Improving Spam Blacklisting Through Dynamic Thresholding and Speculative Aggregation

Sushant Sinha, Michael Bailey, and Farnam Jahanian
University of Michigan, Ann Arbor, MI 48109, USA
{*sushant, mibailey, farnam*} @*umich.edu*

Abstract

Unsolicited bulk e-mail (UBE) or spam constitutes a significant fraction of all e-mail connection attempts and routinely frustrates users, consumes resources, and serves as an infection vector for malicious software. In an effort to scalably and effectively reduce the impact of these e-mails, e-mail system designers have increasingly turned to blacklisting. Blacklisting (blackholing, block listing) is a form of course-grained, reputation-based, dynamic policy enforcement in which real-time feeds of spam sending hosts are sent to networks so that the e-mail from these hosts may be rejected. Unfortunately, current spam blacklist services are highly inaccurate and exhibit both false positives and significant false negatives. In this paper, we explore the root causes of blacklist inaccuracy and show that the trend toward stealthier spam exacerbates the existing tension between false positives and false negatives when assigning spamming IP reputation. We argue that to relieve this tension, global aggregation and reputation assignment should be replaced with local aggregation and reputation assignment, utilizing preexisting global spam collection, with the addition of local usage, policy, and reachability information. We propose two specific techniques based on this premise, dynamic thresholding *and* speculative aggregation, *whose goal is to improve the accuracy of blacklist generation. We evaluate the performance and accuracy of these solutions in the context of our own deployment consisting of 2.5 million production e-mails and 14 million e-mails from spamtraps deployed in 11 domains over a month-long period. We show that the proposed approaches significantly improve the false positive and false negative rates when compared to existing approaches.*

1 Introduction

Recent estimates indicate that as much as 94% of all Internet e-mail is unsolicited bulk e-mail (UBE) or spam [24]. This spam routinely impacts user productivity [21], consumes resources [14], and serves as an infection vector for malicious software [15]. In an effort to reduce these impacts, two major classes of anti-spam approaches have emerged: content-based filtering and blacklisting. Content-based filtering methods (e.g., [22, 10]) rely on classification algorithms to examine the contents of e-mails (i.e., headers, body) and to differentiate legitimate (or ham) and unsolicited (or spam) e-mail. Unfortunately, these methods are easy to evade [27] and can even be co-opted to block legitimate e-mail [16]. In an effort to scalably and effectively reduce the impact of these e-mails, e-mail system designers have increasingly turned to blacklisting. Blacklisting (blackholing, block listing) is a form of course-grained, reputation-based, dynamic policy enforcement in which real-time feeds of spam sending hosts are sent to networks so that the e-mail from these hosts can be rejected or specially marked. Currently, a large number of organizations provide these services for spam detection (e.g., NJABL [1], SORBS [3], SpamHaus [5], and SpamCop [4]).

Unfortunately, current spam blacklist services are highly inaccurate and exhibit both false positives and significant false negatives. For example, in a previous study of four prominent blacklists including SORBS, SpamHaus, SpamCop, and NJABL, false positives ranged from 0.2% to 9.5% and false negatives ranged from 35% to 98.4% [23]. To compensate for these limitations, blacklists are often used in conjunction with content-base techniques to further improve effectiveness [13]. However, the accuracy of spam blacklist services remains important as they are used reduce the cost of executing the more expensive content-based filters and often successfully blacklist e-mail that the content-filters fail to capture. While numerous novel blacklisting systems have been created (e.g., [20, 11]) to address this need, little work has focused on understanding why existing methods fail and how these methods might be directly improved.

In this paper, we explore the factors that affect the accuracy of traditional blacklisting techniques. We evaluate both factors that are inherent to the evolution of spammer behavior (e.g., targeted spam, low rate or "snow shoe"), as well as those integral to the approach itself (e.g., detection delay, over or under aggressive blacklisting). In evaluating these factors, we show that the vast number of false negatives come from IP addresses that have sent limited numbers of e-mail to each domain and very few domains overall, limiting the amount of information available with which to make reputation assignments. Furthermore, many of the false positives are attributable to the blocking of high volume, multi-user domains (e.g., web-mail) or to the lack of appropriate whitelisting to avoid such problems. We believe this fundamental tension between the increasingly small number of events with which to assign reputation to an individual IP and the accuracy of the reputation result must be overcome if these methods are to be improved.

In order to address this tension, we propose two novel techniques: *dynamic thresholding* and *speculative aggregation*. Fundamentally, these techniques work by supplementing spam events with local policy, usage, and routing data and by moving the blacklist aggregation and decision making away from the global collection infrastructure providers to the local network that is enforcing the policy decision. In *dynamic thresholding*, the determination to blacklist a spamming IP is neither global, nor based on a static threshold and whitelist combination, but rather based on the relative importance of a remote IP address to the local network. These local, customized blacklists are created by tracking the ratio of spam events for a remote IP to the number of outbound e-mails from the local network to that remote IP. The value of this technique is that: (i) it allows more aggressive threshold selection for remote domains that are not used often by the local network (ii) it alleviates the need for manual (and sometimes arbitrary or punitive) whitelist selection, as important remote IPs and domains will not be blacklisted unless they become "more trouble than they are worth," and (iii) policies are now local to the networks in which they are applied, allowing unique, dynamic thresholds and whitelists for each and every organization.

In the second approach, *speculative aggregation*, we use global information provided by spamtraps and BGP reachability information to determine the ratio of good (and active) IP addresses within a block to the number of spamming IP addresses within a block. Based on the prevalent notion that spamming IPs are clustered [11, 26], the ratio of spamming to non-spamming hosts in a network block is a good predictor of future spamming activity for a variety of reasons, including shared administrative and security policies, and dynamic hosts. The danger of such an approach, obviously, is that it may block entire prefixes, some of which send legitimate e-mail to the local network. To ameliorate this effect, we layer dynamic thresholding techniques on top of speculative execution to allow e-mail from bad neighborhoods if these neighborhoods are important to the local network. This technique improves the accuracy of generation by: (i) predicting potential new sources of spamming before they hit spam collectors and (ii) limiting the chance that these predicted hosts or networks are of use to the local network.

To validate our techniques, we collected headers from both a production e-mail system of a large academic department, which received 2.5 million e-mails, and our own separate spamtrap deployment, which received 14 million e-mails, during the month-long evaluation period of February-March, 2009. We built blacklists based on e-mails received on the spamtrap and on the e-mails received on the production network. We evaluated the blacklists using a combination of SpamAssassin and manual examination. In our evaluation, we found that these approaches performed significantly better than our implementation of existing approaches—the detection rate for the *dynamic thresholding* approach is three times that of the existing approaches for a false positive rate below 0.5%, and the *speculative aggregation* approach provides five times the detection rate when compared to the existing approach for a false positive rate below 0.5%.

To summarize, the main contributions of this paper are:

- An examination of the root causes of blacklist ineffectiveness. We argue that the decreasing number of observable spam events for a given IP severely hampers the accuracy of these techniques.

- We propose two techniques that address these root causes: *dynamic thresholding* and *speculative aggregation*. We argue that blacklist generation techniques should take into account both *local* usage, policy, and reachability information as well as *global* reputation data when making policy decisions and that these policy decisions should be made locally rather than globally. By shifting the location of these decisions and adding local context, we argue that we improve the accuracy of the reputation assignment.

- An evaluation of these techniques on data collected from a large academic departmental e-mail server and a demonstration of these two techniques that shows significant improvement over existing methods.

The remainder of this paper is structured as follows: We begin in Section 2 by exploring the root causes of existing blacklist failure. The architecture section, Section 3, introduces the speculative aggregation and dynamic thresholding techniques that make up our system and Section 4 evaluates these approaches in our production deployment. Section 5 provides a brief overview of related work. We conclude in Section 6 by discussing the limitations of and future of this work.

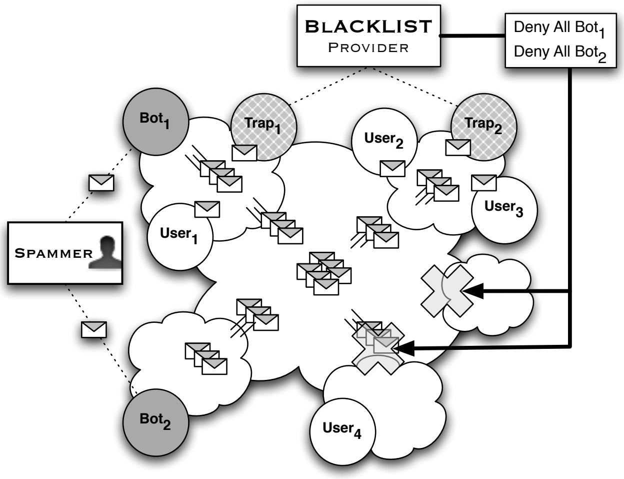

Figure 1: In existing approaches to blacklist generation, spam is sent to both legitimate users as well as unused accounts and domains (spamtraps). Spam is aggregated by a blacklisting provider and global provider configuration (inclusion threshold, whitelisting) is applied to determine blacklist contents. Customer networks may choose to locally implement the published blacklist policy (i.e., block or specially mark e-mail from those IPs).

2 Exploring the Inaccuracy of Blacklists

Spam blacklists serve an important role in blocking unwanted e-mail traffic. In this section, we examine the factors that limit existing spam blacklist accuracy in an effort to understand how to improve them. We begin by describing existing methods for blacklist creation and proceed to discuss the experimental setup used throughout this analysis (and the remainder of this paper) including the oracle we used, the production e-mail network, and our spamtrap deployment. By creating our own blacklist and analyzing its effectiveness in the context of our production e-mail system, we explore the factors (e.g., low-rate, low-volume spam, detection delay) that may impact the accuracy of existing blacklist creation methods. We conclude that trends in spammer behavior limit the number of events used to assign IP reputation and therefore impact the accuracy of these methods.

2.1 Background

A number of organizations generate dynamic blacklists for spam including: SORBS [3], SpamHaus [5] and SpamCop [4]. These spam blacklist providers deploy and monitor a number of unused e-mail addresses called *spamtraps*. There are two general approaches to spamtrap deployment. The first approach is to configure an e-mail server for an unused domain. For example, Project Honeypot [25] takes unused sub-domains, like mail1.umich.edu, within legitimate domains, like umich.edu, and monitors all e-mails to these domains. The second approach is to monitor unused users within a legitimate domain. In this deployment model, the e-mail server delivers all e-mails directed to existing users to their respective folders, but any e-mail directed to a nonexistent user is delivered to a separate account.

E-mails sent to spamtraps are then aggregated by a blacklist provider, as shown in Figure 1. The e-mails are aggregated by source IP and those IP addresses that exceed a threshold number of spamtrap hits within a given time window are blacklisted [7]. Since legitimate e-mail servers, such as yahoo.com, can also be used by spammers, the danger of a threshold-based approach is that it can blacklist legitimate e-mail servers causing widespread e-mail disruption. Therefore, commercial blacklist providers maintain whitelists of popular e-mail services and then use "Received" headers added by those legitimate servers to determine the IP addresses of the sender. Unfortunately, this scheme does not work universally. For example, this scheme does not work with Gmail because Gmail does not add the source IP of the client, if the web interface is used for sending the e-mail [23]. In addition, SpamCop uses a sample of DNS lookups to determine if IP addresses should avoid being blacklisted. However, this may not be a reliable estimate of actual e-mails delivered (e.g., DNS caching).

2.2 Experimental Setup

We evaluate the effectiveness of spam blacklists, as well as the other results in this paper, by observing e-mails to and from a large academic institution. In our experiments, we observed over $7,000$ local hosts during a month-long period from February 10, 2009 to March 10, 2009. We monitored traffic using a traffic tap (i.e., span port) to the gateway router that provides visibility into all of the traffic exchanged between the network and the rest of the Internet. The TCP streams on port 25 were reassembled using libnids [28], and full SMTP formatted e-mails were available for evaluation. During the measurement period we observed a total of $3,999,367$ SMTP connections, out of which $2,575,634$ e-mails were successfully delivered. The remaining SMTP connections failed or were aborted in large part due to non-existent users on the target domain.

2.2.1 Oracle Selection

In order to evaluate blacklist accuracy, we need to determine whether an e-mail on the production network is ham (legitimate e-mail) or spam. At the scale of the above measurement, a hand classification of e-mails was infeasible and so we used SpamAssassin [13] as our oracle classification. SpamAssassin uses a number of spam detectors and assigns scores for each detector. The total score for a message is computed by adding the score of all the detectors that classified the message as spam. If the total score exceeds the default threshold of 5.0, then the message is classified as spam. We used the default SpamAssassin configuration that came with the Gentoo Linux distribution. We configured SpamAssassin with two additional detection modules, Pyzor [2] and Razor [6], for improving SpamAssassin accuracy. We discuss the issue of oracle accuracy and our manual examination to cover the oracle limitation in Section 4.5.

2.2.2 Characterizing the E-mail Seen on the Network

Our month-long observation shows that roughly 75% of the delivered e-mail (i.e., ham and spam, but not failed connections) was spam. This number rises to 84% when failed SMTP connections (due to nonexistent users or domains) are included. We observed 764,248 unique IP addresses during this period in 35,390 distinct BGP prefixes, announced by 85 unique autonomous systems. Most of the spam messages (1,448,680) came from sources external to our network. However, we had a sizable number of spams (392,192) from within the network, which was roughly four times the number of spam messages (98,679) from hosts within the network to the rest of the Internet (we send much more spam to ourselves than to the rest of the Internet). Ham messages were dominated by internal to internal e-mails (369,431), followed by internal to external (151,860), and then by external to internal (114,792). The top five external senders (i.e., autonomous systems) of spam observed during this period at our network were Turk Telekom (69,278), Verizon (34,819), Telecomunicacoes Brazil (34,175), TELESC Brazil (27,360), and Comcast (25,576). The top five destinations (i.e., autonomous systems) for legitimate e-mail from our network were Google (87,373), Inktomi (4,559), Microsoft (3,466), Inktomi-II (2,052), and Merit Networks (1,793). The average message size for all e-mails was 5,301 bytes, with averages of 4,555 bytes, 15,152 bytes, and 1,916 bytes for spam, ham, and failed connections respectively.

Top level domain	E-Mails received	The % of total spamtrap e-mails	Unique sources
.org	289,991	2.1	137,725
.org	449,803	3.2	216,291
.org	571,856	4.1	253,777
.com	1,090,611	7.8	407,838
.net	1,159,353	8.3	439,152
.net	1,306,411	9.4	473,686
.net†	1,321,232	9.5	18
.com	1,458,865	10.5	486,675
.com	1,552,240	11.2	521,321
.net	1,698,295	12.2	513,057
.net	3,004,583	21.6	689,633

Table 1: Our spamtrap deployment by top level domains, number of e-mails received, and number of unique sources. Over 14 million spam e-mails were captured and analyzed. †This domain received between 28,244-58,597 spam e-mails from 2-11 unique source addresses per day. Interestingly, the total number of distinct source IP addresses was small (18) and all of them belong to Gmail. We conjecture the domain was being spammed using numerous compromised Gmail accounts.

2.2.3 Characterizing the Spamtrap Deployment

In order to understand the issues effecting the root causes of the false positives and false negatives produced by existing blacklist aggregation algorithms, we deployed our own spamtrap deployment covering 11 domains during the measurement period. The e-mail server in these domains copied e-mails sent to non-existent users to a separate account for post analysis. In total, we observed $13,903,240$ e-mails from $1,919,911$ unique sources between February 10, 2009 to March 10, 2009. Table 1 shows the number of e-mails received and the number of unique sources observed on each of these domains. Over 14 million spam e-mails were captured and analyzed.

2.3 Factors that May Influence Blacklist Accuracy

Having now described how existing blacklists are created and the context in which we perform our experiments, we now embark on an exploration of the reasons for the

Number of domains	OR of domains		AND of domains	
	FP rate	FN rate	FP rate	FN rate
1	2.2	71.5	2.2	71.5
2	2.2	66.7	1.0	80.58
3	2.2	63.6	1.0	83.54
4	2.3	61.6	0.0	100.0
5	2.3	61.6	0.0	100.0
6	2.3	60.4	0.0	100.0
7	2.3	59.2	0.0	100.0
8	2.4	58.2	0.0	100.0
9	2.4	57.5	0.0	100.0
10	2.4	57.0	0.0	100.0
11	2.4	56.8	0.0	100.0

Table 2: The false positive and false negatives rates when the spamtrap deployment is expanded domain by domain using existing methods. No spamming IP address is seen by more than three spamtraps.

false positives and false negatives we observed. We examine two broad categories of potential reasons for these inaccuracies, including trends in spamming behavior (i.e., targeted spam, low-volume spam) and systemic properties of the blacklist creation methods (i.e, detection delay, static whitelisting).

2.3.1 Targeted E-Mail

One possible explanation for the false negative rates observed by the blacklists is that some of the e-mails are part of a targeted spam campaign. Obviously, if a spammer sends targeted spam to a domain in which there are no spamtraps, it is impossible to blacklist the host. To explore the impact of this potential cause of false negatives, we examined the impact of spamtrap deployment size on accuracy. By building blacklists from spamtrap deployments of size 1, 2, ..., 11 we can explore the targeted nature of spam. Table 2 shows the result of this analysis. We consider two cases for blacklist generation, one in which an IP address is blacklisted if a spam host appears on any spamtrap domain, and one in which it is blacklisted if it appears on every spamtrap domain. The false negative rate for the OR of domains converge to roughly 56.8%, indicating that roughly 57% of the spam does not appear in any of the spam traps—a reasonable upper bound on the amount of targeted e-mail. A lower bound on the amount of global e-mail can be seen in the false negative rate for the AND of domains, 100% after just three spamtrap domains are combined. Clearly, global spam seems to be quite limited. While a precise estimate is difficult without a universal deployment, it is clear that the blacklists are impacted by significant targeted behavior.

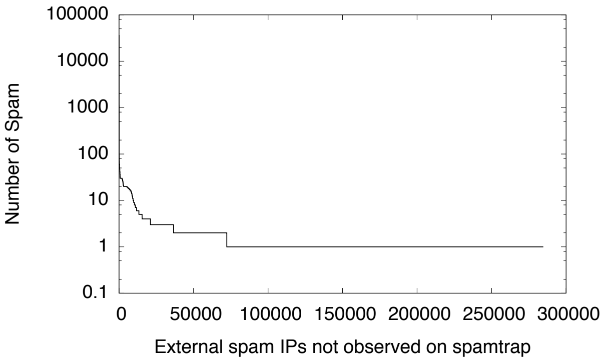

Figure 2: The number of e-mails sent by external spamming sources that were not observed on any spamtrap. Most of these sources sent just one spam to our network.

2.3.2 Low Volume Spam

Another potential explanation of the false negatives observed is that although the campaigns are global, the vast number of hosts available to spammers makes it feasible to send a small handful of e-mails from each host to each target user or domain and still send millions of mails. This contributes to the problem of false negatives, in that most blacklist providers will not blacklist hosts for a single spam sent to a spamtrap. In order to investigate this phenomenon, we examined the spam sent to our network that was not observed on ANY of our spamtraps. For each spamming source, we calculated the number of spams sent to our network over the measurement period. As shown in Figure 2, while some spammers clearly sent numerous spams, the vast majority of sources sending spam to our network only sent a single spam. Therefore, any approach that requires multiple spamtrap hits will never report these high-volume, single-target sources as spammers.

2.3.3 Detection Delay

A third potential source of false negatives is the reactive nature of blacklist generation. By their nature, hosts are not put on blacklists until they send enough e-mail to spamtraps. During a fast global campaign, it is possible that we might receive the spam at the production network before it reaches a spamtrap or before the blacklist provider can send out an update. To explore the impact of this delay, we examined the idea of retroactive detection. That is, we created blacklists as expected, creating blacklist entries for spamming hosts only if they sent spam over a given threshold. We then enabled retroactive detection, that is, we classified hosts as spam if they sent e-mail to the spamtraps *at*

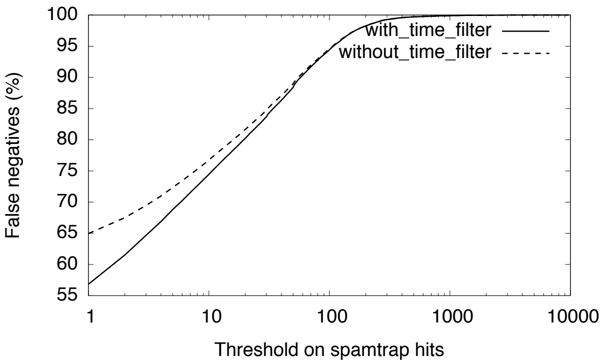

Figure 3: The effect of blacklist detection delay on false negatives for different thresholds using existing blacklisting approaches. The difference between blacklisting only those spamming IP addresses whose spamtrap hits occur before an e-mail is received on the production network is compared to blacklisting the IP address if it appears anywhere in the trace (often in the future). The biggest cost of detection lag is for small threshold values in traditional approaches.

anytime during our observations (potentially several weeks after we observed the spam). Figure 3 shows the result of this analysis. For small threshold values (i.e., blacklisting when we see only one spam) the decrease in false negatives from retroactive detection is 10%. For higher thresholds, this value decreases. Thus 10% approximates a reasonable upper bound on the false negatives caused by delay.

2.3.4 Static Whitelisting

False positives occur from blacklists when legitimate e-mail servers are blocked. Often times this occurs when a legitimate e-mail sever has been compromised or is being used by compromised hosts. In many cases this can be avoided, as the e-mail server can add the IP address of the sending host in the e-mail headers, but this is not always the case. For example, e-mails sent from the Gmail web interface do not include the client's IP address, and as a result, blacklists are only left with the choice of blacklisting the server itself. What these blacklists lack is a notion of what servers are used and not used by a specific network. For example, consider the data in Figure 4. In this figure, we examine the amount of mail we sent to those networks (autonomous systems) that sent us spam and those that sent us ham. Note the stark contrast between the e-mail we sent to legitimate networks and those we sent to spamming networks—90% of ham senders received more than one e-mail from us, while over 60% of spammers never received a single e-mail from our network. A few spamming domains received a large number of e-mails from us. As expected, these are false positives from web hosting sites as in the example above: Google (87,373), Inktomi (4,559), and Microsoft (3,466). These sites could be whitelisted, but without knowing what services a network uses, this whitelisting may create false negatives. What blacklists need is a way to figure out what remote networks are important to a given network.

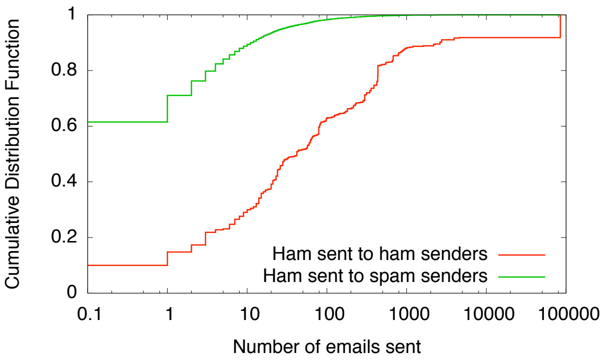

Figure 4: The amount of legitimate e-mail sent by our network to networks that sent us spam and legitimate e-mail. There is a huge difference in how our network uses the IP addresses that sent spam and those that do not.

2.3.5 Putting it Together

In this section, we explored the root causes of traditional threshold-based blacklist creation algorithms. We note that spam is both targeted and in many cases, low-volume. Capturing this spam places pressure to lower detection thresholds to require fewer and fewer spamtrap hits in order to capture the spamming behavior. This pressure places additional burden on the blacklist operators to select the appropriate whitelists to avoid the increasing number of false positives. This tension is further exacerbated by the delay inherent in existing blacklisting these methods, which is most pronounced at precisely the lower thresholds being utilized. What is needed then, are additional sources of information and methods that can be used to determine when and how to be aggressive in blacklisting.

3 Architecture

In this section, we describe our approach to mitigating the limitations discussed in the previous section. Rather than a "one size fits all" method, which is embodied by the generation schemes for existing production blacklists (and shown in Figure 1), our method (shown in Figure 5) decides on blacklisting policy with the help of local information including usage patterns (i.e., e-mail usage), network routing visibility (i.e., BGP information), as well as global information (i.e., spamtraps). With local context in hand,

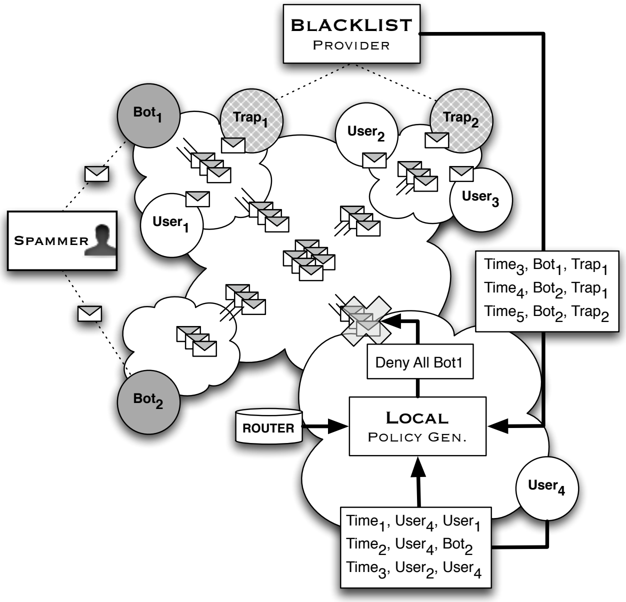
Blacklist Provider
Bot$_1$
Trap$_1$
User$_2$
Trap$_2$
User$_1$
User$_3$
Spammer
Time$_3$, Bot$_1$, Trap$_1$
Time$_4$, Bot$_2$, Trap$_1$
Time$_5$, Bot$_2$, Trap$_2$
Deny All Bot1
Bot$_2$
ROUTER
Local Policy Gen.
User$_4$
Time$_1$, User$_4$, User$_1$
Time$_2$, User$_4$, Bot$_2$
Time$_3$, User$_2$, User$_4$

Figure 5: Our approach to spam blacklist generation. Rather than enforcing a global "one size fits all" policy, a local generation algorithm combines local usage patterns (i.e, e-mail usage), allocation information (i.e., BGP reachability), and global information (i.e., spamtraps) to make localized, tailored blacklist policies.

the policy generation mechanisms can eliminate false positives that occur from blacklisting locally important e-mail servers. In addition, the blacklisting can be more aggressive in blacklisting networks rather than individual sources—if these networks are not important in the local context. In this section, we see how this general idea is applied in two specific improvements to spam blacklist generation: dynamic thresholding and speculative aggregation.

3.1 Dynamic Thresholding

In a simple static, threshold-based approach model of existing methods, a threshold is decided and an IP address is blacklisted if the number of e-mails sent to spamtraps crosses that threshold. However, the simple threshold mechanism can blacklist e-mail servers that are important e-mail servers (e.g., Gmail) if they are used to send even a small amount of spam. One solution to this problem is to compare local network traffic to the spamtrap e-mails. The assumption here is that a valid e-mail server will have significantly more e-mails delivered to valid addresses than to spamtraps, while a spamming source will hit significantly more spamtraps than legitimate users in the live network, as we saw in Figure 4. Therefore, we propose a dynamic threshold approach that computes the ratio of e-mails on the live network to the number of e-mails seen at the spamtrap and blacklists sources if the computed ratio is below a configured ratio. For example, consider that the configured ratio is 1 and a source IP address is observed 5 times on the e-mail server and 10 times on the spamtrap. The ratio is $5/10 = 0.5$, which is lower than the provided ratio of 1, so this source IP address will be blacklisted.

3.2 Speculative Aggregation

While a dynamic threshold approach addresses the false positive issues with the spam blacklists, the blacklists still exhibit a significant amount of false negatives. Recall from the previous section that this may be the result of low volume spammers, targeted spam, or detection delays. In order to attack false positives resulting from sources we have not seen, the only solution we have is to speculate about potentially bad sources. One potential source of information that we have to inform our prediction is the list of previous spamming sources. In Figure 6, we aggregated the spamming sources that have missed the spamtraps by the BGP prefixes (obtained from routeviews.org project). We find that most of these prefixes have a large number of sources that have previously hit spamtraps. We conclude, therefore, that the number of sources that have hit the spamtraps is a good indication of spamming prefixes. Secondly, we find that most of the sources that have missed spamtraps in these prefixes have sent at least one spam as well. Therefore, blacklisting these BGP sources will have little impact on

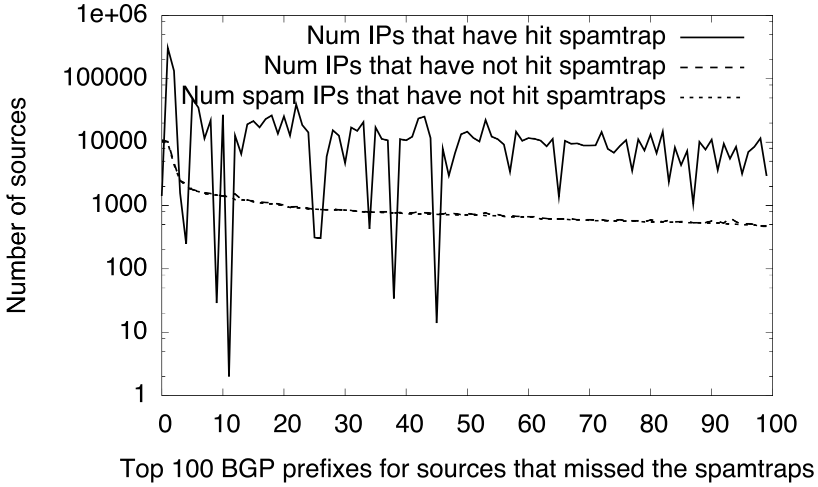

Figure 6: The top 100 BGP prefixes for which we failed to blacklist an IP address. Almost all of the IP addresses in these prefixes have previously sent spam.

legitimate sources.

In order to detect these sources that do not hit spamtraps, we can use the context of local network traffic to determine bad and good neighborhoods with respect to the network. For example, consider a /24 network address range that has 75 active sources. If 50 of them are already blacklisted then it is quite likely that the remaining 25 are also spammers from the perspective of the network and therefore a heuristic that also looks at the bad neighborhood may be able to filter out these sources. In order to enlarge the scope of the blacklists, we leveraged topological information available through Border Gateway Protocol (BGP). BGP is used to exchange routing information between autonomous systems (AS) and helps identify organizational and logical boundaries.

We aggregated traffic from spamtrap feeds and the live network by BGP prefixes and autonomous systems. Then we used three parameters for deciding to blacklist a network as opposed to individual sources. First, the ratio of good to bad e-mails for the network is below the ratio provided in the ratio-based approach. Second the ratio of bad to active sources in a network should be above a provided ratio. This parameter decides when we can speculatively classify an entire network as bad. However, we may over-aggressively blacklist networks when we have seen very sources from that network if we communicate very infrequently with that network and have little insight into the network's activities. Therefore, the final parameter is the ratio of the minimum number of bad sources to total possible sources.

3.3 Implementation

Having described our broad approach, we now discuss how e-mails from live networks and spamtraps are aggregated, how blacklists are generated, how they are applied, and how entries are removed from the blacklists.

3.3.1 Aggregating Sources in a Moving Time Window

The two streams of e-mail messages, the spamtrap e-mails (bad events) and the live network e-mails (good events), are merged together using the e-mail's timestamps. Sources are extracted and fed to the blacklist generation algorithms. We use a jumping window model to store network history. In this model, the events are stored for a given time window and the time window jumps periodically. For example, in a system with a history window size of 10 hours and a periodic jump of 15 minutes, the events are kept for 10 hours and the window jumps by 15 minutes. The counts in the last (oldest) 15 minutes are then aged out.

Note that we do not process or annotate the network e-mails (good events) in any way. As a result this live stream may in fact contain spam e-mails directed at a legitimate user. While this does not appear to have significantly impacted the accuracy of our system (see Section 4), it does leave open the possibility that an attacker could improve the reputation of a spam sending source by only sending spam to legitimate users (hence avoiding our "classifier"). The use of additional "classifiers" (e.g., SpamAssassin) in the reputation assignment and the corresponding error that the may be introduced (see Section 4.5) are interesting areas for further exploration.

3.3.2 Generating Blacklists

For the modeling of existing approaches, we count the number of bad events (i.e., spamtrap hits) for each source IP address and send those sources that cross a given threshold. For the dynamic threshold approach, we calculate the ratio of good events to bad events and send those sources for which the ratio is below a given ratio. The count for each address is taken over the history window. To enlarge blacklists from source IP address to BGP prefixes and autonomous systems, we take into account two additional parameters. A BGP prefix or an autonomous system is blacklisted if all three conditions are satisfied—the ratio of good events to bad events for the prefix is below the given ratio, the number of bad IP addresses to active addresses is above the minimum fraction, and the ratio of bad IP addresses to total possible addresses is above the specified threshold.

3.3.3 Applying Blacklists

The blacklists are generated periodically and the lists are refreshed each time. To save on messaging, the blacklist generation technique only emits new entries or instructions to remove old entries. These blacklists are applied to e-mails from the live network until a new list is refreshed. In our implementation, we maintained the blacklist as a list of IP addresses in the open source database PostgreSql. We used the Postgres GiST index ip4r for quickly checking whether a source IP is blacklisted.

3.3.4 Removal from Blacklists

Finally, we need to define the policy for removing entries from the blacklist. For existing approaches, an IP address is not blacklisted until the network has seen enough bad events from that IP address. When the network history for an IP address goes lower than the threshold, the IP address is removed from the blacklist. For the dynamic threshold approach, an IP address is removed from the blacklist when the ratio of good events to bad events goes above the specified ratio. BGP prefixes and autonomous systems are removed from the blacklist if any of the three conditions fail—the ratio of good to bad events exceeds the specified ratio, or if the number of bad IP addresses to active IPs from the network falls below the provided threshold, or the ratio of bad IP addresses to total possible addresses falls below the threshold.

4 Evaluation

In this section, we compare the three approaches to blacklist generation: the static threshold-based model of existing approaches, the dynamic thresholding approach, and the speculative aggregation approach. These approaches are compared in terms of their false positive rate and false negative rates as well as time and space performance. In addition, we compare the stability of the approaches for a variety of chosen parameters. The comparison is accomplished by using the deployments described in Section 2.

4.1 Comparing the Three Approaches

We now compare the simple model of existing methods with the two approaches proposed in the paper: the dynamic thresholding approach and the speculative aggregation approach. Recall that in the threshold-based model, an IP address is blacklisted if it has more spamtrap hits than the provided threshold. In the dynamic threshold approach, an IP address is blacklisted if the ratio of the number of good events (e-mails to the live network) to the number of bad events (e-mails to the spamtrap) is below the specified ratio. In the speculative aggregation approach, the IP addresses are aggregated by BGP prefixes and autonomous systems. Then BGP prefixes or autonomous systems are blacklisted instead of individual IP addresses if it is found that these networks are not of importance to one's network. Since the speculative approach uses a dynamic threshold technique for blacklisting individual IP addresses, it is essentially a combination of the dynamic threshold and speculation approaches.

Figure 7 shows the trade-off between the false negative rate and the false positive rate for the three approaches. First, we find that the dynamic thresholding approach yields a significantly better false negative rate for any false positive rate provided by the static threshold-based model. Conversely, the dynamic threshold method provides a significantly better false positive rate for any false negative rate provided by the static threshold-based model. For example, the false negative rate for the dynamic threshold approach is roughly 20% better than the existing model for false positive rates below 0.5%, which is roughly three times the detection rate of the existing model.

The speculative aggregation further improves detection rates over the dynamic thresholding approach. The false negative rate improvement of the speculative approach over the existing model is between 30-40% for false positive rates below 0.5%, which is roughly 4-5 times the detection rate of the existing model. For false positive rates greater than 0.5%, the dynamic thresholding approach provides a slight improvement over the existing model. Over this range, the speculative aggregation approach provides almost double the detection rate over the existing approach.

The operational point for an approach is usually the knee in the false negative and false positive curve. For the existing approach, the knee is at 0.67% of false positives and 71% of false negatives, and for the dynamic threshold approach, the knee is at 0.31% of false positives and at 67% of

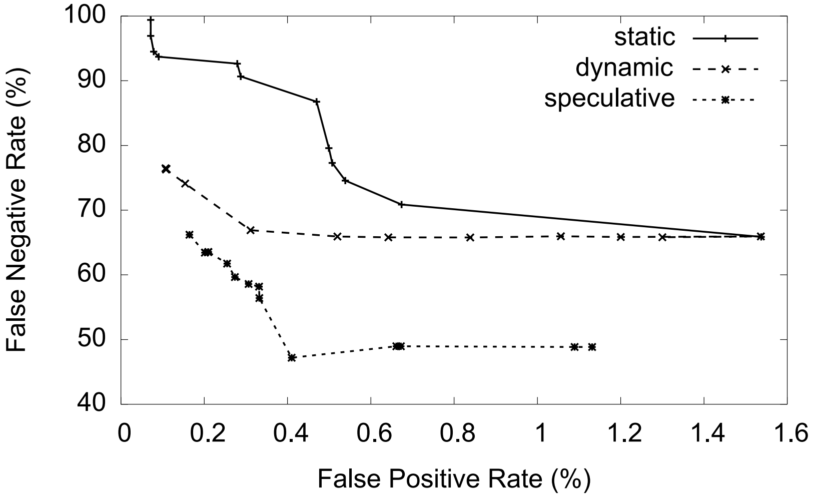

Figure 7: Trade-off curve for the false positive rate and the false negative rate for the three methods for a variety of parameter values. The speculative aggregation approach outperforms both existing methods and dynamic thresholding approach alone.

Existing Model			Dynamic Thresholding		
Threshold	FP	FN	Ratio	FP	FN
1	1.54	65.9	100.000	1.30	65.8
2	0.67	70.9	75.000	1.20	65.8
3	0.54	74.6	50.000	1.05	65.8
4	0.51	77.3	25.000	0.83	65.8
5	0.50	79.6	10.000	0.64	65.8
10	0.47	86.8	5.000	0.52	65.9
15	0.29	90.7	1.000	0.31	66.9
20	0.28	92.6	0.010	0.15	74.1
25	0.09	93.7	0.005	0.11	76.3
30	0.08	96.9	0.001	0.09	76.4

Table 3: The values of the existing static threshold-based model and the dynamic thresholding approaches and the corresponding false positive and false negative rates.

false negatives. For the speculative aggregation approach, the knee is at 0.40% of false positives and 48% of false negatives.

4.2 The Effects of Existing Model and Dynamic Thresholding Parameterization

In both the existing approach and the dynamic thresholding approach, a network operator has to choose the threshold or the ratio for blacklisting. Since the thresholds are chosen by hand, we need to investigate how stable these schemes are for any given threshold. Table 3 shows the false positives and false negatives of the two approaches for different values of the thresholds and the ratios. For the existing approach, the false positive rate increases suddenly from 0.67% to 1.54% when the threshold is reduced from 2 to 1. For the dynamic thresholding approach, the increase in false positives is more gradual. Looking at the data, we find that many mail servers in the network had one spamtrap hit in the time window of 10 hours.

4.3 Impact of Parameters on Speculative Aggregation

Recall that in speculative aggregation, BGP prefixes or autonomous systems are blacklisted if three conditions are satisfied. The first is if the ratio of good events (mails to the live network) to bad events (mails to the spamtraps) is below a specified ratio. The second is if the ratio of bad sources to total active sources is above a given threshold. Finally, the third is if the ratio of bad sources to total size of the BGP prefix or the autonomous system is above a given threshold.

Figure 8 shows the variation in the false positive rate and false negative rate for the speculative approach when the above three parameters are varied. The default ratio was kept at 0.1 and varied from 0.01 to 100. The ratio of bad IPs to total active sources was kept at 0.4 and varied from 0.1 to 0.99. The minimum ratio of bad IPs to total possible IPs in the network was kept at 0.01 and varied from 0.001 to 0.1. First, we find that the first and third parameters have significant impact on the false positive and false negative rates of the speculative aggregation approach. But varying the second parameter has very limited impact on the approach. Second, changing the minimum number of bad IP addresses provides a much better trade-off between the false positive rate and the false negative rate when compared to changing the ratio of good to bad events.

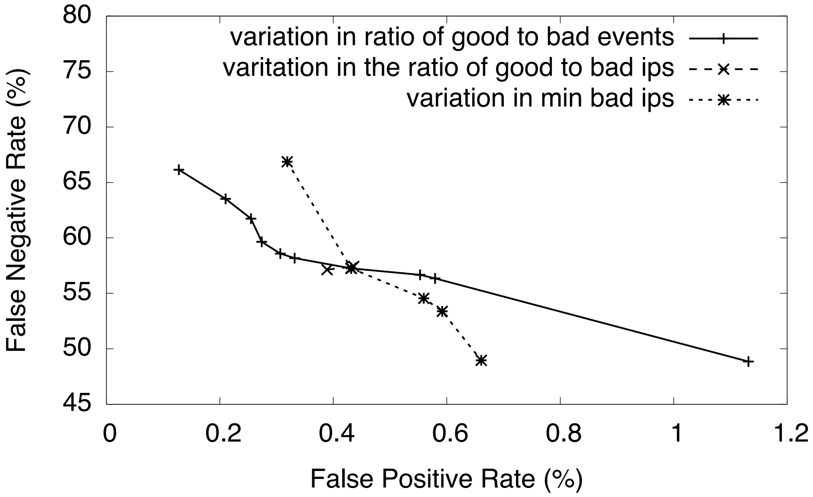

Figure 8: Impact of three parameters on the false positive rate and false negative rate for speculative aggregation.

Blacklist size	Look up time (ms)	Index size
1000	0.045	40 KB
10,000	0.046	264 KB
100,000	0.050	3.7 MB
1 million	0.052	32 MB

Table 4: The impact of blacklist size on the time to look up an IP address. The index size grows linearly, but the lookup time for an entry is very fast.

4.4 Performance

Figure 9 shows the growth of blacklists for the three techniques: existing static threshold-based models, dynamic thresholding and speculative aggregation. We find that the growth of blacklist size is highest for the ratio-based techniques and lowest for the speculative-aggregation technique, as it combines many sources into BGP prefixes. In order to see how blacklist size may impact the performance of the system, we created tables with different blacklist sizes in the database Postgresql (which is what we have used in our system). Then we created an index on the IP addresses and prefixes using GiST index in Postgresql. Table 4 shows the time to look up an entry and the index size for different sizes of the blacklist. We find that the time to look up an entry does not increase significantly, and for the month's operation, the index size is easily manageable.

4.5 Impact of the Oracle on Accuracy

To validate the accuracy of SpamAssassin, we hand classified several e-mail boxes and fed them to SpamAssassin. As published in our previous study [23], SpamAssassin had a false positive rate of less than 1% and a false negative rate of around 5%. Obviously, the error in the oracle is likely to impact the accuracy of our measurements. For example, a false negative for the SpamAssassin oracle (i.e., a spam classified as ham) may appear as a false positive for the blacklist, if, unlike the oracle, the blacklist correctly identified the e-mail as spam.

Given the inaccuracy of SpamAssassin, the accuracy of false positives for the blacklist will be $FP_{blacklist} \pm FN_{spamassassin}$ and the accuracy of false negatives for the blacklist will be $FN_{blacklist} \pm FP_{spamassassin}$. Therefore, given the values of 20% (or greater) for the false negatives of blacklists and 1% for the false positives that appear in this paper, we arrive at or $1\% \pm 5\%$ for the blacklist false positives and $> 20\% \pm 1\%$ for the false negatives. Clearly the small false positive rate of the blacklists is likely to be lost in the noise of the oracle. In order to overcome this problem, we hand classified the false negatives of the SpamAssassin. Instead of manually examining all false negatives of the SpamAssassin (potentially all legitimate e-mail), we only hand classified sources that hit spamtraps (and hence were sent to no legitimate user).

5 Related Work

Recently a number of research papers have looked at the algorithms to generate blacklists. Ramachandran *et al.* [20] proposed a new method to blacklist source IP addresses based on their e-mail sending patterns. However, their experiment is only based on e-mails received on the spamtraps and not on e-mails received on the live network. As a result, they only evaluate the false negatives of spamtrap received e-mail and not the false positives of their approach. In our

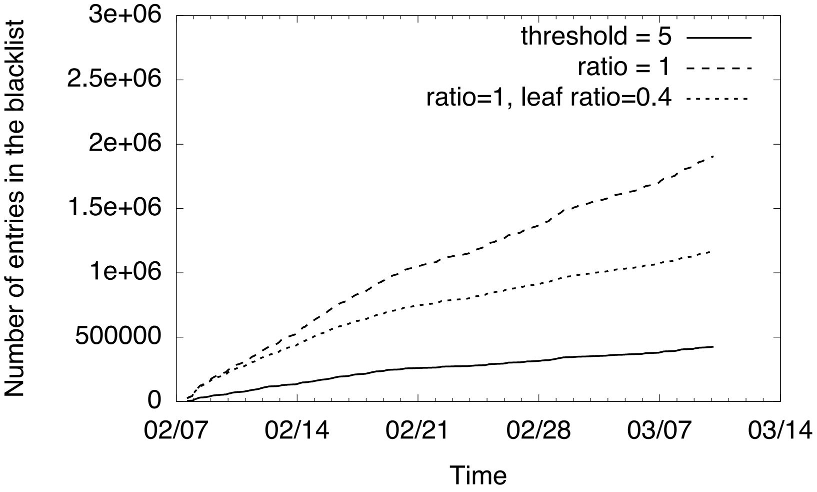

Figure 9: The growth of the blacklist size for the three approaches.

study, we generate blacklists based on spamtrap e-mails and then apply them to the e-mail on the live network, so we evaluate the false positives and false negatives for the e-mail on the live network.

Xie *et al.* [29] have shown that a large number of IP addresses are dynamically assigned and e-mails from these IP addresses are mostly spam, so they recommend adding dynamic IP ranges into blacklists to reduce the false negatives. Zhang *et al.* [31] argued that a common blacklist may contain entries that are never used in an organization. So they proposed an approach to reduce the size of the blacklists and possibly reduce the computational overhead in blacklist evaluation. However, their proposed approach only improves the blacklist "hit-rate" and not the overall false positive rate or the false negative rate of the blacklists.

Ramachandran and Feamster [19] collected spam by monitoring e-mails sent to an unused domain and performed a preliminary analysis of spammers. They observed that the spamming sources are clustered within IP address space and some of these sources are short-lived. Our approach of speculative aggregation automatically identifies bad IP neighborhoods by considering the sources in the neighborhoods that have hit the spamtraps and the sources that have not yet hit the spamtraps.

A number of papers have questioned the effectiveness of blacklists. Ramachandran *et. al.* [18] analyzed how quickly Bobax infected hosts appeared in the SpamHaus blacklists. They found that a large fraction of these hosts were not found in the blacklist and demonstrated the delay in blacklisting. We also demonstrate the problem of delay in blacklisting and show that our proposed speculative aggregation technique is able to address this problem.

E-mail servers can be easily overwhelmed when a significant amount of spam is received. Accordingly, there has been increased interest in developing lightweight measures for reducing the load on an e-mail server. Venkataraman *et al.* [26] proposed coarse IP-based blacklists to reject e-mails and to reduce server load. They monitored e-mails on a mail server and used SpamAssassin score to identify spam and ham. The spam and ham ratio for IP addresses and IP clusters were computed and the history was used for blocking IP addresses in case of server overload. It is important to note that they required an existing spam detector in order to create their IP blacklists. We do not rely on a pre-existing spam detector and instead use e-mails on our deployed spamtraps and the e-mails on the production network to generate blacklists. While their approach is a lightweight technique abstracted from an existing detector, our approach is to build blacklists using traditional spamtraps. Therefore, our blacklists can be used for improving a spam detector rather than just reducing the load on the server.

Xie *et al.* [30] used a spam detector to classify e-mails as spam and ham and then proposed an automated way to generate spam signatures. However, their approach focuses on using message content for developing signatures. Similarly, Beverly *et al.* [8] have used TCP information from spam and ham packet level data to develop TCP level spam features. Kanich *et al.* [12] empirically evaluated the success rate and the monetization from spam. Rajab *et al.* [17] analyzed the botnet behavior across a number of dimensions but did not develop any automated way of blacklisting them.

Most similar to this effort is the recent work of Hao *et al.* [11]. The authors used a spam detector to separate e-mails into ham and spam. By examining these e-mails, they identified a number of network level features that can be used to differentiate ham and spam. Hao *et al.* then used machine-learning models on those network features to build a spam classifier. Similar to Venkataraman *et al.* [26], Hao *et al.* relies on an existing spam detector to build the data streams, which are used to feed the classifier and periodically retrain it. Our efforts are similar to this work in that we have we also identified features that differentiates spam and ham [23]. Rather then examining numerous features and combining them in a classifier, we focus on a small handful of these features (e.g., remote BGP prefix clustering) and explicitly explore the properties and tradeoffs of each. Further, while the work of Hao *et al.* does result in a lightweight spam filter that performs on par with existing approaches, the filter is offered as an alternative and does not explore how existing blacklists fail and can be improved. The key difference in these two systems, however, is the role of local and global information in the classification process. Hao *et al.* is similar to existing blacklist deployment models in that the classifiers are built from global sources of information only and are not customized. We view this work as complimentary to Hao *et al.* in that the local generation and customization approach presented here could likewise be applied to their classifier generation to yield improved accuracy.

6 Discussions

In this paper, we presented a detailed investigation of blacklist generation techniques using 2.5 million e-mails from a large academic network and 14 million e-mails from a spamtrap deployment in 11 domains. We presented a detailed analysis of ham and spam sources, based on our own spamtrap deployment that helps to explain the limitations of existing spam blacklist approaches. We then proposed two improvements to the standard threshold-based blacklist approach. The first one reduces false positives by comparing traffic on the live network to the spamtrap hits for blacklisting sources. The second takes network traffic into account to safely aggregate bad sources into bad neighborhoods. The proposed techniques, when combined together, improved the false negative rate by 4-5x for false positive rates below 0.5% and 2x for false positive rates above 0.5%.

6.1 Ethical Considerations for Spam Blacklist Evaluation

The increasing level of detailed access and subject interactivity of Internet research experiments raise numerous ethical issues for research [9]. Beneficence refers to the process by which a researcher seeks to do good or seeks to maximize benefits while minimizing harm. As mentioned previously, the phenomenon of unsolicited bulk e-mail or Spam is one that routinely impacts user productivity [21], consumes resources [14], and serves as an infection vector for malicious software [15]. The main benefit of this work is to examine techniques for reducing this burden. The greatest risk to the subjects of the study is the loss of privacy. In an effort to minimize this harm, no personally identifiable information of the subjects is published herein. The collected data was restricted to e-mail source and destination servers only, except in the following two cases: (i) we hand classified e-mail contents of four subject's inboxes with their explicit user permission in order to determine the accuracy of our oracle (ii) we hand classified the contents of e-mails not marked by the oracle, but that were sent to our spamtrap where no legitimate users preside. Because this analysis was performed offline, no e-mails were modified during the study.

6.2 Limitations

While effective at its goal of addressing the limitations of blacklist generation, this work has several limitations and opportunities for future work. First, the speculative aggregation technique presented in this paper is somewhat preemptive in nature. While our evaluation shows that the proposed technique provides significantly better trade-offs, it may be unacceptable to block traffic from hosts preemptively. Second, like other reputation-based systems, our blacklist generation system is also exposed to the attacks that increase or decrease the reputation of sources. While the dynamic threshold technique provides protection against attacks to blacklist a mail server, it is still vulnerable to attackers increasing the reputation of sources by sending a large number of e-mails to a legitimate user. Currently, our system only counts the total number of e-mails on the live network and is vulnerable to such an attack. A system that counts the number of unique users to which a source sends mail may be resilient to such an attack and will be explored in the future. Third, blacklist providers often indicate that they are not responsible for the blocking email as they only generate the blacklists, and it is the network administrators who are blocking the e-mails. However, these blacklists currently are generated centrally. The only option a network administrator has is to accept or reject a given blacklist. Our proposed deployment model requires either publication of raw spamtrap data to subscribers or the publication of (aggregate) local network traffic statistics to the blacklist providers, each of which have obvious limitations (or attacks against them). Finally, in our current implementation, we only extracted the first "Received" header in the email messages. In our dynamic threshold mechanism, we could not blacklist sources if we did not blacklist the first source. In the future, we may like to add support for blacklisting of sources in received headers beyond the first one.

7 Acknowledgements

We would like to thank the anonymous reviewers for their comments and extend special thanks to Thorsten Holz, our shepherd, for his efforts in significantly improving this paper. This work was supported in part by the Department of Homeland Security (DHS) under contract numbers NBCHC080037, NBCHC060090, and FA8750-08-2-0147, the National Science Foundation (NSF) under contract numbers CNS 091639, CNS 08311174, CNS 0627445, and CNS 0751116, and the Department of the Navy under contract N000.14-09-1-1042.

References

[1] Not just another bogus list. `http://njabl.org`.

[2] Pyzor. `http://pyzor.sourceforge.net/`.

[3] Sorbs DNSBL. `http://www.sorbs.net`.

[4] SpamCop.net - Beware of cheap imitations. `http://www.spamcop.net/`.

[5] The SpamHaus Project. `http://www.spamhaus.org`.

[6] Vipul's razor. `http://razor.sourceforge.net/`.

[7] What is the SpamCop Blocking List (SCBL)? `http://spamcop.net/fom-serve/cache/297.html`.

[8] R. Beverly and K. Sollins. Exploiting transport-level characteristics of spam. In *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS)*, Aug. 2008.

[9] D. Dittrich, M. D. Bailey, and S. Dietrich. Towards community standards for ethical behavior in computer security research. Technical Report 2009-01, Stevens Institute of Technology, Hoboken, NJ, USA, April 2009.

[10] H. Drucker, V. Vapnik, and D. Wu. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

[11] S. Hao, N. A. Syed, N. Feamster, A. Gray, and S. Krasser. Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine. In *Usenix Security '09*, Montreal, Canada, August 2009.

[12] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamalytics: an empirical analysis of spam marketing conversion. In *CCS '08: Proceedings of the 15th ACM conference on Computer and communications security*, pages 3–14, New York, NY, USA, 2008. ACM.

[13] J. Mason. Filtering Spam with SpamAssassin. SAGE-IE meeting presentation, 2002.

[14] McAfee and I. International. The carbon footprint of email spam report. `http://newsroom.mcafee.com/images/10039/carbonfootprint2009.pdf`, April 2009.

[15] T. Micro. Most abused infection vector. `http://blog.trendmicro.com/most-abused-infection-vector/`, December 2008.

[16] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia. Exploiting machine learning to subvert your spam filter. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats*, April 2008.

[17] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*, pages 41–52, New York, NY, USA, 2006. ACM Press.

[18] A. Ramachandran, D. Dagon, and N. Feamster. Can DNS-Based Blacklists Keep Up with Bots? . In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*, July 2006.

[19] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *SIGCOMM '06: Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 291–302, New York, NY, USA, 2006. ACM Press.

[20] A. Ramachandran, N. Feamster, and S. Vempala. Filtering spam with behavioral blacklisting. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 342–351, New York, NY, USA, 2007. ACM.

[21] N. Research and KnowledgeStorm. Nucleus research: Spam costing us businesses $712 per employee each year. `http://nucleusresearch.com/news/press-releases/nucleus-research-spam-costing-us-businesses-712-per-employee-each-year/`, April 2007.

[22] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 55–62, 1998.

[23] S. Sinha, M. Bailey, and F. Jahanian. Shades Of Grey: On the effectiveness of reputation based blacklists. In *International Conference on Malicious and Unwanted Software (Malware 2008)*, October 2008.

[24] B. Stone. Spam back to 94% of all e-mail, March 2009. `http://bits.blogs.nytimes.com/2009/03/31/spam-back-to-94-of-all-e-mail/`.

[25] Unspam Technologies. Project Honey Pot. `http://projecthoneypot.org`, 2008.

[26] 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*, pages 1–18, Berkeley, CA, USA, 2007. USENIX Association.

[27] G. L. Wittel and S. F. Wu. On attacking statistical spam filters. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004. Available: `http://www.ceas.cc/papers-2004/170.pdf`.

[28] R. Wojtczuk. libnids, June 2004.

[29] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are IP addresses? In *SIGCOMM '07: Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 301–312, New York, USA, 2007.

[30] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: signatures and characteristics. *SIGCOMM Comput. Commun. Rev.*, 38(4):171–182, 2008.

[31] J. Zhang, P. Porras, and J. Ullrich. Highly predictive blacklisting. In *17th USENIX Security Symposium (USENIX Security '08)*, July-August 2008.