Shades of Grey: On the effectiveness of reputation-based "blacklists"

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

Abstract

Malicious code, or malware, executed on compromised hosts provides a platform for a wide variety of attacks against the availability of the network and the privacy and confidentiality of its users. Unfortunately, the most popular techniques for detecting and preventing malware have been shown to be significantly flawed [11], and it is widely believed that a significant fraction of the Internet consists of malware infected machines [17]. In response, defenders have turned to coarse-grained, reputation-based techniques, such as real time blackhole lists, for blocking large numbers of potentially malicious hosts and network blocks. In this paper, we perform a preliminary study of a type of reputation-based blacklist, namely those used to block unsolicited email, or spam. We show that, for the network studied, these blacklists exhibit non-trivial false positives and false negatives. We investigate a number of possible causes for this low accuracy and discuss the implications for other types of reputation-based blacklists.

1 Introduction

Current estimates of the number of compromised hosts on the Internet range into the hundreds of millions [17]. Malicious code, or *malware*, executed on these compromised hosts provides a platform for attackers to perform a wide variety of attacks against networks (e.g., denial of service attacks) and attacks that affect the privacy and confidentiality of the end users (e.g., key-logging, phishing, spam) [10]. This ecosystem of malware is both varied and numerous– a recent Microsoft survey reported tens of millions of computers infected with tens of thousands of malware variants in the second half of 2007 alone.

This scale and diversity, along with an increased number of advanced evasion techniques such as polymorphism have hampered existing detection and removal tools. The most popular of these, host-based anti-virus software, is falling woefully behind–with detection rates as low as 40% [11]. Admitting this failure to completely *prevent* infections, defenders have looked at new ways to defend against large numbers of persistently compromised computers and the attacks they perform. One technique becoming increasingly popular, especially in the network operation community, is that of reputation-based blacklists. In these blacklists, URLs, hosts, or networks are identified as containing compromised hosts or malicious content. Real-time feeds of these identified hosts, networks, or URLs are provided to organizations who then use the information to block web access, emails, or all activity to and from the malicious hosts or networks. Currently a large number of organizations provide these services for spam detection (e.g., NJABL [3], SORBS [6], SpamHaus [8] and SpamCop [7]) and for intrusion detection (e.g., DShield [15]). While these techniques have gained prominence, little is known about their effectiveness or potential draw backs.

In this paper, we present a preliminary study on the effectiveness of reputation-based blacklists. In particular we examine the most prevalent of these systems, those used for spam detection. Using an oracle, a spam detector called SpamAssassin [1], we identify the spam received by a large academic network consisting of 7,000 unique hosts, with millions of email messages, over a period 10 days in June of 2008. We examine the effectiveness, in terms of false posi-

tives and negatives, of four blacklists, namely NJABL, SORBS, SpamHaus and SpamCop and provide an investigation into the sources of the reported inaccuracy. While a preliminary study, this work offers several novel contributions:

- **An investigation of email, spam, and spam tool behavior in the context of a large academic network.** We found that roughly 80% of the email messages received by our network were spam. The network level characteristics of spam were also quite different when compared to the observed ham. For example, individual sources contributed significantly to overall ham but the spam was distributed in small quantities across a large number of sources. Conversely, destinations of spam tend to be very targeted when compared to the ham. Using a small number of hand classified email mailboxes, we also evaluated our oracle, SpamAssassin, to be quite effective with less than 0.5% false positives and 5% false negatives for the default threshold.

- **An analysis of the accuracy of four prevalent spam blacklists.** We found that the black lists studied in our network exhibited a large false negative rate. NJABL had a false negative rate of 98%, SORBS had 65%, SpamCop had 35% and SpamHaus had roughly 36%. The false positive rate of all blacklists were low except that of SORBS, which had an overall false positive rate of 10%.

- **A preliminary study of the causes of inaccuracy and a discussion of the issues as they relate to reputation-based services.** We found that while blacklists agree significantly with each other over what is spam, a significant amount (21%) of the spam is not detected by any of these lists, indicating that the blacklists may not have visibility into a significant portion of spam space. Second, we found that many spamming sources that went undetected sent very little spam to our network and that 90% of the undetected sources were observed on the network for just 1 second. This indicates that it is possible that these blacklists are not able to detect these low volume, short lived spammers. Finally, we found that the blacklists rarely agreed with each other on their false positives and that many critical mail servers were blacklisted, especially by SORBS. This included 6 Google mail servers that sent significant amount of ham to our network.

This paper is structured as follows: Section 2 presents background and related work on blacklists and Section 3 presents our approach to evaluating blacklist effectiveness. Section 2 presents a preliminary evaluation of the blacklists and we conclude in Section 5.

2 Related Work

Access control devices like firewalls enforce reputation that is statically decided. In recent years, more powerful dynamic reputation-based systems in the form of blacklists have evolved. A number of organizations support and generate dynamic blacklists. These organizations include spam blacklist providers like NJABL [3], SORBS [6], SpamHaus [8] and SpamCop [7].

Ramachandran and Feamster [13] collected spam by monitoring mails sent to an unused domain and performed a preliminary analysis of spammers. They observed that the spamming sources are clustered within the IP address space and some of these sources are short lived. Instead of collecting spam on a single domain, we monitored all emails on an academic network, both spam and ham, using an accurate detector SpamAssassin.

Spam blacklists providers set up a number of unused email addresses called *spamtraps*. These spamtraps are not advertised to real users but are infiltrated into spammer lists when they scrape the web looking for email addresses. Then source IPs that have sent mails to more than a threshold number of spamtraps are blacklisted. Recently, new blacklist generation techniques have been proposed. Ramachandran *et. al.* [14] argue that blacklisting based on spamtraps is often late and incomplete. They proposed a new method that blacklists source IPs based on their mail sending patterns. DShield [15] aggregates intrusion detection alerts and firewall logs from a large number of organizations. It then publishes a common blacklist that consists of source IPs and network blocks that cross a certain threshold of events. Zhang *et. al.* [20] 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. Xie *et. al.* [19] have shown that a large number of IP addresses are dynamically assigned and mails from these IP addresses are mostly spam. So they recommend adding dynamic IP ranges into blacklists to reduce the false negatives. While these methods may be more effective, we only evaluated production spam blacklists in our study.

A number of papers have questioned the effectiveness of blacklists. Ramachandran *et. al.* [12] analyzed how quickly bobox infected hosts appeared in the Spamhaus blacklists. They found that a large fraction of these hosts were not found in the blacklist. In this paper, we present the overall consequences of such incompleteness of blacklists. Finally, there has been other innovative uses of blacklists. Venkataraman *et. al.* [16] presented a situation where spammers may send a lot of spam to overwhelm a mail server. They proposed using coarse IP based blacklists to reject mails and to reduce server load.

3 Approach

This section presents our approach for the evaluation of reputation based blacklists. We evaluated the blacklists by deploying them in a large academic network of over $7,000$ hosts. We monitored traffic using a traffic tap (i.e., span port) to the gateway router which provides visibility into all the traffic exchanged between the network and the Internet. The TCP streams on port 25 were reassembled using libnids [18]. The data sent by the client constitutes a full SMTP mail that can be used for blacklist evaluation.

However, there is a small problem in this setup. The email that we see is slightly different than the email received on the server. This is because a mail server adds a **Received** header in the email after receiving the email. The received header contains the senders DNS name (or IP address) and the recipient DNS name (or IP address). In order to overcome this problem, we used the source IP address and the destination IP address to fake a **Received** header and added it to each email.

The emails are then fed to a spam detector and the sources in the legitimate received headers are consulted with the blacklists. A number of spam detectors can be used for our study. The two most popular and open source spam detectors are SpamAssassin [1] and DSpam [4]. DSpam requires manual training of individual mail boxes and so we used SpamAssassin in our experimental setup. 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 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 [2] distribution. We configured SpamAssassin with two additional detection modules namely Pyzor [5] and Razor [9] for improving SpamAssassin accuracy.

Blacklist lookups are done by reversing the IP addressing, appending the blacklist zone (eg, combined.njabl.org) and then making a DNS lookup. Remote DNS look ups cause significant latency, which makes evaluation on a large number of emails quite difficult. Therefore, we maintained a local copy of SORBS and NJABL and forwarded DNS queries for SpamHaus (Zen zone) blacklist to a local mirror. SpamCop queries were sent to the actual servers. We used BIND DNS server for these purposes and rbldnsd for serving local blacklists of SORBS and NJABL. The local copies of SORBS and NJABL were refreshed every 20 minutes.

SpamAssassin can itself be erroneous and so we need to first validate the usage of SpamAssassin as an oracle for spam detection. We do this by evaluating false positive and false negative of SpamAssassin on hand classified data sets of ham and spam.

3.1 Validating SpamAssassin

We evaluated SpamAssassin on email mailboxes that were hand classified into spam and ham. Table 3 shows four email accounts that we used for SpamAssassin evaluation. Account #1 contains all spam and ham collected in a work email account for over three years. Account #2 has been used for communicating with open source mailing lists. Account #3 belongs to a separate user who has used it for work and personal use. Account #4 belongs to another user who has used it for personal purposes for a number of years.

Spam-Assassin Threshold	Account #1 ham: 2,019 spam: 11,912		Account #2 ham: 5,547 spam: 107		Account #3 ham: 897 spam: 873		Account #4 ham: 4,588 spam: 482	
	FP	FN	FP	FN	FP	FN	FP	FN
4.0	1.14	4.17	0.25	3.08	0.89	3.67	0.76	5.39
4.5	0.84	4.47	0.02	3.08	0.56	3.78	0.61	5.60
5.0	0.45	4.88	0.02	4.02	0.56	4.24	0.50	5.60
5.5	0.30	5.80	0.02	4.02	0.45	5.27	0.22	6.22
6.0	0.25	6.06	0.02	4.02	0.33	6.41	0.11	6.85

Table 1. The false positive and false negative rates for SpamAssassin (at different thresholds) on four mail accounts that were manually sorted into spam and ham. Overall, SpamAssassin performs well.

A message is a false positive for SpamAssassin if the message is ham and the SpamAssassin score for the message is greater than the given threshold. On the other hand, a message is a false negative for SpamAssassin if the message is spam and the SpamAssassin score is less than the threshold. The false positive rate is then computed as the ratio of false positives to the number of ham. The false negative rate is computed as the ratio of false negatives to the number of spam.

Table 3 shows the false positive rate and false negative rate of Spam Assassin on the four email accounts. We find that the false positive rate for SpamAssassin is very small and is close to 0.5% for a threshold of 5.0 (the default threshold in SpamAssassin). On the other hand, SpamAssassin has false negative rates of around 5%. Overall, SpamAssassin has very few false positive with manageable false negatives.

4 Evaluation

We deployed the entire system on an academic network for a period of around 10 days in June 2008. Figure 1 shows the number of mails per hour observed on the network. On an average, we observed 8,000 SMTP connections per hour. However, half of these SMTP connections were aborted before the actual mail was transferred. This is because many mail servers in our network were configured to reject a mail if the recipient was not a valid user in the domain. Spam and ham were separated using SpamAssassin and the rate of spam was significantly higher than the ham. In what follows we first present the characteristics of spam and ham observed on the network, then present the results on blacklist effectiveness, and finally conjecture and evaluate possible reasons on the false negatives and the false positives of the blacklists.

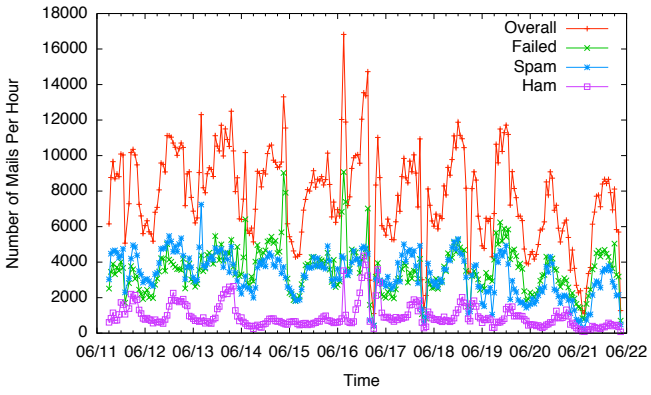

Figure 1. Number of mails per hour observed on the academic network. The overall mail rate is further divided by ham, spam, and failed connections.

4.1 Email characteristics

Over the period of our experiment, we found that a total of 1,074,508 emails were successfully delivered. Figure 2 shows the SpamAssassin score distribution for those mails. We find that roughly 15% of the mails received a score of 0 and around 20% of the mails were below the SpamAssassin threshold of 5.0. Over 70% of the mails received a score of more than 10.

Then we looked at the email sources and destinations. We observed a total of 53,579 mail destinations with 64 of them within the academic network. Overall, we saw 609,199 mail sources with 111 within the academic network. Figure 3 shows the distribution of ham and spam by their sources and destinations. While

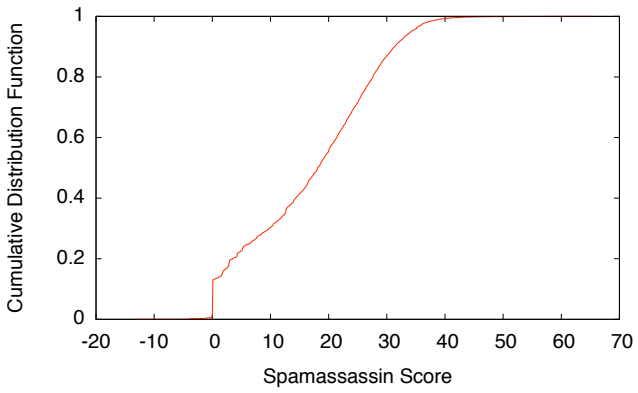

Figure 2. Cumulative distribution of SpamAssassin score for successfully delivered mail on the network (total = $1,074,508$).

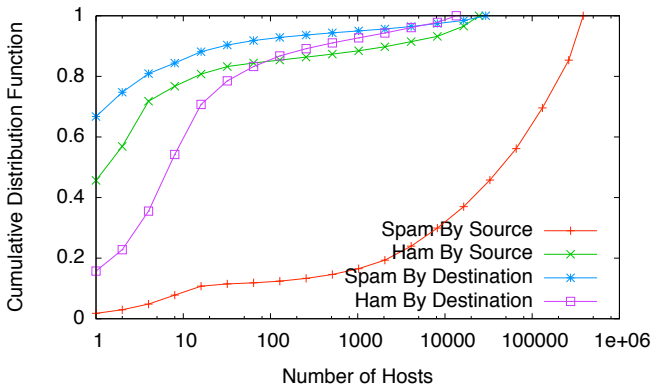

Figure 3. The source IP distribution and the destination IP distribution for spam and ham.

spam was distributed across a large number of sources, the ham was concentrated to a very few sources. For example, while the top 10 hosts covered 80% of ham, the top 10 spamming sources covered less than 10% of spam. On the other hand the targets of spam were very concentrated when compared to ham. For example, while the top 10 destinations covered 80% of the spam, the top 10 destinations covered only 50% of ham. Overall, we find that the spam is well distributed across a large number of sources but targeted towards a few destinations. This is quite in contrast to the network level behavior of ham.

4.2 Blacklists effectiveness

We now evaluate the false positive and false negative rates of four blacklists namely NJABL, SORBS (all zones), SpamCop (main zone) and SpamHaus (Zen zone). Table 4.2 shows the false positive rate of the four blacklists for different SpamAssassin thresholds. First, we find that the NJABL has the least false positives followed by SpamHaus. Second, the false positive rate of SpamCop and SpamHaus increases significantly when the SpamAssassin threshold is increased from 5.0 to 5.5. This indicates that the blacklists were positive for a number of messages that received the overall SpamAssassin score between 5.0 and 5.5. Finally, we look at unique source IPs for determining the false positive and false negative rates. We find that the false positive rates for unique source IPs are significantly higher when compared to the overall false positive rates. For example, SORBS has an overall false positive rate of 9.5%, but when unique source IPs are considered the false positive rate increases to 26.9%. Overall, we find that SORBS has unreasonable amount of false positives but the other blacklists have few false positives.

Table 4.2 shows the false negative rates of the four blacklists for different SpamAssassin thresholds. While NJABL had a very few false positives, it has a huge false negative. For a threshold of 5.0 the false negative rate is 98.4%. SpamCop has the smallest false negative rate at around 36.3%. While the SpamAssassin threshold significantly impacted the false positive rate, its impact on the false negative rate is quite small. The false negative rates are around 59% for SORBS, 35% for SpamCop and 36% for SpamHaus. Overall the blacklists seem to have significantly higher false negative than we expected.

4.3 Exploring blacklist false negatives

It is difficult to come up with reasons behind the large false negative rates of the blacklists because we do not know have access to the spamtrap deployment, and we do not know the precise algorithm used for blacklisting. However, we will look at characteristics of spam messages that the blacklists missed and infer possible causes. We look at two possible causes: lack of visibility and the possibility of low volume or low rate spammers.

4.3.1 Wide visibility

One possible reason may be that the blacklists do not have visibility into the spamming sources. In order to

SpamAssassin Threshold	NJABL		SORBS		SpamCop		SpamHaus	
	total	source IP	total	source IP	total	source IP	total	source IP
4.0	0.1	0.3	9.4	24.8	1.5	8.9	0.5	4.6
4.5	0.1	0.4	9.2	25.6	1.8	11.4	0.5	4.5
5.0	0.2	0.5	9.5	26.9	2.3	13.6	0.6	5.2
5.5	0.2	0.5	10.3	28.0	5.7	26.7	4.0	19.6
6.0	0.2	0.5	10.6	29.1	6.3	28.6	4.5	21.3

Table 2. False positive rate in percentage (overall and unique source IPs) for four different blacklists.

SpamAssassin Threshold	NJABL		SORBS		SpamCop		SpamHaus	
	total	source IP	total	source IP	total	source IP	total	source IP
4.0	98.4	98.1	65.4	59.2	36.4	40.4	38.0	41.4
4.5	98.4	98.1	64.9	59.2	35.4	40.3	36.9	41.2
5.0	98.4	98.1	64.8	59.2	34.9	40.2	36.3	41.0
5.5	98.4	98.1	64.5	59.1	34.7	40.2	36.2	41.0
6.0	98.4	98.1	64.4	59.1	34.5	40.1	35.9	40.8

Table 3. False negative rate in percentage (overall and unique source IPs) for the blacklists. Blacklists have a small false positive rate, but a large false negative rate.

evaluate the coverage of different blacklists, we computed the number of times different blacklists agree on a spam. Figure 4 shows the percentage of spam detected by different blacklists and their mutual overlap. NJABL has been omitted because of its low detection rate. Surprisingly we find that the blacklists agree on a large number of spam. For example, SpamHaus and SpamCop agree on 57% of the spam, SORBS and SpamCop agree on 26% of the spam, and SORBS and SpamHaus agree on 24%. All three agree on 21% of the spam. The exclusive detection rate for the blacklists is small: 4.5% for SpamHaus, 3.8% for SpamCop and 6.8% for SORBS. This implies that the spamtrap deployment for individual blacklists may overlap significantly and may not be diverse enough to capture the remaining 21% of the overall spam.

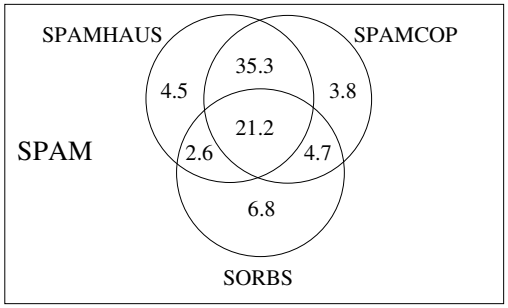

Figure 4. A venn diagram showing the overlap in blacklists for correctly flagged spam (overlap in true positives). There is a significant overlap among the blacklists.

4.3.2 Low volume/short lived spammers

Apart from visibility, another reason that a blacklist may miss spam is because of low volume or short lived spammers. Figure 5 shows the number of spam sent by sources external to the network that did not hit any blacklist. We found that just 100 out of 67,442 such sources sent 20 or more spam to our network. This means that many spamming sources that the blacklists missed may be actually low volume spammers. We then looked at the time interval they were observed on the network. We find that 80% of these sources were observed just for a second, a potential reason they escape blacklisting.

4.4 Exploring blacklist false positives

We earlier observed that the blacklists have a small false positive rate. However, false positive rates for SORBS were significantly higher than the other black-

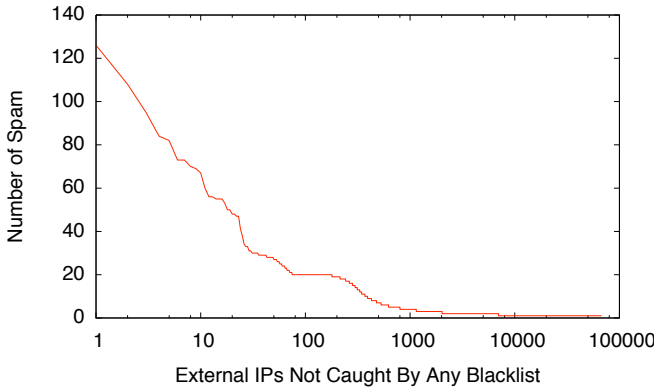

Figure 5. Spam missed by blacklists (false negatives) binned by source IPs external to the network. Most sources sent very few spams to our network.

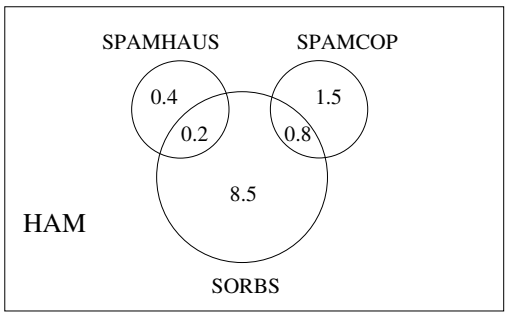

Figure 6. A venn diagram to show the overlap in blacklists in incorrectly flagging ham as spam (overlap in false positive). The blacklists rarely agree on these email messages.

lists. Now we examine two possible reasons behind false positives of the blacklist. The first one is whether SpamAssassin is itself wrong and the blacklists are correctly pointing out the spam. Second, it is likely that prominent mail servers shared by legitimate and illegitimate people are getting blacklisted and ham from these servers are classified as spam by the blacklists.

4.4.1 Errors in SpamAssassin

While validating SpamAssassin we found that SpamAssassin has around 5% of false negatives. So it is likely that the blacklists may be correctly pointing out spam and they are actually false negatives of the SpamAssassin. We checked if the blacklists themselves agree on the false positive, a strong indication that it is a false negative of SpamAssassin. Figure 6 shows the overlap among blacklists for false positives with respect to SpamAssassin. While blacklists do not agree with SpamAssassin for a small number of mails, the blacklists disagree with each other on most false positives.

4.4.2 Aggressive blacklisting

Another possible reason for the false positives of the blacklist is that a mail server shared by legitimate and illegitimate users is blacklisted. If this is the case, then many ham sent by a mail server will be incorrectly flagged by the blacklist. In order to assess this, we aggregated ham that were incorrectly classified by the blacklists. Figure 7 shows the number of ham incorrectly classified by the blacklists and binned by the source IP. First, we find that most of these sources have sent very few ham to our network. Second, NJABL, SpamHaus and SpamCop do not seem to have blacklisted any mail servers. However, SORBS has blacklisted hosts that have significant amount of ham to our network. When we looked at those hosts, we found that five of these hosts are Google mail servers within a /16 and another Google mail server in a separate address block.

While determining the motive behind blacklisting Google mail servers is beyond our scope, we did a short test on three different mail services, namely - Yahoo Mail, Gmail and AOL Mail. If an email is sent through the web interface to Yahoo or AOL mail, we find that these services append the IP address of the sender in the **Received** mail header. So a blacklisting service can choose to blacklist only the IP rather than the mail server itself. Gmail on the other hand does not include the IP address of the sender if one uses the web interface. However, if email is sent through the IMAP interface to Gmail, then the IP address is included in the **Received** header. While refusing to include IP address of the sender may be a reason for blacklisting the entire mail server, we are in no way certain about the real reasons for their blacklisting.

5 Conclusion

The Internet is routinely threatened from a large number of compromised hosts distributed all over the

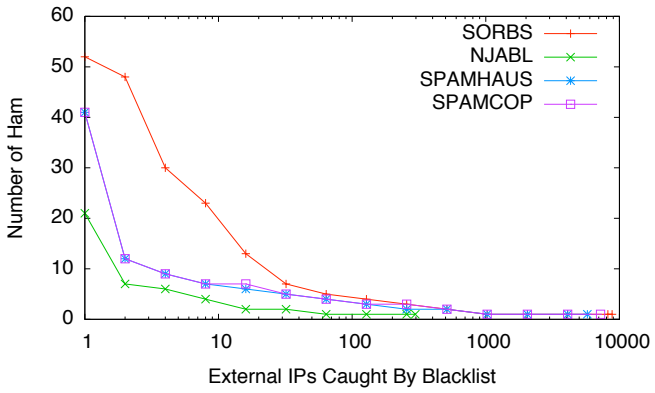

Figure 7. Ham flagged by blacklists (false positives) binned by source IPs external to the network. Six Gmail servers that sent ham to our network are blacklisted by SORBS.

world. A large number of commercial and academic efforts have been made into the detection of malware resident on these hosts. However, the increasing complexity and sophistication of malware have made such efforts increasingly difficult. As a result, defenders are increasingly relying on reputation based blacklists to detect and mitigate new threats. However, little is known about the benefit and the collateral damages of these blacklists. This paper presented a preliminary evaluation of four popular blacklists on an academic network with more than 7,000 hosts. The blacklist evaluation was performed over a period of 10 days on more than a million messages. We found that the blacklists have significant false negative rates and a higher than expected false positive rate. Our analysis of false negatives indicated that the blacklist may not have visibility into a large number of spam. Further, they may not be able to detect low volume spammers and may be late in reacting to them. Our analysis of false positives indicated that blacklists may contain prominent mail servers that are shared with legitimate as well as illegitimate users.

Acknowledgments

This work was supported in part by the Department of Homeland Security (DHS) under contract numbers NBCHC060090 and NBCHC080037, and by the National Science Foundation (NSF) under contract number CNS 0627445. We thank David Watson for providing valuable feedback on the draft and reviewers for useful comments.

References

[1] The apache spamassassin project. `http://spamassassin.apache.org/`.

[2] Gentoo linux. `http://www.gentoo.org/`.

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

[4] Nuclear Elephant: The DSPAM Project. `http://www.nuclearelephant.com`.

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

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

[7] spamcop.net - beware of cheap imitations. `http://www.spamcop.net/`.

[8] The spamhaus project. `http://www.spamhaus.org`.

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

[10] Arbor Networks. Worldwide infrastructure security report, Sept. 2007.

[11] Michael Bailey, Jon Oberheide, Jon Andersen, Z. Morley Mao, Farnam Jahanian, and Jose Nazario. Automated classification and analysis of internet malware. In *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection (RAID'07)*, September 2007.

[12] Anirudh Ramachandran, David Dagon, and Nick Feamster. Can dns-based blacklists keep up with bots? In *CEAS*, 2006.

[13] Anirudh Ramachandran and Nick 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.

[14] Anirudh Ramachandran, Nick Feamster, and Santosh 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.

[15] Johannes Ullrich. DShield. `http://www.dshield.org`, 2000.

[16] Shobha Venkataraman, Subhabrata Sen, Oliver Spatscheck, Patrick Haffner, and Dawn Song. Exploiting network structure for proactive spam mitigation. In *Proceedings of 16th USENIX Security Symposium*, pages 1–18, Berkeley, CA, USA, 2007. USENIX Association.

[17] Tim Weber. Criminals may overwhelm the web. `http://news.bbc.co.uk/1/hi/business/6298641.stm`, January 2007.

[18] Rafal Wojtczuk. libnids, June 2004.

[19] Yinglian Xie, Fang Yu, Kannan Achan, Eliot Gillum, Moises Goldszmidt, and Ted 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.

[20] Jian Zhang, Phillip Porras, and Johannes Ullrich. Highly predictive blacklisting. In *Usenix Security Symposium*, 2008.