Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, December 8-10 2004

Evaluating The Use of Spam-triggered TCP/IP Rate Control To Protect SMTP Servers

Minh Tran, Grenville Armitage
Centre for Advanced Internet Architectures
Swinburne University of Technology
Melbourne, Australia
{mtran, garmitage}@swin.edu.au

***Abstract*- This paper examines an approach to spam mitigation that rate limits incoming TCP/IP connections to an SMTP server based on the real-time detection of spam within the SMTP message exchange. Our approach is motivated by a desire to cause increased resource consumption at the spammer end of each SMTP connection, and to avoid the negative impact of false-positives by eventually allowing all emails through. We call the tool MT Proxy. MT Proxy's spam analysis and traffic differentiation characteristic are analyzed to evaluate the efficacy of this architectural approach to fighting spam.**

Keywords-Spam, Email, SMTP, false positives, false negatives, white lists, black lists, challenge-response, Bayesian, ipfw, dummynet.

I. Introduction

Unsolicited bulk email transmission has becoming a major headache across the Internet for service providers and users alike. Usually of a commercial nature, and often pushing distasteful services or carrying viruses, this type of email is colloquially referred to as "spam" [23]. Almost everyone with an email today has experienced some level of spam – from a few messages a week to hundreds or more a day. The impact on everyone's mailboxes has become so great that many countries are establishing legislative solutions to combat spam, and a wide variety of companies are pursuing technological solutions to reduce or eliminate spam in people's mailboxes. Research by Harris Poll in May 2003 [1] has shown that 93% percent of respondents are dissatisfied with spam.

Although humans often argue that they "know spam when they see it", automating the fight against spam is a non-trivial task. The first goal of automated solutions is to ensure the end user is not required to manually sift through their mailbox deleting spam. A closely related goal is to ensure the spam never gets into the user's mailbox in the first place (especially if the mailbox is on a local machine, and transporting spam from the user's ISP server to a local mailbox incurs excess IP network connection charges). An automated solution should identify and delete (or specially mark) spam emails before they trouble the end user.

Most existing anti-spam methods focus on classifying emails, then deleting the spam or throwing spam emails into 'junk' folders [3]. Unfortunately, automating the detection of spam introduces two sources of misclassification error - false-positives (where a non-spam email is classified as spam) and false-negatives (where spam slips through, incorrectly identified as non-spam). Many anti-spam solutions have been proposed, with varying false-positive and false-negative rates. None of them approach perfection.

False-negatives are a nuisance – the influx of spam into our mailbox is reduced, but never to zero. On the other hand, false-positives are far more problematic – legitimate emails (whether from friends, business partners, etc) are potentially discarded without us ever knowing. False-positives are highly undesirable when the anti-spam system's response is to discard the falsely-identified spam directly, or dump the email into a 'junk' folder then may never be checked.

Inspired by Marty Lamb's exhortation to cause spammers pain [4], we have begun to explore a variation on server-side spam defense. We have two goals – reduce the negative consequences of false-positives, and increase the effort (in time and resources) to which a spammer must go in order to send their spam emails.

Our approach operates at both the application and IP levels. We place an application-layer proxy in front of an ISP's mail server, and force all inbound emails through the proxy. Each email is inspected in real-time as it passes through, and if the email appears to be spam the underlying TCP/IP connection for that specific email transfer is penalized. The net effect is that email believed to be spam (whether correctly, or a false-positive) still gets through, but at a far lower (configurable) bandwidth.

Although we do not stop spam entirely, our approach ties up a spammer's resources (the TCP/IP connection for each piece of spam takes a long time to complete) and false-positives are far less problematic because incorrectly classified legitimate emails will eventually get through. In other words, people sending email in bulk will suffer resource starvation while people who send a few innocent emails every few minutes or hours will hardly notice degradation even if their emails are being incorrectly flagged as spam. We call our prototype implementation MT Proxy.

This paper continues with some background on email transfer protocols and existing anti-spam techniques in section II. The design of MT Proxy is covered in section III, while section IV discusses our experimental validation of MT Proxy. The limitations and future research directions for MT Proxy are provided in section

ISBN 0-646-44190-6

Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, December 8-10 2004

V, which is followed by our conclusion.

II. Background

Vernon Schryver notes (with some wit) that many 'foolproof' anti-spam techniques are developed by people lacking a broad understanding of spam's technical scope [28]. Thus it is important to briefly review the technical context within which our experimental architecture sits.

A. *The Email Transfering System*

An email message relies on the Simple Mail Transfer Protocol (SMTP, defined in RFC 821 [20]) for transferring from the sender's mail client (user agent – UA) to his/her mail server (mail transfer agent – MTA). This MTA in turn uses SMTP to transfer the email to other intermediate mail servers (relaying MTAs) until the email reaches the recipient's mail server (recipient's MTA). Each MTA needs to contact its Domain Name Server (DNS) for the IP address of the next MTA before delivering the email. The end-user's mail user agent (MUA) will normally use POP3 [26] or IMAP4 [27] protocols to retrieve their emails from an ISP's mail server (recipient's MTA).

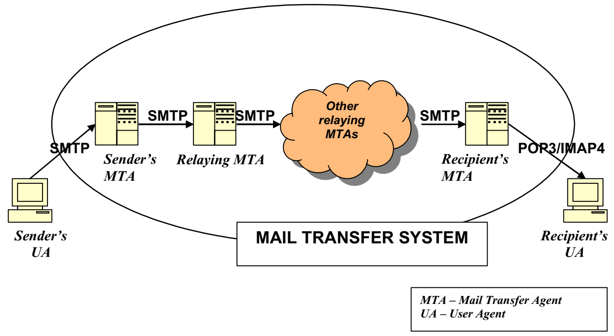

Figure 1 A basic model of an Email Transfering System

MT Proxy works by acting as a proxy for the recipient's MTA – incoming SMTP connections (over TCP/IP) terminate on MT Proxy rather than the recipient's MTA directly. MT Proxy establishes a new SMTP connection to the recipient's MTA for each incoming SMTP from an intermediate MTA. However, as each email is transferred through MT Proxy it is evaluated for evidence of being spam. If an SMTP connection appears to be carrying spam, MT Proxy slows down TCP/IP connection over which that particular SMTP connection is established.

B. *Spam and challenges for anti-spam solutions*

Dictionary.com defines spam as "Unsolicited e-mail, often of a commercial nature, sent indiscriminately to multiple mailing lists, individuals, or newsgroups; junk e-mail" [2].

Spam floods the Internet users' mail box with a bulk of junk messages, wastes their time, money and computer resouces. According to CSO Media, the loss caused by spam is about 20.5 billion US dollars for businesses worldwide in 2003. This number is predicted to rise up to 198 billion dollars in another 4 years [5].

Spam is a serious issue that needs the interception of both law and technology. The Federal CAN-SPAM act of 2003 of the United States [6] has not cut down the amount of spam sent and is of "little impact so far" according to InfoWorld's article [7][8]. As a result, technology is still a key solution in the anti-spam battle.

There is an unavoidable trade-off between false postive and false negative in most anti-spam solutions. Having a more aggressive anti-spam method is equivalent to sacrificing more legitimate emails (false positive). However, reducing this aggressiveness level will allow more spam get into recipients (false negative).

In order to protect the customers from an onslaught of spam, some ISPs have implemented more aggressive spam identification methods [19]. Consequently, many legitimate emails, including important business ones cannot reach their recipients. Research done by Ferris Inc. has shown a loss of exceeding $50 per person per year and $3.5 billions per a U.S. business in 2003 as a negative effect of false positive [9]. This is a real challenge for anti-spam software developers to eliminate possible false positive mistakes whilst still maintaining a small false negative number.

Another common shortcoming in most anti-spam programs is that they normally allow spam to get into recipient's mail server before any classification process happens. This behavior leaves no painful impact on spammers. Marty Lamb says we want to cause spammers pain [4]. We want an anti-spam solution, which is capable of slowing down real-time Internet traffic associated with spammer activities.

C. *Traditional anti-spam methods*

They are generally classified into two main groups – white list/black list, and rule-based filtering.

1. White lists and black lists

White and black lists focus on addresses associated with the email – source IP address, sender's email address or relaying mail servers' address that are involved in the reverse-path of the email message.

When using a white list, any email that arrives from an address present in the white list is considered to be non-spam. All other emails are implicitly considered to be spam. Black lists operate in the reverse fashion. Users receive emails from all sources, unless the source is in the black list. A recipient MTA or MUA can either use its local black list file or query one of a number of Internet black list databases in real-time. Internet black list sites use the DNS protocol to accept queries and provide replies. Typically an anti-spam solution will construct a special DNS query including the address of the incoming SMTP connection, and kill the connection if the black list server returns a special IP address 'code' (for example SpamHaus [10], SORBS [11], ORDB [12] return 127.0.0.2 if the senders are in their blacklist).

Many anti-spam programs are now configured to use both white and black lists as well as an automatic

ISBN 0-646-44190-6

Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, December 8-10 2004

mechanism called challenge-response for updating their local white lists. The email is sent or blocked depending on whether the sender is in the white list or black list. If the sender address is not in either white or black list, the sender is challenged with a reply message. Unless the sender responds to this challenge, the email is not delivered to the recipient. Mail Gate of Corvigo, ASG of Mail Frontier, Perimeter of Postini, Email Thread Management Service of MX Logic, Email Protection Service of Singlefin are commercial anti-spam products that implement this technique [13]. The advantage of this scheme is that it provides an automatic system, which could identify and update white/black list sources. However, if spammers know the challenge-response rule and reply to all of the challenge messages, the scheme can be dangerously vulnerable to spam attacks.

A weakness with list-based filtering is that all addresses carried inside an email can be faked. Only the IP address of the MTA attempting to send you email can be considered accurate. This means spam can get through a white list filter simply by forging the “From” address to be someone in the white list.

2. Rule-based filter (email analysis)

A rule-based filter inspects the actual contents of emails, rather than concerning itself with the address of an email’s source. The entire content or a portion of the email is scanned through the filter. Based on some algorithm, the filter determines the spam level of the email.

An important spam analysis algorithm is known as the Bayesian technique. Bayesian technique works on the assumption that most spam events are dependent [14]. If a word appears in many spam messages, the likelihood that this word indicates spam is high. As a result, it is assigned with a high spam probability number.

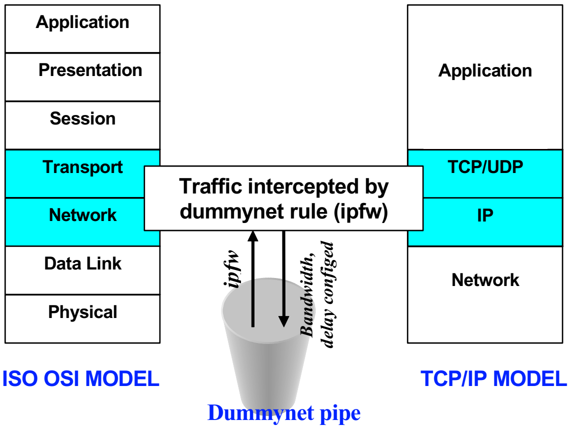

Figure 2 Dummynet in relation to the protocol stacks

Paul Graham’s approach is famous for spam filtering using the Bayesian technique [15]. Paul uses two large hash tables (corpuses) for spam (bad) and non-spam (good) email database. Another hash table is built for the spam probability of each word. When the email is scanned through the filter, each word is tokenized and assigned a spam probability. A final spam value of the email is then computed. If the value is greater than some certain threshold, the email is classified as spam.

Gary Robinson developed a more symmetric approach in spam and non-spam characterization [16]. He suggests an improvement of the bias in Paul’s method by calculating both email’s spam (P) and non-spam (Q) probability. The final spam value (S) is determined from P and Q.

D. *New anti-spam approaches*

New anti-spam techniques have been proposed to provide ISPs with greater capabilities of fighting against spam.

Sender Policy Framework (SPF) is one leading domain authentication technique (merged with Caller ID of Microsoft in May 2004 [25]) to help ISPs identifying spam forgery. SPF uses DNS to maintain records of registered domain names and their associated mail servers. When an email is claimed to come from a particular domain, SPF-enabled receiving server checks if the sending server actually belongs to that domain. Thus, domain identity theft can be discovered. SPF record is growing with over 7,000 registered domains including AOL and AltaVista [24]. SPF solution has showed great help in legal/administrative attack on spammers and is under consideration of being an IETF standard [22].

Whilst SPF provides a mechanism of identifying spam, which can indirectly slow down the amount of spam, some other solutions directly fight spam by reducing the amount of spam messages sent in real-time.

Anti-spam router (ASR) of TurnTide [17] is a typical example. The router protects the network by first categorising the spam possibility of the email traffic, and then allocating different Quality of Service (QoS) for different incoming email traffic. If the sources are spam, they are given a poor level of QoS. Legitimate emails, on the other hand, are assigned good QoS. As a result, TCP traffic are shaped at different speeds. Spam traffic are slowed down relative to non-spam traffic.

Microsoft also reveals a new approach “stamp of approval” to “make spammer pay”. Delay is added to SMTP traffic through cryptographic puzzles. All unknown emails are required to solve the puzzle before reaching the recipient. Microsoft Research Group said that this scheme allows spammers to send at most eight thousands emails a day, compared to millions of emails normally sent everyday. Spammers would need to invest heavily into computers and the cost is now pushed up back to them [18].

MT Proxy is our similar approach to eliminate the negative consequences of false positives and shift back the cost to spammers in the spam equation.

SPF solution reduces the problem of open relay MTAs that are vulnerably used by spammers. This method indirectly reduces spam whilst other solutions (TurnTide, Microsoft’s cryptographic puzzle and MT Proxy) directly fight spam by consuming spammers’

ISBN 0-646-44190-6

Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, December 8-10 2004

time and resources. MT Proxy can be extended to use SPF for its spam forgery identification.

Although TurnTide seems to be the closest method to MT Proxy, there are not published papers about the general design and implementation of TurnTide. Consequently, we cannot make a direct comparison between MT Proxy and TurnTide.

Microsoft's "stamp of approval" method adds an equal delay (through the time to solve the cryptographic puzzle) to all email traffic. MT Proxy analyses the email first before applying bandwidth/delay to only spam traffic. As a result, our approach also makes spammers pay but does not penalize legitimate emails.

MT Proxy design and functionality will be described in the following section.

III. MT Proxy design and functionality

A. *MT Proxy design:*

We implemented MT Proxy on a FreeBSD machine to act as a SMTP proxy server which can intercept email traffic before relaying the traffic to the actual mail server. MT Proxy is written in C/C++ languages by extending an open-source Unix-based SMTP proxy server [21]. The mail server hostname or IP address can be specified when MT Proxy is run.

MT Proxy uses both black listing and content filtering method to analyse spam probability of an incoming email. The black listing method only looks at the IP address of the directly connected MTA. A combination of the two methods reduces false negative. Administrator can flexibly set the weight for each method using a configuration file.

Traffic is shaped at TCP/IP level (Figure 2) using FreeBSD's kernel level functionality known as ipfw (a firewall) and dummynet (a packet processing delay line). Ipfw intercepts traffic based on information such as IP source and destination addresses and TCP or UDP port numbers. Ipfw can be configured to select certain TCP flows and apply unique dummynet rules to each TCP flow. Dummynet rules specify additional constraints (such as bandwidth limits or additional transmission delay) that should be applied to packets to which the rules are applied.

B. *MT Proxy functionality:*

Emails from UAs or MTAs are directed to MT Proxy. Before reading an email message, MT Proxy first checks to see whether the client's IP address is on either a local or Internet black list. If it finds the address in any black list, it triggers ipfw/dummynet to decrease bandwidth and add extra latency for the offending TCP connection. If the client is not found in black list, there is no ipfw/dummynet rule set to the client's connection yet.

As the email begins arriving over the new TCP/IP connection, MT Proxy performs an analysis process of the input. It computes a spam value statistic, which is then used to compute a bandwidth reduction and latency increase through ipfw/dummynet.

The TCP/IP traffic control is done in real-time as the email content is being transferred through MT Proxy to the target MTA. This makes sure that as soon as the software discovers any spam in client's input, the client will experience a significant disadvantage of a slower TCP connection.

MT Proxy sequentially analyses the email content by taking 5 lines (set by default) of client's input for each analysis. The dummynet rule is only updated if the new spam analysis value is greater than the previous ones. This "worst-spam memory" reduces the load on the MT Proxy server since it does not have to update dummynet after every 5 lines of email.

Once the email is transferred and the TCP connection torn down, MT Proxy simply deletes the dummynet rule and clears the pipe configuration for that TCP/IP connection.

IV. MT Proxy evaluation

A.*Spam versus non-spam emails*

Figure 3 shows our testbed for evaluating the efficacy of MT Proxy. The basic goal is to confirm this technique is architecturally viable and that it has a useful impact on the ratio of spam to non-spam that infiltrates the mail spool area of a target (recipient) MTA.

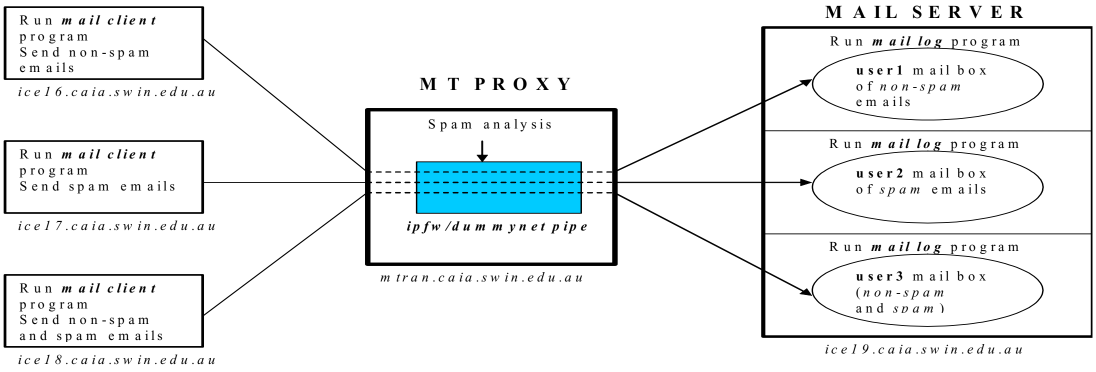

Figure 3 Performance Evaluation Test Model

ISBN 0-646-44190-6

Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, December 8-10 2004

MT Proxy is installed at mtran.caia.swin.edu.au (running FreeBSD4.9) and is configured to forward email traffic to our 'recipient' MTA (a real SMTP server) at host ice19.caia.swin.edu.au. Three email accounts - user1, user2, user3 - are set up at the recipient's MTA (ice19.caia.swin.edu.au).

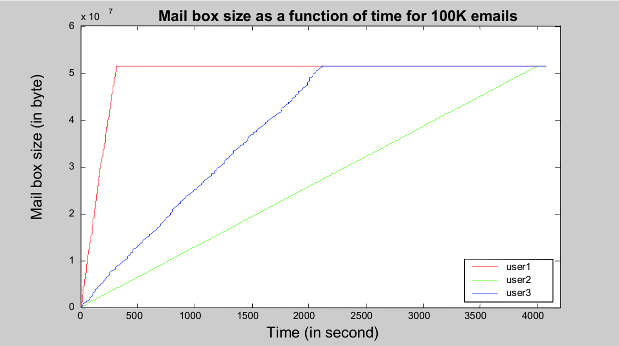

Figure 4 Mail box size as a function of time for 100K emails

Three mail clients (ice16, ice17 and ice18) are setup to originate streams of non-spam, spam and 50:50 spam:non-spam emails to user1, user2, user3 respectively. (We use an open source program, smtpclient, to send custom email payloads.) While the sources are sending their emails we log the size of user1, user2, user3 mail boxes simutaneously every second.

Each mail client sends 500 emails to their corresponding recipient (user1, 2, or 3). Each client sends email in the pattern that right after one email is delivered to the mail server, the next one is sent to MT Proxy. Thus each client sends emails as fast as possible, but does not overlap itself.

Figure 4 plots email account size versus time for three users when 100Kbyte emails are sent.

It has been observed from the graph of Figure 4 that although the emails are sent at the same time from three different clients, non-spam emails reach the mail server at the faster rate compared to spam ones. User1, who receive non-spam emails, has its mail box size grow at a fastest rate while it is lowest for user2, whose all emails come from non-spam sources. The graph of user3, who receives both spam and non-spam emails, as expected, lines in between user1 and user2's graph.

MT Proxy has proved to be capable of slowing down spam emails and giving non-spam email higher priopritY to reach its recipient. In a long run, this can save a significant space at the mail server which would otherwise be wasted by spam.

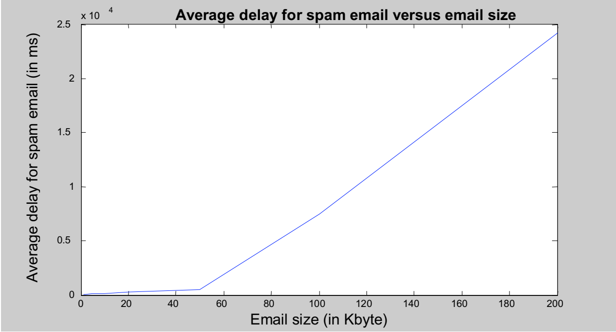

Figure 5 Time delay as a function of email size

B. *Email size does matter:*

Our next test is run with different email sizes of 1Kbyte, 2Kbytes, 5Kbytes, 10Kbytse, 20Kbytes, 50Kbytes, 100Kbytes, and 200Kbytes.

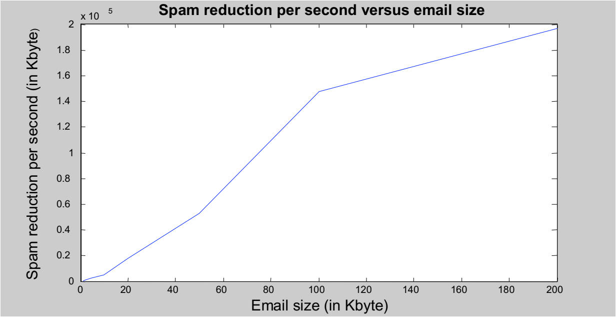

Figure 6 Spam email box reduction as a function of email size

Table 1 below shows average delay of spam email compared to non-spam emails for different emails size. We can observe that when email size increases, the relative time delay between spam and non-spam also increases. Figure 5 plots the relationship between average time delay and email size.

When the email size increases to a threshold of

Email size (in Kbyte)	Distance AB (in second)	**Average delay for 1 spam email (in ms)**	Number of **non-spam** emails received per second	Number of **spam** emails received per second	**Spam mail box size reduction per second**
1	0	**1**	6.59	6.45	**143**
2	27	**54**	4.84	4.32	**1065**
5	54	**108**	2.60	2.06	**2813**
10	60	**120**	2.50	2.04	**4710**
20	120	**240**	2.40	1.54	**17613**
50	225	**450**	2.11	1.08	**52736**
100	3750	**7500**	1.57	0.13	**147456**
200	12100	**24200**	1.00	0.04	**196608**

Table 1 Spam reduction for different email sizes (100% spam case)

ISBN 0-646-44190-6

Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, December 8-10 2004

50KBytes, the slope of the graph sharply increases. After this point, spam emails are received a significant time after non-spam emails. MT Proxy is seen to effectively apply a strict policy of time delay for spam emails at the mail server after the theshold.

We also examine the effect of MT Proxy implementation by inspecting mail box size reduction (due to spam) per second in Figure 6. The graph shows a quite linear relationship between the amount of spam reduction at the mail server in respect to the email size. The slope is slightly reduced after 100Kbytes email size. Thus, MT Proxy works best when email size is around 100Kbytes.

C.*Spam percent structure of the email does make difference:*

Since MT Proxy performs its real-time spam classification process by reading the email from top to bottom in sequence, there will be a different behavior of MT Proxy with different spam structures.

To characterize this behavior, we have set up the test with 4 types of email structure, in which the top part is non-spam and the bottom part is spam. Four types used in the test are 30%, 50%, 70%, 100% (these numbers represent percentage of the bottom part in the email).

Figure 7 illustrates difference between the average time delay versus email size for different spam structures.

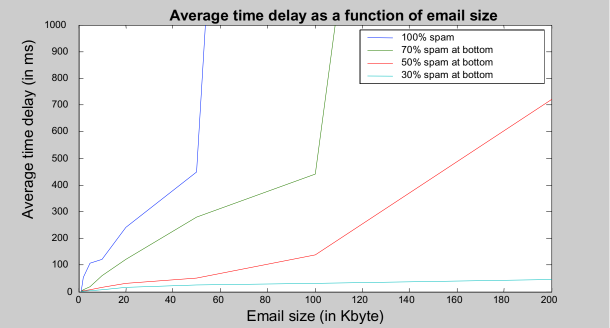

Figure 7 Average time delay as a function of email size for 4 types

The four email types behave differently on the graph. The 30% case has a small delay whilst it is greater for 50%, 70% and 100% case respectively. Thus, the ealier spam appears in the email, the more effective that MT Proxy can reduce spam at the mail server.

D. *Test with overlapping mail transfers*

The previous tests assumed that each inbound SMTP connection carried one email at a time, and there was no overlap in emails arriving from each of the three sources. We also tested the case where each source initiated the transmission of emails at regular intervals, regardless of whether a previous SMTP transfer had completed. This meant that some clients sending spam would end up with concurrent SMTP/TCP/IP open.

The amount of spam reduction as a function of the email sending rate is shown in Figure 8.

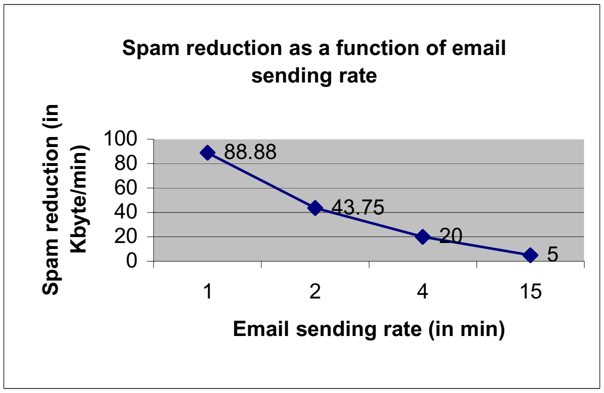

Figure 8 Spam reduction as a function of email sending rate

The graph shows a positive feature of MT Proxy – attempting to send emails faster with parallel sources does not significantly help you push spam into the receipient's MTA.

V. Research limitations and further work

MT Proxy clearly has an impact of slowing down the transfer of spam emails, and thus reducing the rate at which unsuspecting users' mailboxes fill up with spam. The architectural approach of performing spam-detection on a per-SMTP connection basis appears attractive, and certainly allows us to implement targetted response such as slowing down specific TCP/IP connections.

However, our results raise some valid questions too. The most dramatic benefits are seen for quite large emails, 20K to 100Kbyte long. It is fair to question whether the majority of spam falls into that category. Certainly, spam containing viruses tends to be in this range, but the 'real' spam emails are far smaller. Our University's IT department logged 6955 spam emails over a short, representative period and discovered that the sizes ranged from 1Kbyte to 11 Kbytes long (with a mean of 4.64Kbytes). The architecture of MT Proxy does not work very effectively for spam emails in this size range.

It is also worth noting the lesson of Figure 7 – MT Proxy is more effective against email bodies where signs of spam occur early. This is hardly surprising, as our TCP/IP rate reduction only kicks in after our detection algorithm decides the SMTP connection is transferring spam. Note that if the SMTP connection is used to transfer multiple emails without closing, then an early 'spammy' email can cause pain to the rest of the SMTP transfer. However, spammers would soon learn this work-around and simply disconnect and reconnect their SMTP connections for each individual email.

There are two problems with our first implementation. Firstly, short emails (only a few Kbytes long) might be completely transferred in only a small number of TCP/IP packet exchanges (given a likely maximum IP packet size of 1500 bytes, or 1460 bytes of TCP payload per packet). Thus dummynet has limited

ISBN 0-646-44190-6

ability to slow down the TCP transfer. Secondly, we 'forget' the spam statistics associated with a particular source IP address whenever the SMTP connection terminates. Each subsequent connection is treated as 'innocent until proven guilty'.

Our next version of MT Proxy will 'learn' which source IP addresses had attempted to send spam recently, and proactively rate-limit their subsequent connection attempts. In a sense this would be a dynamic, short-term black list created and maintained internal to MT Proxy. This 'learned black list' approach would have two positive consequences – it would fix the limitation reflected in Figure 7, and go a long way to solving the problem of most spam emails being small. Both benefits would accrue from the fact that many spammers use the same IP address (whether legitimate or hijacked zombie machines) for multiple spam emails. Thus the 2nd, 3rd, etc... emails (SMTP connections) from an IP address on the 'learned black list' would be rate-limited right from their beginning. Thus, even if the individual spam emails were roughly 5Kbyte long, the net effect of MT Proxy would be just the same as if the consecutive emails (SMTP connections) were a single, large email.

Since MT Proxy treats real time TCP traffic equally, it does the analysis after every 5 lines by default. Future work can be done on mapping the relationship between number of lines and spam reduction for different emails size. Administrator can decide to use which number of line setting depending on his SMTP server's email size distribution.

VI. Conclusion

Based on a simple, research prototype we have demonstrated a new approach to combating the affect of spam on end-user mailboxes. We describe a tool called MT Proxy that accepts SMTP connections on behalf of a target/recipient SMTP server, monitors the contents of each email passing through, and applies IP level rate limiting and latency penalties to SMTP connections believed to be carrying spam.

MT Proxy has proved to have certain contribution to the arsenal of anti-spam techniques. MT Proxy can effectively slow down traffic from spammers using FreeBSD's kernel resident ipfw/dummynet. As a result, spammers experience significant bandwidth limitation and additional delay, at least partially shifting the cost of sending spam back to the spammer's end.

MT Proxy's approach also avoids the damaging consequences of false positives because all email eventually gets through – MT Proxy errs on the side of letting spam through slowly rather than risk deleting legitimate emails.

VII. References

[1] H. Taylor, "Majority in Favor of Making Mass-Spamming Illegal Rises to 79% of Those Online", The Harris Poll , June 2003, http://www.harrisinteractive.com/harris_poll/index.asp?PID=387

[2] Houghton Mifflin Company, The American Heritage® Dictionary of the English Language, Fourth Edition http://dictionary.reference.com/search?q=spam, (as of August 2004)

[3] N. Krawetz, "Anti-Spam Solutions and Security", February 2004, http://www.securityfocus.com/infocus/1763#_ftn1

[4] M. Lamb, "Using Statistic to cause Spammers Pain", February 2003, http://www.martiansoftware.com/articles/spammerpain.html#note1

[5] J. Surmacz, "Safety in Numbers", CSO, June 2003, http://www.csoonline.com/metrics/viewmetric.cfm?id=563

[6] House of Representatives, "CAN-SPAM Act of 2003" , http://www.spamlaws.com/federal/108s877.html (as of August 2004)

[7] G. Gross, "CAN-SPAM law: Little impact so far", InfoWorld, May 2004, http://www.infoworld.com/article/04/05/20/HNcanspamimpact_1.html

[8] G. Gross, "Is the CAN-SPAM Law Working?", PC World, January 2004, http://www.pcworld.com/news/article/0,aid,114287,00.asp

[9] C. Williams, D. Ferris, "The cost of spam false positive", Ferris Analyzer Information Service. Report #385, August 2003, http://www.brightmail.com/pdfs/Cost_of_Spam_False_Positives___Ferris_Research_8_2003.pdf

[10] SpamHaus, http://www.spamhaus.org/ (as of August 2004)

[11] Spam and Open Relay Blocking System (SORBS), http://www.us.sorbs.net/ (as of August 2004)

[12] Open Relay Database (ORDB), http://www.ordb.org/ (as of August 2004)

[13] J. Snyder, "Test: Spam in the wild", Network World, September 2003, http://www.nwfusion.com/reviews/2003/0915spam.html

[14] H. Thornburng, "Introduction to Bayesian Statistics", Stanford University, April 2001, http://www-ccrma.stanford.edu/~jos/bayes/bayes.html

[15] P. Graham, " A Plan for Spam", August 2002, http://www.paulgraham.com/spam.html

[16] Robinson G. ,"Spam Detection", October 2003, http://radio.weblogs.com/0101454/stories/2002/09/16/spamDetection.html

[17] http://Turntide.com/ (as of August 2004)

[18] J. Twist, "Microsoft aims to make spammers pay", BBC News, December 2003, http://news.bbc.co.uk/2/hi/technology/3324883.stm

[19] G. J. Koprowski., "Spam filtering and the flague of False Positive", TechNewsWorld, September 2003, http://www.technewsworld.com/perl/story/31703.html

[20] J. B. Postel, "RFC 821 Simple Mail Transfer Protocol", August 1982, http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc0821.html

[21] W. Zekoll, smtp.proxy software, http://www.quietsche-entchen.de/software/smtp.proxy.html (as of August 2004)

[22] J. Bennet , "One giant steps towards ending spam", ZDNet UK, March 2004, http://insight.zdnet.co.uk/software/0,39020463,39149351,00.htm

[23] "Guide to Internet Terms: A Glossary", http://www.getnetwise.org/glossary.php (as of August 2004)

[24] A. Conry-Murray, "The Anti-spam Cocktail: Mix It Up to Stop Junk E-Mail", Network Magazine, January 2004, http://www.networkmagazine.com/shared/article/showArticle.jhtml?articleId=22103325&classroom=

[25] G. Keizer, "Microsoft to merge Caller ID with SPF anti-spam scheme", TechWeb News, May 2004, http://www.pcauthority.com.au/news.asp?PageType=ArticleDetail&CatID=1&ID=19815

[26] RFC 1725, "Post Office Protocol – Version 3", November 1994, http://www.apps.ietf.org/rfc/rfc1725.html

[27] RFC 2060, "Internet Message Acess Protocol – Version 4", December 1996, http://www.apps.ietf.org/rfc/rfc2060.html

[28] V. Schryver, "You Might Be An Anti-Spam Kook If...," http://www.rhyolite.com/anti-spam/you-might-be.html (as of August 2004)

ISBN 0-646-44190-6