Quantifying the Impact of Blocklisting in the Age of Address Reuse

Sivaramakrishnan Ramanathan
University of Southern California
satyaman@usc.edu

Anushah Hossain
UC Berkeley, ICSI
anushah@icsi.berkeley.edu

Jelena Mirkovic
USC Information Sciences Institute
sunshine@isi.edu

Minlan Yu
Harvard University
minlanyu@g.harvard.edu

Sadia Afroz
ICSI/Avast
sadia@icsi.berkeley.edu

ABSTRACT

Blocklists, consisting of known malicious IP addresses, can be used as a simple method to block malicious traffic. However, blocklists can potentially lead to unjust blocking of legitimate users due to IP address reuse, where more users could be blocked than intended. IP addresses can be reused either at the same time (Network Address Translation) or over time (dynamic addressing). We propose two new techniques to identify reused addresses. We built a crawler using the BitTorrent Distributed Hash Table to detect NATed addresses and use the RIPE Atlas measurement logs to detect dynamically allocated address spaces. We then analyze 151 publicly available IPv4 blocklists to show the implications of reused addresses and find that 53–60% of blocklists contain reused addresses having about 30.6K–45.1K listings of reused addresses. We also find that reused addresses can potentially affect as many as 78 legitimate users for as many as 44 days.

CCS CONCEPTS

• **Networks** → **Network measurement**; • **Security and privacy** → **Network security**;

KEYWORDS

IP address reuse, blocklists, unjust blocking

ACM Reference Format:
Sivaramakrishnan Ramanathan, Anushah Hossain, Jelena Mirkovic, Minlan Yu, and Sadia Afroz. 2020. Quantifying the Impact of Blocklisting in the Age of Address Reuse. In *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3419394.3423657

1 INTRODUCTION

Consider one user's experience with Cloudflare discussed under a trouble ticket [24]. When the user tried to access any website hosted by Cloudflare, they were unnecessarily blocked and subjected to CAPTCHAs. On further inspection, the user found that their public IP address was shared with many other users via a Network Address Translation (NAT). One of the NAT users was running a spam campaign leading to the NAT's IP address being listed in many blocklists. It is known that Cloudflare uses its own IP reputation with the help of blocklists [21] to protect its customers. Thus users behind reused addresses will be *unjustly blocked* whenever they access websites hosted on Cloudflare [23]. Other hosting providers have similar issues as well [35, 36]. What should legitimate users do when they are unjustly blocked? In fact, Cloudflare's best practice [25] recommends users to obtain a new IP address, by either resetting their device or by contacting their ISP. In reality, obtaining a new untainted static IP address may be impossible or too costly for many users [59, 66, 77]. This type of blocking often gets unnoticed by the network operators because currently there is no way to measure the excessive blocking for a blocking mechanism.

This paper proposes two techniques to measure unjust blocking from IP blocklisting. Blocklists are lists of identifiers (most often IP addresses) that are associated with malicious activities. For a network operator, blocklists provide a simple method to quickly block malicious traffic entering their network. Blocklists can have unjust blocking due to two forms of **address reuse**: 1) NATed addresses where several users share the same IP address at the same time and 2) dynamic addressing where the same IP address is allocated to multiple users over time.

In this study, we make the following contributions:

Detecting reused addresses: We propose two new techniques to identify reused addresses that provide high-confidence detection, leverage only public datasets, and can be replicated by other researchers. While extensive prior work exists on detecting NATed [7, 8, 45, 49, 51, 52, 62, 69, 73] and dynamic addresses [40, 61, 78], we find that they either do not provide sufficiently accurate and fine-grained information per IP address or do not publicly release the final list of reused addresses or prefixes. To detect NATed IP addresses, we implement a crawler using the BitTorrent's Distributed Hash Table (DHT). Our crawler detects when BitTorrent users simultaneously use the same IP address (Section 3.1), and thus we can measure the lower bound of users behind that IP address that would be adversely affected if the address were blocklisted. To detect dynamic addresses, we use the RIPE Atlas probe measurement logs to identify probes whose IP addresses change frequently and thus determine IP prefixes that are dynamically allocated (Section 3.2). By determining dynamic prefixes, we can identify users

IMC '20, October 27–29, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8138-3/20/10…$15.00
https://doi.org/10.1145/3419394.3423657

that would be affected by address reuse when allocated a previously blocklisted IP address. Our techniques have a reasonable overlap with the blocklisted IP address space, where BitTorrent and RIPE addresses are present in 29.6% and 17.1% of autonomous systems that have blocklisted addresses.

Measuring the impact of reused addresses: We apply our detection techniques to identify reused addresses in 151 publicly available *IPv4* blocklists (Section 4). About 60% of blocklists contain at least one NATed address and about 53% of blocklists list at least one dynamic address that will lead to unjust blocking. We find 45.1K and 30.6K listings in blocklists for each type of reuse, respectively. NATed and dynamic addresses in blocklists can have an impact on end-users by blocking as many as 78 users for as long as 44 days.

Finally, we survey 65 network operators (Section 6) on their blocklisting practices and find that IP blocklists are often used to directly block traffic. To assist network operators to avoid unjust blocking, we make our techniques publicly available and also publish a new address list that has all reused addresses we detect[1].

2 RELATED WORK

Existing studies identify *autonomous systems (ASes) or IP prefixes* that may be reused (e.g., that use carrier-grade NATs) using heuristics. However, to estimate the impact of blocklisting reused addresses, we need to accurately identify *IP addresses* that are reused. Müeller et al. [52] use traceroutes to a dedicated server in an ISP to detect middleboxes (including NAT). Other techniques use IPid [7], OS fingerprinting [8] or UDP hole punching [69] to detect NATed addresses. Netalyzr [45] and NetPiculet [73], on the other hand, require users to install Android applications that carry out measurements from the client's device. Though these techniques are effective in detecting NATs, they require many users to install custom applications to achieve significant coverage, and must continuously incentivize them to conduct measurements. These measurements are also no longer active.

Other approaches to reused address identification use private data and cannot be replicated. Richter et al. [61] and Casado et al. [15] observed IP addresses using NAT or dynamic addressing by monitoring server connection log of a CDN. Xie et al. [78] analyzed Hotmail user-login trace to determine dynamic addresses. Metwally et al. [51], use Google's application logs to detect NATed addresses and reduce false positives in detecting abuse traffic.

Cai et al. [13] present an ongoing survey by sending ICMP ECHO messages to 1% of the IPv4 address space. Based on the responses, they define metrics on availability, volatility, and median up-time to determine address blocks that are potentially dynamically allocated. This work produces a public dataset, which we compare against our approach in Section 5. However, this work has several limitations. An ICMP reply from an IP address need not uniquely identify the host using the IP address since firewalls and middleboxes can reply on behalf of hosts. Further, some networks filter outgoing ICMP traffic, potentially leading to undercounting. Finally, this work introduces an ad-hoc estimate of dynamically allocated prefixes based on the address uptime, and we cannot establish its accuracy.

Foremski et al. [34] define Entropy/IP that discovers IPv6 address structures using clustering and statistical techniques on a subset of IPv6 addresses that are known to be active. Using this system, one could identify IPv6 addresses that share similar characteristics (such as a network's address allocation strategy). Although we could use their technique to drive our measurement study to identify reused addresses, our current work focuses only on IPv4 blocklists.

BitTorrent network has been used to identify carrier-grade NATs (CGNs) in autonomous systems [49, 62]. These techniques leverage the fact that CGN public-facing IP addresses are likely to appear more frequently in a time window than non-CGN addresses. However, identifying ASes that use CGN is not useful for our research, since blocklists list IP addresses and CGN's may not be deployed across the entire AS. Thus making it hard to identify IP addresses that are using CGN.

3 TECHNIQUES

We propose two novel techniques to identify reused IP addresses. We use a BitTorrent-based crawler to identify NATed addresses and to estimate a lower bound on the number of users behind a NATed address. To identify dynamically addressed /24 prefixes, we extend Padmanabhan et al. [58]'s idea of using the RIPE Atlas measurement logs. Our priorities in designing these approaches were: (1) IP address granularity, (2) high accuracy of a positive detection, and (3) reasonable coverage. In other words, we accept some loss in coverage to achieve the first two goals: accuracy and fine granularity of detection. Our findings are therefore a lower bound on reused addresses.

3.1 Identifying NATed Addresses

We crawl the BitTorrent network to identify NATed addresses among BitTorrent users. BitTorrent is a popular peer-to-peer network for content exchange. A new BitTorrent user learns about other users as it joins the network. Every user generates its own unique 160-bit *node_id* that is obtained by hashing the (possibly private) IP address of the user and a random number. A new user learns IP addresses and port numbers of eight other users through the BitTorrent protocol – these users become the *neighbors* of the new user. The protocol supports two messages – *bt_ping* to periodically ping active neighbors and *get_nodes* to get a list of neighbors of any given node. We built a crawler that uses *bt_ping* and *get_nodes* messages to crawl the BitTorrent network and identify BitTorrent users using the same IP address at the same time, indicating such users are likely behind a NAT.

Initially, the crawler sends a *get_nodes* message to the BitTorrent bootstrap node, which returns a set of its neighbors. The crawler maintains a list of discovered BitTorrent users, and issues further *get_nodes* messages to the users on the list. The messages are issued in the order of discovery time. Replies to the *get_nodes* messages include the IP address, port number, *node_id* and the BitTorrent version of the node. If the crawler finds a new user with an already discovered IP address, but with a different port number, it tries to establish the reason behind this occurrence: (1) multiple BitTorrent users are using the same IP address (NATed address), or (2) the BitTorrent user has changed the port number and the crawler encountered stale information. We do not use *node_id* to determine multiple BitTorrent nodes using the same IP address, because the BitTorrent user can regenerate a new *node_id* every time their machine reboots.

[1] https://steel.isi.edu/members/sivaram/blocklisting_impact/

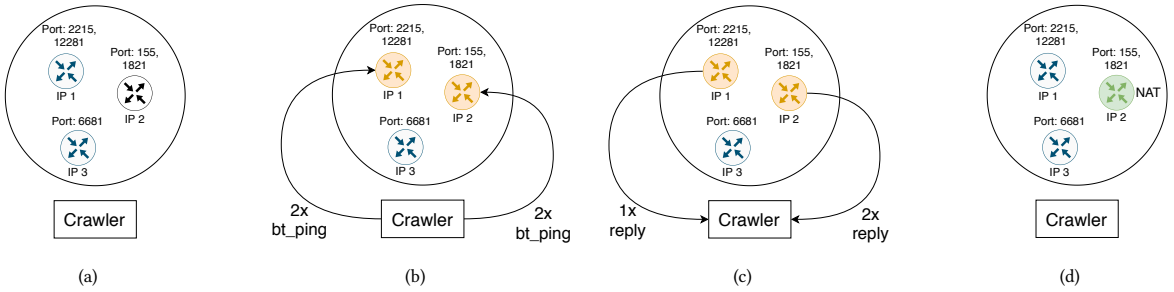

Figure 1: BitTorrent crawler to detect NATed reused addresses. In (a), the crawler identifies BitTorrent users with same IP address and multiple port numbers. In (b) and (c), the crawler sends *bt_ping* to $IP1$ and $IP2$ and receives replies. In (d), the crawler determines $IP1$ is a NATed reused address.

To determine if more than one active BitTorrent users share the same IP address at the same time, the crawler issues *bt_ping's* to all discovered ports behind a given IP address, and waits for responses. If the crawler gets more than two responses with two different *node_id's* and two different port numbers, we conclude that the IP address is shared by multiple BitTorrent users. Our technique provides an underestimation of NATed addresses and users, but with a high precision.

Figure 1 shows an example of how our BitTorrent crawler finds NATed addresses. The crawler encounters two users ($IP1$ and $IP2$) having the same IP address, but with two different ports in Figure 1(a) (with ports 2215, 12281 with $IP1$ and ports 155, 1821 with $IP2$). To verify if multiple active BitTorrent users are using the same IP address, the crawler issues four *bt_ping* messages in Figure 1(b), one for each port across two IP addresses and waits for responses. The crawler receives two replies from $IP2$ and one reply from $IP1$ (in Figure 1(c)), therefore determining that $IP2$ is shared by multiple BitTorrent users and $IP1$ is not (in Figure 1(d)).

BitTorrent *bt_ping* messages are sent over UDP, which means that they could be lost in transit. To compensate for this, the crawler sends *bt_ping* messages every hour for all the IP addresses that have more than one discovered port. The crawler logs all the messages (*bt_ping* or *get_nodes*) sent and all the messages received with the timestamps, which are then processed to determine NATed addresses. Once the crawler sends out a message (*get_nodes* or *bt_ping*) to all discovered ports associated with an IP address, the crawler does not contact that same IP address for the next 20 minutes. Initially, we did not restrict our BitTorrent crawler. However, the ping replies generated tremendous amount of incoming traffic that was undesirable to our network administrators. Therefore, to minimize the disturbance to users we probe and reduce burden on our network due to ping replies, the crawler is rate-limited and restricted only to address spaces where blocklists are present (Section 4). However, we could reduce this burden and have a faster coverage by having the crawler at multiple vantage points in different networks.

Limitations: Our crawler can only detect NATed addresses that have more than one BitTorrent user. Our NAT detection technique is certainly biased towards BitTorrent users. We will miss the NATs in networks where BitTorrent is not popular, or where BitTorrent traffic is filtered. Moreover, we can only detect the NATed addresses that are reused by more than one user using BitTorrent at the same time. We are thus likely to grossly underestimate the number of users behind a NAT.

3.2 Identifying Dynamic Addresses

The RIPE NCC's Atlas project deploys custom devices that conduct various measurement tasks. Every RIPE Atlas probe connects to a central infrastructure to get instructions on new measurement tasks and to update measurement data. All measurements are logged to include the unique *probe ID* and the IP address through which the measurement was made. Probes are usually deployed within the customer premises equipment (CPE) of the user. We use the measurement logs to infer the dynamics of IP address allocation in the covering /24 address prefix. We identify three properties of measurement logs that help us to find dynamic addresses.

1) Dynamic addressing observed over time: We observe RIPE Atlas probe measurement logs for 16 months to determine dynamic addressing. Observing logs for a long duration allows us to understand the frequency of IP address reallocation in probes.

2) Frequency of IP address change: We consider probes that have gone through multiple IP address allocations within the same AS during the monitoring period, to eliminate probes that experienced IP address changes rarely and to remove probes that have changed locations. Among the remaining probes, we consider probes whose average duration between every IP address change is within 1 day. This helps to estimate the risk involved in blocklisting these IP addresses, since blocklisting may be effective only for one day, and will lead to unjust blocking afterward.

3) Extent of dynamic addressing: If we detect that an IP address is dynamic, according to the criteria we described above, we consider that the entire /24 prefix covering this IP address is dynamically allocated. Usually, IP address reallocation occurs from a pool of IP addresses. It is hard to determine this address pool, and network operators do not make such information public. A conservative approach is to consider the entire /24 prefix as dynamic as contiguous addresses are usually administered together

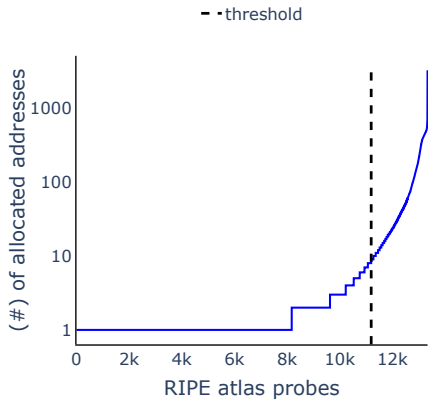

Figure 2: IP addresses allocated to RIPE Atlas probes.

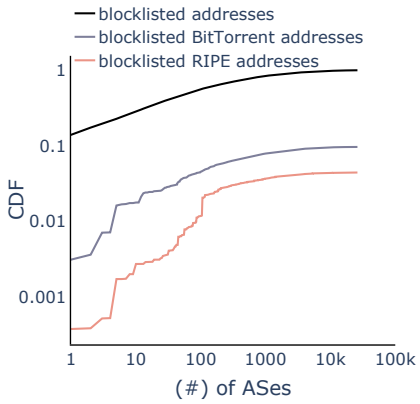

Figure 3: CDF of blocklisted and reused addresses from each AS.

and /24 prefixes have been previously used to identify network characteristics [19, 28, 29, 40, 61].

We use RIPE Atlas connection logs from 1 Jan 2019 to 11 May 2020. In this period, we observe over 15,703 RIPE Atlas probes that were allocated 311K IP addresses (referred to as **RIPE addresses**). About 13.1% (or 2K) of probes go through address changes but have addresses allocated across multiple autonomous systems. Figure 2 shows the remaining 13.6K probes and the number of addresses allocated to them. The majority of the probes (59% or 9.3K) did not go through any IP address change in this period and the remaining 27% (or 4.2K) of probes go through multiple address changes. We determine probes that change IP addresses more frequently than others by obtaining the knee point of this graph. The knee point indicates the number of IP address reallocation required to differentiate probes that change IP address more frequently when compared to remaining probes. We use a technique proposed by Satopää et al. [63] to determine the knee point to be at eight addresses.

This leaves us with 16.6% (2.6K) of the probes that have at least eight IP address allocations during our monitoring period, and where all reallocated IP addresses belong to the same autonomous system. These 2.6K probes cover a total of 204K IP addresses, with an average of 78 IP addresses allocated to each probe. In other words, 16.6% of all RIPE Atlas probes contributed to 65.5% of all IP addresses allocated to all RIPE Atlas probes. As a final step to consider probes that regularly change addresses within one day, we end up with 4% (629) of probes that are in dynamically allocated address spaces. We only consider probes that change IP addresses daily, since these probes are more likely to cause unjust blocking than probes that rarely change addresses.

Limitations: Our detection of dynamic addresses is limited only to IP prefixes that deploy a RIPE Atlas probe. RIPE Atlas probes are predominantly present only in Europe and North America. Therefore, our insights on dynamic address detection are limited to these regions. Another limitation is that our detection technique involves choosing RIPE probes that have been allocated IP addresses from the same autonomous system. However, ISPs may own several autonomous systems and can allocate addresses from multiple autonomous systems. Even within the IP address spaces covered by RIPE Atlas probes, our technique only presents a lower-bound of

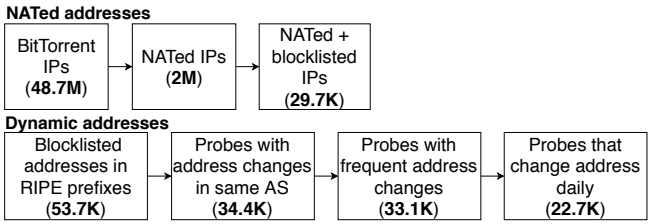

Figure 4: Detecting NATed and dynamic addresses.

prefixes that are dynamically allocated. For probes that change their IP addresses frequently, we consider the entire /24 prefix to be potentially dynamically allocated. However, we may incorrectly gauge these boundaries. Network operators may deploy dynamic addressing within larger prefixes, leading us to underestimate the extent of unjust blocking. In some cases, they may deploy dynamic addressing within smaller prefixes, thereby overcounting the number of dynamic addresses. Estimating boundaries is difficult because ISPs have their own private policies for dynamic addressing. This issue is noted in previous studies as well [40, 61, 78].

4 DETECTION

Blocklists typically list IP addresses that have sent spam, DDoS attacks, dictionary attacks, or malicious scans. To quantify unjust blocking by blocklists, we use 151 IPv4 public blocklists shown in Table 2 in the Appendix B taken from the BLAG dataset [60]. These lists are actively maintained and they monitor a variety of malicious activities including Spam, DDoS, malware hosting or general reputation of IP addresses. This dataset includes popular lists like DShield [43], NixSpam [57], Spamhaus [4], Alienvault [2] and Abuse.ch [1]. We collect blocklist data for 83 days over two measurement periods from 03 Aug 2019 to 10 Sep 2019 (39 days) and 29 Mar 2020 to 11 May 2020 (44 days). During our measurement periods, we observed 2.2M IP addresses and each blocklist, on average, has 30K IP addresses.

Figure 4 shows the number of reused addresses discovered by our techniques. We ran our BitTorrent crawler at the same time

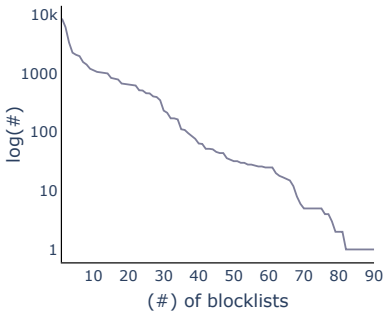

Figure 5: NATed addresses in blocklists.

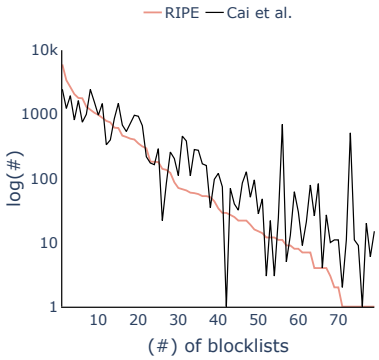

Figure 6: Dynamic addresses in blocklists.

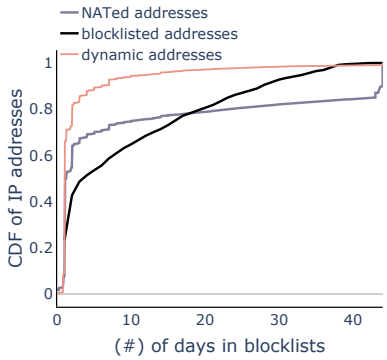

Figure 7: Duration distribution of reused addresses.

as the blocklist collection period. To prevent unnecessary probing, we restrict the crawler to the blocklisted address space of 899K /24 prefixes (as discussed in Section 3.1). During this period, the crawler sent 1.6 billion *bt_ping* messages and received 779M responses (48.6% response rate). The crawler identified a total of 48.7M unique IP addresses that use BitTorrent, belonging to 203M unique *node_id's*. Among the discovered BitTorrent IP addresses, 2M are NATed; out of these, 29.7K are blocklisted. This overlap of blocklisted addresses with BitTorrent addresses is in agreement with previous work [31] that find devices using P2P are likely to be compromised. We use the 311K RIPE addresses obtained from Section 3.2, and convert them into 90.5K /24 RIPE prefixes. About 53.7K blocklisted addresses are in RIPE prefixes. The overlap with RIPE prefixes decreases as we apply our technique. At first, by eliminating all probes that change addresses across multiple ASes, the number of blocklisted addresses reduces to 34.4K. While considering probes that have at least eight address changes, the number of blocklisted addresses further reduces to 33.1K. Finally, after considering probes that change their addresses daily, we have 22.7K blocklisted addresses that are dynamically allocated.

To determine the feasibility of our techniques to discover reused addresses, we estimate (shown in Figure 3) the extent of overlap between the address spaces where BitTorrent or RIPE addresses are present and the address spaces where blocklisted addresses are present. The ASes in Figure 3 are arranged in increasing order of the number of blocklisted addresses present in them. Blocklisted addresses are present in about 26K autonomous systems, therefore, the curve for blocklisted addresses reaches upto 1. Blocklisted addresses using BitTorrent are present in 7.7K (or 29.6%) ASes and blocklisted addresses among RIPE prefixes are present in 1.9K (or 17.1%) ASes. Since our techniques do not cover the entire blocklisted addresses, the curves for blocklisted RIPE and BitTorrent addresses plateau at 7.7K and 1.1 ASes respectively. Our technique has significant coverage in the ten most blocklisted ASes. These ASes contribute to 606K (or 27.7%) of all blocklisted addresses. Among these addresses, 38K (6.4%) use BitTorrent and 4.4K (0.7%) are among RIPE prefixes. AS4134, belonging to China Telecom Backbone, has the highest number of blocklisted addresses (202K or 9%). Among the blocklisted addresses from AS4134, about 3% (or 6.2K) use BitTorrent and 0.4% (or 817) are in RIPE prefixes.

5 ANALYSIS

In this section, we estimate the potential extent of unjust blocking caused by blocklisting reused addresses. Although we identify only 29.7K blocklisted NATed addresses and 22.7K blocklisted dynamic addresses, we observe that reused addresses are present in many blocklists (up to 60%). Blocklisting reused addresses can have serious impact on users. Blocklisting NATed addresses can lead to unjust blocking of multiple users having the same IP address and dynamically allocated addresses can lead to unjust blocking of a user that has been newly allocated a previously blocklisted address. We find that a reused address can be present in blocklists for as many as 44 days and can potentially block as many as 78 users.

Reused addresses are present in many blocklists: Figure 5 and Figure 6 shows blocklists that list reused addresses. There are 61 blocklists (40% of all blocklists) that do not list any NATed addresses and 72 blocklists (47% of all blocklists) that do not list any dynamic addresses. We discover 45.1K listings[2] that include 29.7K IP addresses that are NATed. We also discover 30.6K listings that include 22.7K IP addresses that are dynamic addresses. On average, a blocklist lists 501 NATed IP addresses and 387 dynamic addresses. Our techniques of reused address detection is not perfect, therefore, blocklists that do not show any reused addresses can contain reused addresses but are not identified by our techniques.

Some blocklists list more reused addresses than others: The top 10 blocklists contribute 65.9% of all listings of NATed addresses and 72.6% of all listings of dynamic addresses. This is expected, as the top 10 blocklists among NATed and dynamic addresses contribute to 53.4% and 70.3% of all blocklisted addresses. The three highest presence of NATed addresses are from spam or reputation blocklists – *Stopforumspam*, *Nixspam* and *Alienvault* listing about 3.3K-8.6K (or 0.01%–0.03% of blocklists) such IP addresses. Similarly, the three highest presence of dynamic addresses are from *Stopforumspam*, *Nixspam* and *Bad IPs*, primarily used to

[2] An IP address can be present in different blocklists, therefore the number of listings need not be equal to the number of reused IP addresses.

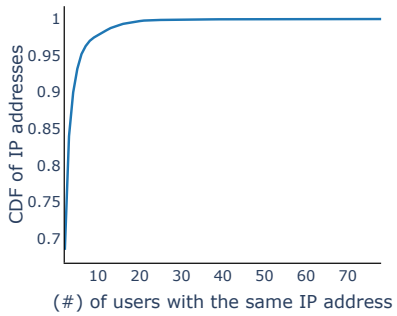

Figure 8: Number of users behind NATed addresses in blocklists.

mitigate spam and identify IP addresses with a poor reputation. These blocklists list about 2.6K–5.9K (or 0.01%–0.02% of blocklists) dynamic addresses.

Most reused addresses are removed quickly: Figure 7 shows the CDF of all blocklisted addresses, blocklisted NATed/dynamic addresses and the duration in days that they were present in a blocklist. On average, blocklisted addresses are removed within nine days, NATed IP addresses are removed within ten days, and dynamic addresses are removed within three days from blocklists. Compared to all blocklisted addresses, reused addresses are removed much faster and among reused addresses, dynamic addresses are removed faster than NATed IP addresses. Within two days, 77.5% of all dynamic addresses are removed from blocklists, compared to only 60% of NATed IP addresses. On the other hand, only 42% of all blocklisted IP addresses are removed in two days. In the worst case, reused addresses are present in blocklists for the entire monitoring period of 44 days.

Blocklisting NATed addresses impact many users: The BitTorrent crawler described in Section 3.1, helps us quantify the lower bound of active users using the same IP address that would be affected by blocklisting. Figure 8 shows the CDF of NATed blocklisted addresses and the lower bound of affected users. For most of these IP addresses, we detect only two active users (68.5%). 97.8% of the IP addresses have fewer than ten active users. The remaining 2.2% of the IP addresses are shared by many more users. At the maximum, we detect 78 active users behind an IP address.

Exploring other techniques: As discussed in Section 2, there are many other techniques for detecting reused addresses. The only technique that can be reproduced at scale is Cai et al. [13]. We use their techniques and the datasets (*IT86c* and *IT89w*) that most closely match our monitoring periods to compare dynamic addresses. Figure 6 (black line) shows the number of blocklisted addresses that overlap with their study. Cai et al. have broader coverage in some blocklists compared to this study; this is likely due to the absence of RIPE Atlas probes in those address regions. We find that the total number of listings discovered by Cai et al. is roughly the same as our technique: they detect 29.8K listings compared to 30.6K listings using our technique.

	Question	Response
Blocklist usage	External blocklists	85%
	Paid-for blocklists	Avg:2 Max:39
	Public blocklists	Avg:10 Max:68
Active defense	Directly block IPs	59%
	Threat intelligence system	35%
Issues	Dynamic addressing*	76%
	Carrier-grade NATs*	56%

Table 1: Summary of survey responses on usage of blocklists. Questions in (*) were only taken by 34 out of 65 respondants.

6 UNDERSTANDING BLOCKLISTS USAGE

We survey 65 network operators (see Appendix A for the full survey), on how they use blocklists to identify malicious traffic, the role blocklists play in traffic filtering, and their perceptions of limitations of blocklists due to reused addresses. Our survey indicates that 85% of respondents use blocklists, and 59% of respondents use blocklists to directly block malicious traffic (Table 1). 34 survey participants responded to direct questions about the impact of reused addresses. Of these, 56% believe that blocklists are inaccurate due to NAT and 76% believe dynamic addressing introduces inaccuracies. From these responses as well as open-ended comments made within the survey, it is evident that network operators use blocklists and are aware of unjust blocking caused by reused addresses.

We make our crawler and scripts to determine reused addresses public[3]. We believe that our lists can assist network operators in many ways. Depending on the blocklist type, a network operator could take necessary action on their incoming traffic with our list. For instance, operators that use DDoS blocklists to reduce intensity of the attacker should block all traffic listed in DDoS blocklists, even if there is collateral damage due to reused addresses. On the other hand, network operators using application-specific blocklists (such as spam blocklist) that require more accuracy, can use our list to implement greylisting [46], which is already built into popular spam filtering systems like Spamassassin [74] or Spamd [14].

Our lists can also provide incentives to blocklist maintainers to maintain more accurate blocklists. They may identify malicious reused IP addresses in a separate greylist to their customers. Finally, our lists can assist services such as Google or Cloudflare, to warn users that their IP address is reused along a compromised device. This could help users to clean up their home network, or even request help from their ISP to identify other compromised users.

7 CONCLUSION

We present two techniques to identify reused addresses and analyze 151 publicly available IPv4 blocklists to quantify their impact. We find 53–60% of blocklists list at least one reused address. We also find 30.6K–45.1K listings of reused addresses in blocklists. Reused addresses can be present in blocklists for as long as 44 days affecting as much as 78 users. Finally, to assist blocklist maintainers to reduce unjust blocking, we made our discovered reused addresses public.

[3]https://steel.isi.edu/members/sivaram/blocklisting_impact/

ACKNOWLEDGMENT

We thank our shepherd Andra Lutu and anonymous reviewers for their helpful comments. We also thank Philipp Richter for his help with the BitTorrent crawler and useful discussions on identifying reused addresses. This material is based on research sponsored by the Department of Homeland Security (DJ-IS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency (HSARPA), Cyber Security Division (DHS S&T/HSARPA CSD) BAA HSHQDC-14-R-B0005, and the Government of United Kingdom of Great Britain and Northern Ireland via contract number D15PC00184. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security, the U.S. Government, or the Government of United Kingdom of Great Britain and Northern Ireland. Minlan Yu is supported by CNS 1834263 by the National Science Foundation. Sadia Afroz is supported by CNS 1518918 by the National Science Foundation. Additional thanks to Zhiying Xu, Matthias Marx and Krutika Jain for their inputs in the earlier draft of the paper.

REFERENCES

[1] Abuse.ch. 2020. Swiss Security Blog - Abuse.ch. https://www.abuse.ch/. (May 2020).
[2] Alienvault. 2020. Alienvault Reputation System. https://www.alienvault.com/. (May 2020).
[3] Antispam. 2020. ImproWare. http://antispam.imp.ch/. (May 2020). (Accessed on 05/13/2020).
[4] Charles Arthur. 2006. Can an American judge take a British company offline? (October 2006). https://www.theguardian.com/technology/2006/oct/19/guardianweeklytechnologysection3
[5] BadIPs. 2020. badips.com | an IP based abuse tracker. https://www.badips.com/. (May 2020).
[6] Bambenek. 2020. Bambenek Consulting Feeds. http://osint.bambenekconsulting.com/feeds/. (May 2020).
[7] Steven M Bellovin. 2002. A technique for counting NATted hosts. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM, 267–272.
[8] Robert Beverly. 2004. A robust classifier for passive TCP/IP fingerprinting. In *International Workshop on Passive and Active Network Measurement*. Springer, 158–167.
[9] Blocklist.de. 2020. Blocklist.de fail2ban reporting service. https://www.blocklist.de/en/index.html. (May 2020).
[10] Botscout. 2020. We catch bots so that you don't have to. https://www.botscout.com. (May 2020).
[11] Botvrij. 2020. botvrij.eu - powered by MISP. http://www.botvrij.eu/. (May 2020).
[12] Malware Bytes. 2020. hpHosts - by Malware Bytes. https://hosts-file.net/. (May 2020).
[13] Xue Cai and John Heidemann. 2010. Understanding block-level address usage in the visible internet. In *Proceedings of the ACM SIGCOMM 2010 conference*. 99–110.
[14] Calomel. 2017. Spamd tarpit and greylisting daemon. https://calomel.org/spamd_config.html. (Jan 2017).
[15] Martin Casado and Michael J Freedman. 2007. Peering through the shroud: The effect of edge opacity on IP-based client identification. In *4th {USENIX} Symposium on Networked Systems Design & Implementation ({NSDI} 07)*.
[16] Taichung Education Center. 2020. Taichung Education Center. https://www.tc.edu.tw/net/netflow/lkout/recent/30. (May 2020).
[17] CIArmy. 2020. CINSscore. http://ciarmy.com/. (May 2020).
[18] Cisco. 2020. Cisco Talos - Additional Resources. http://www.talosintelligence.com/. (May 2020).
[19] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. 2009. Internet mapping: from art to science. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*. IEEE, 205–211.
[20] Cleantalk. 2020. Cloud spam protection for forums, boards, blogs and sites. https://www.cleantalk.org. (May 2020).
[21] Cloudflare. 2020. Understanding Cloudflare Challenge Passage (Captcha). https://support.cloudflare.com/hc/en-us/articles/200170136. (Feb 2020).
[22] GPF Comics. 2020. The GPF DNS Block List. https://www.gpf-comics.com/dnsbl/. (May 2020).
[23] Cloudflare Community. 2018. Getting Cloudflare capcha on almost every website I visit for my home network. Help! https://community.cloudflare.com/t/getting-cloudflare-capcha-on-almost-every-website-i-visit-for-my-home-network-help/42534. (Nov 2018).
[24] Cloudflare Community. 2019. Blocked IP address: Sharing IPs. https://community.cloudflare.com/t/cloudflare-blocking-my-ip/65453/57. (Mar 2019).
[25] Cloudflare Community. 2019. Community Tip - Best Practices For Captcha Challenges. https://community.cloudflare.com/t/community-tip-best-practices-for-captcha-challenges/56301. (Jan 2019).
[26] CruzIt. 2020. Server Blocklist / Blacklist - CruzIT.com - PHP, Linux & DNS Tools, Apache, MySQL, Postfix, Web & Email Spam Prevention Information. http://www.cruzit.com/wbl.php. (May 2020).
[27] Cybercrime. 2020. CyberCrime Tracker. http://cybercrime-tracker.net/. (May 2020).
[28] Alberto Dainotti, Karyn Benson, Alistair King, KC Claffy, Michael Kallitsis, Eduard Glatz, and Xenofontas Dimitropoulos. 2013. Estimating internet address space usage through passive measurements. *ACM SIGCOMM Computer Communication Review* 44, 1 (2013), 42–49.
[29] Alberto Dainotti, Karyn Benson, Alistair King, Bradley Huffaker, Eduard Glatz, Xenofontas Dimitropoulos, Philipp Richter, Alessandro Finamore, and Alex C Snoeren. 2016. Lost in space: improving inference of IPv4 address space utilization. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1862–1876.
[30] Binary Defense. 2020. Binary Defense Systems | Defend. Protect. Secure. https://www.binarydefense.com/. (May 2020).
[31] Louis F DeKoven, Audrey Randall, Ariana Mirian, Gautam Akiwate, Ansel Blume, Lawrence K Saul, Aaron Schulman, Geoffrey M Voelker, and Stefan Savage. 2019. Measuring Security Practices and How They Impact Security. In *Proceedings of the Internet Measurement Conference*. 36–49.
[32] DYN. 2020. Index of /pub/malware-feeds/. http://security-research.dyndns.org/pub/malware-feeds/. (May 2020).
[33] IP finder. 2020. IP Blacklist Cloud - Protect your website. https://www.ip-finder.me/. (May 2020).
[34] Pawel Foremski, David Plonka, and Arthur Berger. 2016. Entropy/ip: Uncovering structure in ipv6 addresses. In *Proceedings of the 2016 Internet Measurement Conference*. 167–181.
[35] Comcast Forums. 2018. Dirty (blacklisted) IPs issued to Comcast Business Account holders. https://forums.businesshelp.comcast.com/t5/Connectivity/Dirty-blacklisted-IPs-issued-to-Comcast-Business-Account-holders/td-p/34297. (Mar 2018).
[36] Verizon Forums. 2020. IP address blocked by SORBS, Verizon will do nothing. https://forums.verizon.com/t5/Fios-Internet/IP-address-blocked-by-SORBS-Verizon-will-do-nothing/td-p/892536. (Feb 2020).
[37] Daniel Gerzo. 2020. Daniel Gerzo BruteForceBlocker. http://danger.rulez.sk/index.php/bruteforceblocker/. (May 2020).
[38] Greensnow. 2020. Greensnow Statistics. https://greensnow.co/. (May 2020).
[39] Charles B. Haley. 2020. SSH Dictionary Attacks. http://charles.the-haleys.org/. (May 2020).
[40] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. 2008. Census and survey of the visible internet. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. 169–182.
[41] Project Honeypot. 2020. Project Honeypot. https://www.projecthoneypot.org/. (May 2020).
[42] IBM. 2020. IBM X-Force Exchange. https://exchange.xforce.ibmcloud.com/. (May 2020).
[43] SANS Institute. 2019. Internet Storm Center. https://dshield.org/about.html. (Sept 2019).
[44] My IP. 2020. My IP - Blacklist Checks. https://www.myip.ms/info/about. (May 2020).
[45] Christian Kreibich, Nicholas Weaver, Boris Nechaev, and Vern Paxson. 2010. Netalyzr: illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 246–259.
[46] M Kucherawy and D Crocker. 2012. *Email greylisting: An applicability statement for smtp*. Technical Report. RFC 6647, June.
[47] Snort Labs. 2020. Sourcefire VRT Labs. https://labs.snort.org/. (May 2020).
[48] Malware Domain List. 2020. Malware Domain List. http://www.malwaredomainlist.com/. (May 2020).
[49] I. Livadariu, K. Benson, A. Elmokashfi, A. Dainotti, and A. Dhamdhere. 2018. Inferring Carrier-Grade NAT Deployment in the Wild. In *IEEE Conference on Computer Communications (INFOCOM)*.
[50] Malc0de. 2020. Malc0de Database. http://malc0de.com/database/. (May 2020).
[51] Ahmed Metwally and Matt Paduano. 2011. Estimating the number of users behind ip addresses for combating abusive traffic. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 249–257.
[52] Andreas Müller, Florian Wohlfart, and Georg Carle. 2013. Analysis and topology-based traversal of cascaded large scale NATs. In *Proceedings of the 2013 workshop on Hot topics in middleboxes and network function virtualization*. ACM, 43–48.
[53] Blocklist NET. 2020. BlockList.net.ua. https://blocklist.net.ua/. (May 2020).

[54] Normshield. 2020. Normshield - Cyber Risk Scorecard. https://www.normshield.com/. (May 2020).
[55] NoThink. 2020. NoThink Individual Blacklist Maintainer. http://www.nothink.org/. (May 2020).
[56] Nullsecure. 2020. nullsecure. https://nullsecure.org/. (May 2020).
[57] Heise Online. 2020. Nixspam Blacklist. https://goo.gl/jsyksA. (May 2020).
[58] R. Padmanabhan, A. Dhamdhere, E. Aben, k. claffy, and N. Spring. 2016. Reasons Dynamic Addresses Change. In *Internet Measurement Conference (IMC)*.
[59] Spectrum Partners. 2020. Spectrum Static IP. https://partners.spectrum.com/content/spectrum/business/en/internet/staticip.html. (May 2020).
[60] Sivaramakrishnan Ramanathan, Jelena Mirkovic, and Minlan Yu. 2020. BLAG: Improving the Accuracy of Blacklists. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020 (NDSS '20)*. The Internet Society. https://doi.org/10.14722/ndss.2020.24232
[61] Philipp Richter, Georgios Smaragdakis, David Plonka, and Arthur Berger. 2016. Beyond Counting: New Perspectives on the Active IPv4 Address Space. In *Proceedings of ACM IMC 2016*. Santa Monica, CA.
[62] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. 2016. A Multi-perspective Analysis of Carrier-Grade NAT Deployment. In *Proceedings of ACM IMC 2016*. Santa Monica, CA.
[63] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*. IEEE, 166–171.
[64] Sblam. 2020. Sblam! http://sblam.com/. (May 2020).
[65] Stop Forum Spam. 2020. Stop Forum Spam. https://stopforumspam.com/. (May 2020).
[66] ARS Technica. 2020. ATT raises prices 7% by making its customers pay ATT's property taxes. https://arstechnica.com/tech-policy/2019/10/att-raises-prices-7-by-making-its-customers-pay-atts-property-taxes/. (Oct 2020).
[67] Threatcrowd. 2020. Threat Crowd - Open Source Threat Intelligence. https://threatcrowd.org/. (May 2020).
[68] Emerging Threats. 2020. Emerging Threats Rules. https://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt. (May 2020).
[69] Kazuhiro Tobe, Akihiro Shimoda, and Shegeki Goto. 2010. Extended UDP Multiple Hole Punching Method to Traverse Large Scale NATs. *Proceedings of the Asia-Pacific Advanced Network* 30 (2010), 30–36.
[70] Turris. 2020. Greylist :: Project:Turris. https://www.turris.cz/en/greylist. (May 2020).
[71] URLVir. 2020. URLVir: Monitor Malicious Executable Urls. http://www.urlvir.com/. (May 2020). (Accessed on 05/13/2020).
[72] VX Vault. 2020. VX Vault ViriList. http://vxvault.net/ViriList.php. (May 2020).
[73] Zhaoguang Wang, Zhiyun Qian, Qiang Xu, Zhuoqing Mao, and Ming Zhang. 2011. An untold story of middleboxes in cellular networks. In *ACM SIGCOMM Computer Communication Review*, Vol. 41. ACM, 374–385.
[74] Apache Wiki. 2019. Other Trick For Blocking Spam. https://cwiki.apache.org/confluence/display/SPAMASSASSIN/OtherTricks#OtherTricks-Greylisting. (Jul 2019).
[75] Wikipedia. 2019. Internet network operators' group — Wikipedia, The Free Encyclopedia. (June 2019). https://en.wikipedia.org/w/index.php?title=Internet_network_operators%27_group&oldid=906511356
[76] Chris Wilcox, Christos Papadopoulos, and John Heidemann. 2010. Correlating spam activity with ip address characteristics. In *2010 INFOCOM IEEE Conference on Computer Communications Workshops*. IEEE, 1–6.
[77] Xfinity. 2020. Business Class Internet at Home. https://www.xfinity.com/hub/business/internet-for-home-business. (May 2020).
[78] Yinglian Xie, Fang Yu, Kannan Achan, Eliot Gillum, Moises Goldszmidt, and Ted Wobber. 2007. How dynamic are IP addresses?. In *ACM SIGCOMM Computer Communication Review*, Vol. 37. ACM, 301–312.
[79] ZeroDot1. 2020. CoinBlockerLists. https://gitlab.com/ZeroDot1/CoinBlockerLists. (May 2020).

A USAGE AND PERCEPTIONS OF BLOCKLISTS

In this section, we survey network operators to understand blocklist usage to filter suspicious traffic. This is when users of reused addresses could be unjustly blocked. Further, we try to understand the operator's anecdotal experiences on blocklisting reused addresses. This helps us to establish the importance of this issue from a human perspective.

We circulated an online questionnaire to regional network operator groups by posting to all groups that published open-access mailing lists (identified via [75]). We received responses from members of forty groups. Network operators' mailing lists are forums for network engineers, operators, and other technical professionals to coordinate and disseminate information about network security, peering, routing, and other operational Internet issues. We chose this strategy to maximize outreach to the relevant communities, offering apologies in advance for potential overlap as operators may subscribe to more than one list (e.g. NYNOG for New York, USA, in addition to NANOG for North America). Mailing list subscribers were informed that the purpose of the study was to better understand current blocklisting practices and challenges. Participants could complete the survey anonymously and were offered the option to subscribe to receive findings from the completed study. The survey included 24 questions on what blocklists are used, the role they play in filtering (e.g. indirect blocking or as an input to a threat intelligence system), and their perceived benefits and limitations (see Section C of Appendix). Between July and August 2019, 65 respondents finished and submitted survey answers. Survey participants operate networks in five continents, including end-user and enterprise ISPs and content providers. Sizes of networks vary from 100 to over 10 million users. Our survey did not require IRB, since we did not collect any personal data and our results are not based on human subjects. Our key findings are as follows:

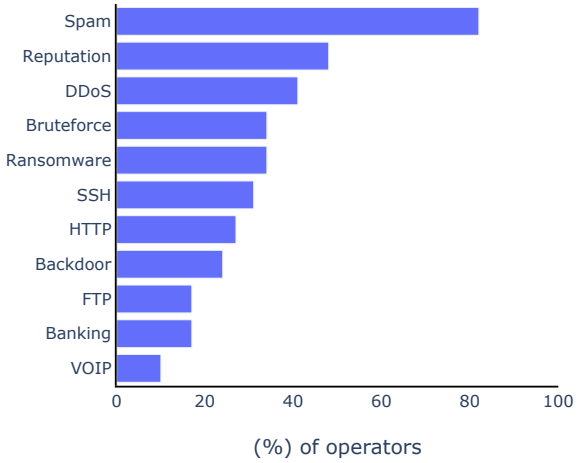

Figure 9: Types of blocklists used by operators that have faced issues with reused addresses in blocklists.

Blocklists are widely used and are used for active defense: Network operators use two types of blocklists – operator curated internal blocklists and external blocklists that includes paid-for or publicly available blocklists. About 70% of operators maintained internal blocklists and 85% used external blocklists. Network operators often use multiple blocklists to defend their networks. 55% of respondents used two or more different types of blocklists. On average, network operators subscribed to 2 paid-for lists and 10 publicly available blocklists and can use up to 39 paid-for and 68

public blocklists. Usage of blocklists can have consequences as network operators often use them to block traffic. 59% of surveyed network operators use blocklisted addresses to directly block traffic, and fewer than 35% of network operators use blocklisted addresses as an input to other threat intelligence systems. Therefore, depending on the blocklists used, network operators could unjustly block users in reused addresses. Our survey also finds that some network operators set manual filters to override blocklisting in address spaces that they believed were dynamically allocated. We also find that blocklists have usage beyond blocking. One of the surveyed network operators checks its own addresses on blocklists before assigning them to new customers, to avoid unjust blocking.

Perceived inaccuracies due to reused addresses: When asked directly, only 34 of the survey respondents answered this question. 56% of the respondents (19 out of 34) believed carrier-grade NAT (CGN) affected the accuracy of blocklists, citing cases where legitimate users were getting blocked because of a shared address. About 76% of respondents (26 out of 34) said that dynamic addressing affected the accuracy of blocklists. As a part of the survey, operators identified the type of external blocklists used in their network. Figure 9 shows the type of blocklists used operators that have faced issues with blocklists due to reused addresses. Among the blocklists subscribed by these operators, we find spam and reputation blocklists to have the highest consequences of blocking reused addresses. Although our findings are anecdotal, previous studies have shown malicious activities such as spamming to be correlated with dynamically allocated address spaces [76, 78].

B BLOCKLIST DATASET

A network operator could use its own set of blocklists to determine reused addresses. We use 151 public blocklists taken from BLAG dataset [60] as shown in Table 2. We collected blocklist data for 83 days over two measurement periods from 03 Aug 2019 to 10 Sep 2019 (39 days) and 29 Mar 2020 to 11 May 2020 (44 days). As a part of the survey (Section A), some network operators manually listed external blocklists used by them. There are 27 such blocklists indicated by a * in Table 2.

Maintainer	# of blocklists
Bad IPs [5]	44
Bambenek [6]	22
*Abuse.ch [1]	10
Normshield [54]	9
*Blocklist.de [9]	9
Malware bytes [12]	9
*Project Honeypot [41]	4
CoinBlockerLists [79]	4
NoThink [55]	3
Emerging threats [68]	2
ImproWare [3]	2
Botvrij.EU [11]	2
IP Finder [33]	1
*Cleantalk [20]	1
Sblam! [64]	1
*Nixspam [57]	1
Blocklist Project [53]	1
BruteforceBlocker [37]	1
Cruzit [26]	1
Haley [39]	1
Botscout [10]	1
My IP [44]	1
Taichung [16]	1
*Cisco Talos [18]	1
Alienvault [2]	1
Binary Defense [30]	1
GreenSnow [38]	1
Snort Labs [47]	1
GPF Comics [22]	1
Turris [70]	1
CINSscore [17]	1
Nullsecure [56]	1
DYN [32]	1
Malware domain list [48]	1
Malc0de [50]	1
URLVir [71]	1
Threatcrowd [67]	1
CyberCrime [27]	1
IBM X-Force [42]	1
VXVault [72]	1
*Stopforumspam [65]	1
Total	**151**

Table 2: Each row shows the number of blocklists provided by the blocklist maintainer. We monitor 151 blocklists to determine the number of blocklisted addresses using NAT or are dynamically allocated. Blocklists used by network operators who took our survey are marked with (*).

C QUESTIONNAIRE ON PERCEPTIONS OF BLOCKLISTS

Questions that permitted open-ended responses are denoted with asterisks.

(1) What is your company's name and AS number if available?*
(2) What is your position / your role in network management?*
(3) What is your email address?*
(4) May we reach out to you via email: to inform you once the results of this survey are publicly available
(5) May we reach out to you via email: with further questions
(6) What type of network do you run? (more than one choice possible)
(7) How many subscribers do you connect to the Internet?
(8) In what geographic region(s) do you operate?
(9) Do you maintain internal blocklists?
(10) How and why did you develop internal blocklists? How do they compare to third-party blocklists?*
(11) How many third-party blocklists do you use?
(12) Which of the following types of third-party blocklists do you use? (Please select all that apply)
(13) What factors determine which third-party blocklists you use?*

(14) Do you use third-party blocklists to directly block malicious activity?
(15) Do you use third-party blocklists as an input to a threat intelligence system?
(16) In your experience, do third-party blocklists provide accurate information on threats?
(17) What are the shortcomings of any third-party blocklists you are familiar with?*
(18) What are the strengths of any third-party blocklists you are familiar with?*
(19) How do your filtering practices vary according to type of attack or blocklist?*
(20) To help us map your responses to the blocklists we are monitoring, please list the third-party blocklists you use.*
(21) Do you see the quality of blocklists being affected by: Dynamic addressing
(22) Do you see the quality of blocklists being affected by: Carrier grade NATs
(23) Do you see the quality of blocklists being affected by: Other*
(24) How could blocklists be improved?*
(25) Do you donate data from your network to community blocklist sources (such as Project Honeypot or DShield)?
(26) Is there anything else you would like to share with us?*