identify trends that can benefit future botnet detection and defense mechanisms. In our analysis, we make use of an email dataset collected from a large email service provider, namely, MSN Hotmail. Our study not only detects botnet membership across the Internet, but also tracks the sending behavior and the associated email content patterns that are directly observable from an email service provider. Information pertaining to botnet membership can be used to prevent future nefarious activities such as phishing and DDoS attacks. Understanding the email sending behavior of botnets can facilitate the development of new botnet detection techniques.

Our investigation is based on a novel framework called *AutoRE* that identifies botnet hosts by generating botnet spam *signatures* from emails. AutoRE is motivated in part by the recent success of signature based worm and virus detection systems (e.g., [12, 21, 16, 15, 13]). The framework is based on the premise that botnet spam emails are often sent in an aggregate fashion, resulting in content prevalence similar to the worm propagation case. In particular, we focus primarily on URLs embedded in email content because they form the most critical part of spam emails – URLs play an important role in directing users to phishing Web pages or targeted product Web sites [2] [1].

However, the following two observations make it challenging to derive URL signatures that distinguish botnet spam from others. First, spam emails often contain multiple URLs, some of which are legitimate and very general (e.g., `http://www.w3.org`). The mixture of legitimate and spam URLs in an email requires us to clearly separate them. Second, spammers deliberately add randomness into URLs to evade detection. Therefore, sifting through polymorphic URLs to identify common patterns is a critical task.

AutoRE addresses the first challenge by iteratively selecting spam URLs based on the *distributed* yet *bursty* property of botnets-based spam campaigns. AutoRE does not require labeled data or whitelists, a common necessity in most previous solutions. AutoRE further outputs *regular expression* signatures that are different from traditional worm signatures that consist of either fixed strings or token conjunctions (token1.*token2.*token2). Compared with complete URL (fixed string) based signatures, regular expression signatures are *more robust* and can detect 10 times more spam emails. Compared with token conjunction based signatures, regular expression

[1] Based on an analysis of sampled emails sent to Hotmail, we found that 74.1% of spam emails contained at least one URL (with the remainder mostly geared towards campaigns for penny stocks)

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.

signatures can significantly *reduce the false positive rate* of detecting polymorphic URLs (by 10 to 30 times in our experiments).

Furthermore, AutoRE uses the generated spam URL signatures to group emails into *spam campaigns*, where a campaign refers to a targeted spam effort to a single product or service. In this paper, we identify spam campaigns originating from botnets. Using three months of sampled emails from Hotmail, AutoRE successfully detected 7,721 spam campaigns that originated from 340,050 distinct botnet host IP addresses spanning 5,916 ASes. Below, we briefly summarize several desirable features of AutoRE:

- *Low false positive rate:* Using AutoRE signatures, we identified 580,466 spam emails with a false positive rate of 0.002. AutoRE's false positive rate in detecting botnet hosts is less than 0.005.
- *Ability to detect stealthy botnet-based spam:* AutoRE detects 16-18% of spam that bypassed well known blacklists (e.g., Spamhaus [22]) that are deployed by major mail providers.
- *Ability to detect frequent domain modifications:* Using domain-agnostic signatures, AutoRE is able to capture spam URLs even if spammers adopt new domains. This enables us to filter 15 times more spam than using domain specific signatures.

Another important contribution of our work is an in-depth analysis of identified spamming botnet characteristics and their activity trends. Our key findings include:

(1) By comparing botnet statistics in July 2007 to those obtained in Nov 2006, we noticed that the number of spam campaigns doubled, while the total number of botnet IPs increased by only 10%. This indicates that botnets are becoming an increasingly popular mechanism for spam delivery and that one botnet host is involved in multiple attacks.

(2) Even though a spam campaign directs email users to the same/similar set of destination Web pages, the text content of the emails in a campaign varies significantly. When viewed individually, a botnet host does not exhibit distinctive sending patterns compared to a legitimate host. These observations suggest that detecting botnet hosts individually based on their email text or sending features is difficult.

(3) However, as an aggregate, spam emails from botnets are often sent in a highly synchronized fashion. Hosts within the same campaign also exhibit similar sending patterns (e.g., the number of recipients per email and connection rate). Surprisingly, many distinct spam campaigns behave similarly, suggesting that they may all utilize the same spam sending tools. This implies that we may detect botnet hosts by looking for aggregated common features from concurrent email sending activities.

(4) Finally, we correlate the botnet activities with the network telescope data [7]. Our analysis reveals that botnet attacks might have different phases. Thus, it may be possible to identify them by exploring network scanning patterns.

In the rest of this paper, we first discuss related work and the challenges in our work (Section 2). We then present the AutoRE data processing flow (Section 3) and elaborate on our regular expression generation scheme (Section 4). We present our experimental results in Section 5, and describe our evaluation in Section 6. Using the inferred botnet membership information, we study their characteristics in Section 7. Finally, we discuss the limitations of the AutoRE framework (Section 8) before we conclude.

2. BACKGROUND AND CHALLENGES

The term *botnet* refers to a group of compromised host computers that are controlled by a small number of commander hosts referred to as *Command and Control (C&C) servers*. Due to the increased use of botnets for launching large-scale network attacks, several recent studies have looked at different aspects of bot activities: infection process [6], communication channels between bots and C&C servers [4, 17], and propagation strategies [5, 10, 11].

Identification and prevention of email spam that originate from botnets is the primary focus of this paper. In this problem space, Ramachandran et al. [19] performed a large scale study of the network-behavior of spammers, providing strong evidence that botnets are commonly used as platforms for sending spam. Anderson et al. studied interesting characteristics of Internet scam infrastructures using Spamscatter [2], a system that analyzes spam email URLs. Webb et al. studied the link between spam emails and Web spam using a large Web spam corpus [23]. More recently, Ramachandran et al. proposed ways to infer botnet membership and identify spammers by monitoring queries to DNSBL [18] and by clustering email servers based on their target email destination domains [20].

These approaches have all provided insight into various aspects of spamming activities and successfully explored opportunities for monitoring different spam traffic. In contrast, our work focuses on the problem of not just detecting botnet hosts, but also correctly grouping them based on spam campaigns. We hope such a collective view can shed light on how botnets operate and evolve as an aggregate to facilitate the detection of future attacks. We adopt the generic perspective of an email server receiving incoming traffic destined to a single domain without additional communication to other infrastructure servers or services. Any solution in such a setting could potentially be adopted autonomously by email service providers or ISPs.

In a similar context, Zhuang et al. showed that the similarity of email texts can help identify botnet-based spam campaigns [25]. Li and Hsish performed a measurement study where they examined various content features including spam URL links [14]. They found that spam emails with identical URLs are highly clusterable and are often sent in a burst. These observations motivated us to develop techniques that extract spam URL signatures for large-scale spamming botnet detection and analysis.

The spam URL signature generation problem is in many ways similar to the content-based worm signature generation problem that has been extensively studied (e.g., [21, 12, 16, 15, 13]). Despite the fact that botnet spam exhibits content prevalence like the worm propagation case, two challenges remain in practice, preventing us from directly adopting existing solutions:

First, spammers often add random, legitimate URLs to content in order to increase the perceived legitimacy of emails. Furthermore, HTML-based emails often contain URLs generated by standard software (e.g. compliance to HTML standards). Figure 1 shows an example of three emails all sharing the highlighted URL, but are also mixed with a number of other URLs. In our dataset, a total of 203 emails containing this highlighted URL were sent on the same day from 70 different IP addresses spanning 15 ASes. We suspect the corresponding hosts were from the same botnet.

Due to the mixing of legitimate and spam URLs in the email content, we cannot adopt the approach taken by many existing solutions (e.g., [12, 16, 15]) where traffic is pre-classified into legitimate and suspicious pools. Instead, AutoRE takes an approach similar to the one used in Earlybird [21] by seeking both content prevalence and source address dispersion. However, Earlybird uses a white list to remove false positives such as common protocol headers or P2P traffic. Although white listing is effective in the worm signature extraction case, we do not use this approach here as spammers can easily abuse legitimate Web sites. It was reported that Google's *feeling lucky* feature was exploited by spammers as a

Figure 1: Multi-URL spam emails that we suspect were sent from the same botnet. These emails were from different IP addresses, but were sent almost simultaneously.

Time	URLs	Source ASes	URLs
2006-11-02	66	38	http://www.lympos.com/n/?167&carthagebolets http://www.lympos.com/n/?167&brokenacclaim http://www.lympos.com/n/?167&acceptoraudience
2006-11-15	72	39	http://shgeep.info/tota/indexx.html?jhjb.cvqxjby,hvx http://shgeep.info/tota/indexx.html?ikjija.cvqxjby,hvx http://shgeep.info/tota/indexx.html?ivvx_ceh.cvqxjby,hvx

Figure 2: Examples of polymorphic URLs.

mechanism for redirection [2]. Instead, AutoRE ensures a low false positive rate by using an iterative approach to identify spam URLs, detailed in Section 3.

The second challenge arises from spammers' extensive use of URL obfuscation techniques to evade detection. Additionally, spammers often customize URLs to reflect recipients' email address, with the goal of tracking users that visit spamming web-sites. Figure 2 shows two examples of polymorphic spam URLs: the first group contained 66 URLs (only 3 are shown in the illustration) with random words inserted at the tails; these were sent from 38 ASes on a single day. The second group had 72 URLs, each attached with an encrypted email address.

Previous systems also looked at the problem of detecting polymorphic worms. These systems output keyword/token conjunction signatures like token1.*token2.* (e.g., [16, 15]). However, token conjunction based signatures cannot be directly applied to the URL case as URL strings are typically much shorter than worm binary executables. Furthermore, URL strings mostly contain human readable words and substring segments, suggesting that keywords (tokens) extracted from spam URLs may largely be short, regular, and predictable substrings. Looking at these substrings alone without checking the structure of the URLs could potentially result in a high false positive rate.

AutoRE goes one step further to generate regular expression signatures. As we will show in Section 5, compared to token conjunctions, regular expressions significantly increase the expressive power of signatures and in fact reduced the false positive rate by 30 times. To the best of our knowledge, this is the first successful attempt to automatically generate regular expression signatures.

3. AUTORE: SIGNATURE BASED BOTNET IDENTIFICATION

In this section, we present *AutoRE* – a framework for *automatically* generating URL signatures to identify botnet-based spam campaigns. As input, AutoRE takes only a set of unlabeled email messages (messages are not tagged as spam/non-spam), and produces two outputs: a set of *spam URL signatures*, and a related list of *botnet host IP addresses*. The resulting URL signature(s) could be either in the form of a complete URL string or a URL regular expression. These signatures can be used to identify both present and future spam emails that originate from botnets. The knowledge of botnet host identities can help filter other spam emails that could potentially originate from these infected hosts. In this paper, we did not consider using AutoRE in a real time fashion, though we discuss such a scenario in Section 8.

We emphasize that AutoRE is completely automatic. It does not require labeled training data in order to generate signatures. AutoRE operates by identifying unique behaviors exhibited by botnets – in particular it seeks to discover email traffic patterns that are *bursty* and *distributed*. The notion of "burstiness" reflects the fact that emails originating from botnet hosts are sent in a highly synchronized fashion as spammers typically rent botnets for a short period. The notion of "distributed" captures the fact that botnet hosts usually span a large and dispersed IP address space. AutoRE employs an iterative algorithm to identify botnet based spam emails that fit the above traffic profiles. Additionally, it generates "specific" regular expression signatures, where the learned signatures strive to encode maximal information about the matching URLs that characterize the underlying spam emails.

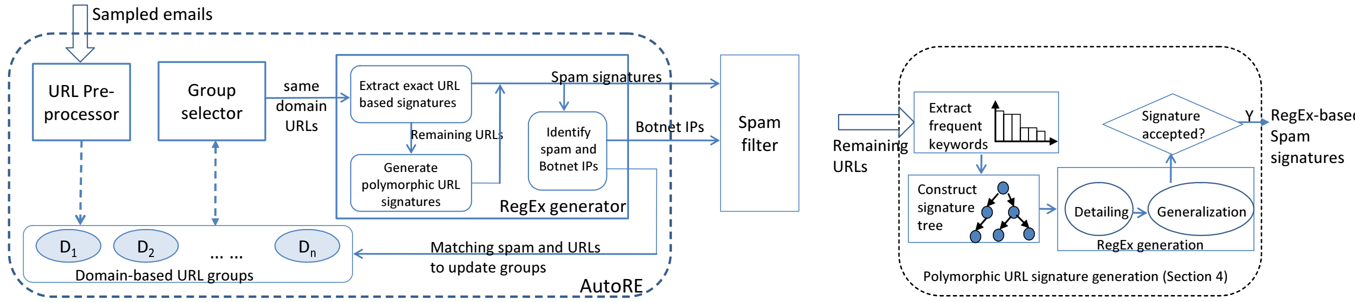
At a high level, AutoRE is comprised of the following three modules (Figure 3): a *URL preprocessor*, a *Group selector* and a *RegEx generator*. The URL preprocessor extracts URLs and other relevant fields from input emails and groups them according to Web domains. Each URL group is then treated as a candidate while identifying spam campaigns. The *Group selector* selects URL groups with the highest degree of burstiness in sending time and feeds such groups to the RegEx generator. Finally, the *RegEx generator* module extracts signatures by processing one group at a time. Every time a signature is generated by the RegEx generator, all its matching emails and associated URLs are discarded from the pool of remaining URL groups to avoid further consideration. This process is continued in an iterative fashion until all the groups are processed.

3.1 URL Pre-Processing

Given a set of emails, AutoRE begins by extracting the following information: *URL string, source server IP address* and *email sending time*. In addition, AutoRE assigns a unique *email ID* to represent the email from which a URL was extracted. During this process, we discard all forwarded emails (about 17% of total emails) from our analysis as this avoids mistakenly identifying a legitimate forwarding server as a botnet member.

The URL preprocessor then partitions URLs into groups based on their Web domains. This partitioning is motivated based on the observation that emails originating from the same spam campaign tend to advertise the same product or service from the same domain (we discuss URL redirection cases in Section 8). By grouping URLs from the same domain together, the search scope for bot-

[2]This problem at Google has been fixed after it was found. Google now warns users about the redirection.



(a) (b)

Figure 3: (a) AutoRE modules and processing flow chart. (b) Algorithmic overview of generating polymorphic URL signatures.

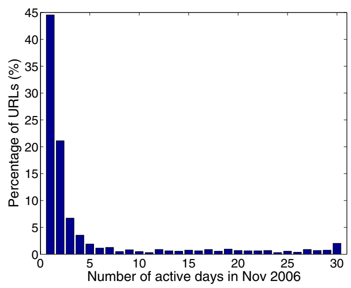


Figure 4: Active period of URLs sent from at least 20 ASes.

net signatures is significantly reduced. Later, these domain-specific signatures can be further merged to produce domain-agnostic signatures (see Section 4.2).

3.2 URL Group Selection

After preprocessing, each email might be associated with multiple groups, as the email may contain multiple URLs pertaining to different domains. A key question is, which group best characterizes an underlying spam campaign? To address this issue, AutoRE explores the bursty property of botnet email traffic. At every iteration, the Group selector greedily selects the URL group that exhibits the strongest temporal correlation across a large set of distributed senders. To quantify the degree of sending time correlation, for every URL group, AutoRE constructs a discrete time signal S, which represents the number of distinct source IP addresses that were active during a time window w. The value of the signal at the n-th window, denoted by $S_i(n)$, is defined as the total number of IP addresses that sent at least one URL in group i in that window. Intuitively, sharp signal spikes indicate strong correlation, meaning a large number of IP addresses all sent URLs targeting a common domain within a short duration. With this signal representation, we can compute a global ranking of all the URL groups at each iteration by selecting signals with large spikes. In this paper, for simplicity, at every iteration we favor the URL group with the narrowest signal width (breaking tie with the highest peak value).

3.3 Signature Generation and Botnet Identification

Given a set of URLs pertaining to the same domain, the RegEx generator returns two types of signatures: *complete URL based signatures* and *regular expression signatures*. Complete URL based signatures are geared towards detecting spam emails that contain an identical URL string. Regular expression signatures are more generic and powerful, as they can be used to detect spam emails that contain polymorphic URLs. In both cases, the generated signatures are required to meet the previously defined signature criteria: "distributed", "bursty", and "specific".

The "distributed" property is quantified using the total number of Autonomous Systems (ASes) spanned by the source IP addresses. Here, we choose the number of participating ASes rather than the number of IPs as it is possible for a large company to own a set of email servers with different IP addresses. In this paper, we primarily focus on detecting large botnets and conservatively require a signature to be associated with at least 20 ASes. This parameter is discussed later in Section 8.

We quantify the "bursty" feature using the inferred duration of a botnet spam campaign. In this paper, we enforce that the set of matching URLs should be sent within 5 days. As shown in Figure 4, the majority of URLs groups were sent within 5 days. Notice that this step does not discard URL groups even if their sending times are wide spread (>5 days). This is because that each group could potentially correspond to different spam campaigns, with each being individually bursty. Our iterative approach can clearly separate these campaigns and output different signatures.

The "specific" feature is quantified using an information entropy metric pertaining to the probability of a random URL string matching the signature. In the complete URL case, each signature, by definition, satisfies the "specific" property since it is a complete string and can not be more specific. For polymorphic URLs, we further discuss this metric in Section 4.3.

When AutoRE successfully derives a URL signature (satisfying all the three quality criteria), it outputs it as a spam signature. This signature characterizes the set of matching emails as botnet-based spam and the originating mail servers as botnet hosts. Note that if these spam emails contain additional URLs from multiple domains, those URLs will be removed from the remaining groups before the Group selector proceeds to select the next candidate group.

Using these three features, generating complete URL based signatures is straightforward: AutoRE considers every distinct URL in the group to determine whether it satisfies these properties, and then removes the matching URLs from the current group. The remaining URLs are further processed to generate regular expression based signatures.

4. AUTOMATIC URL REGULAR EXPRESSION GENERATION

In this section, we present a detailed view of the module in AutoRE that generates regular expression signatures. The input to the module is a set of polymorphic URLs from the same Web domain. The signature generation process involves constructing a keyword-based signature tree, generating candidate regular expressions, and finally evaluating the quality of the generated expressions (signatures) to ensure they are specific enough.

4.1 Signature Tree Construction

Our method begins by determining a candidate set of substrings from the pool of all frequent substrings; the candidate set serves as a basis for regular expression generation. We leverage the well-known suffix-array algorithm [1] to efficiently derive all possible substrings and their frequencies. To ensure that a keyword is not too general, we only consider substrings of length at least two.

The key question now is, what combinations of frequent substrings constitute a signature? At a high level, our idea is to start with the most frequent substring that is both bursty and distributed (based on the thresholds introduced earlier). We then incrementally expand the signature by including more substrings so as to obtain a more specific signature. To this end, AutoRE constructs a *keyword-based signature tree* where each node corresponds to a substring, with the root of the tree set to the domain name. The set of substrings in the path from the root to a leaf node defines a keyword based signature, each associated with one botnet-based spam campaign.

Initially, there is only the root node corresponding to the domain string with all the URLs in the group associated to it. Given a parent node, AutoRE looks for the most frequent substring; if combining this substring with the set of substrings along the path from the root satisfies the preset AS and sending time constraints, AutoRE creates a new child node. Consequently, all matching URLs will be associated with this new node. We repeat this process on the same parent node using the remaining URLs and popular substrings until there is no such substring to continue. We then iteratively proceed to the child nodes and repeat the process.

Figure 5 shows an example signature tree constructed using a set of 9 URLs [3], all associated with the domain `deaseda.info`. Notice that we have two signatures corresponding to nodes N_3 and N_4, each defining a botnet spam campaign.

There are two reasons for a tree to generate multiple signatures: (1) they correspond to different campaigns, hence different signatures, and (2) multiple signatures map to one campaign, but each of them occurs with enough significance to be recognized as different ones.

4.2 Regular Expression Generation

Given the keyword-based signatures, we now proceed to derive regular expressions based on them. There are two major steps involved: *detailing* and *generalization*. *Detailing* returns a domain-specific regular expression using a keyword-based signature as input. This step encodes richer information regarding the locations of the keywords, the string length, and the string character ranges into the target regular expression. In fact, this step is important to significantly increase the quality of URL signatures from the perspective of reducing false positive rates. *Generalization* returns a more general domain-agnostic regular expression by merging very similar domain-specific regular expressions. As we will show in Section 6.1.4, this step is helpful to increase the coverage of botnet spam detection.

The detailing process assigns the derived frequent keywords as fixed anchor points, and then applies a set of predefined rules to generate regular expressions for the substring segments between anchor points. The final regular expression is the concatenation of the anchored keywords and segment-based regular expressions. Each regular expression for a substring segment has the format $\mathcal{C}\{l_1, l_2\}$ (in Perl Compatible Regular Expression notation), where $\mathcal{C}$ is the character set, and l_1 and l_2 are the minimum and maximum substring lengths. Without loss of generality, we include all frequently used character sets (e.g., [0-9], [a-zA-Z]) and special characters (e.g., '`.`', '@') according to the URL standard [3]. The bounds on the substring length are derived using the input URLs. Notice that the resulting regular expressions are domain-specific. Figure 5 shows two example signatures.

The generalization process takes domain-specific regular expressions and further groups them. The rationale behind this is that we found scenarios where spammers sign up for many domains, sometimes with one IP address hosting more than 100 domains. If one domain gets blacklisted, spammers can quickly switch to another. Although domains are different, interestingly, the URL structures of these domains are still quite similar, maybe because they use a fixed set of tools to set up web servers and send out emails. Therefore, if two regular expressions differ only in the domain name and substring lengths, we merge them by discarding domains, and taking the lower bound (upper bound) as the new minimum (maximum) substring length. In the first example, shown in Figure 6, generalization preserves the keyword `/n/?167&` and the character set `[a-zA-Z]`, but discards domains and adjusts the substring segment lengths to `{9,27}`.

4.3 Signature Quality Evaluation

The generalization process may produce overly general signatures. AutoRE quantitatively measures the quality of a signature and discards signatures that are too general.

Our metric, defined as *entropy reduction*, leverages information theory to quantify the probability of a random string matching a signature. Given a regular expression e, let $B_e(u)$ and $B(u)$ denote the expected number of bits used to encode a random string u with and without the signature respectively. The entropy reduction $d(e)$ is defined as the difference between $B_e(u)$ and $B(u)$, i.e., $d(e) = B(u) - B_e(u)$. The entropy reduction $d(e)$ reflects on the probability of an arbitrary string with expected length allowed by e and matching e, but not encoded using e. We can write this probability as

$$P(e) = \frac{2^{B_e(u)}}{2^{B(u)}} = \frac{1}{2^{B(u)-B_e(u)}} = \frac{1}{2^{d(e)}}$$

Given a regular expression e, its entropy reduction $d(e)$ depends on the cardinality of its character set and the expected string length. Intuitively, a more specific signature e requires fewer bits to encode a matching string, and therefore $d(e)$ tends to be larger. In our framework, AutoRE discards all signatures whose entropy reductions are smaller than a preset threshold (set to 90 in our experiments; viewed another way, this means the probability of a random string matching a signature is $\frac{1}{2^{90}}$). For example, based on our metric, a signature `AB[1-8]{1,1}` is much more specific than `[A-Z0-9]{3,3}` even though they are of the same length.

5. DATASETS AND RESULTS

Our study is based on randomly sampled Hotmail email messages, excluding those that originated from blacklisted IPs, such as the ones published by Spamhaus [22]. In particular, the dataset was collected in November 2006, June 2007, and July 2007, with a total of 5,382,460 sampled emails (sampling rate 1:25000). All the email messages in our sample were pre-classified as either spam or non-spam by a human user. However, in our experiments, we ignored these classification labels while using the AutoRE framework to generate a list of botnet URL signatures and the corresponding botnet IP addresses. These labels were used later to evaluate the false positive rate of results obtained using AutoRE.

[3]We used these 9 URLs for illustration purposes only. In practice, the number of URLs that match a signature could be much larger.

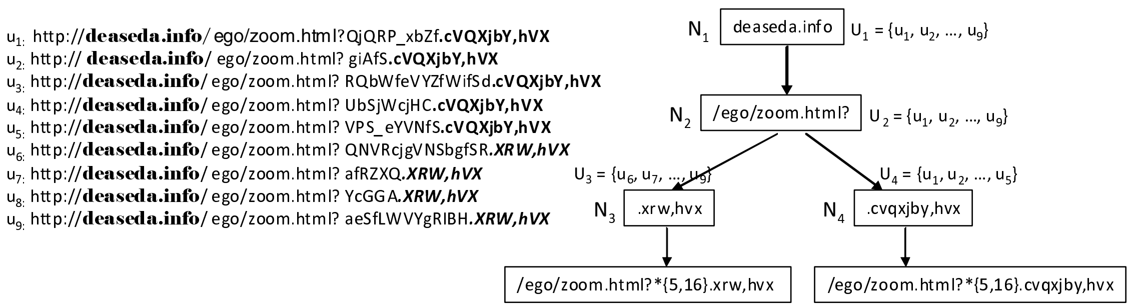


Figure 5: Example input URLs and the keyword-based signature tree constructed by AutoRE.

Figure 6: Generalization: Merging domain-specific regular expressions into domain-agnostic regular expressions.

Using the three months input data, AutoRE identified a total of 7,721 botnet-based spam campaigns. These campaigns together include 580,466 spam messages, sent from 340,050 distinct botnet host IP addresses spanning 5,916 ASes. Table 1 shows the statistics for the different months. We use *CU* to represent the set of complete URL based signatures and *RE* to denote the set of regular expression signatures. From the table, we observe that the majority (70.3-79.6%) of these campaigns belong to the CU category. About 20.4-29.7% of the campaigns have adopted polymorphic URLs. Comparing results across three months, we can clearly see a steady upward trend in the number of the identified campaigns – we see a 100% increase in the number of campaigns identified in July 2007 when compared to the number in Nov 2006. Consequently, the spam volume increased significantly by around 50% from Nov 2006 to June/July 2007. Interestingly, the total number of botnet IPs per month does not increase proportionally, suggesting that each botnet host is used more aggressively now.

The distribution of botnet size in terms of the number of unique IP addresses participating in a campaign is shown in Figure 7(a). We did not see any substantial difference in the shape of the distribution for the various months. Most botnets have tens to hundreds of IP addresses, with the largest having 1384 IPs. Since our identification used only sampled emails, the reported botnet sizes are expected to be much smaller than the actual sizes.

For the RE category, recall that AutoRE merges domain-specific regular expressions into domain-agnostic regular expressions. As shown in Figure 7(b), this step reduced the number of regular expressions by 4 to 19 times. In particular, for the month of July 2007, this grouping merged 717 regular expressions to 39. From these results, we hypothesize that spammers very likely used a limited number of automatic spam generation programs for generating polymorphic URLs.

We further use the generated signatures to examine how many of our sampled emails were sent from botnet hosts. As shown in Figure 7 (c), around 16-18% of spam emails match the derived URL signatures. Note that the spam campaigns we captured are more likely large ones, which have a higher probability of being sampled initially and subsequently being identified by AutoRE. Hence, we expect this 16-18% to be the lower bound on the botnet spam emails received. In the two next sections, we focus on evaluating our results (URL signatures and botnet IP addresses) and also analyze the botnet distribution and sending patterns.

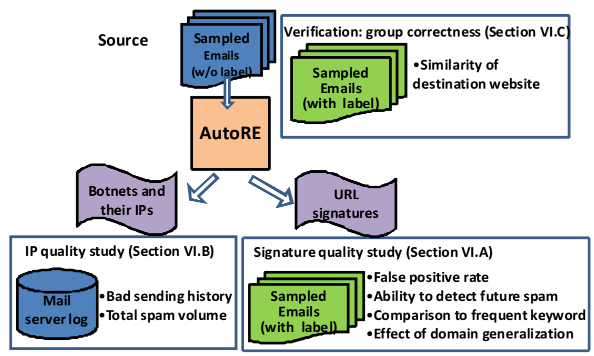


Figure 8: Overview of experiments and evaluation.

6. BOTNET VALIDATION

Ideally, the AutoRE identification results should be validated by comparing them against known URL signatures and botnet host identities. However, in the absence of such information, our validation is based on the following three methods (Figure 8 illustrates the overall evaluation setup):

We first study the quality of the extracted URL signatures. We used the human classified labels to compute the spam detection false positive rate. To better understand the effectiveness of using signatures for future spam detection, we performed cross-month evaluation by applying signatures generated in a previous month to emails received in a later month. Our experiments also demonstrated the importance of having regular expression signatures.

Second, we examined whether the identified botnet hosts were indeed spamming servers – to this end, we used the Hotmail server log that records the sending history of *all* email servers that communicate with Hotmail over time. This log includes the email volume and the spam ratio [4] of each server on a daily basis. In this paper, we use these statistics to evaluate the identified botnet hosts.

[4] The spam ratio was computed using the existing spam filtering system configured by Hotmail. The current filter leverages both email content and email server sending history for spam detection.

Month	Nov 2006		June 2007		July 2007		Total
	CU	RE	CU	RE	CU	RE	
Num. of spam campaigns	1,229	519	1835	591	2826	721	7,721
Num. of ASes	3,176	1,398	4,495	1,906	4,141	1,841	5,916
Num. of botnet IPs	88,243	23,316	113,794	19,798	85,036	29,463	340,050
Num. of spam emails	118,613	26,897	208,048	26,637	159,494	40,777	580,466
Total botnet IPs	100,293		131,234		113,294		340,050

Table 1: Some statistics pertaining to the botnets identified by AutoRE.

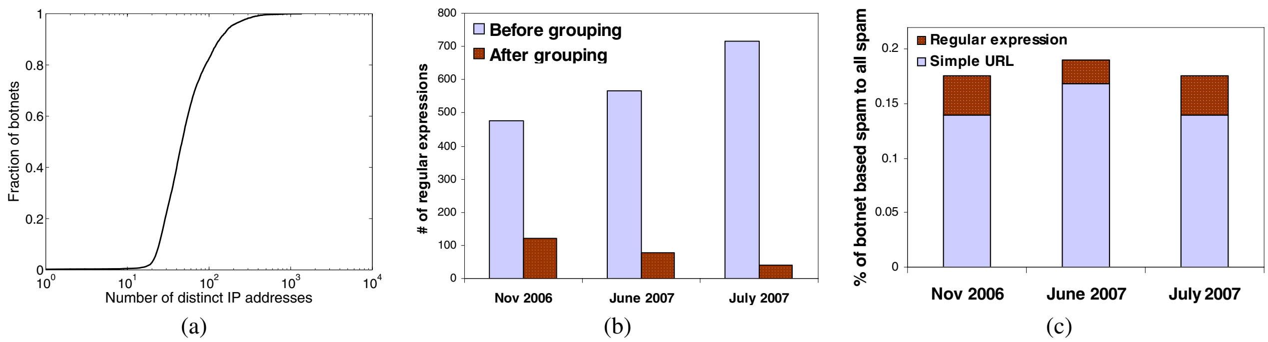


Figure 7: (a) Cumulative distribution of botnet size in terms of number of distinct IPs involved. (b) Number of regular expression patterns before and after generalization. (c) Percentage of spam captured by AutoRE signatures.

Finally, we are interested in finding whether each set of emails identified from the same spam campaign were correctly grouped together. To answer this question, for every set, we examine the similarity between the corresponding destination Web pages. In previous work [2], destination web pages were shown to be strongly correlated to the corresponding spam campaign.

6.1 Evaluation of Botnet URL Signatures

Recall that every email in our dataset has been pre-classified as either spam or non-spam by a human user; we now use these labels to evaluate the effectiveness of the generated botnet URL signatures.

6.1.1 False Positive Rate

We begin by presenting the spam detection false positive rate: for every signature, we compute its spam detection false positive rate as the fraction of non-spam emails matching the signature to the total number of non-spam emails (see Figure 9(a)). For CU signatures, the false positive rates lie between 0.0001 to 0.0006. For RE signatures, the rates are between 0.0011 and 0.0014. The aggregated false positive rates vary between 0.0015 and 0.0020.

6.1.2 Ability to Detect Future Spam

In Section 5, we showed that URL signatures generated by AutoRE can be used to detect between 16% to 18% of spam on a monthly basis (Figure 7(c)). We now proceed to evaluate the effectiveness of using AutoRE signatures for future spam detection – in other words, we are interested in determining if AutoRE signatures are effective over time. For this experiment, we applied the signatures derived in Nov 2006 and June 2007 to the (sampled) emails collected in July 2007.

From Table 2, we find signatures generated in Nov 2006 are not useful in detecting botnet spam sent in July 2007. In contrast, signatures from June 2007 are highly effective, matching 50,529 spam emails sent in July 2007 with a low false positive rate. The big difference between the detection rates of Nov 2006 and June 2007 signatures indicate that spam URL patterns evolve over time. Furthermore, we observe that RE signatures are much more robust over time than CU signatures. Specifically, the RE signatures generated in June 2007 have comparable detection capabilities to the RE signatures generated in July 2007. In the next subsection, we further demonstrate the effectiveness of regular expression signatures by comparing them against frequent keywords.

Month	Nov 2006			June 2007		
	CU	RE	Total	CU	RE	Total
# of spam emails	2	3	5	6,751	43,778	50529
# of non-spam emails	10	0	10	154	561	715

Table 2: Number of spam and non-spam emails from July that match signatures derived from previous months.

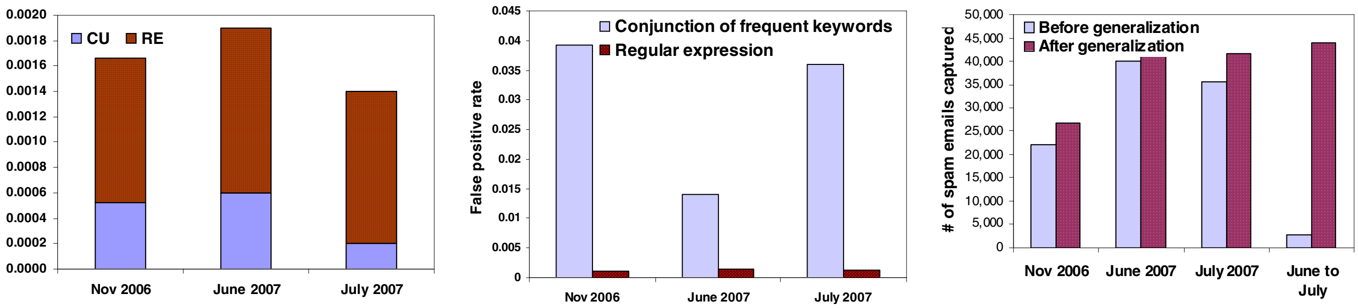
6.1.3 Regular Expressions vs Keyword Conjunctions

To understand the benefits of using regular expressions vs. keyword conjunctions (e.g., token1.*token2.*token3), we compare them in terms of their spam detection rate and false positive rate. The frequent keyword based signatures were generated using paths from the root of the keyword-based signature tree (Section 4.1) to its leaves. If a URL string contains all frequent keywords in a signature (regardless of order), then we consider it a match.

Both types of signatures produce almost identical spam detection rate. However, their false positive rates differ dramatically. Figure 9(b) shows that regular expressions reduce the false positive rates by a factor of 10 to 30. As we discussed in Section 2, URL strings are often human readable strings with English words and substring segments. Thus merely using frequent keywords results in a high positive rate. Using regular expressions can greatly reduce the chance of legitimate URLs matching a signature.

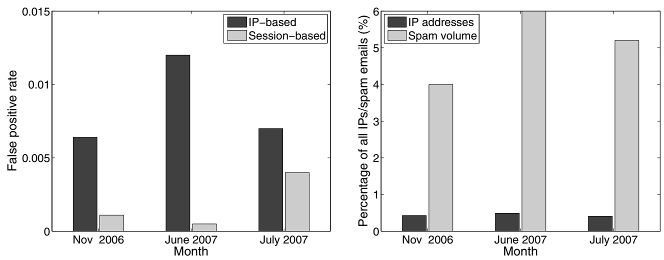
6.1.4 Domain-Specific vs Domain-Agnostic Signatures

An important step of regular expression generation is to merge domain-specific signatures into domain-agnostic ones through *gen-*



(a) (b) (c)

Figure 9: (a) False positive rate of AutoRE signatures. (b) False positive rate: RE signatures vs keyword-based signatures. (c) Number of spam emails captured by RE signatures: before and after generalization.



(a) (b)

Figure 10: Spam detection performance using botnet IPs identified by AutoRE. (a) False positive rate. (b) Total botnet-based spam volume.

eralization. Figure 9(c) shows that after generalization, AutoRE can detect 9.9-20.6% more spam without affecting the false positive rates. More importantly, generalization is critical to detecting future botnet spam sent from new domains. When we apply June 2006's domain-specific RE signatures to data in July 2007, we identified only 2749 spam emails. However, with domain-agnostic RE signatures, the number of spam emails captured increased sharply to 43,778. Thus, generalization effectively preserves the stable structures of polymorphic URLs, yet removing the volatile domain substrings.

In summary, our signature evaluation demonstrates the applicability of using AutoRE signatures for botnet spam detection. Using only a small number of automatically generated signatures, we detect 16-18% of total spam with a low false positive rate (the remainder could be attributed to non-botnet based spam, or sent by small botnets that AutoRE failed to detect due to our aggressive sampling rate). We emphasize that this 16-18% belongs to the "stealthy" category – emails that slip through the sophisticated spam filtering system employed by Hotmail. Compared with exact URLs or frequent keyword based signatures, regular expressions are much more robust for future spam detection and also achieve a low false positive rate. Finally, *domain-agnostic* signatures are more effective in detecting future botnet spam than domain-specific ones.

6.2 Evaluation of Botnet IP Addresses

In this section, we are interested in evaluating the identified botnet IP addresses to determine if these hosts are indeed spammers; if so, we are also interested in quantifying the total amount of spam that is received by Hotmail from them. Our evaluation leverages the email server log on *all* emails and the human classified labels on the *sampled* emails. As described earlier, every record in the email server log contains aggregated statistics about the email volume and the spam ratio of each IP address on a daily basis. For clarity, we refer to each record in the log as a session.

For those email servers setup using botnet IP addresses, we hypothesize that the spam ratio should be 100% for all its sessions. However, since existing spam filters do have false positives, it is possible that a spamming server does not always have a 100% spam ratio in our log. In such cases, we examine whether users unanimously classify those emails (in our sampled email data) as spam. If not, we consider this IP address as a possible legitimate IP in the false positive category. Figure 10 (a) shows that the false positive rate over the total identified IPs (sessions) is very small.

Although the identified botnet IPs constitute only a very tiny percentage (less than 0.5%) of all the IP addresses that were used as email servers in our log, their total spam volume is non-trivial — up to 6% of all spam received (Figure 10 (b)). By using a larger set of incoming emails, we expect the number of the identified botnet hosts and the fraction of detected spam to increase.

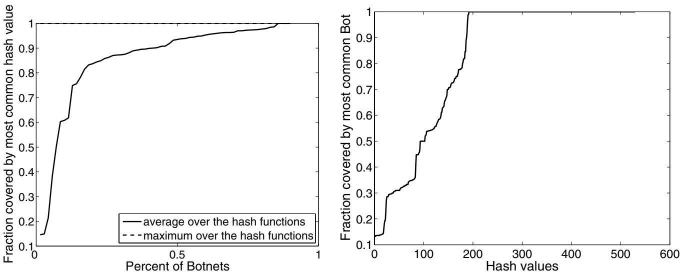
6.3 Is Each Campaign a Group?

In the previous sections, we evaluated the signatures and the corresponding botnet IP addresses individually. In this section, we proceed to verify whether each spam campaign is correctly grouped together by computing the similarity of destination Web pages. Our verification focuses on polymorphic URLs generated using the Nov 2006 dataset. We crawled all the corresponding Web pages [5] and applied text shingling [9] to generate 20 hash values (shingles) for each Web page. Note that the shingling process strips common HTML headers and tags.

Figure 11(a) shows the average ($favg$) and maximum ($fmax$) number of Web pages covered by the most common shingle in each campaign. For most spam campaigns, 90% of the destination Web pages had a $favg$ value of larger than 0.75, meaning these pages are at least 75% similar. The difference in content may be attributed to random advertisements and customized user contents. The value of $fmax$ was always 1, meaning there exists at least one identical hash value.

Next we analyze whether the destination Web pages advertised by different campaigns are dissimilar. If so, for each campaign, we expect the common shingle (i.e., $favg$) to occur infrequently at Web pages associated with other campaigns. To validate this, we measure the ratio of this hash value occurrence within the campaign to the occurrence across all the campaigns (see Figure 11(b)). For most hash values, the ratio was exactly 1, indicating they occurred

[5]We intentionally crawled only one month's polymorphic URLs. This is because crawling is an intrusive process that might let spammers believe certain groups of users are more vulnerable to spam emails and thus send more spam to them in the future.



(a) (b)

Figure 11: (a) The fraction of Web pages covered by the most common hash value in each campaign. The two curves are presented in an ascending order independently ($fmax$ is always 1). (b) The ratio of the average hash value occurrence within a campaign to the occurrence across all campaigns.

in exactly one campaign. This validation shows that the Web pages pointed to by each set of polymorphic URLs are similar to each other, while pages from different campaigns are different.

7. UNDERSTANDING SPAMMING BOTNET CHARACTERISTICS

In this section, we study the characteristics of botnets that have been associated with the identified spam campaigns. We first analyze their geographic distribution over the Internet and their overall sending patterns. We then examine each spam campaign by studying their individual behavior. Following this, we analyze the similarity and overlap in behavior across different campaigns. Since it has been hypothesized that botnet hosts are often used to actively scan and infect other hosts, we correlate the botnet spamming activities with the network scanning activities using the distributed telescope data [7].

7.1 All Botnet Hosts: A General Perspective

We begin our analysis by examining the distribution of botnet hosts across the Internet and their spamming patterns by treating them as one population.

7.1.1 Distribution of Botnet IP Addresses

Figure 12(a) shows the top five ASes ranked according to the number of unique botnet IP addresses. Notice that all five ASes are Internet service providers which offer residential network access. Although countries like China and Korea are often regarded as having a large number of vulnerable home computers, interestingly, we see a significant fraction of botnet IPs in the U.S. The latter observation suggests that botnet menace is indeed a global phenomenon.

Figure 12(b) shows a scatter plot of the number of ASes vs. the number of IPs for each spam campaign. We observe that botnet IP addresses are typically spread across a large number of ASes, with each AS on average having only a few participating hosts. The largest spam campaign we identified had hosts that spanned 362 ASes, indicating the importance of employing a network-wide view for botnet detection and defense.

Previous study [24] has shown that email servers set up on dynamic IP ranges are more likely to be zombie spam servers. Motivated by this, we compared the list of botnet IPs identified by AutoRE to the list of dynamic IP addresses identified in [24] and the list of Dynablock IP addresses [8].

On average, 69% of botnet IP addresses were dynamic, confirming the earlier observation that dynamic IP based hosts are popular targets for infection by botnets. Figure 12 (c) shows the CDF of the percentage of dynamic IP addresses per campaign. Across all three months, more than 80% of campaigns have at least half of their hosts in the dynamic IP ranges [6]. What is surprising, however, is that the (observed) spam emails from botnets are switching away from dynamic IP ranges to static IP addresses. In particular, the percentage of spam campaigns that had at least 80% of dynamic IP addresses dropped significantly from 52% in November 2006 to 14% in June 2007. The absolute number of static IP based botnet hosts increased from 21,010 in November 2006 to 44,790 in July 2007. This could be associated with the increased adoption of dynamic IP-based blacklists (e.g., Spamhaus [22]). Spammers could query the blacklist before using an IP address to send spam emails; thus they tend to use static IP addresses not present on the blacklist. On the other hand, this indicates a potential opportunity to capture and track these bots as their IP addresses remain static.

7.1.2 Spam Sending Patterns

In this section we explore the potential for detecting botnet hosts using content independent features. To begin with, we are interested in the following question: do botnet hosts exhibit distinct email sending patterns when analyzed individually? Taking the standpoint of a server receiving incoming emails from other servers, we select the following three features (collected at the SMTP protocol level) to describe the sending patterns of each incoming server in our study:

- *Number of recipients per email*: For modelling purposes, we use the reciprocal of this feature, which is a number between 0 and 1. A value of one indicates that any particular email is sent to only one recipient. A value close to zero means that the email has a large number of recipients.
- *Connections per second*: The frequency of incoming connections received from the host (log scale).
- *Nonexisting recipient frequency*: We track the rate of observing an invalid recipient normalized by the number of valid emails received from the host.

We use the *Number of recipients per email* and *Connections per second* features because they reflect the aggressiveness of a spammer. The *nonexisting recipient frequency* feature roughly provides a measure on the amount of traffic destined to invalid email addresses, indicating whether spammers are scanning the email address space, trying to obtain valid email addresses. We map each of the above features to a coordinate system and represent each record as a point in a three-dimensional space. We found that both the sending patterns of the identified botnet hosts and other hosts are well spread in the space. In other words, when viewed individually, botnet hosts do not exhibit distinct sending patterns for them to be identified.

7.2 Per Campaign: An Individual Perspective

Here, we study individual botnet-based spam campaigns identified by AutoRE and examine whether botnet hosts within one campaign exhibit varied behavior.

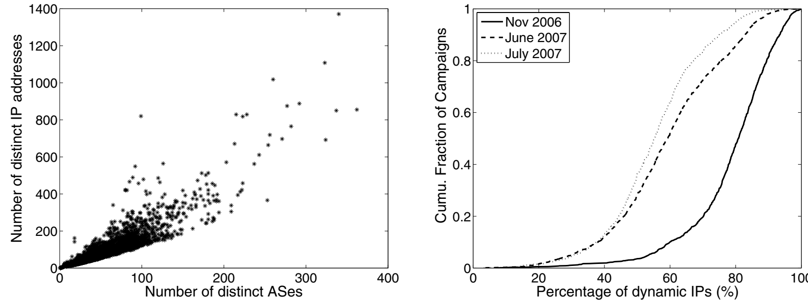
7.2.1 Similarity of Email Properties

As a first step, we analyze the content similarity of botnet emails. For each email that matches a given signature, we shingle its contents. Figure 13 (a) shows the percentage of emails that share the

[6] For each spam campaign, given its activity burstiness, the likelihood of a host changing from one dynamic IP to another is small. In our analysis, a majority of the dynamic IP ranges contained only one botnet IP address.

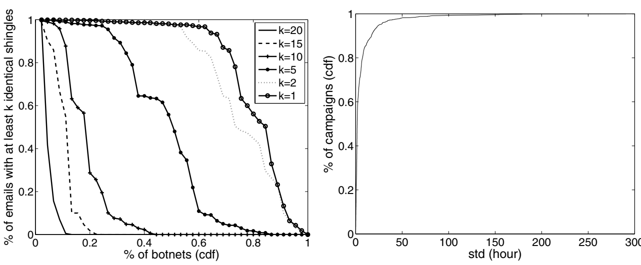
AS description	AS Number	Number of bot IPs
Korea Telecom	4766	15757
Verizon Internet service	19262	11426
France Telecom	3215	11303
China 169-backbone	4837	9960
Chinanet-backbone	4134	8113

(a)



(b) (c)

Figure 12: (a) Top five ASes that account for most botnet IPs (identified by AutoRE). (b) Number of ASes vs. number of IP addresses in each spam campaign. (c) Cumulative fraction of spam campaigns using dynamic IPs.



(a) (b)

Figure 13: (a) Similarity of email content shingles (b) CDF of sending time standard variations (in hours) for each campaign.

most common k shingles (where k varies as showed in the figure). For the majority of campaigns ($> 60\%$), most emails share at least one shingle. However, the likelihood of these emails sharing all shingles is very low. In fact, around 50% of the campaigns have no two emails sharing 10 common shingles, suggesting that the contents are quite different even though their target web pages are similar.

7.2.2 Similarity of Sending Time

We proceed to examine the synchronous degree of spam sending time for each campaign. For each campaign, we compute the standard deviation (std) of spam email sending time (Figure 13(b)). 50% of campaigns have std less than 1.81 hours, meaning they sent almost simultaneously and were likely triggered by a single command. The rest of the campaigns have a larger variation, suggesting those bots might start sending whenever they come online. Overall, 90% of campaigns have stds less than 24 hours and were likely located at different time zones.

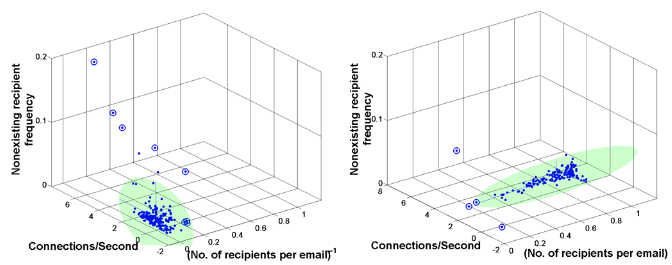
7.2.3 Similarity of Email Sending Behavior

We now broaden our analysis to the set of email sending features discussed in Section 7.1.2. Our goal is to systematically investigate whether botnet hosts could be grouped into a well-formed cluster (in the previously defined three-dimensional coordinate space). For each campaign, we use a Gaussian model with full covariance matrix to model the data and learn the Gaussian parameters.

Table 3 lists the percentage of outliers that do not fit into the learned Gaussian models. We see that for each spam campaign, the host sending patterns are generally well clustered (with $<10\%$ outliers). Figure 14 shows two such clusters. Figure 14 (a) involves 191 botnet hosts with 9 outliers. The majority of the hosts are tightly clustered by having a similar number of recipients per email. These hosts sent emails with a long *To* or *Cc* list. The second example in Figure 14 (b) shows a campaign with 162 hosts spanning 80 ASes. A unique aspect with regard to this particular example is that the participating hosts (except for the 4 outliers) shared a constant connection rate (3 connections per second) in their communication with the server, suggesting that the botnet software may have applied rate-control in initiating connections.

% of outliers	$< 5\%$	$5-10\%$	$10-15\%$	$> 15\%$
Nov 2006, CU	59%	27%	8%	6%
Nov 2006, RE	69%	21%	6%	4%
June 2007, CU	74%	23%	2.5%	0.5%
June 2007, RE	44%	42%	9%	5%

Table 3: Percentage (%) of campaigns that have different ratios of outliers after clustering.



(a) (b)

Figure 14: Two examples of well-clustered botnets. Outliers are shown using circles.

For the few cases with a high number of outliers, we found that many of them are bi-modal. We are investigating these cases further to understand whether this could be attributed to the heterogenous nature of bot hosts in terms of computation power, network access speed, etc.

7.3 Comparison of Different Campaigns

In this section, we study the overlaps among different spam campaigns and compare the botnet host email sending patterns. Section 5 showed that a large number of campaigns share the same domain-agnostic regular expression signatures (refer to Figure 7 (b)). So the first question we explore is whether the corresponding botnets essentially correspond to the same set of hosts. For each domain-agnostic signature, we identify the set of spam campaigns (say a total of k) that all share this signature. We then plot in Figure 15 the ratio of the number of unique IPs across the k botnets to the sum of their IPs as a function of k. Quite surprisingly,

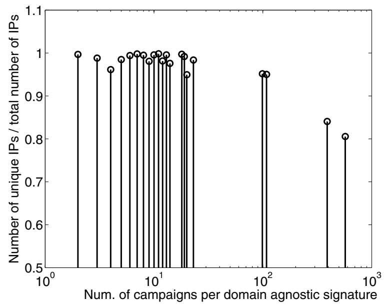
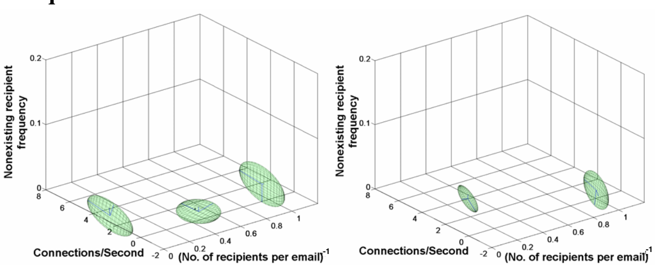


Figure 15: For each domain-agnostic signature, we show the number of botnets that share the signature vs the ratio of unique IPs to all IPs.



(a) Non-polymorphic case (b) Polymorphic case

Figure 16: Super clusters obtained by aggregating the June 2007 botnet sending patterns.

the ratio is close to 1 when k is small, meaning botnets sharing a domain-agnostic signature barely overlap with each other in most of the cases. With k increasing, the ratio increased gradually to 0.8, meaning 20% of the botnet IPs participated in multiple campaigns characterized by the same signature.

Finally, we examine the similarity of sending patterns across botnet campaigns using the learned Gaussian models in Section 7.2.3. Specifically, we group individual botnet clusters into super clusters based on the similarity of the estimated mean; we discard clusters whose covariance matrices are not compact (and hence the data is well spread out). Interestingly, for botnets that sent non-polymorphic URLs, the resulting super clusters correspond to three specific operating modes (Figure 16 (a)): in two cases, the number of recipients per email feature is held at a constant value, while the connection rate feature varies significantly. The converse of the above is evident in the third (middle) cluster. For the botnets that sent polymorphic URLs, they map to only two models. The small number of super clusters suggests spammers may all utilize a few common programs to launch botnet spamming attacks.

7.4 Correlation with Scanning Traffic

We now analyze the network scanning behavior of the identified botnet hosts using the distributed telescope data. In particular, we use the Dshield trace collected in 2006 over a large network of more than 400,000 hosts [7]. This log contains failed connection attempts rejected by firewalls and scanning traffic to non-existing hosts. In our study, we focus on the source IP address and the port number fields and we consider the botnet IPs generated using the dataset from November 2006.

Due to dynamic IP address assignment, using IP address as a host identifier for correlation is not robust. Therefore, for dynamic botnet IP addresses, instead of correlating the exact addresses, we check all the scanning activities from the corresponding dynamic IP ranges obtained from [24]. Using the dynamic IP ranges together with the remaining “likely static" IPs, we plot in Figure 17 the number of scans originating from these IP addresses into a set of popular scanning destination ports in Aug 2006 and Nov 2006, respectively. Besides ports 1026, 1027 and 25, all other ports are used for exploiting host vulnerabilities. For these ports, the amount of scanning traffic in August is higher than in November, when these botnet IPs were actually used to send spam. One reason could be that botnet attacks have different phases. In August, they were used to actively seek victim computers with the purpose of expanding the botnet size. In November, these botnets reached their target size and were used to launch spam attacks. Hence, monitoring scanning traffic in advance could be potentially helpful in defending against botnet-based spam attacks and is an interesting future topic to investigate.

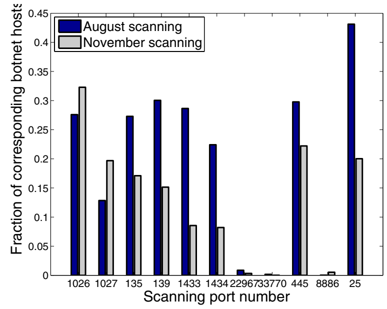


Figure 17: Scanning traffic from botnet IPs captured by Dshield.

8. DISCUSSION

Although, in this work, AutoRE serves as a post-mortem tool for botnet spam detection, it has the potential to work in real time mode. Due to the aggressive sampling rate (1:25000), the number of data points in our dataset was not sufficient to perform real-time experiments. But given a live mail feed, AutoRE can be designed to produce signatures as soon as there is enough information to conclude that a distributed botnet spam campaign has commenced. Indeed, we have demonstrated that the signatures of June 2007 caught a non-trivial portion of July 2007's spam, suggesting that AutoRE can potentially stop a large portion of botnet spam in real-time service. The success of this approach depends on how quickly signatures can be generated and deployed, and how long a spam campaign lasts. The extension of AutoRE into a real time setting is left for future work.

Spammers may attempt to craft emails to evade the AutoRE URL selection process. For example, they may add legitimate URLs to confuse the URL selection process. Since spammers have no control of the sending frequency of legitimate URLs, it will be hard for them to select which URLs to include. A popular URL would be discarded as background noise, and a rarely used URL will stand out as a spike for identifying the botnet [7]. Spammers may also wish to pollute the "bursty" feature by sending a spam URL from a few hosts before launching a large-scale attack. Such pollution can be easily detected by a more robust signal processing methodology that captures signal spikes in the existence of low frequency background noise.

In the extreme case, spammers may wish to evade detection by having no patterns in their URLs. For example, each URL points to just a domain string (e.g., `a.com`, `b.com`, etc). We expect such a scenario to be rare as the cost of registering domains makes this economically less attractive to spammers.

[7]In this case, the legitimate URL itself servers as a signature

AutoRE leverages the "bursty" and "distributed" features of botnet attacks for detection. Legitimate emails sent by a big company advertising a product or event could also be bursty. But they will be unlikely sent from hosts spanning more than a few ASes. One false positive case could be email flash crowd, where people forward each other a few popular URL links. We expect such events to be very rare. In our experience of using three months of data and the source AS threshold of 20, we did not encounter a single such event. Studying legitimate email traffic can potentially guide the source AS number threshold selection in the AutoRE framework.

More sophisticated spammers may leverage URL redirection techniques to hide the real spamming Web sites. In this case, hosts from a botnet may send out seemingly unrelated URLs, but these URLs all redirect to the same final destination. To detect such botnets, we can potentially construct an entire redirection path consisting of a sequence of IP addresses and intermediate URLs, and then apply AutoRE in a similar way to the redirection paths. We do not advocate this approach because constructing the redirection paths requires extensive querying of the destination URLs. Such process might encourage spammers to send even more spam, as they might view the visiting traffic to be from users (and regard these users as vulnerable to spam or phishing).

9. CONCLUSION

In this paper, we presented AutoRE, a framework that automatically generates URL signatures for spamming botnet detection. AutoRE requires neither pre-classified inputs nor other training data or white lists. Furthermore, AutoRE generates regular expression signatures, which were previously written by human experts only. Using sampled emails from Hotmail, AutoRE identified 7,721 botnet-based spam campaigns, comprising 340,050 distinct IP addresses spanning 5,916 ASes. The false positive rate of applying AutoRE signatures for botnet spam detection is less than 0.002, and the false positive rate of botnet host detection is less than 0.005. We expect the generated spam signatures and the botnet membership information to be useful for capturing future spam and reducing other malicious Internet activities.

Our extensive analysis of the identified botnets revealed several important findings. First, our exploration showed botnet hosts are wide-spread across the Internet, with no distinctive sending patterns from normal servers when viewed individually. This suggests that detecting and blacklisting individual botnet host will continue to remain a challenging task. Second, in our work, we demonstrated the existence of botnet spam signatures and the feasibility of detecting botnet hosts using them. Our analysis also shows that botnet host sending patterns, such as the number of recipients per email, connection rates, and the frequency of sending to invalid users, are clusterable and their sending times are synchronized. Thus an interesting future direction is to further explore mechanisms that capture aggregated activities of botnets. Finally, comparison of spam traffic patterns from 2007 to 2006 clearly showed that botnets are evolving and getting increasingly sophisticated. For example, the adoption of polymorphic URLs increased significantly, and the number of static IP address based bots doubled from Nov 2006 to July 2007. These trends for evading existing detection systems suggests that we need to take a holistic view of various mechanisms and explore the invariable attack features in order to get an upper hand in the spam arms race.

10. REFERENCES

[1] M. I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *J. of Discrete Algorithms*, 2(1), 2004.

[2] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: Characterizing Internet scam hosting infrastructure. In *14th conference on USENIX Security Symposium*, 2007.

[3] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifiers (URI): Generic syntax. *RFC 2396*, 1998.

[4] K. Chiang and L. Lloyd. A case study of the Rustock rootkit and spam bot. In *The First Workshop in Understanding Botnets*, 2007.

[5] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *Proc. of the 13th Annual Network and Distributed System Security Symposium (NDSS)*, 2006.

[6] N. Daswani, M. Stoppelman, and the Google click quality and security teams. The anatomy of Clickbot.A. In *The First Workshop in Understanding Botnets*, 2007.

[7] Dshield: Cooperative network security community.

[8] Dynablock dynamic IP list. `http://www.njabl.org/`, recently aquired by spamhaus, http://www.spamhaus.org/pbl/index.lasso, 2007.

[9] D. Fetterly, M. Manasse, M. Najork, and J. L. Wiener. A large-scale study of the evolution of web pages. *Softw. Pract. Exper.*, 34(2), 2004.

[10] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm. In *LEET 08: First USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[11] C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, and S. Savage. The Heisenbot uncertainty problem: Challenges in separating bots from chaff. In *LEET '08: First USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[12] H.-A. Kim and B. Karp. Autograph: Toward automated, distributed worm signature detection. In *the 13th conference on USENIX Security Symposium*, 2004.

[13] C. Kreibich and J. Crowcroft. Honeycomb: Creating intrusion detection signatures using honeypots. In *2nd Workshop on Hot Topics in Networks (HotNets-II)*, 2003.

[14] F. Li and M.-H. Hsieh. An empirical study of clustering behavior of spammers and group-based anti-spam strategies. In *CEAS 2006: Proceedings of the 3rd conference on email and anti-spam*, 2006.

[15] Z. Li, M. Sanghi, Y. Chen, M.-Y. Kao, and B. Chavez. Hamsa: Fast signature generation for zero-day polymorphic worm with provable attack resilience. In *IEEE Symposium on Security and Privacy*, 2006.

[16] J. Newsome, B. Karp, and D. Song. Polygraph: Automatically generating signatures for polymorphic worms. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, 2005.

[17] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006.

[18] A. Ramachandran, D. Dagon, and N. Feamster. Can DNS based blacklists keep up with bots? In *Conference on Email and Anti-Spam*, 2006.

[19] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proceedings of Sigcomm*, 2006.

[20] A. Ramachandran, N. Feamster, and S. Vempala. Filtering spam with behavioral blacklisting. In *Proceedings of the 14th ACM conference on computer and communications security*, 2007.

[21] S. Singh, C. Estan, G. Varghese, and S. Savage. Automated worm fingerprinting. In *OSDI*, 2004.

[22] Spamhaus policy block list (PBL). `http://www.spamhaus.org/pbl/`, Jan 2007.

[23] S. Webb, J. Caverlee, and C. Pu. Introducing the web spam corpus: Using email spam to identify web spam automatically. In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS)*, 2006.

[24] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are IP addresses? In *ACM Sigcomm*, 2007.

[25] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, I. Osipkov, G. Hulten, and J. Tygar. Characterizing botnets from email spam records. In *LEET 08: First USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.