A Monitor Tool for Anti-spam Mechanisms and Spammers Behavior

Danilo Michalczuk Taveira and Otto Carlos Muniz Bandeira Duarte
UFRJ - PEE/COPPE/GTA - DEL/POLI
P.O. Box 68504 - 21945-970, Rio de Janeiro, RJ, Brazil
http://www.gta.ufrj.br

***Abstract*—Sending spam is a profitable activity for spammers and more than 95% of the Internet messages will be spams in a near future. This paper presents a tool that helps developers of anti-spam systems to monitor the current spammer behavior, monitor the performance of current anti-spam systems, and analyze new anti-spam mechanisms developed. Performance analyses for the mostly used anti-spam mechanisms are provided and the tool can be easily extended to analyze new anti-spam systems. Some characteristics of the processes used by the spammers to harvest electronic addresses, create the messages, and send them are also evaluated. The results show the low efficiency of the analyzed anti-spam mechanisms. Moreover, results also show important characteristics of the harvest and email sending processes, such as the high delay between the two processes and the long period of time that addresses are kept on spammers' lists.**

I. Introduction

The number of unsolicited messages, called spam, is rising every day, representing more than 70% of all the mail messages [1]. The financial losses caused by this activity reach billions of dollars per year due to waste of resources like bandwidth, storage, time, and also loss of productivity. This activity is, however, very attractive for the spammers as they have a high profit due to the low cost of sending spam [2]. Statistics shows that the percentage of spam messages will soon be greater than 95% of all the messages [3].

The process to send spam is composed of three main steps. The first step is the electronic address harvesting, where many addresses are harvested typically from websites to build a list of recipients. The second step is the construction of the message that will be sent to the list of recipients. The last step is the sending of spam messages, which is most of the times accomplished by hijacked servers, open-relay mail servers, or also by zombie machines that makes it more difficult to identify the spammer.

To monitor the performance of anti-spam mechanisms a fast real time analysis is required and each mechanism has to be monitored independently or else the result of one mechanism will influence the result of the others. To meet these characteristics, a system was developed and its architecture is also presented on this paper. The developed system can help the developers of anti-spam systems to monitor the current spammer behavior, monitor the performance of current anti-spam systems and analyze new anti-spam mechanism developed. This paper also presents the techniques used by spammers to send spam and also some common anti-spam mechanisms. The system developed meets several restrictions to not impact negatively the user experience and also does not require many resources. The system analyzes every message as soon as it is received and before its delivery to the user. Therefore, to avoid message queuing and delay, a fast analysis is necessary.

Furthermore, memory requirements are taken into account because the processed data is on the order of magnitude of tens of gigabytes. The total time required to analyze the messages as well as the rates of false positives and false negatives are presented for each mechanism. A false positive happens when a legitimate message is classified as spam and a false negative happens when a spam is not correctly classified. The tool developed also monitors the characteristics of the process that spammers use to send spam, like the lifetime of email addresses on spammer lists and the period of time between the spammer harvests the addresses and the first message is sent to the harvested addresses. These characteristics are obtained from a honeypot that is also part of the tool.

The paper is organized as follows. In section II related works are discussed. Techniques used by spammers to send the messages are presented in section III. In section IV the most commonly used anti-spam mechanism are discussed. Section V presents the tool developed to analyze the anti-spam mechanism and the process spammers use to send the messages. The results of the analysis are presented on section VI and in section VII final remarks are discussed.

II. Related Work

The TREC system [4] was designed to analyze anti-*spam* mechanisms. This system, however, just analyzes the mechanisms based on the past messages the user received and cannot analyze the messages in real time. This characteristic is important because some anti-*spam* rapidly change the lists and/or rules they use to take the decisions. To overcome this restriction, our system analyzes every message at the moment they are received.

Gomes *et al* [5], [6] analyzes network traffic characteristics of the spams. The analyses of the time and size distribution of the messages, number of recipients per message, and other characteristics are shown. Some of these characteristics, like the distribution of the message size, however, have changed due to the fast evolution of the techniques used by spammers to bypass spam filters.

Schryen [7] intentionally published email addresses on web pages and newsgroups about different subjects to analyze if the spams sent to the addresses take into consideration the context from where the email was harvested. Similarly, Andreolini *et al* [8] implements a honeypot with the intent to flood spammers' lists with invalid addresses. Besides that, there was also a fake SMTP server, which appeared to be an open-relay server but, in fact, did not really relay the messages. On both works, however, the process used by spammers to send the messages is not analyzed.

The project Honey Pot [9] is a global project which attempts to track spammers using honeypots installed on globally distributed web sites. Participants of this project deploy honeypots on their servers but do not receive the messages sent to the addresses published on the honeypots, so they will not receive more spam. However, to be able to monitor the anti-spam mechanisms, it is better to receive a large set of messages. To achieve this, our system has a honeypot that publishes email addresses whose messages are sent to our mail server.

Ramachandran and Feamster [10] show characteristics of the spam messages and also some techniques used by spammers to send the spams. The most common technique nowadays to send spam is by using botnets of zombie machines that are controlled by the spammer.

III. Sending Spam

Spammers are always attempting to bypass anti-spam systems to reach the maximum number of recipients. Email harvesting process aims at obtaining a large number of addresses of possible recipients to send spam. The process is conducted by a robot that visit web pages, search for email addresses on the page and then moves on to the next page. Even if the user does not publish his/her email address on his/her site many sites publish email addresses of their members or of users somehow related to the site, most of the time even without the user acceptance. Email harvesting is also performed by hijacking servers and stealing the users' information. Virus and spywares are also used to steal users' contacts.

The creation of the messages has attained more importance with the introduction of anti-spam mechanisms that analyzes messages contents. In the beginning, the spam messages were not created taking into account the anti-spam filters and were easily detected just by looking for some words like Viagra, free, etc.

The last stage is the sending of messages to the recipients. One method used by the spammers to send spam is to abuse third-party mail or proxy servers that are misconfigured or have been hijacked. Currently, however, most of the messages are sent by botnets, which are networks of zombie machines infected by some virus or Trojan horse [10]. The zombie machines usually connect to a central server controlled by the spammers that send the orders to the zombie machines to send spam. The advantage for the spammer to use botnets is that his/her resources to send spam are multiplied by the number of the zombie machines and the traceability is difficult.

IV. Anti-Spam Mechanisms

Anti-spam systems are characterized by the false positive and the false negative rates. The false positives have a greater impact on users, causing financial loss and delays on the communication. On the other hand, false negatives should not be high or else users lose a lot of time reading the spams that reach his/her mailbox. Another important aspect of an anti-spam system is the required level of user interaction. The greater interaction that is needed the users will likely give up using it. This section discusses commonly used anti-spam systems that our tool monitors.

A. Blacklists

The efficiency of a blacklist can be measured by the rate in which the list is updated [11]. As soon as a machine is detected as a spam source, its IP address should be included on the blacklist. The period of time in which addresses that are no longer sending spam are removed is also important, to reduce false positives. Blacklists can be either managed by users or by organizations responsible for the list management. User-managed blacklists are rare, as they require constant user interaction to remove and add addresses. The blacklists commonly used are managed by organizations. The lists are queried using the DNS protocol and are called DNSBL (Domain Name System Blacklist) [12]. The DNS protocol is used to query these lists because it is a well established protocol, its implementation is already mature and it also provides cache of queries, reducing the bandwidth usage.

B. Rule-based

On rule-based mechanisms, the content of the messages is analyzed and a list of rules is checked. Each rule is composed of a logical test that verifies some characteristics typically present on either legitimate or spam messages. Each rule also has a weight that can be either positive or negative. Rules that tests for spam characteristics have positive weights and the ones that tests for legitimate characteristics have negative weights. To classify a message, all the rules are tested and the message is classified according to the sum of the weights of all the rules that matched. If this sum is greater than a predefined threshold, the message is classified as spam. The most commonly used system that implements this mechanism is *Spamassassin* [13].

C. Bayesian Filters

Bayesian filters are an evolution of rule-based systems because the rules to classify the messages are automatically created when training the filter. The filter uses a Bayesian classifier, which is trained with some messages previously classified as either legitimate or spam [14]. Using these training messages, the classifier can automatically discover characteristics that are present on both legitimate and spam messages. To classify the messages it is verified if the characteristics already learned by the filter are present on the message. The probability of a message be classified as spam given its characteristics is calculated by the product of the probability that the characteristics are present on spam messages multiplied by the probability of a message be spam divided by the probability that the characteristics are present on all the messages used to train the filter.

D. Reverse DNS

To make the traceability harder, spammers did not correctly configure their reverse DNS address. Thus, another anti-spam mechanism is to verify if the reverse DNS of the IP address of the machine that sent the message is correctly configured. This mechanism, however, has a high rate of false positives because many network administrators do not correctly configure the reverse DNS of the network servers. Nowadays, this mechanism is inefficient because most of the spam is sent by hijacked servers, open relay servers or zombie machines, which are likely to have their reverse DNS properly configured, generating high false negative rates.

E. Sender Policy Framework (SPF)

The main goal of this mechanism is to reduce the spams by making it harder to send messages with a fake sender address. This mechanism requires that every domain specify through an SPF record what machines can send messages with the sender address from the domain. The SPF record consists of a series of tests that must be executed to verify if the machine sending the message can really send email for the domain. The tests can verify if the message was sent from an specific IP address space, if it was sent from one of the servers listed on the MX DNS records from the domain and other similar tests [15]. When a message is received, this mechanism checks if the domain of the sender has published an SPF record. If the SPF record is present, the tests specified by the record are executed. The message is discarded if the tests indicate that the message was sent from a machine not allowed to send message for the domain.

V. Anti-Spam Monitor Tool

Many anti-spam systems are available nowadays, but some of them do not give the option to just tag the messages as spam, instead of discarding them. It is important just to tag the message because if the message is discarded by one mechanism it is not going to be analyzed by the other mechanisms, giving false results. Most of the mail servers use a combination of anti-spam mechanisms, making it hard to monitor the performance of each mechanism individually. This section describes the tool developed to monitor the performance of the most commonly used anti-spam mechanisms and to analyze the process used by spammers to harvest addresses and send spams. This tool was developed because there was not any other tool available that would permit this analysis. The architecture of the monitoring tool is shown on Figure 1. The system developed has a module that analyzes each of the anti-spam systems implemented and instead of discarding the messages, it just tags each message as spam or not according to each mechanism. The system is also composed of a module that acts as honeypot, publishing email addresses on a web page as a trap for spammers. There is also a module that is responsible for analyzing all the data from the honeypot module and also from the module that implements the anti-spam mechanisms.

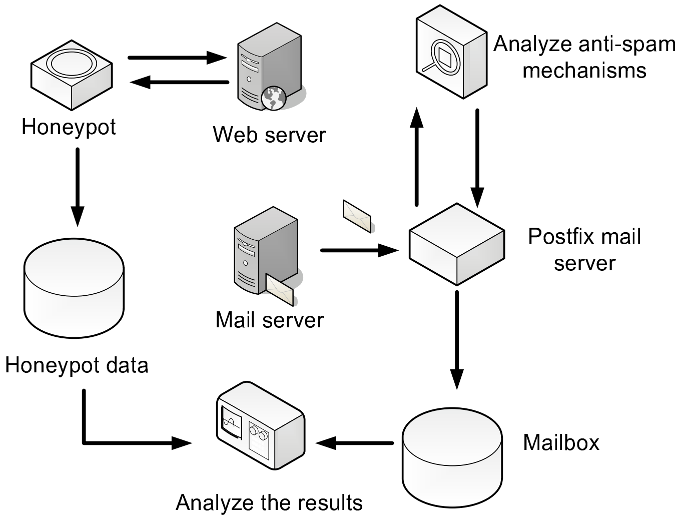


Figure 1. Architecture of the monitoring tool.

The honeypot module publishes email addresses that are randomly generated and does not belong to any legitimate user. The addresses are published at the first page of our research group website with a white color on a white background, making it difficult for a person to notice that the addresses are on the page but easily available for spammers' harvesting robots. The honeypot implementation publishes each address only once and they are changed every time the page is visited. This procedure allows our tool to discover the time between the spammer harvests the address and the time he/she sends the messages, how many messages are sent to each harvested address, and how long the address continues to receive messages, which is a direct indication of the address lifetime on spammers' lists.

The module that analyzes the anti-spam mechanisms, implements the following anti-spam mechanisms: blacklist, reverse DNS, and Sender Policy Framework. The mechanisms of blacklist queries the five [1] most commonly used blacklists [10]. This module was implemented using the Perl programming language as a Postfix policy server [16] which is a framework for developing Postfix extensions that can add information to the header of the message and control the message delivery. The rule-based mechanism *Spamassassin* and the Bayesian filter mechanism are also analyzed by our tool, but as the implementation used just tags the messages and does not discard them, they were not included on this module. The default configuration of the mechanism *Spamassassin* was used

[1] sbl-xbl.spamhaus.org, cbl.abuseat.org, dnsbl.sorbs.net, list.dsbl.org, and bl.spamcop.net

together with a collaborative spam database called Razor [17]. This configuration was chosen because it is the configuration commonly used. Our implementation analyzes every message according to each anti-spam mechanisms implemented and add on the header of the message a line with the result of each mechanism. The line added to the message is based on the extension field standard [18] which states that extension fields should begin with "X-" followed by a name and a value, separated by a colon. We used the name SPAM and the value is composed by parts separated by the symbol @. Each pair of parts represents the name of the mechanism and its result. Figure V shows the structure of the line added to the header. To monitor the performance of a new mechanism, we just need to verify the message received with the new mechanism and append more two parts to the line added to the header of the message. This flexibility makes it easy to analyze new mechanism using our tool.

X-SPAM: @name$_1$@result$_1$@...@name$_n$@result$_n$@

Figure 2. Structure of the line added to the message header.

The last module of the tool is responsible for analyzing the results of the two previous modules. Based on the information added on the header of each message by the module that checks the anti-spam mechanisms, it can calculate the false positive and false negative rates of each mechanism, comparing the result with a previous manual classification of the messages as spam or legitimate. One important characteristic of this module is that it has a strong limitation of the data that can be stored on the memory during the analysis, as we are analyzing every message and the sum of all the messages is on the order of magnitude of tens of gigabytes. This module also permits the analysis of spam characteristics that differ from characteristics of legitimate messages, like the time distribution, IP address distribution, and size distribution.

VI. Results

To be able to monitor the performance of the mechanisms, messages received by ten different users are manually classified as spam or not. Besides that, messages to the addresses published on the honeypot and also messages to recipients which did not exist on our domain are taken into consideration on the analysis as spam messages, because they do not belong to any valid user, so all the messages received are most likely spam. The analyses are based on a 1-year data which corresponds to 41,042 legitimate messages and 791,574 spams.

To benchmark the bayesian filters the implementation of bayesian filters from Mozilla Thunderbird is used. On this test, the group of legitimate and spam messages is separated on two groups sorted by the date the messages were received. The group of spam and legitimate messages that are older is used to train the filter and the newer groups are used to benchmark the filter. This way we can simulate the scenario where some messages already received are used to train the filter and the filter is used to classify new messages that are received.

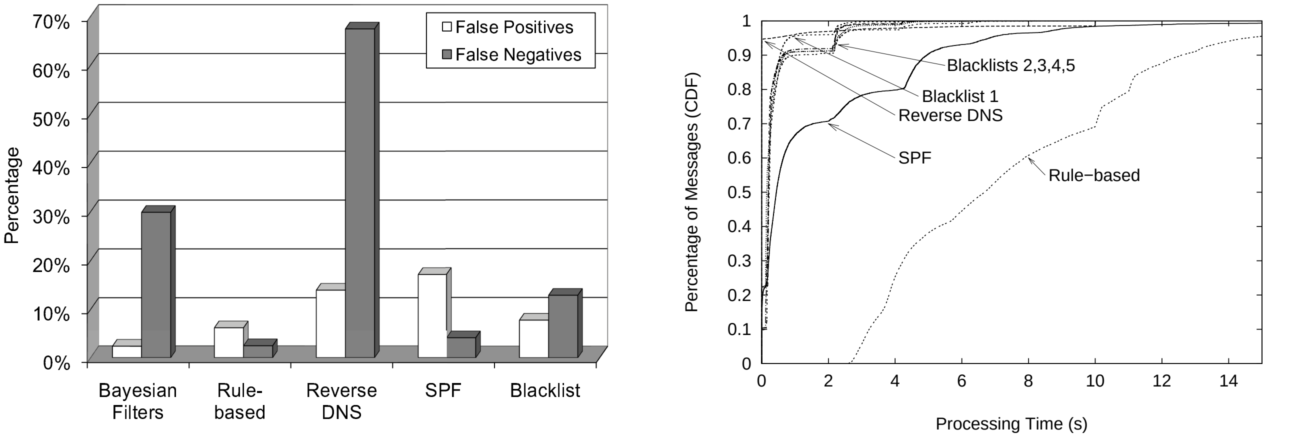
A. Anti-spam mechanisms

Figure 3(a) shows the percentage of false positives and false negatives for the analyzed anti-spam mechanisms. The SPF mechanism requires that the domain of the sender publishes an SPF record to be able analyze the message. As many servers still do not publish SPF records, this mechanism was only able to analyze 53.6% of the legitimate and 19.9% of the spam messages. We can observe that the false positive rate for the reverse DNS mechanism is considerably large, indicating that many legitimate messages would be discarded if this mechanism was used to block the messages. The high rate of false positives for the reverse DNS mechanism also shows that many legitimate servers do not properly configure the reverse DNS. The other mechanisms show lower false positive rates but high enough to still impact the users. Considering the false negative rates, all the mechanisms present high rates. The high rate of false negatives for the reverse DNS mechanism is a consequence of the use of zombie machines and third-party servers that have the reverse DNS properly configured.

Figure 3(b) shows the cumulative distribution function (CDF) of the percentage of messages versus the period of time required by each mechanism to analyze the message. This time is affected by many factors, such as the machine capacity and network capacity, but as they were all run on the same machine, we can still compare then. SPF, black lists, and reverse DNS mechanisms use DNS queries, but the reverse DNS presents the fastest results. The reason is because the mail server already does the query of the reverse DNS when it receives a message. This way, when the reverse DNS mechanism makes the query again the result is already cached, reducing the time it takes to analyze the message. Among all the lists, blacklist number one, `sbl-xbl.spamhaus.org`, presents the best results and, possibly due to network and processing capacity of the servers responsible for this list. The rule-based mechanism presents the worst result, taking up to 12 seconds for 90% of the cases. This result is due to the high processing time required for each message and also because some external collaborative spam identification databases are also queried. The time required to analyze the messages using the bayesian filter could not be analyzed because the implementation used does not offer any way to measure the time required to classify each message.

B. Spams characteristics

Spams have characteristics that are different from legitimate messages. Spammers have a different relation with the users, as they try to send messages to the maximum number of recipients and most of the users do not send any message to the spammers. This relation is different for legitimate messages where users typically reply the messages and they are not sent to a large number of people. Legitimate users also habitually send the messages during the morning and afternoon while the spammers send messages all the day long.



(a) Percentages of false positives and false negatives.

(b) The cumulative percentage of messages versus the processing time.

Figure 3. Benchmark of anti-spam systems.

Figure 4(a) shows the percentage of messages versus the hour the message was received. We can see the number of legitimate messages received between 12AM to 6AM is low, while the spam distribution is more constant, indicating that the messages are sent continuously, as we would expect from an automated mechanism to send spam.

Figure 4(b) shows the percentage of messages versus the size of the messages. Most of the spams have a smaller size than legitimate messages. Initially, the spammers sent very small messages, to send the messages to more users in less time. Nowadays, however, spammers are sending messages with images trying to bypass anti-spam filters that analyze the content of the messages. This technique ends up making the size of each message larger. When comparing this result with results found on [5], when the images in spams were not common, the average size of the spams has increased. Nonetheless, the spams are still smaller than legitimate messages.

Another characteristic that is different between legitimate and spam messages is the distribution of the IP addresses that send the messages. Figure 4(c) shows the percentage of messages versus the IP address that sent the messages. We can see that the subnet *146.164.0.0/16* is responsible for a large number of legitimate messages, because this is the subnet of our university and a large number of legitimate messages comes from people inside the university. The distribution of IP addresses that send spams is more distributed along the IP space. To compare this distribution with the distribution of machines on the Internet, we sent $500,000$ ping probes to random IP addresses. Figure 4(d) show the percentage of machines that send spam and reachable machines that answered the ping probes versus the IP address. Both distributions are similar, indicating that the process of sending spam is globally distributed, due to the use of botnets.

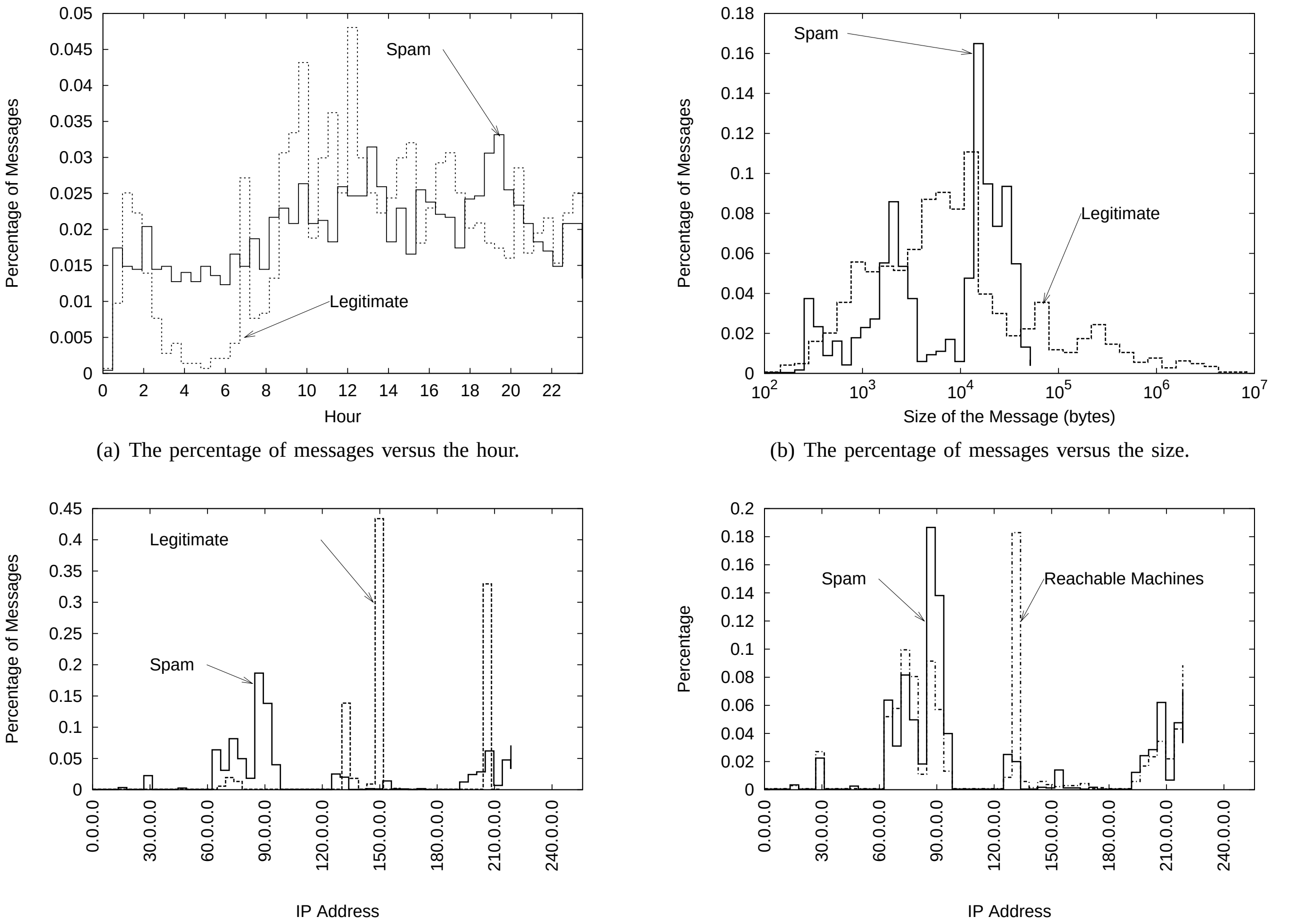
C. Honeypot Analysis

The honeypot that is part of our tool monitors some characteristics of the processes used by spammers to harvest email addresses and send the spams. The addresses published on the page are changed on every visit to the page, ensuring that we can later know exactly when the address was harvested and when the first message was sent to the address that was harvested. Out of the 41,119 different recipients of the messages received by the honeypot, 9,740 or 23.69% were addresses published on the honeypot. All the other addresses are due to random guessing or dictionary attacks. We also analyzed the user agents mostly used by the robots that harvested the site and determined that 94.5% of the robots use the string “Mozilla” on the user agent. This string is present on the user agent claimed by most of the commonly used web browsers, making the detection of the robots more difficult.

Figure 5(a) shows the CDF of the percentage of the time between the address is harvested and the first message is received. This period of time is normally large, as about 70% of the messages is sent just after two months the address is harvested. The email addresses generated appear only once in the honeypot, so the time between the address is harvested and the first message is received can be precisely determined. Real-world addresses which appear on web pages will start to receive messages earlier, because 30% of the spammers send the messages in less than two months.

Figure 5(b) illustrates the CDF of the percentage of addresses that received a given number of messages. The results show that about 13% percent of the addressees received more than 20 messages, indicating that the number of spams sent to each address harvested is low.

To analyze the time the addresses continues to receive messages, we initially calculated the difference between the first and the last message received. This difference, however, is not representative of the time the addresses continues to receive spam. If an address is published and receive messages near the end of the period analyzed the difference will be low but this address may continue to receive messages on the future. To better evaluate the time the addresses are kept on the lists, we normalized the difference between the first and the last message received by the difference between the



(a) The percentage of messages versus the hour.

(b) The percentage of messages versus the size.

(c) The percentage of messages versus the IP address.

(d) The percentage of reachable machines and machines that send spam versus the IP address.

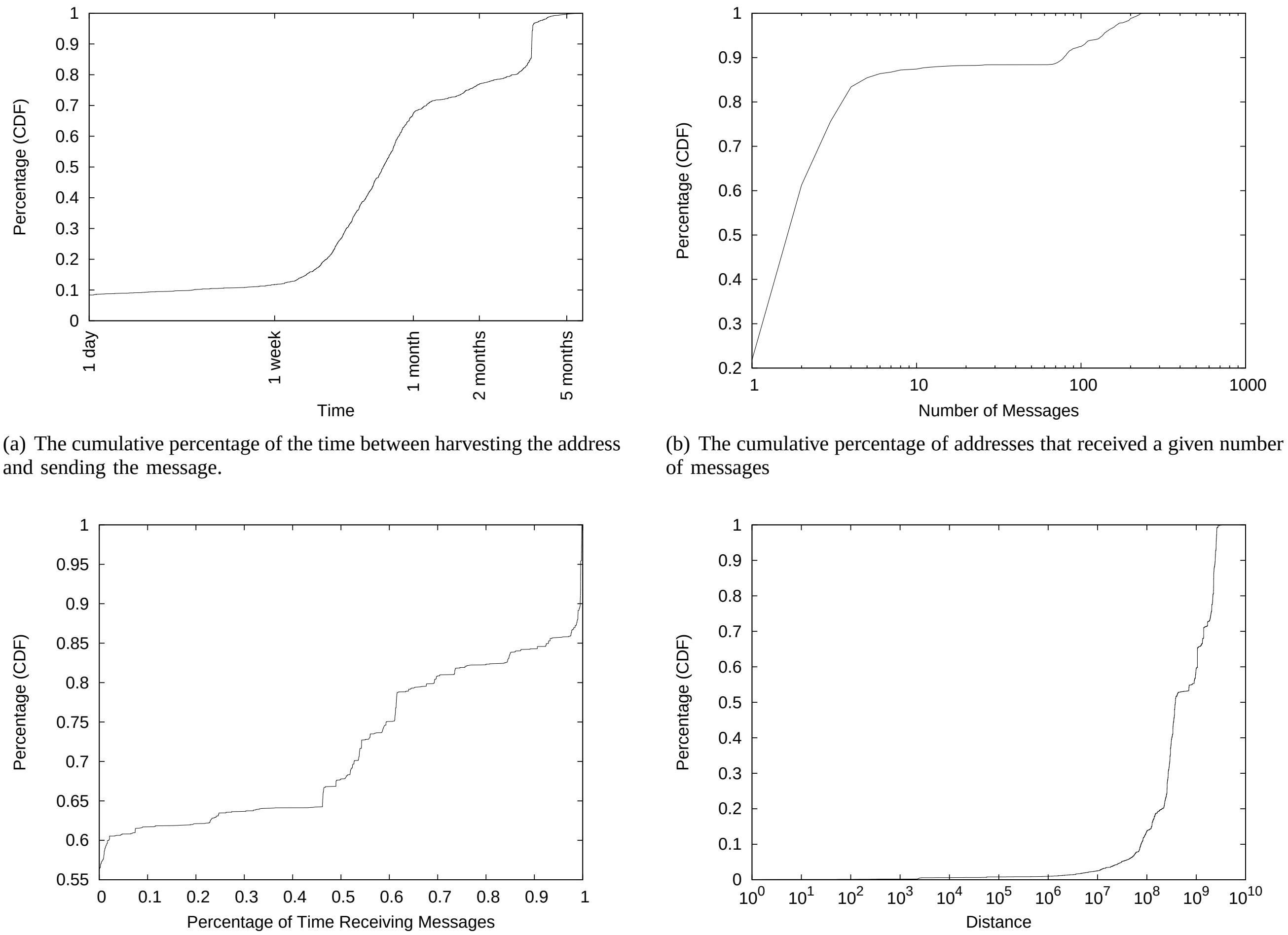
Figure 4. Spams characteristics.

time the address was published and the time of the end of observation. This way we can calculate the percentage of time the published address receives spam during the observation. Figure 5(c) shows the CDF of the percentage of the time the address received spam. We can observe that about 55% of the addresses only received messages for a small period of time. On the other hand, about 15% of the addresses received messages during the whole period. This indicates that different spammers use different strategies to send spam.

We also analyzed if the processes of harvesting and sending the messages are carried out by the same machine or by different machines. Figure 5(d) shows the CDF of the percentage of the distance between the IP address used to harvest the address and the IP address that sent the first message. The distance between two different IP addresses was calculated as the absolute value of the difference between the decimal representations of both IP addresses. The graph shows that in about 90% of the cases the distance is greater than 10^8. This result indicates that most of the spams are sent from different subnets because the difference between two addresses like $x.y.0.0$ and $x.y.255.255$ which represents the start and the end of one class C networks is 65,535. This might be the result of a distributed process between zombie machines, where some of them are responsible to harvest addresses and some others to send the messages.

VII. Conclusion

This paper proposes a tool that monitors the performance of the most commonly used anti-spam mechanisms, monitor the characteristics of spam messages, and also monitor the process used by spammers to harvest and send the messages. The tool developed can also be used to monitor new anti-spam systems which can be easily incorporated on our tool. The results show a high rate of false negatives for all the mechanisms, between 2.4% and 67.4%. All the mechanism analyzed had false positives greater than 2.3% which is high, considering the negative impact for the users. Users receive about 3000 legitimate messages per year. So if the false positive rate is 2.3%, then 69 legitimate messages will be incorrectly classified each year, which is considerably high. The results show that the reverse DNS mechanism is highly incorrect, having a false positive rate of 13.9% and a false



(a) The cumulative percentage of the time between harvesting the address and sending the message.

(b) The cumulative percentage of addresses that received a given number of messages

(c) The cumulative percentage of time the addresses received messages.

(d) The cumulative percentage of the distance between the IP addresses used to harvest and send the message.

Figure 5. Honeypot analysis.

negative rate of 67.4%. Bayesian filters had the best results, having 2.3% of false positives.

The honeypot developed also showed that the period of time between the address harvesting and the effective use of these addresses by sending spams to them is relatively large, on the magnitude of weeks or months. The analyses also showed that the process of harvesting and sending the messages, most of the times, are accomplished by different machines. The long time between the spammers harvest the email address and the time the message is received may be due to the spammers who collect email addresses just to make email lists and then these lists are sold to other spammers who send the messages. This way, an approach of trying to detect the IP addresses used to harvest addresses and include them on some kind of blacklist is likely to be inefficient, because on more than 90% percent of the cases the IP address used to harvest addresses is not even on the same network of the IP address that send the messages. Besides that, the lifetime of the addresses on spammers' lists is often long. This result indicates that once an address is published on a website or is harvested by the spammers by any other mean, it is included on spammer lists for many months, reducing the efficiency of removing addresses published on websites. This procedure may help reduce the number of spams received because the address will not be included on new spammer lists that are being created, but the address will be kept for months on existing lists.

The monitoring tool presented on this paper can be used to better understand the spammer behavior and based on this behavior new anti-spam mechanism can be developed.

Today we do not have any indicatives that the spams will reduce on the next years. In contrast, we just expect that the spammers will send more and more messages. Spammers are in constant evolution, trying to bypass new anti-spam mechanisms or sometimes even trying to get ahead of the capacity the mechanisms have to classify the messages as spam. This evolution is expected to exist for a long time and presents a challenge for the designers of anti-spam systems. To help the developers of anti-spam systems on this nonstop challenge, the monitoring tool presented permits them to monitor the current spammer behavior, monitor the performance of current anti-spam systems and analyze new anti-spam mechanism developed.

REFERENCES

[1] S. L. Pfleeger and G. Bloom, “Canning spam: Proposed solutions to unwanted email,” *IEEE Security & Privacy Magazine*, vol. 3, no. 2, pp. 40–47, Mar. 2005.

[2] M. Wendland, “Spam king lives large off others’ e-mail troubles,” *Detroit Free Press*, Nov. 2002.

[3] B. Hoanca, “How good are our weapons in the spam wars?” *IEEE Technology and Society Magazine*, vol. 25, no. 1, pp. 22–30, Apr. 2006.

[4] G. Cormack and T. Lynam, “TREC 2006 spam evaluation kit,” http://plg.uwaterloo.ca/∼gvcormac/jig/, accessed on January 04 2008.

[5] L. H. Gomes, C. Cazita, J. M. Almeida, V. Almeida, and J. Wagner Meira, “Characterizing a spam traffic,” in *ACM SIGCOMM conference on Internet measurement (IMC’04)*. ACM Press, 2004, pp. 356–369.

[6] ——, “Workload models of spam and legitimate e-mails,” *Perform. Eval.*, vol. 64, no. 7-8, pp. 690–714, 2007.

[7] G. Schryen, “An e-mail honeypot addressing spammers’ behavior in collecting and applying addresses,” in *Systems, Man and Cybernetics (SMC) Information Assurance Workshop*, 2005, pp. 37–41.

[8] M. Andreolini, A. Bulgarelli, M. Colajanni, and F. Mazzoni, “Honeyspam: Honeypots fighting spam at the source,” in *SRUTI05: Steps to Reducing Unwanted Traffic on the Internet Workshop*, 2005, pp. 77–83.

[9] Project Honey Pot, “Distributed spam harvester tracking network,” http://www.projecthoneypot.org/, accessed on January 04 2008.

[10] A. Ramachandran and N. Feamster, “Understanding the network-level behavior of spammers,” in *SIGCOMM ’06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2006, pp. 291–302.

[11] A. Ramachandran, N. Feamster, and S. Vempala, “Filtering spam with behavioral blacklisting,” in *CCS ’07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2007, pp. 342–351.

[12] J. Jung and E. Sit, “An empirical study of spam traffic and the use of DNS black lists,” in *ACM SIGCOMM conference on Internet measurement (IMC’ 04)*. ACM Press, 2004, pp. 370–375.

[13] Apache, “The apache spamassassin project,” http://spamassassin.apache.org/, accessed on January 04 2008.

[14] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A bayesian approach to filtering junk e-mail,” in *Learning for Text Categorization: Papers from the 1998 Workshop*. AAAI Technical Report WS-98-05, 1998.

[15] M. Wong and W. Schlitt, *Sender Policy Framework (SPF) for Authorizing Use of Domains in E-MAIL, Version 1*, RFC 4408, Apr. 2006.

[16] Postfix, “Postfix SMTP access policy delegation,” `http://www.postfix.org/SMTPD_POLICY_README.html`, 2006, accessed on January 04 2008.

[17] V. V. Prakash, “Vipul’s razor,” http://razor.sourceforge.net/, accessed on January 04 2008.

[18] D. H. Crocker, *Standard for The Format of Arpa Internet Text Messages*, RFC 822, Aug. 1982.