A Multi-Faceted Approach towards Spam-Resistible Mail ♣

Ming-Wei Wu* Yennun Huang† Shyue-Kung Lu‡ Ing-Yi Chen§ Sy-Yen Kuo*

* Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

ABSTRACT

As checking SPAM became part of our daily life, unsolicited bulk e-mails (UBE) have become unmanageable and intolerable. Bulk volume of spam e-mails delivering to mail transfer agents (MTAs) is similar to the effect of denial of services (DDoS) attacks as it dramatically reduces the dependability and efficiency of networking systems and e-mail servers. Spam mails may also be used to carry viruses and worms which could significantly affect the availability of computer systems and networks. There have been many solutions proposed to filter spam in the past. Unfortunately there is no silver bullet to deter spammers and eliminate spam mails. That is, in isolation, each of existing spam protection mechanisms has its own advantages and disadvantages. In this paper, we analyze the shortcomings of existing anti-spam solutions and propose a multi-faceted approach using the spam-resistible mail agent (SRMA), which provides the most advantages and the least disadvantages of existing anti-spam solutions. Our experiments show that the proposed SRMA is immune to existing spambots and the prototype proves to be effective, feasible and deployable.

Keywords

Spam, anti-spam, denial of service attacks, unsolicited bulk e-mail (UBE), e-mail dependability

1. INTRODUCTION

Spam refers to any message or e-mail, irrelevant of its "*junkiness*", that was sent unsolicited and in bulk. Jupiter Research estimates the average e-mail user will receive more than 3,900 spam mails per year by 2007, up from just 40 in 1999, and Ferris Research estimates spam costs U.S. companies 10 billion in 2003 and a user spends on the average 4 seconds to process a SPAM mail. As bulk volume of spam mails overtakes legitimate mails, as reported by ZDNet Australia, the effect of spam mails is similar to denial of service attacks (DOS) on computer servers as the dependability and efficiency of networking systems and e-mail servers are dramatically reduced. Spam is also used to disseminate virus and spyware which may severely affect the dependability of computer systems and networks.

There is no silver bullet, unfortunately, to deter spammers and eliminate spam as each of existing spam protection mechanisms has its own advantages and disadvantages. In this paper, we analyze the shortcomings of existing anti-spam solutions and propose a multi-faceted approach towards spam-resistible mail by coordinating layers of shields, namely 1) listing good senders, 2) labeling the message, 3) collecting known spam digests and 4) challenging probable spammers, in a spam-resistible mail agent (SRMA). We implement and evaluate a prototype of SRMA and the results show that the SRMA is indeed effective, feasible and deployable.

The remainder of this paper is organized as follows. Section 2 provides taxonomy of existing anti-spam solutions and analyzes the shortcomings of each approach. We then describe the SRMA multi-faceted approach towards spam-resistible mail in Section 3

♣ Supported by the National Science Council under the grants No. NSC 94-2213-E-002-082.

† AT&T Labs, Florham Park, NJ

‡ Department of Electronic Engineering, Fu Jen Catholic University, Taipei, Taiwan

§ Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan

Email addresses: benson@ee.ntu.edu.tw (M.-W. Wu), yen@research.att.com (Y.-N. Huang), sklu@ee.fju.edu.tw (S.-K. Lu), ichen@ntut.edu.tw (I.-Y. Chen), sykuo@cc.ee.ntu.edu.tw (S.-Y. Kuo)

and evaluate the effectiveness and performance of the prototype implementation of SRMA in Section 4.

2. ANALYSIS OF RELATED ANTI-SPAM SOLUTIONS

Existing anti-spam solutions for all sorts of mail agents including mail transfer agents (MTAs), mail delivery agents (MDAs) and mail user agents (MUAs) could be classified as 1) munging, 2) listing, 3) filtering. 4), shaping, 5) pricing, 6) challenging and 7) identity-hopping.

2.1 Munging

Munging is to deliberately alternate an e-mail address to make it unusable for e-mail harvesters, who build e-mail lists for spamming purposes. For example, benson@ieee.org could be munged as *benson at ieee dot org*.

Intuitively speaking, munging only provides a weak defense line in preventing e-mail addresses from being harvested. It could temporarily fool most of the web-based spambots, which are programs designed to collect e-mail addresses from Internet in order to build mailing lists to send spam mails. However, it is not hard for spammers to adapt all sorts of munging tricks.

2.2 Listing

The idea is simple - permitting the whitelist, blocking the blacklist, and holding (pending) the greylist.

Blacklist maintains a list of spammers (or potential spammers) and whitelist maintains a list of senders that are legitimate to send mails. Senders who are not in either the blacklist or the whitelist are put into the greylist.

Although RBLs (Real-time Blackhole Lists)[4] provide a way to block possibly spammers in real-time, they have a significant problem - an overwhelming amount of address lists in RBLs could block many legitimate users and even entire geographic regions. Whitelisting therefore should be a better alternative than blacklisting even though e-mails which are not sent by senders in a whitelist would be significantly delayed or accidentally discarded.

Greylisting would temporarily reject mails from unfamiliar senders (e.g. not in a whitelist) and require the rejected mails to be retransmitted [1]. A properly-configured MTA, as suggested in IETF RFC 2821, should retransmit mails in 30 minutes after a failure. However, spam mails sent through an open proxy or a non-properly-configured MTA will not be retransmitted. As an example, Matador from Mail-Frontier holds incoming emails in a greylist until senders respond with correct answers on certain questions (such as how many animals are there in a randomly chosen picture).

Since greylisting holds all non-whitelisted e-mails and requires retransmission, it could effectively deter spammers using a non-properly-configured MTA. Wietse Venema, author of Postfix, observed that most spam e-mails don't try to retransmit after being rejected [2]. However, spammers using properly-configured MTA would simply experience some delay.

As most spam mails are originated from spoofed senders, listing which merely verifies the legitimacy of IP and/or DNS addresses becomes ineffective. Increasing number of research projects, namely RMX/RMX++ (Reverse Mail eXchange), DMP (Designated Mailers Protocol), DomainKeys and RMX-derivatives (e.g. Sender Policy Framework and Microsoft Sender ID), take a more aggressive approach in dealing with address forgery by verifying the sender address in each e-mail header. Techniques employed in these solutions generally take advantage of the DNS mechanism such as RMX and existing TXT (Text) records, combining with a cryptographic public key mechanism that allows a recipient to verify a signed email.

2.3 Filtering

Filtering is a common anti-spam feature that can be added or installed at e-mail applications (e.g. the junk mail control of Microsoft Outlook or Mozilla Thunderbird) or at the edge MTAs (e.g. scoring of SpamAssassin – a mail filter, written in Perl, to identify spam mails using a wide range of heuristic tests on mail headers and body texts). The Filter-based approach mainly takes advantage of text categorization techniques such as

1) Naive Bayes is the most commonly used method. Plenty of works have shown that this is one of the most effective approaches and is capable of achieving high junk precisions and recalls [3]. Some studies also suggested that using multinomial model can achieve a higher accuracy than using the multivariate Bernoulli model [4].

2) Boosting Trees (a multi-classifier), which was proposed by Schapire and Singer for addressing

4 RBL includes MAPS (http://www.mail-abuse.org/), DSBL (http://dsbl.org/), AHBL, NJABL, SpamCop, Spamhaus, etc.

multi-class and multi-label classification problem by combining many base hypotheses. Later, Carreras and Marquez [5] implemented the AdaBoost algorithm for anti-spam e-mail filtering. They concluded that Boosting Trees outperforms Naive Bayes, Decision Trees and the k-NN algorithms based on two public corpuses, the PU1 corpus and the Ling-Spam corpus. However, Nicholas argued that the superiority of Boosting Tree and AdaBoost, using decision stumps, are inferior to the Naive Bayes in terms of both accuracy and speed [6].

3) Support Vector Machines was implemented by Drucker et al. [7] for spam filtering. Their study showed that both SVM filter and Boosting Trees filter achieved the lowest error rates and Boosting Tree offers a higher accuracy but a longer training time.

4) Memory-based classifier, which was suggested to combine multiple ground-level classifiers, a.k.a. stacked generalization to induce a higher-level classifier for improving overall performance in anti-spam filtering [8]. The solution is a hierarchical approach where the high-level classifier can be considered as the president of a committee with the ground-level classifiers as members.

Although filtering seems effective to most spammed users, it is far from satisfactory. There are many key limitations of existing filtering approach in fighting against spam problems:

2.3.1 Easily-sneaked

Spammers tend to alter e-mails slightly so that content filtering programs are fooled while human beings are still able to interpret. For example, a few variations for the term Viagra could as follows:

V i a g r a V*i*a*g*r*a V-i-a-g-r-a V1agra \./iagra

2.3.2 Format-dependent

Most filters can only understand text-based content while images and other rich media mails can easily bypass the filters.

2.3.3 Passive approach

Filters react only after spam mails have already spammed many users and consumed network and storage resources.

2.3.4 Predictable behavior

As filters focus on known content (signature-based), spammers can change and disguise their e-mails to get through most sophisticated junk mail filters [9].

2.3.5 False Positives

It is very undesirable to falsely tag a legitimate e-mail as a spam e-mail and delete it. However, even the most widely-used Bayesian filters [10] have only 92 to 95 percent accuracy in identifying spam mails. Because of this accuracy problem, users often hesitate to use more aggressive and more comprehensive e-mail filters. As a result, many spam mails can still get through impotent mail filters.

A significant characteristics of spam e-mails is that identical (or nearly identical when there's slight personalization for each recipient) copies of mails are delivered to a group of recipients. Systems in a network could therefore detect and block spam mails collaboratively by exchanging the sender identity of each message. Distributed Checksum Clearinghouse (DCC) and Vipul's Razor/Pyzor/Cloudmark are some of the well-known community-based peer-to-peer filtering examples.

Scalability could be a significant problem to the community-based filtering approach since the introduced delivery overhead such as exchanging message checksums requires frequent flooding of checksum information. Choosing the frequency of exchanging and updating checksums will significantly impact the tradeoff between effectiveness and performance.

2.4 Shaping

TCP damping [11] is a TCP level mechanism that slows down spammers by increasing delay and resource consumption based on spam likelihood estimation, which is the application level information from SpamAssassin.

Taming IP packet [12] to resist mail flooding attacks requires either 1) some degree of DNS hacking on an indirection layer (i.e. Internet Indirection Infrastructure) to identify hosts without using IP or 2) installing a set of fine-grained network filters on edge routers.

However, neither TCP damping nor taming IP packet is applicable to filter spam mails since e-mail forwarding is commonly deployed by most edge MTAs and the taming IP may interfere with the mail forwarding function of MTAs. As a result, taming IP packet might cause more problems for the edge MTAs rather than the spammers.

2.5 Pricing

A recent study [13] argued that senders, because they want mail receivers to read their messages, pay fees

for bulk e-mails. The study showed that different pricing models strongly influence the behavior of mail recipients to read and reply e-mails. The study also suggested both free e-mail model and flat-fee pricing model would not benefits either senders or recipients because recipients still waste valuable time to distinguish useful mails from junk mails. On the other hand, usage-based pricing (variable rate pricing) model forces the senders to target potential recipients more wisely. Therefore, mail recipients could read a higher percentage of mails when they have fewer mails to read [14]. Turner and Havey [15] also proposed a similar monetary approach that has MTAs to make payments with the Lightweight Currency Protocol (LCP) when delivering emails to other mail domains. As an example, Daum Corporation, the largest Internet portal in Korea, provides an online stamp service that charges bulk e-mailers a fee to send a bulk mail to its customers.

Although pricing with monetary cost seems to be a panacea to reduce spam communication dramatically, the success of this solution requires all e-mail service providers to deploy a standard pricing model. Otherwise, partial or regional deployment of pricing approach might shift the spam population and traffic rather than mitigate it – a problem similar to a partially-deployed QoS network among ISPs.

2.6 Challenging

Quite a few works adapt a challenge-response mechanism which requires e-mail senders to provide "proofs-of-works" (POWs) instead of monetary stamps. Senders are required to spend some CPU processing time by resolving POWs puzzles, a.k.a. cryptographic puzzles, and POWs have been applied to various applications including 1) combating junk e-mail with computational pricing functions [16] or Hashcash, 2) metering web visits, 3) providing incentives in P2P systems, 4) mitigating DDoS attacks, 5) rate limiting TCP connections, 6) protecting SSL/TLS, and many other usages [17]. As the amount of processing power available to users can vary enormously, some recently proposed puzzles rely on accessing random access memory since they have considerably more constant performance across different machines as compared to CPU speeds [18]. Some research projects use Completely Automated Public Turing test to Tell Computers and Humans Apart (CAPTCHA) - humans can pass, but most computer programs including bulk e-mailers will fail these "hard" AI tests [19]. CAPTCHAs are very effective to current program bots and are therefore being used ubiquitously in Web applications. There are also various kinds of similar tests, namely Gimpy, Bongo, Pix, Sounds and Byan to deal with spam mails.

However, Laurie and Clayton showed that POW puzzles would not work [20] sufficiently - as they were not properly analyzed to consider how much money the spammers may spend and how much resources they may acquire in order to solve cryptographic puzzles. Furthermore, the POW puzzles may not be effective as shown in Mori and Malik work where they developed a program that can pass the visual CAPTCHA tests with over 80% accuracy [21].

2.7 Identity-hopping (aliasing)

Vendor like Spamgourmet.com offers self-destructing disposable email addresses (DEA) by encapsulating policy in e-mail addresses [22] that look like

someword.x.user@spamgourmet.com

,where someword is a word you have never used before (as a way to unique identify the sender or mailing list provider), x is the maximum number of email messages you want to receive at this address and user is your username at spamgourmet.com.

For example, if your user name is "bensonwu", and BigCorp wants you to give them your email address (on the web, on the phone, at a store, etc.), instead of giving them your real address, you can give them:

frombigcorp.2.bensonwu@spamgourmet.com

This disposable email address will be created the first time when BigCorp uses it (transparent to spamgourmet's subscriber), and you'll receive at most 2 mails on this account which will then be forwarded to your real e-mail address.

Instead of generating human-readable DEAs, Jetable.com conceals a real e-mail address with a hashed one. For example, dhgoyd90ucxjiag@jetable.com is referred to benson@lion.ee.ntu.edu.tw.

A few non-commercial solutions of DEA concept, including our proposed anti-spam solution discussed later in this paper, are TMDA (Tagged Message Delivery Agent), ASK (Active Spam Killer) and SFM (Spam-Free Mail). A general introduction to the concept of DEAs and DEA services can be found at http://email.about.com/cs/disposableaddr/.

Although identity-hopping allows anyone to throw away a spammed disposable e-mail address easily, existing practices have some shortcomings. First, most users won't choose a "hard" prefix and subsequent prefix tends to follow a similar pattern (e.g. frombigcorp and fromsmallcorp). Simply concatenating the prefix with the username (e.g. someword.x.user@spamgoumet.com) to form an alias might be vulnerable to dictionary attacks and thus DEAs are likely to be spammed soon - a short expiration date. On the other hand, a hard DEA, which might acquire through fully hashing (e.g. dhgoyd90ucxjiag@jetable.com) or partial hashing (e.g. fyqmecyz.benson@sfm.cs.ualberta.ca), would be much robust to dictionary attacks but it is hard for users to memorize it too. Given the disposable nature, DEA is only applicable in disposable e-mail communication situations, DEA is not likely to be used as a long-term e-mail address. As a consequence, personal e-mail addresses, which need to be permanent, are still at risk of being spammed.

3. A Multi-faceted Approach towards Spam-resistible Mail

As discussed in the previous section, there is no silver bullet for spam resolution. Therefore, we propose and implement a multi-faceted approach towards spam-resistible mail by coordinating layers of shields, namely 1) *listing* good senders, 2) *labeling* the message, 3) *collecting* known spam digests and 4) *challenging* probable spammer, using a spam-resistible mail agent (SRMA).

3.1 Architecture of SRMA

SRMA inspects mails after they are stored in an incoming mail queue. MTA receives unfiltered mails using a standard protocol such as SMTP (e.g. with Postfix smtpd server) and delivers unfiltered mails to SRMA (e.g. with the Postfix pipe delivery agent). SRMA injects filtered mails back into MTA (e.g. with the Postfix sendmail command), so that MTA can deliver them to their final destinations. SRMA assists MTA in forwarding legitimate e-mails to user mail inboxes immediately and delaying spam-likely e-mails for further processing. Most existing MTAs today provide some primitive anti-spam mechanisms which reject non-existent recipients, invalid hostnames, non-fqdn (fully-qualified domain) forms, and unknown hostnames in which they provide no DNS A (address) record or MX (Mail eXchange) record. Therefore, existing MTAs are used to be the first line of defense to *remove* certain spam mails. Meanwhile, since mails coming from a mailing list are very similar to spam mails in a way that both may be generated by programs or scripts, we also deploy DEA through the use of address aliasing so that local users could subscribe remote mailing lists without disclosing their real/master e-mail addresses. Mails delivered to an aliased address are identified by MTA and redirected to DEA. Although it is possible for SRMA to apply some temporal heuristic methods in differentiating junk mails from legitimate mails, spammers could quickly learn and disguise their addresses as coming from some legitimate mail servers. Therefore, DEA support is the second line of defense. After DEA processing, MTA delivers all remaining mails to SRMA where spam likelihood is examined and the mails determined by SRMA as “no-spam” are re-inject back into the MTA for immediate delivery to users.

3.2 Inbound e-mail

All inbound e-mails may arrive with a real address or with an alias. SRMA focuses on real mail addresses which are intended for long-term communication while DEA deals with alias mail addresses which are intended for disposable and temporary communication. Since a real address is likely to be stable and permanent, the presence of a white-listed sender address and a valid passphrase are crucial for SRMA in assuring the legitimacy of a mail. A passphrase refers to a special hashed (e.g. MD5 or SHA1) string labeled by SRMA to mails that had responded to a previously issued challenge correctly, indicating that the mail is probably not a spam. SRMA generates a valid passphrase by associating it with any e-mail address found in "To", "From" and "CC" header. Figure 1 is the lifecycle of a mail passing through SRMA. For example, an inbound mail with a valid passphrase could be delivered immediately while a non-whitelisted mail with no valid passphrase is queued for further processing. Note that to avoid being spoofed, loop delivery (sender address is identical to the recipient) from an un-trusted MTA is discarded immediately.

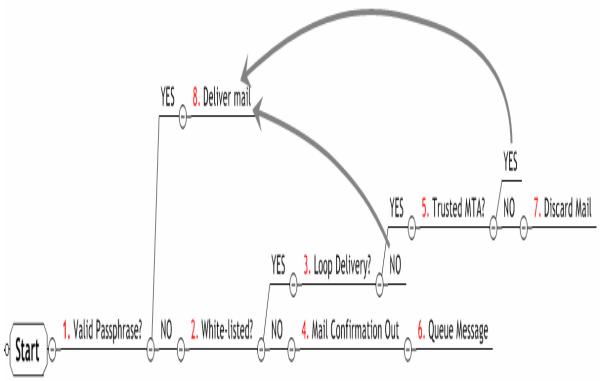

Figure 1. Flowchart of message processing in SRMA

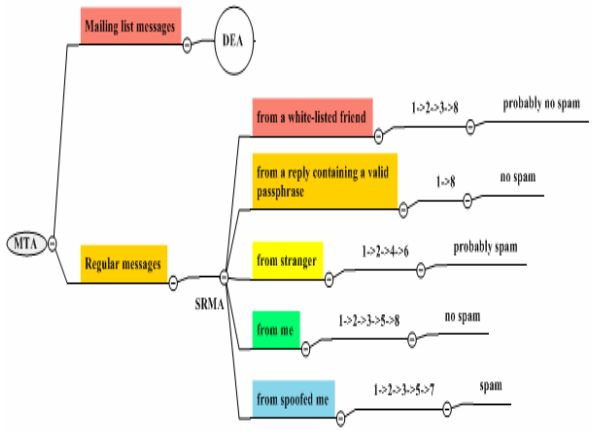

Figure 2. Different sources yield different spam likelihood

Figure 2 illustrates how regular mails redirected by MTA to SRMA are processed and how the spam likelihood of a message is logically classified (after running through its life cycle in SRMA) into 1) no-spam, 2) probably-no-spam, 3) probably-spam, and 4) spam.

3.2.1 From a white-listed friend

In this case, the mails are sent by some legitimate senders (white-listed) but may not contain a valid passphrase. SRMA assumes all mails coming from a white-listed friend as probably-no-spam and should be delivered immediately.

3.2.2 From a reply containing a valid passphrase

As most MUAs (Mail User Agents) tend to quote the original mail when replying an e-mail, SRMA could potentially minimize the number of confirmations sent by looking for the presence of a valid passphrase in the message body.

3.2.3 From strangers

Since strangers are not white-listed and cannot present a valid passphrase, their mails are queued by SRMA until a confirmation reply is received.

3.2.4 From "me" (loop delivery)

When a sender address and a recipient address are identical, proper authentication should be taken to avoid being spoofed. SRMA only allows loop delivery of mails from trusted MTAs and discards all others.

3.2.5 From spoofed "me"

As mentioned previously, spammer could have the sender address forged to be the same as the recipient one. Such mails are apparently white-listed but contain no valid passphrase. SRMA takes a step further to make sure the mails are delivered from trusted MTAs or else the mails are discarded.

3.3 Outbound e-mail

All outbound e-mails are labeled with a valid passphrase allowing that future replies could be treated specially by SRMA. Whenever SRMA finds a valid passphrase of an e-mail in its association table, the e-mail is assumed to be no-spam and is forwarded immediately. This minimizes the number of confirmation requests sent to non-white-listed senders.

4. PROTOTYPE STATUS AND FUTURE WORK

To evaluate the effectiveness of the proposed SRMA and its performance impact on mail delivery, we implemented a prototype of the SRMA system. The prototype SRMA is implemented as an external program (e.g. an external filter) to Postfix, a standard well-known MTA. A stable version of SRMA will be made available for download soon. Interested readers can contact the primary author for a licensing agreement.

4.1 Munging, CAPTCHA and Spampots

Internet crawlers could be any spybot software that is capable of performing spyware activity. There are three kinds of crawlers, namely web crawler, e-mail harvester, and host-based spyware. The basic concept behind trapping spybots with a honeypot-similar spampot is to better understand the frequency of spybots activity and its behavioral patterns. It is also interesting to investigate the effectiveness of munging

and CAPTCHA in identifying and rejecting e-mail crawlers. A total of 30 email boxes (20 are spampots, 5 are munged inboxes and 5 are CAPTCHAed inboxes) are generated in the SRMA platform. From Nov. 18, 2004 to Jan. 17, 2005 (a test duration of 60 days), the testbed successfully counted 44683 visitations and identified 3651 unique e-mail crawlers. Figure 3 shows the number of crawler visits per day during the test period. We found that our e-mail pool generally attracted more than 700 e-mail crawlers a day. One interesting investigation is that we found these e-mail crawlers would return on a 5 day period, which is much more frequent than those of well-behaved web crawlers. For example, from our experiments lasting from February to September, 2004, we found that on the average a unique web crawler revisited our testbed every 19 days. Figure 4 shows the effectiveness of munging and CATPCHA. We set up a few "honeypot" e-mail accounts, which are not protected by munging or CAPTCHA, and they began to receive spam e-mails after 110 hours (about 5 days) and gathered an average of 151 spam e-mails for the past six month (Sep. 10, 2004 to May 10, 2005). This number is lowered than our expectation. We suspect that some e-mail crawlers might perform pruning mechanisms such as removing e-mail addresses containing strings like "spampot" or "removed". In addition, we found 40% of incoming e-mails received by different honeypot inboxes are identical. This suggests that spam e-mails did possess some property of time locality. Our investigation shows that inboxes deploying munging or CAPTCHA techniques are immune to all robots since we received no spam e-mails during the experiment and even until today.

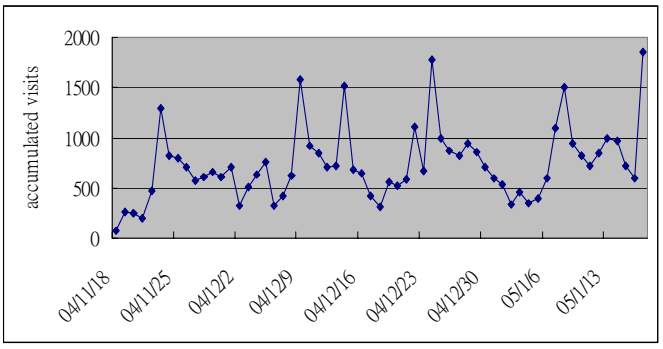

Figure 3. Daily visits by e-mail crawlers

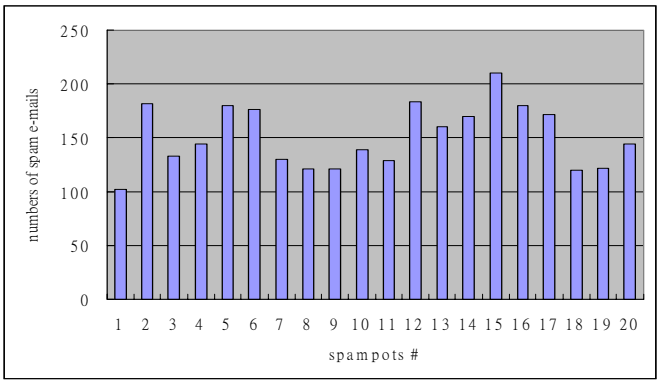

Figure 4. Effectiveness of munging and CAPTCHA

4.2 Effectiveness of SRMA

The multi-faceted approach towards spam-resistible mail makes SRMA to immune to several shortcomings of the existing anti-spam solutions mentioned previously. First, spammers could not adapt SRMA by alternating the content of a mail body in any way. Second, SRMA is deployed transparently by adding an additional external filter to existing MTA rather than introducing new protocols and frameworks. Third, SRMA guards a permanent e-mail address by 1) listing good senders in a white-list, 2) labeling a legitimate e-mail with a valid passphrase, 3) collecting spam digest from honeypot mail accounts and 4) challenging a probably spam mail with a confirmation request. Although the effectiveness of SRMA heavily depends on the intelligence of a spambot, the complexity of a confirmation request (or more generally a Turing test) could be adaptive. Our primitive evaluation of SRMA employs the simplest form of challenges – a mail body containing a valid passphrase for a sender to quote in his/her reply. If the reply is valid, the queued e-mail (by SRMA) is forwarded immediately. The spam set with a total of 2000 e-mails was collected from our specially created honeypot inboxes. SRMA sent 2000 confirmation requests with a specially crafted mail header to 1) identify invalid address bounces and 2) receive returning replies.

Table 1. The Effectiveness of SRMA

	Number of messages
Malformed e-mails*	199
Invalid address**	1244
No response	557
Returned confirmation	0

* E-mails that failed SRMA's trivial mail header inspections
** Bounces due to unknown recipient and/or invalid domain

Surprisingly, Table 1 shows more than 70% of spam e-mails are either malformed, which was identified by a trivial header inspection, or contain a forged sender address that was discovered by SRMA challenges. Interestingly, the concept of such challenge-response confirmation is quite similar to how a 3-way TCP handshake that could prevents IP spoofing. The experiment last for 72 hours and received no returned confirmation. We intend to do more elaborate evaluation of SRMA by collecting more spam sets and by releasing the SRMA code for public trials.

4.3 Performance

As mails arrive at a MTA, run through SRMA, and return back to the MTA via SMTP, our initial performance study showed 50% throughput degradation of a mail server (Postfix), compared with the test cases where no SRMA was used. More bottleneck analysis and fine-tuning of mail queues (maildrop/hold/incoming/active/deferred) will be examined to minimize mail server performance degradation.

5. Summary

Similar to denial of service attacks, bulk volume of spam e-mails delivering to mail transfer agents (MTAs) reduces the dependability and efficiency of computer networks systems and e-mail servers. Spam mails may also be used to carry viruses and worms which could significantly affect the availability of computer systems and networks. As there is no silver bullet to defend spam mails, this paper describes a multi-faceted approach towards spam-resistible mail by coordinating layers of shields, namely 1) listing good senders, 2) labeling the message, 3) collecting known spam digests and 4) challenging probable spammer, in a spam-resistible mail agent (SRMA). Our evaluation on the prototype shows both munging and CAPCHA techniques are immune to existing spambots and the experiment results proves that SRMA is effective, feasible and deployable.

6. REFERENCES

[1] Evan Harris, The Next Step in the Spam Control War: Greylisting, Aug. 21, 2003, available at http://projects.puremagic.com/greylisting/whitepaper.html

[2] Backup MX usefulness, http://archives.neohapsis.com/archives/postfix/2002-03/0859.html

[3] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., and Spyropoulos, C. D., An evaluation of Naive Bayesian anti-spam filtering. In Proceedings of the workshop on Machine Learning in the New Information Age, pp. 9-17, 2000.

[4] Schneider, K.-M., A comparison of event models for Naive Bayes anti-spam e-mail filtering. In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics. Budapest, Hungary, pp. 307–314, 2003.

[5] Carreras, X. and Marquez, L., Boosting trees for anti-spam e-mail filtering. In Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing, 2001.

[6] Nicholas, T., Using AdaBoost and Decision Stumps to Identify Spam E-mail, June 4, 2003, available at http://nlp.stanford.edu/courses/cs224n/2003/fp/tyronen/report.pdf

[7] Drucker, H., Wu, D., Vapnik, V. N., Support Vector Machines for Spam Categorization, IEEE Transactions on Neural Networks, Vol. 20, No. 5, Sep. 1999.

[8] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D., Stamatopoulos, P., Stacking classifiers for anti-spam filtering of E-mail. In: Proceedings of EMNLP-01, 6th Conference on Empirical Methods in Natural Language Processing, Pittsburgh, 2001

[9] Krim, J., "A spammer speaks out: In Hill testimony, bulk e-mailer says Internet providers use same tactics," Washington Post, p. A01., May 22, 2003.

[10] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E., A Bayesian Approach to Filtering Junk EMail. In Learning for Text Categorization – Papers from the AAAI Workshop, pp. 55–62,

Madison Wisconsin. AAAI Technical Report WS-98-05, 1998.

[11] Li, K., Pu, C., and Ahamad, M., Resisting spam delivery by TCP damping. In the first Conference on Email and Anti-Spam (CEAS 2004), Mountain View, CA, Jul. 2004.

[12] Lakshminarayanan, K., Adkins, D., Perrig, A., and Stoica, I., Taming IP packet flooding attacks. 2nd Workshop on Hot Topics in Networks (HotNets-II), Nov. 2003.

[13] Loder, T., Van Alstyne, M., & Walsh, R., An economic solution to the spam problem. Downloaded from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=488444, 2003.

[14] Kraut, R. E., Sunder, S., Telang, R. and Morris, J. H., Pricing Electronic Mail to Solve the Problem of Spam, Jul. 2003, available at http://ssrn.com/abstract=417621

[15] Turner, D. and Havey, D., Controlling spam through lightweight currency. In Proceedings of the Hawaii International Conference on Computer Sciences, Honolulu, HI., 2004.

[16] Dwork, C. and Naor, M., Pricing via processing or combatting junk mail. In Lecture Notes in Computer Science 740 (Proceedings of CRYPTO'92), pp. 139-147., 1993.

[17] Jacobsson, M. and Juels, A., Proofs of Work and Bread Pudding Protocols. In Proceedings of the IFIP TC6 and TC11 JointWorking Conference on Communications and Multimedia Security (CMS '99), Kluwer, 1999.

[18] Dwork, C., Goldberg, A., and Naor, M., On memory-bound functions for fighting spam. In Lecture Notes in Computer Science 2729 (Proceedings of CRYPTO'03), pp. 426-444, 2003.

[19] Ahn, L. von, Blum, M., Hopper, N.J., and Langford, J., CAPTCHA: Telling humans and computers apart. In Advances in Cryptology, Eurocrypt '03, volume 2656 of Lecture Notes in Computer Science, pp. 294-311, 2003.

[20] Laurie, B. and Clayton, R., "Proof-of-Work" Proves Not to Work. The Third Annual Workshop on Economics and Information Security (WEIS04), May 2004.

[21] Mori, G. and Malik, J., Recognizing objects in adversarial clutter - Breaking a visual CAPTCHA. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Jun. 2003

[22] Ioannidis, J., Fighting spam by encapsulating policy in E-Mail addresses. In Proceedings of NDSS'03. San Diego, CA, 2003.