Taking on Internet Bad Neighborhoods

Giovane C. M. Moura*, Ramin Sadre†, and Aiko Pras‡
* Delft University of Technology
Email: g.c.moreiramoura@tudelft.nl
† Aalborg University
Email: rsadre@cs.aau.dk
‡University of Twente
Email: a.pras@utwente.nl

***Abstract*—It's known fact that malicious IP addresses are not evenly distributed over the IP addressing space. In this paper, we frame networks concentrating malicious addresses as bad neighborhoods. We propose a formal definition and show this concentration can be used to predict future attacks (new spamming sources, in our case), and propose an algorithm to aggregate individual IP addresses can bigger neighborhoods. Moreover, we show how bad neighborhoods are specific according to the exploited application (e.g., spam, ssh) and how the performance of different blacklist sources impacts lightweight spam filtering algorithms.**

I. Introduction

The impact of malicious activities on the Internet, such as spam, phishing and distributed denial-of-services (DDoS) extrapolates the Internet borders: it is estimated that losses caused by spam are in the magnitude of US$ 20 billion, only for the United States [1]. More recently, the blacklists provider SpamHaus suffered the biggest DDoS ever observed on the Internet, peaking at 300 Gbps, causing degradation on the performance of many networks all over the world [2].

Such attacks are typically carried out by a large number of distributed computers, usually part of botnets, which are networks of compromised machines (computers at home, schools, businesses) under control of a botmaster [3]. These bots (or zombies) can be found all over the world; however, they tend to be concentrated in certain networks instead [4]. For example, Figure 1 shows the concentration of IP addresses per /8 prefix (or netblock, in CIDR notation [5]) that have spammed Provider A, a major hosting provider in The Netherlands, on April 5th, 2011.

This concentration of malicious addresses resembles actual crime distribution in the real world: it occurs in many places, but tends to be concentrated in certain areas, which are sometimes labeled as "bad neighborhoods". Analogously, on the Internet, malicious activities are statistically more likely to be originated from networks that concentrate most of malicious hosts. Taking this into account, van Wanrooij *et al.* have introduced the term *Internet Bad Neighborhoods* [6] to /24 prefixes having been observed sending spam, and employed it in a spam filter that evaluates suspicious URLs and IP source addresses in individual spam messages.

Even though the idea behind bad neighborhoods (BadHoods hereafter) was employed in [6], the very concept was not proposed or even scrutinized. That then led to the Ph.D. dissertation of one of the authors [7].

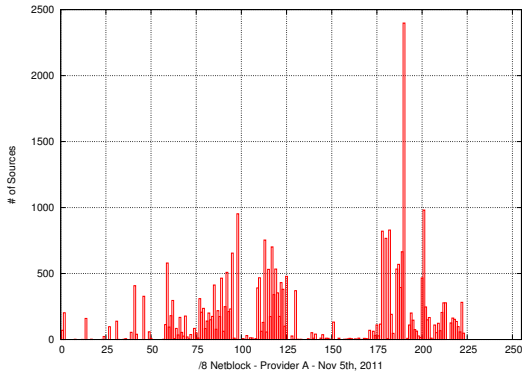


Fig. 1. Number of spam sources per /8 netblock

The motivation to carry out this investigation is to unravel patterns and characteristics associated with BadHoods that can be employed to better secure networks, by means of improved attack prediction, lightweight spam filtering [6], [8] and, additionally, to incentivize botnet mitigation initiatives. The approach employed in the dissertation and in this paper consists in analyzing various datasets from real world production networks.

In this paper, we put the Internet BadHoods under scrutiny. In Section II, we present our definition for Internet BadHoods and illustrate how it can statistically be used to predict attacks. Then, in Section III, we propose and evaluate an algorithm to aggregate /32 IP addresses into BadHoods of various sizes (/24–/8), while in Section IV we evaluate whether BadHoods are application-specific. Next, in Section V, we cover the performance of different BadHood blacklists in spam filtering. Finally, our findings are summarized in SectionVI.

II. Defining Internet Bad Neighborhoods

We define an *Internet Bad Neighborhood* as a set of IP addresses clustered according to an **aggregation criterion** in which **a subset of IP addresses** perform **a certain malicious activity** over a specified **period of time**.

In this definition, *aggregation criterion* stands for the basic mechanism used to cluster malicious IP addresses into Bad Neighborhoods (e.g., by network prefixes, autonomous system numbers (ASN), countries, etc.). A *certain malicious activity*, in turn, refers to the application that the bad neighborhood is abusing or conducting attacks on (e.g., spam, phishing). Finally, *period of time* refers to the time frame used to define a

bad neighborhood (e.g, day, weeks), which is important since bad neighborhoods are expected to change over time – since machines are expected to get compromised and cleaned up regularly.

It is important to emphasize that the malicious IP addresses might not be the one of the real attackers, which, in fact, might employ a series of intermediate computers to hide their own identity [9]. We focus on the *attribution of the last host* in the logical path of the attacks. As a consequence, hosts flagged as malicious might not represent the behavior of the host's owners, who actually might be unaware that their computer is involved in such attacks (the ethical issues related to this research were both covered in the dissertation as well as in [10]). We choose to focus on the attribution of the last host because we assume the point of view of a network administrator who wants to protect a network from malicious sources. For him/her, knowing the identity of the attacker does not help to better protect the network he/she maintains, since blocking traffic from the attacker IP address to the network the administrator maintains does not stop spam messages. Also, tracing back the original attacker may involve different ISPs in different countries, an effort that currently is far from being done in real-time. In contrast, we see the attribution of the responsible attacker as a task of law enforcement agencies instead.

Verifying the BadHood Assumption

Previous research works have shown that malicious IP addresses tend to be concentrated in certain networks. For example, in 2006 Ramachandran *et al.* [11] showed that the majority of spam was sent from a small fraction of the IP address space. Collins *et al.* [4], on the other hand, have defined the term "spatial uncleanliness" for clusters of compromised hosts.

Essentially, the BadHood concept provides an indirect approach to predict new sources of attacks, by e*xtending the reputation of malicious IP addresses to their neighboring ones*, assuming that neighboring hosts are more likely to be malicious as well and, therefore, more likely to carry out attacks. In this subsection, we carry out an experiment to verify this assumption.

We consider a simplistic mail filter that classifies a message as spam *if the sender IP address* is listed in a previously obtained blacklist (`if SenderIP ∈ Blacklist then SPAM`). For this mail filter, we need two inputs: the `SenderIP` address of the mail and a `Blacklist`. In this experiment, we obtained the sender IP addresses from the mail servers of the Electrical Engineering, Mathematics, and Computer Science Faculty of the University of Twente (UT/EWI). In total, 3,198,936 messages were classified by the mail filter `SpamAssassin` [12] as spam and used as ground truth.

We have then evaluated the performance of four blacklists in detecting spam, as shown in Figure 2. `CBL32-STD` curve shows the performance of the Composite Blacklist (CBL) [13], a publicly available spam blacklist used in many mail filters. As can see, by employing this blacklist, our simplistic mail filter was to able to filter, on average, 54.33% of all the spam (we match the spammmers of UT/EWI of a n day to CBL of day

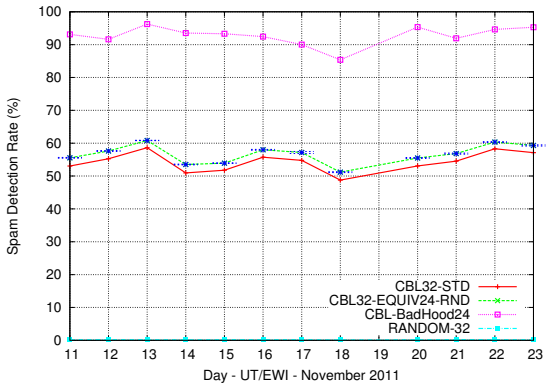


Fig. 2. Performance of various blacklists

$n-1$). One important remark is that UT/EWI's `SpamAssassin` does not employ CBL in the message classification.

To verify whether the BadHood concept provides an advantage over traditional /32 blacklists (*i.e.,* if BadHood-based blacklists are able to predict new sources), we have then generated a second blacklist – `CBL-BadHood24` – which consists of the entire `CBL32-STD` blacklist plus all its /24 neighboring IP addresses. For example, if 10.10.10.4 was listed in `CBL32-STD`, we consider its entire /24 prefix as malicious (10.10.10.0–10.10.10.255, or 10.10.10/24) in `CBL-BadHood24`. As can be seen in Figure 2, `CBL-BadHood24` blacklist provides a much better performance, delivering, on average, 92.74% spam detection.

One may argue that this result was expected, since `CBL-BadHood24` blocks many more IP addresses than `CBL32-STD` (up to 256 times more, due to /24 aggregation [14]). To verify this, we generated a third blacklist – `CBL32-EQUIV24-RND`. Instead of blacklisting neighboring hosts from `CBL32-STD`, we blacklisted *randomly* chosen IP addresses. The total number of blacklisted addresses is then the same as in `CBL-BadHood24` (we have created 10 random blacklists and show the average results, standard deviation shown as well). As can be seen in Figure 2, `CBL32-EQUIV24-RND` blacklist performs far worse than `CBL-BadHood24`, delivering an average performance of 56.64% spam detection. That means that even though both blacklists contain an equivalent number of /32 entries, *blocking randomly chosen hosts will not significantly improve spam detection*, while blocking *neighboring IPs* (leveraging the network reputation) does. A fourth blacklist confirms this result: `RANDOM-32`, which is a /32 blacklist that has the same number of entries as `CBL32-STD`, yields to almost 0% detection.

III. From IP addresses to Bad Neighborhoods

In the previous section, we have defined Internet BadHoods and shown how they can be used to predict attacks. In order to use BadHood-based blacklists efficiently in real intrusion detection systems (IDS) or filters, it is often required to keep such blacklists as short (in number of entries) as possible. In this section, we present an algorithm for BadHood aggegration called variable-prefix aggregation and study its performance using real world security data (we refer to [14] for a complete analysis of BadHood aggregation). The principle behind the algorithm is analogous to the reduction of entries in routing tables by Classless Inter-Domain Routing (CIDR) [5].

A. BadHoods Evilness Metrics

Typical input data for identifying Intenet BadHoods are lists of individual IP addresses which have performed malicious activities [15]. They can be obtained from defense and monitoring mechanisms, such as intrusion detection systems (IDSs) [16], [17] or honeypots. In some cases, third parties provide blacklists containing IP addresses of malicious hosts. A major example are DNS Blacklists [18], built by harvesting spamming IP addresses using *spamtraps* (specialized honeypots to collect spam) distributed over different domains, such as CBL [13], and PSBL [19] and Spamhaus [20].

Given a list of malicious IP addresses (/32), we define a $/n$ BadHood as a $/n$ netblock (or prefix) A^n with a score $score(A^n)$, where the score is the number of malicious hosts in the block:

$$score(A^n) = \#\{\text{malicious hosts in block } A^n\} \quad (1)$$

The score value leads to an intuitive definition of the "evilness" of a netblock: the higher the score, the higher the probability that a single /32 host address from the $/n$ block is a source of malicious activities. We define the *infection rate* of A^n as

$$p_n(A^n) = \frac{score(A^n)}{max\_hosts(A^n)}, \quad (2)$$

where $max\_hosts(A^n) = 2^{32-n}$ is the maximum number of IP addresses in a $/n$ netblock (neglecting the addresses reserved for broadcasting and network identification).

The starting point for the aggregation algorithm described in section III-C will be /24 BadHoods. This prefix size was already used by us in [15] to build BadHoods on spammer sources. The reason for this is the fact that /24 is the minimum prefix "routable on the Internet" [21]. Table I provides a short example of BadHoods, showing their /24 address and score.

#	/24 netblock	Score
1	10.10.10.0	22
2	10.10.11.0	21
3	10.10.12.0	20
4	10.10.13.0	41
5	20.20.24.0	130
6	20.20.25.0	1
7	30.30.34.0	60

TABLE I. EXAMPLE OF /24 BADHOODS AND THEIR SCORES

B. Basic Aggregation Operation

Two $/n$ BadHoods A^n and B^n can be aggregated into the $/(n-1)$ BadHood $A^n \oplus B^n$ only if A^n and B^n have a common address prefix of $n-1$ bits. The aggregated BadHood $A^n \oplus B^n$ spans the IP addresses of A^n and B^n. For example, in Table I, blocks #1 and #2 can be aggregated from /24 to /23, while blocks #1 and #7 can not.

Consequently, the infection rate of the aggregated BadHood $A^n \oplus B^n$ is as follows:

$$p_{n-1}(A^n \oplus B^n) = \frac{score(A^n) + score(B^n)}{max\_hosts(A^n \oplus B^n)} = \frac{1}{2}\left(p_n(A) + p_n(B)\right). \quad (3)$$

C. Variable Prefix Aggregation

The main idea is to merge two BadHoods only if they satisfy a *merging condition*. Intuitively, the merging condition should ensure that the BadHoods to be merged are sufficiently similar and, therefore, the aggregated BadHood is, to some extend, representative for them.

Algorithm 1 presents the pseudocode for the proposed aggregation strategy. The algorithm takes as input the initial list S_{24} of /24 netblocks B_i^{24} with $score(B_i^{24})$ and the largest desired aggregation level m. Then, for each aggregation level n (line 2), the algorithm merges all $/n$ BadHoods B_i^n, B_j^n which would form a valid aggregated BadHood according to the basic aggregation operation (see Section III-B) that satisfy the merging condition *merge* (line 3). BadHoods that do not fulfill those conditions are not aggregated and therefore not considered further for aggregation in this or the next iterations.

Algorithm 1 Variable prefix aggregation

Input: $S_{24} = \{(B_i^{24}, score(B_i^{24})), i = 1 \ldots num\_entries\}$
Input: largest aggregation level m
Input: merging condition parameter β
Output: S

1: $S := S_{24}$
2: **for** $n = 24 \rightarrow m+1$ **do**
3: **for all** $B_i^n, B_j^n \in S, i \neq j$ with common $n-1$ prefix $\wedge$ $merge(A^n, B^n)$ **do**
4: $S := S \setminus \{(B_i^n, score(B_i^n)), (B_j^n, score(B_j^n))\} \cup \{(B_i^n \oplus B_j^n, score(B_i^n \oplus B_j^n))\}$
5: **end for**
6: **end for**

The merging condition is defined as

$$merge(A^n, B^n) = p_{n-1}(A^n \oplus B^n) \geq \beta \cdot \max(p_n(A), p_n(B)). \quad (4)$$

The condition is such that we allow a merge only if the resulting infection rate $p_{n-1}(A^n \oplus B^n)$ is at least equal to a fraction β of the rate of the most malicious of the blocks to be merged. The parameter β prevents therefore the aggregation strategy from merging dissimilar BadHoods. This value can be tuned according to the scenario and application. β ranges between 0.5 and 1.0: smaller values make the aggregation less strict, thus allowing more BadHoods to be merged. Values close to 1 will instead lead to a less permissive aggregation strategy.

Finally, at line 4, the algorithm progressively builds the new BadHood set by removing BadHoods and replacing them with the merged one. To illustrate the strategy, we apply it to the example given in Table I. For $\beta = 0.8$, we merge blocks #1 and #2, because $p_{24}(\#1) = \frac{22}{254}$, $p_{24}(\#2) = \frac{21}{254}$, and $p_{23}(\#1 + \#2) = \frac{43}{510}$, so $p(\#1+\#2) > 0.8 \cdot max(\cdot) \Rightarrow 0.086 > 0.069$. The other blocks, on the other hand, do not match the condition, so they are not aggregated. After the first iteration, the list contains both /23 and /24 entries. In the next iterations, no further aggregation occurs, and the final result contains entries using mixed prefixes (/23 and /24).

D. Experimental Results

We now discuss the performance of the variable prefix aggregation strategy. We have applied it to the following datasets:

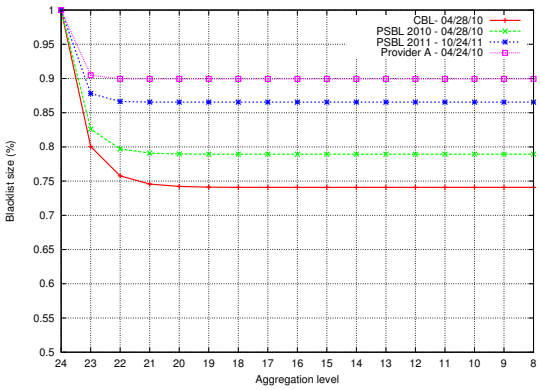


Fig. 3. Performance of the variable prefix aggregation strategy (β=0.8)

- Composite Blocking List (CBL) [13] – an online Spam DNS blacklist. CBL maintains four large spamtrap infrastructures from where the source IP addresses of spammers are harvested. We have obtained the list for the April 28th, 2010. On this day, CBL listed 8,177,138 /32 IP addresses.
- Passive Spam Block List (PSBL) (2010) [19], obtained on April 28th, 2010: the list consists of more than 2.8M /32 distinct IP addresses;
- Passive Spam Block List (PSBL) (2011) [19], obtained on October 24th, 2011: the list consists of more than 283K /32 distinct IP addresses;
- Mail server logs from Provider A: Provider A is a major hosting provider in the Netherlands. We have obtained the IP addresses of spammers on April 28th, 2010, having $\sim$ 256K distinct /32 IP addresses.

Figure 3 shows the number of lines of the result blacklists relative to the original sizes of the /24 data sets, as computed by the variable prefix aggregation for varying aggregation level and $\beta = 0.8$. We observe that our aggregation strategy is able to reduce the blacklist size for each of the considered data sets. For $\beta = 0.8$, the data sources experience a reduction on the number of entries from 10% for the Provider A data set to 26% for the CBL.

A second observation is that the two largest lists, namely CBL and PSBL 2010 (April 28th), clearly benefit more from the aggregation than the smaller lists. This is expected because the BadHoods in the smaller lists are more sparsely distributed over the Internet address space and, hence, are harder to aggregate. In addition, the Provider A data set experiences the smallest reduction of all four traces. By having 0.8 for β, we could reduce the number of entries by 10% to 26%, depending on the data source. However, the best value for β remains an application- and management-dependent decision.

We have also covered other aspects of the aggregation strategy in [14]. One of the most important aspects is the aggregation error, i.e., the fact that the infection rate of the aggregated BadHood $A^n \oplus B^n$ is different from the rate of the individual BadHoods A^n and B^n. Interested readers are referred to [14] for a detailed analysis. Alternatively, other aggregation criteria can be employed, such as country, Autonomous System, which can be also found in the dissertation.

IV. BadHoods and Exploited Applications

In the real world, some bad neighborhoods are known for a certain type of crime incidence (e.g., car theft, robbery, etc.), but not for all of these crimes at the same time. In this section, we investigate whether the BadHoods are the same ones for different types of attacks on the Internet. The motivation is to avoid carrying out unnecessary network measurements: If we find that the same set of Internet BadHoods are responsible for different types of attacks (e.g, spam, SSH attacks, etc), we could then avoid having to generate application-tailored BadHood blacklists and employ the currently available ones to protect targets running different applications.

To proceed with that, we first chose a variety of publicly available data sets covering several applications. For the three data sets, we have collected data for a one week period (November 11th to 18th, 2011). Then, we have generated a single list of /32 IP addresses for each data set. Subsequently, each list was aggregated into a /24 BadHood blacklist. The evaluated data sets are:

- CBL [13], as described in Section III.
- Phishtank: Phishtank is an open community web site in which anyone can "submit, verify, and track phishing websites" [22]. It provides a blacklist of URLs that contain forged websites. The URLs were resolved to IP addresses using Google Public DNS [23]. In case of a URL was resolved to multiple IP addresses, we have considered all of them.
- DShield: DShield [24] is a community shared firewall log system. Volunteers submit their firewall logs from more than 600 contributors, which encompass more than "500,000 IP addresses (firewalls) in over 50 countries" [25]. It is maintained by the SANS Institute [26], and contains security logs from many applications.

Since DShield provides data for more than 100K types of applications, we chose a subset of these for our analysis. We have ranked the most frequently attacked applications (`Port` and `Proto` fields) in terms of number of attacking IP addresses. In addition, many entries did no list any protocol and others used high port numbers (unassigned). Therefore, we have focused only on attacks on the "well-know ports" (port number < 1024, according to IANA terminology and the list [27]) that have the protocol field (`Proto`) different from `NULL`. By filtering out such entries, we filter out potential false positive entries found in the DShield data set, and focus on the most repeated ones. Table II shows the Top 10 ports according to these criteria.

From the top 10 ports shown in Table II, we have chosen the top 5 ports to carry out our experiments (excluding Telnet[1]), plus a high port having most of the attacks (5559). Therefore, six ports from DShield were chosen: TCP 445 (T-445), UDP 5559 (U-5559), TCP 25 (T-25), TCP 443 (T-443), TCP 80 (T-80), and UDP 53 (U-53).

[1]We have deliberately excluded Telnet since this application should have already been phased out and replaced by SSH. In addition, it does not make much sense protecting an application that is intrinsically vulnerable, since no encryption is employed and credentials are transmitted in clear. See more in http://www.networkworld.com/news/2011/012711-hackers-turn-back-the-clock.html

# of /32 IPs	Dst Port	TCP/UDP	Description
553,139	**445**	**TCP**	**Windows/Samba shares**
40,498	**25**	**TCP**	**S (SMTP)**
28,293	**443**	**TCP**	**https**
16,624	**80**	**TCP**	**http)**
11,164	23	TCP	Telnet
8,979	**53**	**UDP**	**DNS**
4,517	161	UDP	(SNMP)
4,469	137	UDP	NetBIOS
3,722	22	TCP	SSH
3,401	80	UDP	unassigned

TABLE II. TOP 10 PORTS < 1024, PROTOCOL "NOT NULL"

Dataset	# /32 IPs	# /24 BadHoods
CBL	13,668,909	1,123,492
PSBL	3,301,159	714,466
Provider A	1,498,991	522,522
UT/EWI	377,571	228,445

TABLE IV. TRAINING DATA SETS, APRIL 19TH–25TH, 2010

Dataset	# /32 IPs	# /24 BadHoods	# spam
Provider A	296,596	206,980	879,856
UT/EWI	68,748	59,739	221,179

Dataset	# /32 IPs	# /24 Hoods	# Ham
HAM: UT/EWI	1,540	978	7,950

TABLE V. TEST DATA SETS, APRIL 26TH, 2010

In order to evaluate whether the same BadHoods are active for different applications, we perform an intersection ($\cap$) operation between the different datasets. Table III shows the results. Note that, for two blacklists, we only compare the one which has observed less BadHoods (row) to the one which has observed more BadHoods (column), since we want to compare what is the intersection of a smaller BadHood blacklist to a bigger one. In Table III, we show the number of BadHoods that were found intersecting between two applications; the percentage values refer to the total number of matching BadHoods divided by the number of entries observed by the list specified in the row. As an example, consider the second row and second column. It is to be interpreted as follows: of all BadHoods that have attacked using UDP Port 5559 (U-5559), 29.8% were also found attacking TCP 445 application (T-445).

Analyzing this table, we can observe that, for only two cases (U-5559 and T-25, both against CBL) we have an intersection rate above 90% (relative to U-5559 and T-25 data sets sizes). That means that more than 90% BadHoods that carry out attacks on port 5559 and on port 25 also carry out spam attacks (we would expect such a high rate for T-25, since it monitors the default SMTP port), however, port UDP 5559 is not assigned by IANA, which means no official application is supposed to run on this port.

However, for the rest of the applications, we can see the matching rate between any two data sets is below 51%, being the majority below 30%. These are very low values if one intends to use BadHood blacklists from one application to secure another application. Therefore, we can conclude that, for most of the cases, the BadHoods attacking two different applications differ, and *therefore it is necessary to carry out measurements for distinct applications*.

V. BLACKLIST SOURCES AND SPAM FILTERING PERFORMANCE

In the previous section, we have studied whether BadHood blacklists from one application can be used to secure another application. In this section, we will study whether the blacklist created at one location can be used to secure *the same application at a different location*. Again, the main motivation is to avoid carrying out unnecessary network measurements: If we know that a BadHood blacklist created from measurements at site X can be also used to protect site Y, we do not need to generate a location-specific blacklist for Y.

To this end, we propose a simple spam detection system that implements a threshold-based filter. Consider L_S as the /24 BadHood blacklist to be used for spam detection. Whenever a new message M arrives, the mail filter extracts the source /24 prefix address of the sender ($M_{/24}$) and checks it against the list L_S. If $M_{/24}$ is found in L_S, then the mail filter will classify the message as spam if $nHosts(M_{/24}) > \theta$, where θ $(0 \leq \theta <= 256)$[2] can be seen as a threshold on how malicious a BadHood is. It should be emphasized that a real-world BadHood-based mail filter, like the one in [6], should combine different techniques, including whitelisting, in order to optimize the overall detection performance.

We create /24 BadHood blacklists from the datasets CBL, PSBL, Provider A, and UT/EWI, as introduced in the previous sections, and use them to filter spam from the mail servers of Provider A and UT/EWI. To evaluate the effectiveness of the different BadHood blacklists, we split each data set into a training data set of seven full days (April 19th to April 25th) and a test set of 1 day (April 26th). Table IV shows the number of malicious hosts (distinct /32 hosts) and the number of BadHoods in each training data set. The training blacklists are then used by our spam detection system to filter spam in the test sets of Provider A and UT/EWI. The achieved spam detection rate is defined as the ratio between the number of spams detected and the total number of spams received.

The total number of spam mails received by the different targets on the test day are shown in the first two rows of Table V. The table also gives the number of spammers (/32 IP addresses) and the number of observed BadHoods on that day. For UT/EWI, we also know the number of Ham messages received, as shown in the fourth row of the table.

Experimental Results and Discussion

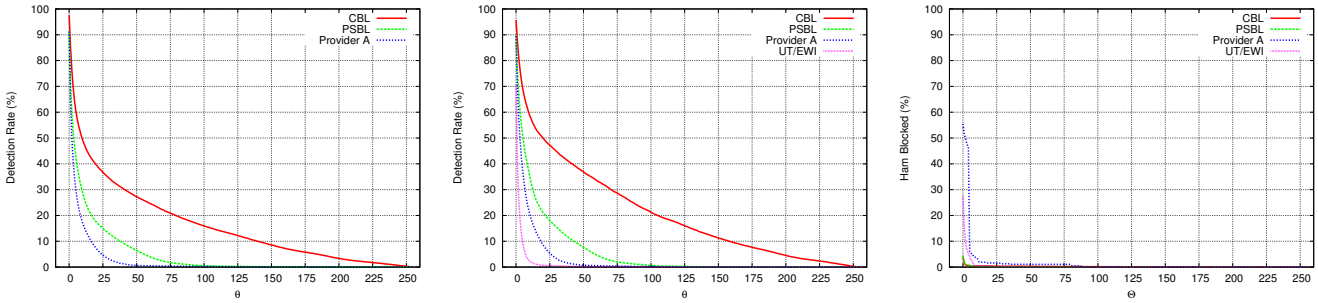
Figures 4(a) and 4(b) show the achieved detection rates for detecting the spam directed to Provider A and UT/EWI, respectively, as function of the threshold θ, using the different training blacklists. As in Section IV, we have only used a training blacklist if it is larger than the target's training blacklist (i.e., we have not applied the UT/EWI list to the Provider A target).

The figures indicate that it is possible to effectively detect spam messages based on the different BadHood blacklists. This is especially true for large blacklists, like CBL, which always provides the best detection rate. However, and especially for the smaller lists, the figures also show that the rate decreases

[2] A /24 prefix can have up to 256 malicious IP addresses depending on how addresses are allocated. *E.g.*, if an ISP allocates addresses as /22, as in 130.89.10.0/22 (which covers the /32 addresses 130.89.8.0 — 130.89.11.255), the addresses 130.89.10.255 and 130.89.10.0 are valid "routable" IP addresses.

	CBL	T-445	U-5559	T-25	T-443	T-80	U-53
T-445	69.4 % (166,789)						
U-5559	91.7% (39,660)	29.8% (12,894)					
T-25	93.0% (31,012)	26.7% (8,928)	19.5% (6,504)				
T-443	51.02% (12,694)	18.2% (4,547)	3.8% (950)	3.5% (884)			
T-80	32.1% (4,658)	11.2 % (1,623)	2.5% (375)	2.6% (387)	9.5% (1,377)		
U-53	28.5% (1,269)	8.2% (368)	3.89% (177)	6.9% (307)	1.8% (84)	3.1% (140)	
Phishtank	23.43% (413)	0.03% (54)	0.01% (2)	2.4% (43)	1.7% (22)	1.7% (23)	0.2% (5)

TABLE III. BADHOODS INTERSECTION FOR DIFFERENT APPLICATIONS, RELATIVE TO THE BIGGER LIST (COLUMN)



(a) Provider A (b) UT/EWI (c) Percentage of Ham erroneously blocked at UT/EWI

Fig. 4. Spam detection rate for varying values of the threshold θ in Fig. (a) and (b) and the rate of ham wrongly flagged in Fig. (c).

fast with increasing values of θ, a fact that most likely is due to the presence of high-volume spammers in the data sets.

A second insight provided by these results is that the value of θ should be adjusted to the considered BadHood blacklist. For the same θ, the detection rate changes considerably among BadHood blacklists. At first sight, this seems to suggest that the best choice for an administrator is the largest BadHood blacklist, just due to the fact that it has observed a higher number of spamming hosts. However, as we show in [8], the performance of blacklists can be adjusted, compensating for the blacklist size.

However, a different picture is obtained when calculating the number of legitimate mail traffic erroneously flagged as spam – that is, the number of false positives. Figure 4(c) shows the percentage of legitimate mail messages received by the mail server of UT/EWI that are labeled as spam for varying values of the threshold θ. While for CBL and PSBL the percentages of blocked Ham is less than 5% and rapidly falls to zero, for UT/EWI and Provider A we observe that up to approximately 60% of legitimate mail would be labeled as spam if a very low value of θ is chosen. On the other hand, also in the case of Provider A and UT/EWI, the percentage of blocked Ham is decreasing rapidly for increasing values of θ.

Our results highlight therefore a trade-off between (i) the size of the blacklist, (ii) the spam detection rate and (iii) the percentage of blocked Ham. Very large lists, such as CBL and PSBL, achieve a high spam detection rate with a low percentage of blocked Ham but contain a large number of irrelevant entries. In contrast, small and mid-sized lists, that is, Provider A and UT/EWI, contain much less irrelevant entries and can achieve detection rates comparable to those of the larger lists. However, for $\theta < 100$, a relatively high number of false positives can be expected.

These results leads to the conclusion that spam should be treated in a multi-layer mail filtering approach. At the first layer, a BadHood-based algorithm is employed to filter out e-mail from the most dangerous neighborhoods (using high θ values), that, at the same time, keeps false positives rate low. The second layer would comprise analysis of URLs and/or contents within the messages. Similarly, we have learned from sources at IBM and Google that subnetwork-based techniques are used in their mail filters.

VI. SUMMARY

Malicious IP addresses tend to be concentrated in certain networks instead of being evenly distributed over the IP address space. We have presented a formal definition for these areas, labeling them as Internet Bad Neighborhoods, and we have summarized some of the findings of the Ph.D. dissertation of one of the authors [7], addressed in other publications by the same team as well [8], [10], [14], [15], [28].

We have shown how neighboring IP addresses of malicious ones are more likely to be involved in future attacks than randomly distributed networks. Moreover, we have proposed and evaluated an algorithm to aggregate malicious IP addresses (/32) into BadHoods of various sizes (/24–8). We have then explored the relation between BadHood and exploited application and shown how they are application-specific. Finally, we have shown how BadHoods from third-party sources impact the performance of spam filters.

As future work, we envision the development of algorithms for spam filters and intrusion detection that take into account the findings and algorithms here presented , as well as other covered in the dissertation as well in our other publications.

Acknowledgments

The authors would like to thank Marc Berenschot, Provider A, and UT/EWI for their support for this research. Special thanks to the maintainers of CBL, PSBL, Phishtank and DShield.

REFERENCES

[1] J. M. Rao and D. H. Reiley, "The economics of spam," *The Journal of Economic Perspectives*, vol. 26, no. 3, pp. 87–110, 2012.

[2] BBC News, "Global Internet slows after 'biggest attack in history'," March 2013. [Online]. Available: http://www.bbc.co.uk/news/technology-21954636

[3] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: understanding, detecting, and disrupting botnets," in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*. Berkeley, CA, USA: USENIX Association, 2005, pp. 6–6.

[4] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane, "Using Uncleanliness to Predict Future Botnet Addresses," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 93–104.

[5] V. Fuller and T. Li, "RFC 4632: Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan," August 2006. [Online]. Available: http://tools.ietf.org/html/rfc4632

[6] W. van Wanrooij and A. Pras, "Filtering Spam from Bad Neighborhoods," *International Journal of Network Management*, vol. 20, no. 6, pp. 433–444, November 2010.

[7] G. C. M. Moura, "Internet Bad Neighborhoods," Ph.D. dissertation, University of Twente, Enschede, The Netherlands, March 2013. [Online]. Available: http://dx.doi.org/10.3990/1.9789036534604

[8] G. C. M. Moura, A. Sperotto, R. Sadre, and A. Pras, "Evaluating Third-Party Bad Neighborhood Blacklists for Spam Detection," in *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, Ghent, Belgium, May 2013.

[9] D. A. Wheeler and G. N. Larsen, "Techniques for cyber attack attribution," Institute for Defense Analyses, Alexandria, VA, USA, Tech. Rep., 2003. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA468859

[10] A. L. van Wynsberghe and G. C. Moreira Moura, "The concept of embedded values and the example of Internet Security," Responsible Research and Innovation in ICT, Oxford, Technical Report 1101, June 2013. [Online]. Available: http://torrii.responsible-innovation.org.uk/resource-detail/1101

[11] A. Ramachandran and N. Feamster, "Understanding the Network-level Behavior of Spammers," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '06. New York, NY, USA: ACM, 2006, pp. 291–302.

[12] SpamAssassin, "The Apache SpamAssassin Project," 2013. [Online]. Available: http://spamassassin.apache.org/

[13] CBL, "Composite Blocking List," 2012. [Online]. Available: http://cbl.abuseat.org/

[14] G. C. M. Moura, R. Sadre, A. Sperotto, and A. Pras, "Internet Bad Neighborhoods Aggregation," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, April 2012, pp. 343 –350.

[15] G. C. M. Moura, R. Sadre, and A. Pras, "Internet Bad Neighborhoods: the Spam Case," in *Proceedings of the 7th International Conference on Network and Services Management (CNSM)*, October 2011, pp. 56–63.

[16] Snort, "Snort: A free lightweight network intrusion detection system for UNIX and Windows," 2011. [Online]. Available: http://www.snort.org

[17] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-Based Intrusion Detection," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 3, pp. 343 –356, 2010.

[18] J. Levine, "DNS Blacklists and Whitelists," RFC 5782 (Informational), Internet Engineering Task Force, Feb. 2010.

[19] Passive Spam Block List, 2013. [Online]. Available: http://psbl.surriel.com/

[20] The Spamhaus Project, 2013. [Online]. Available: http://www.spamhaus.org

[21] RIPE NCC, "PI Assignment Size ," July 2006. [Online]. Available: http://www.ripe.net/ripe/policies/proposals/2006-05

[22] PhishTank, "PhishTank: Join the Fight Against Phishing," 2012. [Online]. Available: http://www.phishtank.com

[23] Google, "Google Public DNS," 2013. [Online]. Available: https://developers.google.com/speed/public-dns/

[24] DSHIELD.org, "dshield Home — DShield; Cooperative Network Security Community - Internet Security," May 2013. [Online]. Available: http:/www.dshield.org

[25] ——, "About the Internet Storm Center— DShield; Cooperative Network Security Community - Internet Security," May 2013. [Online]. Available: http://www.dshield.org/about.html

[26] SANS, "SANS Information, Network, Computer Security Training, Research, Resources," May 2013. [Online]. Available: http://www.sans.org

[27] IANA, " Service Name and Transport Protocol Port Number Registry," 2013. [Online]. Available: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt

[28] G. C. M. Moura, R. Sadre, and A. Pras, "Internet Bad Neighborhoods Temporal Attack Strategies *(to appear)*," in *Network Operations and Management Symposium (NOMS) Miniconf, 2014 IEEE*, May 2014.