

Blocking Spam Sessions with Greylisting and Block Listing based on Client Behavior

Pin-Ren Chiou*, Po-Ching Lin*, Chun-Ta Li*

*Department of Computer Science and Information Engineering

National Chung Cheng University, Chiayi, Taiwan 62102

Email: {cpj97u,pclin,lcta99m}@cs.ccu.edu.tw

Abstract—*Greylisting and real-time block listing are two common mechanisms for spam filtering. The former can efficiently reduce the number of spam sessions, but can be easily evaded if spammers retry spam delivery within a period of time. The latter can detect and reject the clients with poor reputation during SMTP sessions by looking up an external blacklist. Its weaknesses are the high false-positive rate and a lack of locality. We design a method of dynamically updating the greylist and block list based on the delivery behavior of spammers to block spam sessions in time. The method is implemented on a mail gateway in a senior high school. In our experiments with the real-world mail traffic for a month, the method can block 70.29% and 69.21% of the known and possible messages in the spam sessions with greylisting and block listing. The forward and reverse DNS lookups are also adopted to further reduce the false-positive rate to under 0.01%. Therefore, the required system resources for further spam filtering on mail servers can be greatly saved by blocking most of the spam sessions in time.*

keyword: greylist, block list, spam, IP reputation, delivery behavior.

I. INTRODUCTION

It was reported that over 90% of email messages are spam [1]. Numerous spam messages have been a serious problem to both users and IT administrators for decades. Most anti-spam products are post-filtering systems, which store incoming mail messages in a queue before scheduling them for spam filtering [2]. Therefore, spam filtering for a large number of spam messages will significantly consume the computation and storage resources of mail servers, and cause delays to mail delivery.

Most anti-spam techniques can perform a reputation check on the clients by looking up an external real-time block list (RBL) before accepting incoming mail messages [3]. Nevertheless, spammers can leverage open relays, botnets, or poorly configured servers to relay spam rather than deliver spam by itself. It is therefore difficult for an RBL to track so many possible IP addresses in time. Moreover, when a client starts spam delivery, there will be a window of time before it is captured by a spamtrap or reported from users [4], [5]. Therefore, the spam delivered in the window of time will bypass the RBL and consume the system resources for spam filtering. Greylisting is another common spam-filtering technique that rejects newly arriving clients which are not white listed. A mail message will be accepted when the client retries after a period of time. Spammers therefore can easily evade greylisting by retrying later.

This work presents a method to determine whether a client is likely to be a spammer or not by analyzing the client behavior recorded in the mail transaction logs. The method of building a local RBL is collecting the error messages during the SMTP sessions, the results returned from the spam filter and the DNS record of the client with several validation mechanisms. The RBL can block spam sessions in time and save the system resources for spam filtering. The client's IP address extracted from the mail transaction logs is also sent to the enhanced greylist daemon for evaluation, and the final result is compared with that of the RBL. The comparison will help us to understand better the right time, the order and the effectiveness to apply the greylist and RBL. In our experiences, using the public RBL services occasionally results in mail losses due to false positives. We use a DNS lookup method to verify the relationship between the host name of the client and the sender domain to reduce the false-positive rate. This work also queries several famous RBL services with the IP addresses in the local RBL we have built. On average, 63.74% of the IP addresses listed in the local RBL database are not listed in the RBLs such as SpamCop, Passive Spam Block List (PSBL), Spam and Open Relay Blocking System (SORBS), Composite Blacking List (CBL).

The remainder of the paper is organized as follows. Section II reviews the common error codes of mail delivery, the mail processing in Amavisd-new and related works. Section III presents the method of analyzing the behavior of spammers and the automatic generation of the greylist and the RBL on the mail gateway. Section IV presents the validation results and compares the performance of this work with that of the popular RBL services. Section V concludes this work.

II. BACKGROUND AND RELATED WORK

Spam sessions are likely to raise error messages due to their unusual behavior. In this section, we therefore review the error messages in SMTP first, and then introduce the Amavisd-new integrated in the gateway system. Finally, we review the related work of sender-based spam-filtering techniques.

A. Error messages in SMTP

The RFC 3463 [6] describes the error codes in SMTP. The most common one about the security or policy status is 5.7.1, meaning that the mail transfer agent (MTA) refuses to relay the mail for the client. We developed the system based on the

RFC to analyze and extract the error codes recorded in the mail transaction logs of Postfix. According to the observation of real-world mail traffic, we summarize three events that cause the error code 5.7.1 with an error message appended in the mail transaction logs as follows.

- 1) *Illegal relay*: The client asks the MTA to relay mail for an unauthorized destination. The error code 5.7.1 is appended with the error message `Relay access denied`.
- 2) *Illegal sender Address*: The client uses an unauthorized sender address and is caught by the MTA. The error code 5.7.1 is appended with the error message `Sender address rejected`.
- 3) *Unknown client*: The reverse DNS lookup of the clients IP address fails, or the result of forward and reverse DNS lookup is a mismatch. The error code 5.7.1 is appended with the error message `Client host rejected`.

We ignore the third error message due to possible timeout when Postfix performs DNS queries under heavy SMTP traffic. We only watch the first and the second events, and exclude the third to avoid rejecting normal clients. In the second event, if the clients ask the mail service provider to send mail for themselves, the system can additionally validate the sender policy framework (SPF) record or the domain keys (www.dkim.org) to avoid false positives.

Moreover, we also consider the other error codes, but track only the error code 4.1.8 with the error message `Domain not found` because the other error codes may be appended with other irrelevant error messages such as `non-smtp command`. Some spammers fake the domain name in the `MAIL FROM` column in the mail header, so the domain name may not even exist [7]. The domain name provided by the client or sender should be official and public, according to RFC 821. Some administrators will enable the sender domain checking mechanism to ensure the sender domain exists and is acceptable.

B. Amavisd-new

We use Amavisd-new to cooperate with SpamAssassin [8] and ClamAV, which can help us perform spam filtering and virus checking before delivering the mail to end-user mailboxes. Figure 1 presents the internal mail processing flow. Amavisd-new can parse for the IP addresses of the client and the original sender, and record them into the mail transaction logs. After the client passes the sender-based check, Postfix stores the mail into a queue and then passes it to Amavisd-new for spam filtering and virus checking. If a mail message passes all of the checks, it will be sent back to Postfix and scheduled for local delivery; otherwise, Amavisd-new will generate the information about the spam in the mail transaction logs.

The latest version of Amavisd-new allows the administrators to customize the scheme of notification mail with the predefined parameters corresponding to the information of spam such as the sender's address, the recipient's address, etc. Amavisd-new outputs the IP addresses of the sender and the client into the mail transaction logs with the final result of

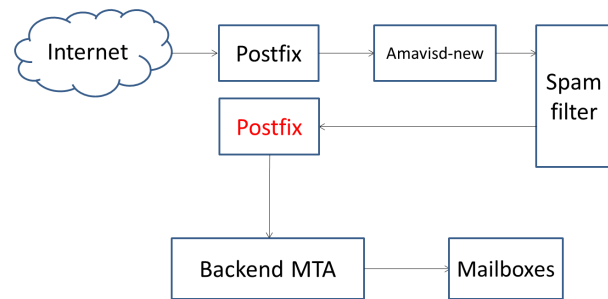


Fig. 1. The internal mail processing flow.

spam filtering. We confirm how Amavisd-new extracts the two addresses by writing to the author of Amavisd-new. According to him, the two IP addresses in the mail transaction logs are derived as follows:

- 1) The client's IP address is the immediate SMTP client or MTA connected to our gateway.
- 2) The best guess of the sender's IP address is derived from the received trace by the bottom-up mail header parsing.

The two IP addresses differ from each other only when the spammers use a third-party mail server to relay spam instead of relaying by itself [9]. Spammers can confuse the spam filter by adding a fake mail header containing the IP address of the innocent mail server into the mail header. The problem can be overcome by checking the abnormal path in the mail header and performing two-way DNS validation.

C. Related Work

Spam detection has been an active research topic for decades. Despite the existence of numerous techniques, this work focuses on reviewing the related work of sender-based techniques. Ramachandran et al. proposed ways to infer botnet membership and identify spammers by clustering email servers based on their target email destination domains [4] and by monitoring queries to DNSBL [5]. T. Sochor presented the long term analysis of the anti-spam effect on using greylisting with case studies [10]. Qian et al. [11] presented their work on spam detection using network-based clustering, showing that network-based clusters can increase the accuracy of spam-oriented blacklists. G. Stringhini et al. [12] presented their work on clustering botnets by analyzing the DNS query sent to DNSBL, mail transaction log and the spam mails collected on the spamtrap. An appliance, namely Email Security Virtual Appliance (ESVA, see www.esvacommunity.com), also aims at spam filtering. The difference between our work and ESVA is that we integrate Amavisd-new with Postfix for processing incoming email messages on a physical machine. Amavisd-new helps us to parse useful information from each incoming mail messages because we need to analyze the mail log and change the inner routing path between the parser and the MTA.

Most blacklisting mechanisms establish a list of suspicious IP addresses. As mentioned in Section I, the lack of locality and the window of time before the spammers' IP addresses are listed in the RBLs is a great limitation. Replacing individual

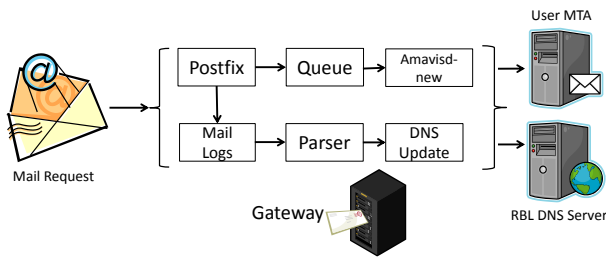


Fig. 2. The system architecture of the gateway.

IP addresses with clusters [13] can improve the coverage of spamming IP addresses [14], but may increase the false-positive rate in spam filtering. This work focuses on analyzing the behavior and DNS information of the clients during the SMTP sessions. With the validation described in Section III, we can generate and update the local RBL in time before the spammers come again. Like the project presented in [10], we also integrate the checking mechanism in building the local RBL into the greylist design.

III. SYSTEM ARCHITECTURE

The gateway system can parse the mail transaction logs for the error messages introduced in Section II-A and refer to the results from Amavisd-new to update the RBL built on a DNS server. The detailed methods are introduced in the following subsections.

A. Mail Processing Flow

We implemented the gateway system in a senior high school in the Taiwan Academic Network. Figure 2 presents the system architecture of the gateway. The MX record entry of the domain is modified to redirect the mail traffic to the Postfix (www.postfix.org) daemon on the gateway. The following information of a client is checked before its mail is forwarded to Amavisd-new via the queue for spam filtering and virus checking. If one of the checks fails, Postfix will refuse the delivery request from the client.

- 1) *Destination check*: We check the recipients' email addresses that the client tries to send to when an SMTP session is established. If the recipient does not belong to the domain we serve, the MTA will generate the error code 5.7.1 with the error message `Relay access denied`.
- 2) *Local RBL check*: The MTA will refuse to talk to the client already listed in the local RBL, and generate the error code 5.7.1 with error message `Service unavailable; Client host [IP] blocked by rbl.domain %Reason`.
- 3) *Sender address verification*: If the client passes the above two checks, we will verify the sender's address to prevent unauthorized use of our domain by spammers. The MTA will generate the error code 5.7.1 with error message `<username@domain.name>: Sender`

address rejected: You are not from domain.name, if the verification fails.

- 4) *Sender domain verification*: We will verify the domain by the method mentioned in Section II-A. The MTA will generate the error code 4.1.8 with error message `<username@domain.name>: Sender address rejected: Domain not found, if the verification fails.`

B. Validation of the Sender's IP address

If a spam messages is from a mail service provider, listing in the RBL is dangerous because other normal messages from the mail service provider will be blocked as well. The following checks can verify whether the sender's IP address is from a mail service provider or not. If it passes the checks, the IP address will not be listed in the local RBL.

- 1) *FCrDNS validation*: Also known as forward-confirmed DNS, the verification first performs a reverse DNS lookup on the clients IP address. The lookup will return a list of zero or more PTR records. Second, the forward DNS lookup is performed for each domain name in the returned PTR records. The IP addresses returned from the forward DNS lookup will be compared with the clients IP addresses. The reasons that the FCrDNS validation will fail are listed as follows.
 - The client's IP address has no PTR records.
 - The PTR records cannot be resolved.
 - The IP address returned from the forward DNS lookup is a mismatch with the client's IP address.
- 2) *Dynamic IP verification*: We collected a list of common keywords, including `dynamic`, `static`, `NAT`, `pppoe`, etc., in the host names of the IP addresses allocated dynamically by an Internet service providers (ISP). Moreover, some ISPs also combine the ASCII representation of an IP address with the ISP's domain name as the host name of the IP address. For example, the host name of the IP address 94.44.147.76 is `apn-94-44-147-76.vodafone.hu`. We can search the keywords or the ASCII representation of the IP address in the host name to verify whether an IP address is allocated dynamically by an ISP.

C. Implementation of the Parser

We duplicate the log stream with pipe into a FIFO file with `syslog-ng`. Our parser reads the FIFO file as input in the non-blocking mode, and the log stream will be analyzed in real-time to avoid consuming the disk space. The parser makes the `popen` system call to read line by line from the FIFO file and search the following keywords: (1) `Relay access denied`, (2) `You are not from domain.name`, (3) `Domain not found`, and (4) `Blocked SPAM`.

The local RBL is built with the BIND (www.isc.org/software/bind) DNS system. The utility `dnsupdate` is called to update the RBL with the IP addresses extracted from the log stream when the first or the second keyword is matched. The parser will try to look up the host name of the client when the

TABLE I
THE STATISTICS OF THE EVENTS ON THE SYSTEM.

Event	Count	Rate
SMTP sessions from clients	237,697	N/A
Processed mail messages	208,667	N/A
SMTP sessions rejected by block listing	67,853	28.54%
SMTP sessions rejected by greylisting	71,426	30.05%
Distinct IP addresses between block listing and greylisting	8,672	N/A
Quarantined spam messages	88,304	42.31%
Possible spam messages	2,263	1.08%
Normal mail messages	118,100	56.59%

third keyword is matched. The domain part is extracted from the host name, and compared with the sender domain. If the host name does not belong to the sender domain, we consider the client a spammer using a fake sender domain and add the client's IP address into the RBL. When the fourth keyword is matched, the parser will perform the checks described in Section III-B to avoid listing the IP addresses from a mail service provider.

As mentioned in Section II, the error message `Client host rejected` is ignored because the DNS queries may be likely to expire when Postfix performs DNS lookup with heavy mail traffic to process. Nevertheless, the parser can perform the DNS lookup, and the timeout limit depends only on the default setting on the local DNS cache server.

We develop the validation program with Perl. This program reads the mail transaction logs and extracts the IP address after the beginning keyword, `Connected from`, in the mail transaction logs. Two checks described above are performed to determine if we should greylist the client. If the checks on the client's IP address fail, the enhanced greylisting will reject the session from the client temporarily.

IV. EVALUATION

We evaluate how effective the enhanced greylist and the local RBL can block spam sessions in this section. Moreover, we also summarize the domains and the sources of the spammers observed in this work, and discuss the consideration of implementing the purge mechanism.

A. Analysis of Experimental Results

We deployed the gateway system described in Section III from June 20, 2011 to July 20, 2011. Table I summarizes the statistics of the events on the system over the period. The system handled totally 237,697 requests to establish SMTP sessions from external clients. The local RBL blocked 67,583 out of the requests for the gateway. In our observation, the spam delivery in each session contains about three recipients on average, meaning that if a delivery succeeds, the gateway will have to process a spam message and duplicate it to three recipients. Therefore, the RBL mechanism can save the storage space for more than 200,000 spam messages on the gateway.

Table II lists the average number of connection attempts from each IP address over the period. According to the analysis, each spammer that appeared only in one day made 2.2

TABLE II
NUMBER OF CONNECTION ATTEMPTS FROM EACH IP ADDRESS.

Duration (in days)	Average number of attempts
1	2.2
2~9	9.3
>10	139

TABLE III
TOP FIVE DOMAINS OF THE SPAMMERS.

Domain name	# of spam messages	# of fake domains
returns.groups.yahoo.com	6,308	0
yahoo.com	1,578	802
yahoo.com.tw	1,019	617
googlegroups.com	865	2
gmail.com	623	615

connection attempts on average, each spammer that appeared in from two to nine days made 9.3 connection attempts on average, and so on. The analysis implies that the clients can revisit the mail server in short time.

Because of the frequent revisiting from some spammers, greylisting can reject more SMTP sessions than the RBL. Nevertheless, the parser performs more checks for building the local RBL than for the greylisting validation. The RBL thus can detect 8,672 more IP addresses than greylisting. According to Table I, we can reject more spam sessions by applying the greylisting before block listing.

We also summarize the domains of the spammers' IP addresses and the senders' domains over the observation period. Totally 4,066 unique domains were found. Table III lists the top five domains of the spammers and the number of spam messages from those domains. We also verify whether the senders' domains are faked by spammers with the domain keys, and list the number of them in the table. Up to 24.15% of spam messages were sent from the top five domains. The results indicate that spammers like to steal or register accounts provided by email service providers. Few mail servers block mail messages from these service providers due to the numerous users on them. Therefore, we will extend the current method to detect such spammers behind the scenes in the future work.

Totally 24,170 unique IP addresses of the spammers were observed in this work. According to the GeoIP database (www.maxmind.com), the spam sources by country are sorted and summarized in Figure 3. The top four sources where the spammers come from are China, India, Russia and Taiwan. They differ significantly from the statistical results provided by the well-known RBL provider, CBL. The results indicate that spammers choose the targets of spam delivery according to the affinity in terms of languages and areas, and thus the RBL has strong locality, meaning that referring to an external RBL for blocking spam sessions will miss many spammers' IP addresses that appear locally only on the system.

We compare the local RBL built in this work with other contemporary RBLs from well-known RBL organizations by querying them with the IP addresses in the local RBL. Ta-

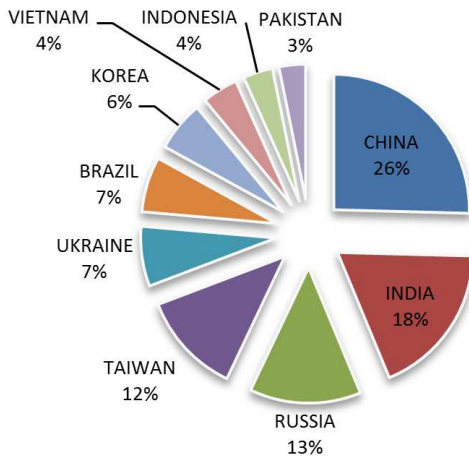


Fig. 3. Sources of the spammers.

TABLE IV
COMPARISONS WITH OTHER CONTEMPORARY RBLs.

RBL organization	Number of listed IP addresses	Percentage
cbl.abuseat.org	10,120	41.87%
fivetensg.com	9,331	38.60%
nsbl.sorbs.net	8,020	33.18%
zen.spamhaus.org	16,300	67.44%
psbl.surriel.com	5,831	24.12%
bl.spamcop.net	2,988	12.36%
this work	24,170	100%

ble IV summarizes the numbers and the percentages of the IP addresses found in those RBLs. The collection methods in some of the RBLs we know are summarized as follows:

- Passive Spam Block List (PSBL) deploys spamtraps to collect spammers' IP addresses. When the spamtraps receive mail messages, they will list the mail senders in the RBL.
- Spamcop computes the weights of the reports from the spamtraps and users in diverse areas. When the weight of a report exceeds a certain threshold, the mail senders will be listed in the RBL.
- Spamhaus provides several RBLs, including SBL, XBL, PBL, ZEN and so on. Among the RBLs, ZEN combines the lists from SBL, XBL and PBL (www.spamhaus.org/zen/index.lasso), so it has a better detection rate than the others.

According to Table IV, only 36.26% of the IP addresses in the local RBL are listed in other RBLs on average. The result proves that collecting a local RBL like this work, rather than just relying on external RBLs, is important for efficiently blocking spam sessions.

B. Effectiveness of Greylisting and Block Listing

We enhanced the mechanism of greylist and tested with the IP addresses extracted from the mail transaction logs for a month. According to the numbers of rejected SMTP sessions in Table I, the enhanced greylist can help to reject more than 1.5% of SMTP sessions. Also, the enhanced greylist

can prevent the spammers from establishing more than one SMTP session within a short time, before the spammers are listed in the local RBL. Assume that S_i denotes the total of quarantined and possible spam messages, B_i denotes the number of spam sessions rejected by the RBL, G_i denotes the number rejected by the greylist, and α denotes the constant of 3, the approximate average number of recipients in each SMTP session. The following equations indicate how we calculate the percentages of the total messages blocked by the RBL and the greylist, denoted by P_B and P_G , respectively:

$$P_B = \frac{B_i * \alpha}{B_i * \alpha + S_i} * 100\%, \quad P_G = \frac{G_i * \alpha}{G_i * \alpha + S_i} * 100\%. \quad (1)$$

According to the results in Table I, P_B is around 70.29% and P_G is around 69.21%. In other words, the two lists can block around 70% of spam messages, and save the consumption of the system resources by rejecting the clients before a mail is accepted by the MTA.

C. Consideration of the Purge Mechanism

The purge mechanism means flushing part or all of the IP addresses in the RBL and restarting the learning. We observed that about 90% of the IP addresses appeared only in one day over the observation period, while the other 10% appeared in more than one day. No matter how frequent the IP addresses are, the clients all attempt spam delivery for multiple times, so we can successfully block them for a large number of times, as presented in Table I. According to the entry of DNSBL in wikipedia, the expiration time of IP addresses in the operating RBLs vary from several minutes to several days. Some organizations allow the users to apply for removing an IP address from RBLs (tool.ietf.org/rfc5782).

Given that totally 24,170 unique IP addresses of the spammers were observed in the observation period, and each IP address takes just four bytes in the storage, it is trivial to keep the total addresses in the memory or disk storage. The expiration time can theoretically be much longer than a month for broad coverage of IP addresses. Nevertheless, only about 10% of the IP addresses appeared in more than one day according to our analysis, meaning that most IP addresses in the local RBL will not be referred to again after one day since their entry, and thus keeping a large number of IP addresses is not particularly useful. Considering these points, we think the expiration time of a few months is acceptable. Moreover, when a spammer revisits the system, the expiration time of its IP address can be restored to the maximum threshold. Therefore, the spammers that revisit frequently will be kept long in the local RBL.

V. CONCLUSION

We present a method to build an enhanced greylist and a local RBL based on the analysis of the client behavior to effectively block spam sessions in time, instead of relying on collecting spam messages on the spamtrap or reports from users. This method can efficiently block around 70% of spam sessions with the false-positive rate lower than 0.01%. The

performance is verified with the real-world mail traffic. On average, 63.74% of IP addresses in the local RBL are not listed in well-known external RBLs, implying that the targets of spamming have locality and relying on external RBLs is insufficient. The enhanced greylist mechanism can reject the spam sessions earlier than the RBL. This method can save the consumption of system resources for spam filtering and effectively speed up the mail processing.

ACKNOWLEDGMENT

This work is supported by the National Science Council (NSC) Project 101-2219-E-194-001-.

REFERENCES

- [1] Messaging Anti-Abuse Working Group, Email metrics program: Report #15 – first, second and third quarter 2011, Tech. rep., http://www.maawg.org/sites/maawg/files/news/MAAWG_2011_Q1Q2Q3_Metrics_Report_15.pdf (First, Second and Third Quarter 2011).
- [2] D. Twining, M. Williamson, J. Miranda, F. Mowbray and M. Rahmouni, “Email prioritization: reducing delays on legitimate mail caused by junk mail,” In Proceedings of annual conference on USENIX, Jun.-Jul. 2004.
- [3] A. Ramachandran, N. Feamster and D. Dagon, “Revealing botnet membership using DNSBL counter-intelligence,” In Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), Jul. 2006.
- [4] A. Ramachandran, N. Feamster and S. Vempala, “Filtering spam with behavioral blacklisting,” In Proceedings of the 14th ACM conference on Computer and Communications Security (CCS), Oct.-Nov. 2007.
- [5] A. Ramachandran, D. Dagon and N. Feamster, “Can DNSbased blacklists keep up with bots?,” In Proceedings of the third Conference on Email and AntiSpam (CEAS), Jul. 2006.
- [6] G. Vaudreuil, “Enhanced Mail System Status Code,” <http://www.ietf.org/rfc/rfc3463.txt>.
- [7] A. Pitsillidis, K. Levchenko, C. Kreibich, C. Kanich, G. Voelker, V. Paxson, N. Weaver and S. Savage, “Botnet Judo: fighting spam with itself,” In Proceedings of Network and Distributed System Security Symposium (NDSS), Feb. 2010.
- [8] The Apache SpamAssassin Project, <http://spamassassin.apache.org>.
- [9] K. Ehrlich, A. Karasaridis, D. Liu and D. Hoein, “Detection of spam hosts and spam bots using network flow traffic modeling,” In Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats (LEET), Apr. 2010.
- [10] T. Sochor, “Greylisting long term analysis of anti-SPAM effect,” Risks and Security of Internet and Systems (CRiSIS), 2009 Fourth International Conference, 19-22 Oct. 2009.
- [11] Z. Qian, Z. M. Mao, Y. Xie and F. Yu, “On Network-level Clusters for Spam Detection,” In Proceedings of Network and Distributed System Security Symposium (NDSS), Feb.-Mar. 2010.
- [12] G. Stringhini, T. Holz B. Stone-Gross C. Kruegel and G. Vigna, “BOTMAGNIFIER: Locating Spambots on the Internet,” In Proceedings of Annual Conference on USENIX, Aug. 2011.
- [13] S. Venkataraman, S. Sen, O. Spatscheck, P. Haffner and D. Song, “Exploiting network structure for proactive spam mitigation,” In Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, Aug. 2007.
- [14] S. Sinha, M. Bailey and F. Jahanian, “Shades of Grey: On the Effectiveness of Reputation-based “Blacklists”,” In Proceedings of Malware, Oct. 2008.