IPv6 Hitlists at Scale: Be Careful What You Wish For

Erik Rye
University of Maryland

Dave Levin
University of Maryland

ABSTRACT

Today's network measurements rely heavily on Internet-wide scanning, employing tools like ZMap that are capable of quickly iterating over the entire IPv4 address space. Unfortunately, IPv6's vast address space poses an existential threat for Internet-wide scans and traditional network measurement techniques. To address this reality, efforts are underway to develop "hitlists" of known-active IPv6 addresses to reduce the search space for would-be scanners. As a result, there is an inexorable push for constructing as large and complete a hitlist as possible.

This paper asks: what are the potential benefits and harms when IPv6 hitlists grow larger? To answer this question, we obtain the largest IPv6 active-address list to date: 7.9 billion addresses, 898 times larger than the current state-of-the-art hitlist. Although our list is not comprehensive, it is a significant step forward and provides a glimpse into the type of analyses possible with more complete hitlists.

We compare our dataset to prior IPv6 hitlists and show both benefits and dangers. The benefits include improved insight into client devices (prior datasets consist primarily of routers), outage detection, IPv6 roll-out, previously unknown aliased networks, and address assignment strategies. The dangers, unfortunately, are severe: we expose widespread instances of addresses that permit user tracking and device geolocation, and a dearth of firewalls in home networks. We discuss ethics and security guidelines to ensure a safe path towards more complete hitlists.

CCS CONCEPTS

• **Networks** → **Network measurement**; **Network privacy and anonymity**.

KEYWORDS

IPv6, hitlists, passive measurement

ACM Reference Format:
Erik Rye and Dave Levin. 2023. IPv6 Hitlists at Scale: Be Careful What You Wish For. In *ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23), September 10, 2023, New York, NY, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3603269.3604829

1 INTRODUCTION

ZMap [19] revolutionized Internet scanning in 2013, enabling scans of the entire IPv4 Internet in under an hour. Since then, Internet-wide scanning has become one of the most powerful and commonly used tools for measurement researchers and practitioners, leading to previously inaccessible findings in security [4, 12, 15, 18, 55], topology discovery [9], IoT measurement [30], outage detection [53], and more.

Unfortunately, the continued, accelerating deployment of IPv6 represents an existential crisis for Internet-wide scanning. In contrast to IPv4, brute-force scanning of every IPv6 address is impossible owing to IPv6's exponentially larger address space (2^{128} compared to IPv4's 2^{32}).

Without the prospect of iterating over all IPv6 addresses, some scanning tools instead rely on "hitlists" that identify addresses that are likely to be active and in-use, and probe only those. Others have introduced IPv6 target generation algorithms that emit potentially-active IPv6 addresses as probing candidates; these models must be trained on *some* hitlist and are biased to the types of addresses contained in their training data. Thus, the larger and more representative these hitlists are, the more complete the view of the IPv6 Internet available to measurement efforts [58].

As a result, the "holy grail" of Internet scanning is a complete list of all active IPv6 addresses. There are ongoing efforts approximate such a list. To date, the largest public list is the *IPv6 Hitlist* [24, 75]. The IPv6 Hitlist uses a variety of active measurement techniques—namely ZMap6 [70] and Yarrp [9]—and target generation algorithms to discover new responsive IPv6 addresses.[1] Their most recent efforts in 2022 nearly tripled IPv6 Hitlist's size to a total of 8.8M addresses [75].

In this paper, we ask: what are the potential benefits and harms when IPv6 hitlists grow larger? To answer this question, we obtain the largest list of active IPv6 addresses to date: 7.9 billion addresses, 898 times larger than the current IPv6 Hitlist snapshot and 370 times larger than the Hitlist over the same time interval. Our corpus contains nearly 15 times more active IPv6 addresses than *all* of the active IPv6 addresses discovered in the IPv6 Hitlist's four-year history though it was collected in about 15% of the time. In contrast with the active measurements employed by the IPv6 Hitlist, we passively collected the nearly 8 billion active IPv6 addresses in our corpus by running a set of 27 geographically-distributed NTP servers as part of the NTP Pool [3], which devices worldwide use to synchronize their clocks.

Although our hitlist is not comprehensive—for instance, we are likely missing most modern Android devices because they are not configured to use the NTP Pool by default—it is a significant leap forward from the current state of IPv6 hitlists. As such, it provides a glimpse into what the future holds as the community moves toward its goal of a more comprehensive hitlist.

Our glimpse into the future of IPv6 hitlists produces two broad results: that network measurement insights are indeed improved through a bigger list (especially one that is passively collected like

ACM SIGCOMM '23, September 10, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0236-5/23/09.
https://doi.org/10.1145/3603269.3604829

[1] We use capitalization to differentiate between hitlists in general (lowercase) and the IPv6 Hitlist in particular (uppercase).

ours), and that large hitlists poses significant security and privacy risks.

Benefits: New insights from larger hitlists What is gained from having larger IPv6 hitlists? What do current hitlists lack, and does filling in those gaps facilitate a deeper understanding of the IPv6 Internet?

We find that a significant missing portion of prior IPv6 hitlists is *end-host addresses*. The IPv6 Hitlist, for example, relies primarily on ZMap6 and Yarrp, and as a result discovers many infrastructure nodes (especially routers and CPE), but has difficulty identifying end-hosts due to firewalling and frequently unpredictable and ephemeral client addresses [50]. Conversely, our dataset is comprised largely of end-hosts, though not exclusively (virtually all Internet devices must synchronize their clocks).

We demonstrate that when a hitlist has more end-host addresses, it enables investigations that are impossible with existing hitlists, such as studying client address entropy, IoT detection, and address assignment patterns. However, these new insights are not without cost.

Threats: New privacy leaks from larger hitlists Conventional wisdom dictates that IPv4 addresses do not constitute personally identifiable information (PII). This is largely because the link between a device and an IPv4 address is very weak: addresses are assigned randomly and are frequently reassigned, and multiple devices often map to the same address at any time. However, the same is not always true for IPv6 addresses. It is now well-known that IPv6 addresses can risk uniquely identifying a client device—e.g., if the device embeds its MAC address into the lower-order bits of its IPv6 addresses, then it could potentially be tracked as it moves across networks.

These privacy issues were not a major concern for previous IPv6 hitlists, as they focused primarily on infrastructure nodes, which typically have no active user and do not move much across networks. However, more complete hitlists will inevitably include more client devices.

We perform what we believe to be the first analysis of the privacy risks inherent to large, client-rich IPv6 hitlists. We observe nearly 15 million clients in multiple networks across four distinct types of address tracking, and apply recent precision IPv6 geolocation techniques to hundreds of thousands of addresses.

Collectively, our results show that there is significant promise in store for larger IPv6 hitlists and Internet scanning, but also potential harm in making larger hitlists public.

Contributions To summarize, we make the following contributions:

- We collect and report on the largest publicly-obtained IPv6 hitlist to date: over three orders of magnitude larger than prior public hitlists.
- We perform a thorough comparison with the two premiere hitlists in use today—the *IPv6 Hitlist* [75] and CAIDA's routed /48 dataset [14]—finding that our NTP-derived dataset is far larger and comprises more end-hosts, but that each dataset provides a complementary perspective on active IPv6 addresses.
- We show that larger, more client-centric IPv6 hitlists enable new discoveries that prior hitlists do not, particularly in address patterns and aliased network discovery.
- We also show that larger, more client-centric IPv6 hitlists enable new *attacks* on privacy in the form of tracking and geolocation.
- We discuss the ethical ramifications of our findings, and provide guidelines that we hope will help shape future efforts in obtaining and sharing more complete IPv6 hitlists.
- We make the active /48 prefixes we discovered publicly available at `https://v6-research.cs.umd.edu`

2 BACKGROUND AND RELATED WORK

2.1 Why creating IPv6 hitlists is hard

Applications ranging from census and adoption studies [17, 52], to vulnerability identification and remediation [41], to outage detection [39, 54, 59, 67] all rely on an understanding of what IP addresses are assigned to live hosts. In IPv4, identifying these hosts is tractable due to stateless scanners like ZMap [19] and Yarrp [9] that can probe the IPv4 Internet in minutes. However, several factors complicate live address discovery and hitlist creation in IPv6.

The first and most obvious complicating factor is the immensity of the IPv6 address space. This leads to large prefixes—at least /64 and frequently /56 or larger [61]—being delegated to even residential customers. Thus, the average Internet subscriber has 4 billion or more times the number of IP addresses in the entire IPv4 Internet available to her, all of which are publicly routable; contrast this with IPv4, in which she commonly has one public IP, with her home network hidden behind a NAT. With such large prefixes delegated to customers with only tens or hundreds of devices, IPv6 is considerably more sparse than IPv4. Further, some service providers delegate prefixes to their customers for only short periods (e.g., 24 hours) before recuperating them and issuing new ones [64].

The lower 64 bits of a 128-bit IPv6 address is called the Interface Identifier (IID), and its assignment adds further complexity to creating IPv6 hitlists. Some IPv6 addresses are manually assigned: these are often infrastructure devices that network administrators prefer to assign easily memorable IIDs for troubleshooting [62], like `::1` or `::2`. Occasionally, some network operators will embed the IPv4 address assigned to the same interface in the IID [10], but there is no requirement to do so and these embeddings are relatively uncommon. Often, hosts self-assign IIDs via one of several processes. Extended Unique Identifier - 64 (EUI-64) Stateless Address Autoconfiguration (SLAAC) [49] embeds the interface's Media Access Control (MAC) address in the IID, after inverting the Universal/Local bit of the MAC and inserting a `0xFF 0xFE` in-between the third and fourth bytes. Because these addresses embed a static, link-layer identifier in the IPv6 address that allows for device identification and tracking, as well as attacks tailored to device manufacturers, generating ephemeral random addresses has instead been encouraged since 2001 [50]. However, ephemeral addresses are problematic for servers, which should possess stable yet privacy-preserving addresses for long periods of time. In response, standards have been proposed for generating stable, random addresses [28]. Finally, DHCPv6 [46], the less-ubiquitous cousin of its IPv4 analog, also exists to assign addresses to hosts. However, given such large prefix allocations, it is up to the DHCPv6 implementation and operator configuration how client addresses are assigned.

Even responsive addresses in IPv6 may not indicate a live host. In IPv6, *aliasing*, in which a single device replies to probes to an entire network, is a relatively common practice. This necessitates a filtering step in hitlist creation, wherein responsive addresses from aliased networks are removed.

2.2 Related Work

There have been extensive efforts to discover new live IPv6 addresses through both active and passive measurements.

Active approaches Beverly developed the Yarrp high-speed, stateless traceroute tool to improve host and topology discovery in both IPv4 and IPv6 [9], and Beverly et al. used Yarrp to discover significant IPv6 Internet core topology [11]. Similarly, Gasser et al. developed ZMap6 [70] IPv6 extensions of the ZMap [19] high-speed scanner to enable fast IPv6 scanning without tracing to intermediate hops [25]. Gasser et al. used a combination of ZMap6, scamper traceroutes [38], and public data sources to develop an IPv6 hitlist and identify aliased networks [24]. They continue to publish a weekly hitlist of responsive addresses and known aliased and non-aliased networks [1]. Rye and Beverly used Yarrp and ZMap6 to focus discovery of topology at the network periphery (Customer Premises Equipment (CPE) devices in customer ISP networks) [65] and characterized high-frequency customer network changes [64]. Others have enumerated reverse DNS zones to discover active IPv6 addresses [13, 21, 69]. Foremski et al. [22] aggregated datasets comprising more than 3.5 billion IPv6 addresses from cloud providers and ISPs to develop new candidate addresses for active measurements. Numerous machine learning models [68] have been trained on responsive addresses in order to generate candidate addresses for active measurements, using Bayesian networks [47], reinforcement learning [32], generative adversarial networks [16], divisive hierarchical clustering [37], and ensemble learning [72].

As we will demonstrate, our passive approach is largely complementary to these active efforts, exposing portions of the in-use IPv6 address space that the active efforts alone do not reach.

Passive approaches Numerous passive IPv6 efforts also have studied IPv6 addressing. Gasser et al. [24] and Huz et al. [33] both crowd-source small numbers of IPv6 client addresses via Mechanical Turk [6] and Prolific Academic [2]. A major barrier to conducting large-scale passive IPv6 measurements, however, is access to proprietary datasets. Plonka and Berger [56] use IPv6 addresses gathered from a large CDN's webservers to determine how customer addresses change over time. Using data obtained from Facebook, Li and Freeman [36] examine how client IPv6 addresses change over time and consider the problem of handling abusive or malicious actors in IPv6. Saidi et al. [66] examine aggregated IPv6 client traffic, including NTP, provided by a large European ISP to track customer subnet allocations over time by tracking clients employing EUI-64 IPv6 addresses. Enayet and Heidemann use data from the DNS B root to detect outages in IPv4 and IPv6 [20], while Fukuda and Heidemann use DNS backscatter from the B root to detect IPv6 scanning [23].

Comparison of passive and active approaches Several works compare active and passive measurement approaches, albeit only in IPv4 and on a far smaller scale than our work. Bartlett et al. compared passive monitoring and active probing to discover services running on a university network [8]; Heidemann et al. similarly used passive and active approaches to discover IPv4 end hosts in 2008 [31]. Zander et al. used several passive sources of active IPv4 addresses, including Wikipedia edits and NetFlow records, to augment active measurements to estimate IPv4 address space utilization [73].

Unlike prior active efforts, we avoid sending millions of unsolicited probes in search of active addresses. Unlike many prior passive efforts, our technique does not require access to privileged datasets obtained by private organizations. Rather, we demonstrate that enormous amounts of IPv6 data can be obtained by contributing to an open service—the NTP Pool—which we review next.

2.3 NTP and the NTP Pool

The Network Time Protocol (NTP) is one of the Internet's oldest protocols, standardized in 1985 in RFC958 [44]. NTP synchronizes a device's clock with a remote time server, even in variable-delay networks. Keeping accurate time is of immense importance to a variety of applications ranging from TLS certificate validation [40, 74], to authentication [35, 40, 45], to DNS cache entries [40]. Most devices on the Internet today synchronize their clocks using NTP.

Where a device looks for its time is typically a function of its operating system. Windows clients and servers, for instance, synchronize their time with `time.windows.com` [43] by default when not joined to a domain. Likewise, Apple devices synchronize with `time.apple.com`. Android clients until Android 8 (Oreo) used the NTP Pool (`pool.ntp.org`, discussed next); later versions now use `time.android.com` [27]. NTP servers can additionally be specified via DHCP [5] and DHCPv6 [26] options.

The NTP Pool Project [3] provides NTP service through a worldwide set of geographically distributed NTP servers, many of which are contributed by volunteers. Any host with a publicly reachable IP address can serve in the NTP Pool. The Pool preferentially directs clients to servers geographically near them, using a combination of IP geolocation and DNS round robin. The NTP Pool Project further provides various "vendor zones" to equipment and software vendors to use as defaults on their devices; vendor zones for Android, Ubuntu, and CentOS exist, among many others (`android`, `ubuntu`, and `centos.pool.ntp.org`, respectively.)

3 METHODOLOGY

This section first highlights the measurement infrastructure we established to conduct our experiments. Then, it introduces the other IPv6 address datasets that we compare to our NTP-derived corpus. Finally, we discuss how we geolocate NTP client addresses and our methodology for probing back to active NTP clients that query our servers.

Vantage Points In order to measure the effectiveness of using NTP servers as large-scale, longitudinal, passive IPv6 measurement infrastructure, we operated 27 NTP servers from 25 January 2022 through 31 August 2022. We chose Virtual Private Servers (VPSs) from 20 countries across 6 continents to obtain geographic diversity.

Dataset	Dates	IPv6 Addresses		ASNs		/48s		Avg. Addrs
		Num.	Common	Num.	Common	Num.	Common	per /48
NTP Pool (This paper)	Jan–Aug '22	**7,914,066,999**	–	9,006	–	7,205,127	–	**1,098**
IPv6 Hitlist [1]	Feb–Aug '22	21,409,629	277,026	**18,184**	7,560	431,851	267,908	50
CAIDA Routed /48 [14]	Feb–Apr '22	11,613,494	3,117	13,770	6,957	11,111,563	102,864	1

Table 1: Comparison IPv6 datasets considered. Our NTP corpus is passively collected, while the comparison datasets used active techniques. "Common" denotes the intersection of the comparison data with our data.

Specifically, we ran 6 servers in the US, 2 in Japan, 2 in Germany, and 1 server in each of: Australia, Bahrain, Brazil, Bulgaria, Hong Kong, India, Indonesia, Mexico, Netherlands, Poland, Singapore, South Africa, South Korea, Spain, Sweden, Taiwan, and the United Kingdom.

Though we ran servers in 20 countries, the NTP Pool's load balancing allowed us to collect data from 175 countries[2] in total. The majority of the IPv6 addresses we discovered came from India (1.9B), China (1.6B), US (1.2B), Brazil (700M), and Indonesia (630M), collectively accounting for 76% of our entire dataset. The other 170 countries accounted for 24%.

We emphasize that each VPS was minimally provisioned and cost on average *less than $7 per month* to operate. We typically used one virtual core, 500MB–2GB of RAM, and a Linux OS available from the VPS vendor (Ubuntu, Amazon Linux, or CentOS).

Each of these VPSs was configured as a stratum-2 NTP server and joined to the NTP Pool. Because the NTP Pool directs clients to its NTP servers via a DNS round-robin that incorporates the client's coarse-grained IP geolocation, our globally-distributed NTP servers were visited by a wide range of clients around the world.

Comparative Datasets In order to compare our passive NTP results with contemporaneous, state-of-the-art IPv6 measurements, we acquired two external datasets. First, we compare against the IPv6 Hitlist [24], which provides a list of responsive addresses and networks that the operators detect as being aliased networks (responsive on all addresses) and those that are not. The Hitlist is updated on roughly a weekly basis, so in order to best compare our dataset to their data, we consider all Hitlist responsive IPv6 addresses published during our study's time frame. IPv6 Hitlist data concurrent with our study was published first on 16 February 2022 and runs through 29 August 2022. While our study collects only NTP requests, the IPv6 Hitlist is obtained through active measurements using ICMPv6, TCP ports 80 and 443 (HTTP and HTTPS), and UDP ports 53 (DNS), 161 (SNMP), and 443 (QUIC). Our IPv6 Hitlist comparison dataset consists of 21,409,629 unique IPv6 addresses.

Second, we leverage a large dataset of 1,083,188,032 Yarrp traces conducted by CAIDA from their Archipelago distributed measurement system [34] between 3 February and 6 April 2022. Their measurement methodology splits each prefix of length /32 or longer into /48s and probes the `::1` address of each /48. For prefixes of length less than /32, only a single `::1` address is probed, with no splitting into constituent /48s. These traces discovered 11,613,494 live addresses. We refer to this measurement as the "CAIDA routed /48" dataset throughout.

Table 1 lists the relevant details of each dataset involved in our study. We will explore these numbers in more detail in §4, but even at a glance it is clear that the NTP data corpus comprises three orders of magnitude more addresses, with a greater density of addresses on average in each /48.

Geolocation We consider two different types of geolocation in our results. First, we used MaxMind's GeoLite2 City database [42] to geolocate the NTP client addresses we observed. While fine-grained IP-geolocation is often error-prone, particularly in IPv6, we consider only the country reported by MaxMind in aggregated results and do not use the more granular geolocation data it reports.

Second, for IPv6 addresses with embedded MAC addresses in the form of EUI-64 Interface Identifiers (IIDs), we attempt to link this embedded MAC address with a wireless MAC address from the same device obtained from a geolocation service. EUI-64 IIDs are constructed by first inverting the seventh least significant bit of the most significant byte of the interface's MAC address. Then, a static `0xFF 0xFE` is inserted between the third and fourth bytes of the MAC address to create a 64-bit identifier; this is then used as the lower 64 bits of the IP address in an EUI-64 IPv6 address. Recovering interface MAC addresses is a simple process — the `0xFF 0xFE` bytes of an EUI-64 IIDs are removed, followed by the seventh bit's inversion.

Both Google and Apple both offer geolocation APIs [7, 29], and other individual and community projects also collect wireless geolocation information [48, 51, 57, 71]. Our methodology for linking wired EUI-64-derived MAC addresses to wireless MAC addresses follows that of Rye and Beverly [63], wherein they form a linkage between the most commonly-arising offsets between pairs of wired and wireless identifiers from within the same vendor-assigned three-byte address prefix, called an Organizationally Unique Identifier (OUI). This is often, but not always, the closest match between wired and wireless MAC addresses within the same OUI. This methodology is also limited to devices that have wireless and wired MAC addresses from the same OUI.

Backscanning In order to compare and contrast active and passive methodologies for compiling IPv6 hitlists, we actively probed NTP clients that visited five of our 27 NTP servers over the course of a week during January 2023. During this week, we recorded the source addresses of NTP clients that queried the servers over ten minute intervals. When the interval concluded, we initiated traceroutes from the servers back to the clients using Yarrp and sent ICMPv6 Echo Requests to the clients with ZMap6. The probe targets were both the NTP client address that had queried the NTP server, as well as a random IPv6 address within the same /64 as the client. All probes used ICMPv6 in order to minimize potential

[2] We count ISO-3166-1 two-letter country codes and use the term "countries," although some are dependent territories.

disruption to the probed addresses; no IP was probed more than once during a 10 minute interval.

Ethical Considerations During this study, we accumulated nearly 8 billion unique IPv6 addresses by adding 27 NTP servers to the NTP Pool. Users of the NTP Pool had no way of knowing their NTP request data could or would be used in our study. That said, our study follows the same general principles of prior IPv6 Hitlist generation [24] as well as other peer-offered infrastructure, such as studies that use BitTorrent to collect and study IP addresses [60]. Like those studies, we do not collect any PII that might be included in the application-layer data (NTP requests do not contain PII).

Novel to our findings, however, is that large sets of IPv6 addresses may in and of themselves contain enough information to track and geolocate users. These attacks on privacy are made possible through the lower-order bits (specifically, we make use of EUI-64 IIDs). Thus, to avoid spreading this potentially sensitive information, we will only be releasing our dataset at the /48 level. This is an ethical consideration that future IPv6 hitlists must contend with: what is an appropriate way to share hitlists so as to enable Internet scanning tools to use them?

We communicated with the project owners of the NTP Pool to inform them of our experiments and to ensure that we were abiding by both the terms of service and acting in a way that preserved the privacy of the NTP Pool users. They concurred that we were not violating any NTP Pool policy or community standard and requested that address data released be aggregated to protect user privacy.

We also submitted our study to our institution's IRB for review. Much like other institutions' IRBs [24], they did not consider IP addresses as constituting human data. However, after informing them of our privacy results (§5), they agreed that further future consideration would be appropriate. Our hope is that this paper can be a first step towards the networking community helping to guide appropriate methods for ethically sharing IPv6 hitlists as they continue to grow.

Finally, contrary to active measurements that introduce immense volumes of superfluous data for the sole purpose of eliciting responses from remote devices (e.g., `traceroute` and `ping`), our experiments actually *provided a beneficial service* to the devices we measured. All of our servers provided stratum-2 NTP service and are located in cloud providers with exceptionally high availability, providing a reliable source of accurate time for NTP clients. Therefore, we believe that the benefits of our work outweigh the potential harm or risk that it may present.

4 BENEFITS OF LARGER HITLISTS

We begin our analysis by evaluating whether larger IPv6 hitlists confer benefits: as the community races to obtain larger hitlists, is it worth it? To this end, we first show that our NTP-based dataset is not only larger, but nearly disjoint with and complementary to other state-of-the-art IPv6 measurements. Then, we evaluate various applications of this larger hitlist: measuring aliased networks and analyzing IPv6 addressing patterns.

4.1 How Do the Datasets Compare?

First, we compare our dataset to the IPv6 Hitlist and a large-scale active measurement conducted by CAIDA.

In terms of size Table 1 compares the aggregate number of live IPv6 addresses we observed during our study to the number of responsive addresses collected by the IPv6 Hitlist project and during a large-scale, active measurement campaign by CAIDA. Running 27 IPv6 NTP servers from January through August 2022, we observed 7.9 billion unique IPv6 source addresses—370 times more than the IPv6 Hitlist's collections over a comparable period of time. The CAIDA measurement discovered 11,613,494 addresses during its two-month run, 681 times fewer than the NTP corpus.

In terms of addresses discovered Despite the massive difference in size, our dataset did not subsume either of the active datasets; we only discovered 1.3% (277,026) of the addresses that IPv6 Hitlist found, and a mere 0.02% of the IPv6 addresses CAIDA's routed /48 dataset discovered. This shows that the datasets are indeed complementary, and suggests that the kinds of devices we are finding are in fact distinct. We confirm this hypothesis later in this section.

In terms of Autonomous Systems (ASes) While the number of raw addresses in the NTP corpus dwarfs the other dataset address counts by several orders of magnitude, this trend is reversed in the number of ASes we observe. Our NTP corpus contains 65.3% of the number of ASes observed in the CAIDA scan (9,006 vs 13,770) and 49.5% of the number discovered in the IPv6 Hitlist data (9,006 vs 18,184). This discrepancy is likely due to the nature of the two active datasets, which use `traceroute`-like tools to discover Internet infrastructure between their vantage point and their probe targets. Our data, coming from NTP clients, is concentrated in ASes where NTP Pool clients exist, typically in customer ISPs. This hypothesis is strengthened by examining the types of ASes the different corpora addresses originate from, as classified by ASdb [76]. While the top AS type is consistent between all three datasets ("Computer and Information Technology","Internet Service Provider (ISP)"), an additional 14% (1,146,709,677) of our NTP Pool corpus originates from "Phone Provider" ISP subtype. By contrast, only 2% of the IPv6 Hitlist addresses originate from "Phone Provider" ASes, indicating that the NTP Pool corpus consists of a higher percentage of mobile clients than do either of the two active datasets.

In terms of prefixes Though the NTP Pool corpus IPv6 addresses are concentrated in fewer ASes than either of the two active measurements, our NTP dataset exhibits the highest number of addresses discovered per /48 (Table 1). The NTP dataset discovers a mean of 1,098 addresses per /48, while the IPv6 Hitlist uncovers 50 and the CAIDA scanning only 1. This "address density" has at least two potential root causes. First, NTP Pool clients may more commonly be client devices that frequently change their (random) IID in order to prevent tracking, as is considered best practice for client devices [50]. This phenomenon would manifest as many different addresses originating from the same prefix—for instance, a /56 or /64 allocated to a residential deployment by a customer ISP. A second possibility is that our passive NTP methodology detects more customer deployments than either of the active probing methods. While we detect devices in customer deployments so long as they visit an NTP Pool server, these networks are often highly subnetted

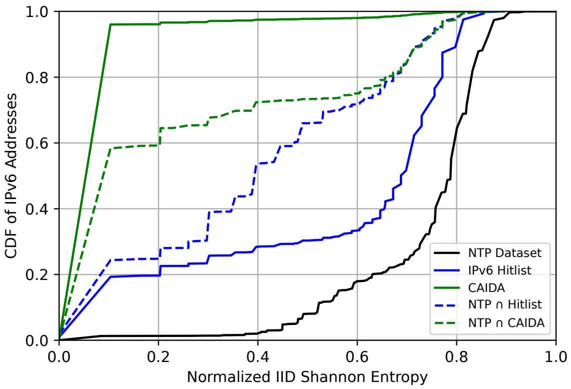


Figure 1: IID entropies of the IPv6 addresses from the NTP Pool, IPv6 Hitlist, and CAIDA routed /48 corpora, as well as the IID entropies of their intersections.

and would require significant active probing to discover the CPE devices, which may or may not respond. Neither the IPv6 Hitlist nor CAIDA scanning is specifically calibrated to do such fine-grained, residential deployment discovery. Note that these two root causes are not mutually exclusive; both may manifest in our data.

In terms of device type It is in general very difficult to perform device fingerprinting at scale. However, we can fortunately make use of features of the IPv6 addresses themselves to gain some insight into the kinds of devices that comprise the respective datasets.

We examined the IIDs, or lower 64 bits, of the addresses in each dataset. Because IIDs uniquely identify a host interface on a network, and because different types of hosts are subject to different concerns (e.g., preventing client tracking, static server addresses, or infrastructure addresses easily memorable by network administrators), the randomness (or lack thereof) of an IID may vary significantly between device types.

Figure 1 plots the CDF of all addresses found in each dataset versus IID entropy, using the normalized Shannon entropy as a metric. The NTP Pool corpus exhibits significantly higher entropy than the other two datasets, with a median normalized Shannon entropy of approximately 0.8. The IPv6 Hitlist has a somewhat lower median entropy of about 0.7, while almost the entirety of the CAIDA dataset has extremely low entropy.

These data points reinforce the hypothesis that *our NTP Pool dataset consists primarily of client devices*, which often use ephemeral, random addresses to defend against long-term tracking. The CAIDA dataset, by contrast, discovers mainly core Internet infrastructure, which is often manually addressed by operators with an incentive to create easily-memorable addresses. The IPv6 Hitlist occupies a middle ground of sorts, discovering both core Internet infrastructure, as well as some higher-entropy addresses assigned to CPE devices at the network periphery. In the next subsection, we further investigate address entropies within our dataset to illuminate differences in IPv6 addressing schemes between service providers.

Observed address durations The durations over which we observe distinct IPv6 addresses vary significantly, as shown in Figure 2. Figure 2(a) displays a CCDF of the lifetimes with which we observe all of the 7.9 billion addresses in our dataset. More than 60% of them are observed only once (a "lifetime" of 0 seconds in the plot). With purely passive measurements, we cannot determine whether this is because the devices had highly ephemeral addresses, or simply because they only contacted our (or anyone's) NTP servers only once. This demonstrates the importance of combining passive collection with active scans; so long as a device shows up even once at our servers (e.g., at boot-up), it can be used in subsequent backscanning.

At the opposite extreme, 95,780,865 (1.2%) IPv6 addresses are observed for a week or longer, 32,985,774 (0.4%) IPv6 addresses for a month or longer, and 2,218,998 (0.03%) IPv6 addresses for more than six months. Figure 2(b) is a CDF of the 670,737,407 unique IIDs we observed, binned by the entropy of the IID. This figure shows that while ∼10% more of the low normalized entropy (< 0.25) IIDs appear only once in our corpus than medium or high entropy IIDs, low entropy IIDs are more likely to persist for long periods of time. In fact, 10% of all low entropy IIDs are observed for a week or more, as compared to 5% or less of the medium and high entropy IIDs.

Summary Collectively, these results show that our hitlist of IPv6 addresses has little overlap with prior datasets, and in particular adds more end-host devices than any prior efforts. This is a natural progression in IPv6 hitlists, but one that would have been much more difficult with purely active measurement-based approaches.

4.2 Backscanning and Aliased Networks

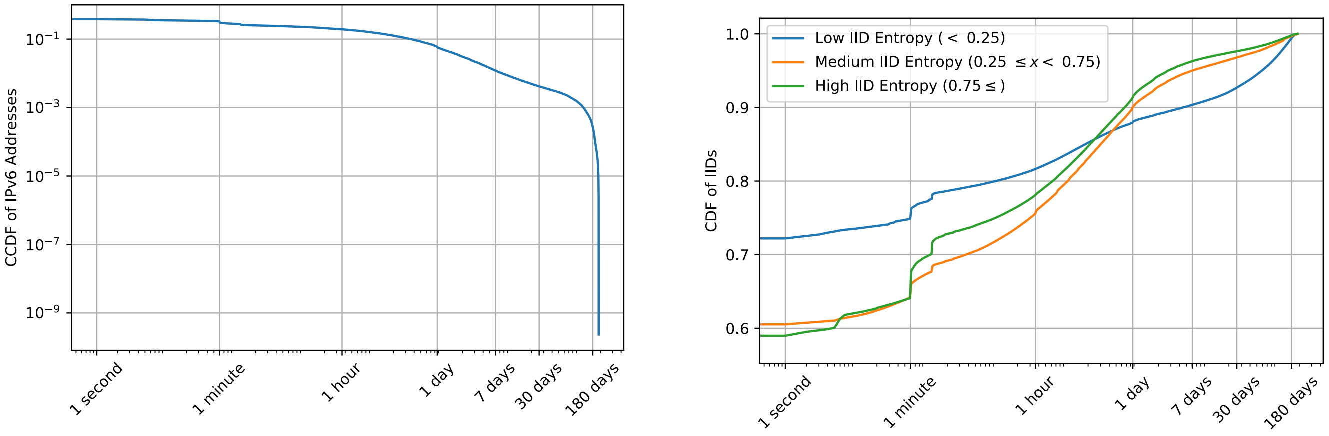
The IPv6 addresses we discovered are only useful for Internet scanning insomuch as they are responsive to outside scans. Moreover, if the addresses we learned are merely aliases of one another, then the volume of them would not be useful. To test both of these, we initiated Yarrp traces and ZMap6 probes from five NTP servers back to the clients that contacted them, as well as to a random address in the same /64 (see §3).

Responsiveness to backscanning As described in §3, we initiated a limited number of ICMPv6 scans back to NTP clients over the course of a week in January 2023. About two-thirds of the 71,341,581 NTP clients that were probed from the NTP servers responded to Yarrp or ZMap6 probes. This shows that the addresses we obtained can be used as scan targets.

In contrast to the NTP clients we probed, the random targets we probed in the same network as the NTP clients responded only 3.5% of the time. Because it is exceedingly unlikely that we guessed a live random-IID IPv6 address, these responses almost certainly originate from aliased networks.

Figure 3 displays the IID entropy of the responsive addresses broken down by whether the address was responsive to backscanning ("NTP hit") or not ("NTP miss") or if the address was randomly chosen yet responsive ("Random"). The responsive NTP client addresses exhibit the lowest median normalized IID Shannon entropy, although they still exhibit higher entropy than addresses in our comparison datasets.

The higher the IID entropy, the less likely the end-host was to respond. Nearly 70% of the unresponsive NTP clients had normalized entropy greater than 0.75, compared to only ∼50% of the responsive



(a) CCDF of distinct address lifetimes over all IPv6 addresses observed during our study.

(b) CDF of IID lifetimes by IID entropy category.

Figure 2: Address and IID lifetimes vary significantly. While most addresses and IIDs are observed only once, many persist for days or months.

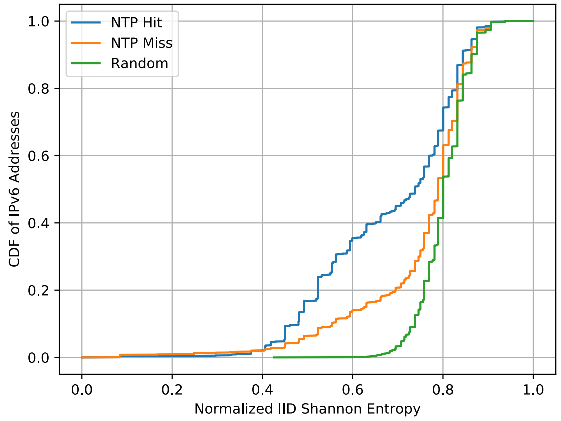


Figure 3: CDF of the IID entropy of NTP clients probed back with Yarrp and ZMap6. The clients that we passively detect from running an NTP client are responsive to back scanning, particularly those with slightly lower entropy in their IID.

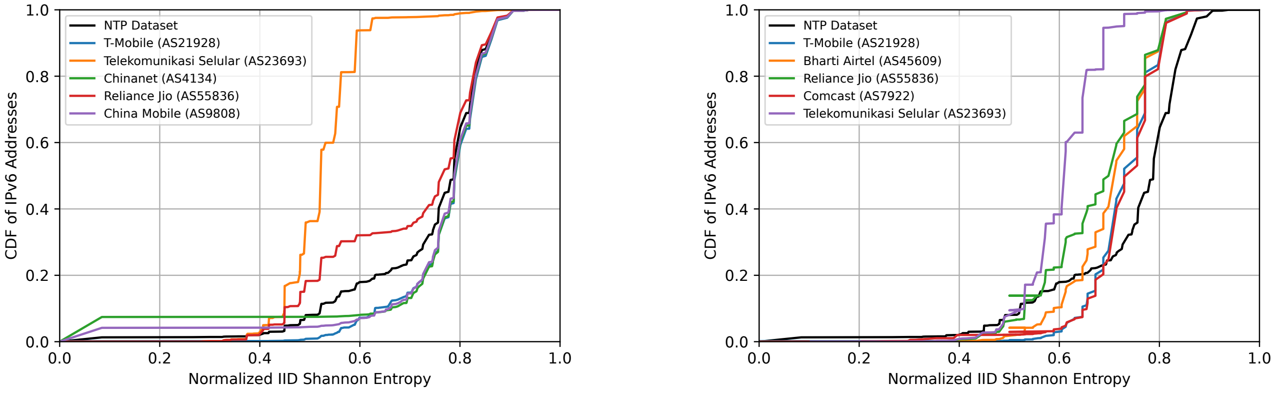
clients. We speculate that this can be attributed to infrastructure devices, which typically have stable, low-entropy IIDs, and are more likely to be responsive than client devices, such as mobile phones and personal computers. Client devices are more likely to reside behind a router or CPE device, which are often configured to block unsolicited inbound traffic, such as our ICMPv6 Yarrp and ZMap6 probes. Also, because IPv6 client addresses are often ephemeral, it is conceivable that some clients change addresses after querying our NTP servers and are no longer assigned that address when we probe it after the ten minute interval expires. Mobility may also play a factor, with devices switching to a new network in the period between sending an NTP request and when our probing began.

Discovering aliased networks As part of our backscanning, we received ICMPv6 responses to 4,476,089 unique, random IPv6 addresses we probed. These responses originated from 3,740,619 unique /64s. Because these addresses were chosen randomly from within active /64s, it is much more likely that the network in question is aliased rather than we randomly chose the address of a live host in an unaliased network. We compared our inferred aliased networks to those within the IPv6 Hitlist, which maintains a list of aliased networks. Of the 4.5 million aliased addresses we probed, the IPv6 Hitlist also categorized 4,425,001 (98%) as aliased. However, we discover an additional 46,512 aliased addresses in prefixes the IPv6 Hitlist does not recognize as aliased. This suggests that backscanning NTP clients is a potential avenue for discovering additional aliased IPv6 networks.

Finally, we examine the NTP clients in networks that we later determined were aliased via our backscanning technique. We find 3,841,751 NTP client addresses are part of aliased /64s as discovered by backscanning. These NTP client addresses originate from 36 different ASes.

We note that because the prefixes these NTP clients originate in are aliased, an active measurement campaign would be unable to distinguish these live hosts from aliased responses. That is, if it even attempted to scan the prefix in the first place—filtering known aliased networks is a best practice first step when conducting active measurements. Indeed, we searched for these addresses in a contemporaneous IPv6 Hitlist and found *only 23* addresses in our backscanning-detected /64s, while our NTP corpus contains 3,841,751.

Summary These results show that the majority of the addresses learned from our NTP-based dataset are responsive to scanning, despite the fact that most of them are clients and thus likely behind CPE devices. While this is a boon for Internet scanning, the fact that many of the devices are clients means that it is also a potential security issue; most would likely benefit from being behind firewalls. We also find that many of the addresses in our dataset are unlikely



(a) CDF of IPv6 address IID entropies observed in the top five ASes observed between Jan-Aug 2022.

(b) CDF of IPv6 address IID entropies for the top five ASes observed on 1 July 2022.

Figure 4: A comparison of the normalized Shannon entropies of collected IPv6 addresses over two time periods.

to ever be discovered with today's active measurement techniques: guessing active random IPv6 addresses or differentiating active addresses in aliased networks is impractical. Collectively, these results motivate complementing active measurements with passive data collection to obtain more complete IPv6 hitlists.

4.3 IPv6 Addressing Patterns

Our collection of IPv6 addresses permits insight into address allocation patterns, both on a macro scale, which has been previously studied [22, 24], and also at an AS level. The magnitude of addresses we obtain and the breadth of the networks from which they originate allows us to observe interesting phenomena specific to individual service providers. These phenomena are not visible from active measurements, and may help inform active measurement studies by illuminating the types of addresses present in various ASes.

Figure 4 displays two plots; each is a CDF over the total number of IPv6 addresses observed during a time interval plotted against the normalized Shannon entropy of address IID. While an imperfect proxy for randomness (e.g., the IID `0123:4567:89ab:cdef` has a normalized Shannon entropy of 1.0, but such a clear pattern might conceivably appear as the result of a network operator manually assigning it), IID entropy allows us to make generalizations about the types of addresses found within specific ASes.

Variability in entropy across ASes Figure 4(a) displays CDFs of the normalized IID Shannon entropies of the top five most commonly-observed ASes in our entire dataset, collected between January and August 2022. Three of the top five ASes (T-Mobile, ChinaNet, and China Mobile) exhibit entropy behavior that closely tracks with the NTP aggregate curve shown in Figure 1. The remaining two entropy curves, representing Reliance Jio and Telekomunikasi Selular (an Indian and Indonesian mobile provider, respectively), exhibit much lower median entropies. Further, while about 60% of the Reliance Jio addresses have high entropy (>0.75), approximately one-third exhibit a lower entropy below 0.6. This indicates that there are perhaps multiple classes of IPv6 addresses reaching our NTP servers from this network. Closer inspection reveals that at least two addressing patterns exist for hosts on Reliance Jio's network: one that randomizes all eight bytes of the IID, and another that uses the only lower four IID bytes with the remaining four set to 0.

Addressing strategies This result inspired us to investigate different patterns of addresses within ASes. To permit comparison to IPv6 Hitlist data, which is released in snapshots at intervals, we limit this analysis to a single day: 1 July 2022. Limiting this analysis to a single day also minimizes the influence that numerous random, ephemeral addresses from the same host might have over longer observation windows. Figure 4(b) shows the entropy of the top five ASes from that day in our dataset.

We compare the 46,195,900 NTP addresses we collect on 1 July 2022 to the IPv6 Hitlist's 2,970,366 IPs released on 1 July for the week prior across seven categories: (1) All zero IIDs ("Zeroes"); (2) IIDs with only the least significant byte set ("Low Byte"); (3) two least significant bytes ("Low 2 Bytes"); (4) IPv4 mapped addresses; and (5) high-entropy (>0.75); (6) medium-entropy (between 0.25 and 0.75); and (7) low-entropy (<0.25) IIDs. For IPv4 mapped addresses, which embed an IPv4 address in the IPv6 IID, we check whether any of three different embedded address encodings produce an IPv4 address in the same AS as the IPv6 address they are embedded in. We accept IPv4 embedded addresses only when i) there are at least 100 instances of them in the AS, and ii) more than 10% of the AS's total addresses are IPv4 embedded. These steps reduce false positives from random IIDs that coincidentally produce embedded IPv4 addresses in the same AS as the IPv6 address.

Figure 5 compares the frequency of these seven categories between our NTP-derived dataset and the IPv6 Hitlist. We find that the distributions of address types vary significantly between the two datasets. For the day considered, the NTP dataset is two-thirds high-entropy, with an additional 21% medium entropy. The IPv6 Hitlist, on the other hand, is approximately 20% medium and high entropy over the same time period. The fraction of IPv6 Hitlist Low

Manufacturer	Count	Manufacturer	Count
Unlisted	126,789,603	Sunnovo International Limited	1,193,746
Amazon Technologies Inc.	19,090,527	Hui Zhou Gaoshengda Technology Co.,LTD	1,067,459
Samsung Electronics Co.,Ltd	2,683,846	Huawei Technologies	876,083
Sonos, Inc.	1,633,209	Shenzhen Chuangwei-RGB Electronics	861,122
vivo Mobile Communication Co., Ltd.	1,330,987	Skyworth Digital Technology (Shenzhen) Co.,Ltd	723,316

Table 2: Number of MAC addresses extracted from EUI-64 IPv6 NTP clients by manufacturer (N = 171,611,786)

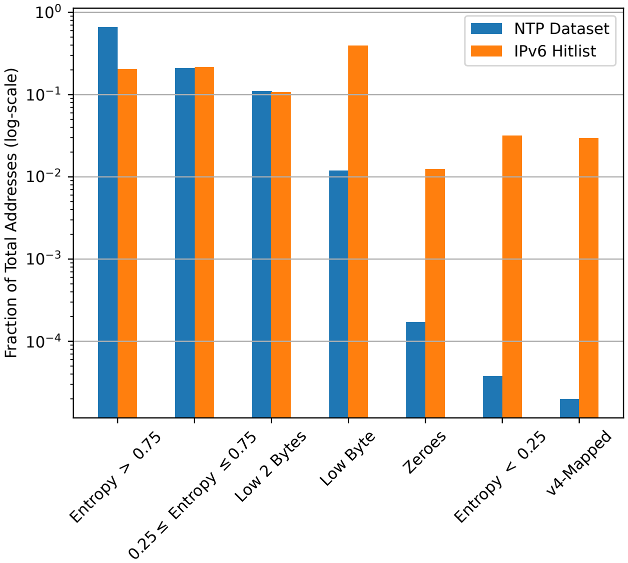


Figure 5: Fraction of the NTP corpus and IPv6 Hitlist that fall into each of seven categories for 1 July 2022. Note: y-axis is log-scale.

Byte addresses, however, is nearly 33 times that of the NTP corpus, and it contains 3% IPv4 mapped addresses compared to the NTP corpus' 0.00002%.

Taking only the IPv6 Hitlist into consideration, it would appear that the preponderance of IPv6 addresses' IIDs have only the least significant bytes set ("Low Byte"). But in our NTP dataset, the majority of active IIDs are in fact high-entropy. We believe that the Low Byte are more likely to be routing infrastructure; it is far easier for a network operator to manually set, read (e.g., in logs), and remember an IID with few bytes set (e.g., `::100`) than random ones. Indeed, because the IPv6 Hitlist relies heavily on `traceroute`-like techniques, we suspect it comprises many such routers. Nevertheless, the reality of the IPv6 Internet is that the majority of addresses are randomly set by clients: precisely the kinds of addresses that our prior results show are impractical for active measurements to obtain at scale.

Summary This example application of our NTP-derived dataset shows that *larger, more client-rich hitlists stand to improve future network measurement studies.* The push for larger hitlists is justified, at least from a measurement perspective. In the next section, we turn to whether larger hitlists come at increased security cost.

5 PRIVACY ISSUES OF LARGER HITLISTS

In this section, we perform what is, to our knowledge, the first empirical analysis of the privacy leakages in large, public hitlist datasets. We study two sources of privacy leakage: tracking via EUI-64, and geolocation by matching MAC addresses to geolocated BSSIDs (Basic Service Set Identifier, the MAC address of a WiFi access point), as in Rye and Beverly [63]. Our analysis here shows the potential harms inherent in larger hitlists.

5.1 Prevalence of EUI-64 IIDs

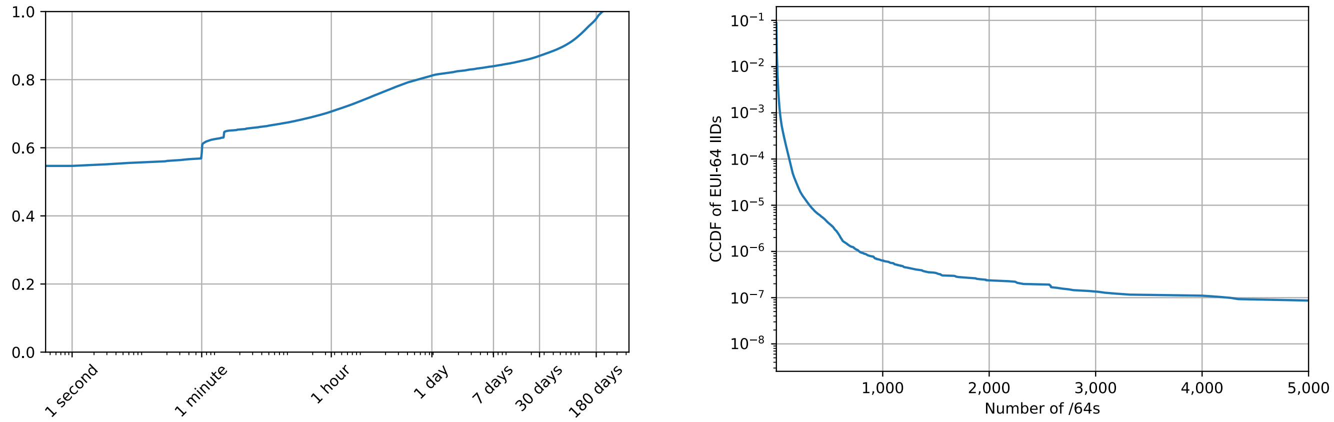
Both of the techniques we use to track and geolocate users make use of EUI-64 IIDs. EUI-64 IPv6 addresses have long been considered privacy vulnerability and have recently been studied extensively in CPE devices [63, 64] and in traffic from an ISP [66]. As our corpus consists primarily of client devices (§4) from a global network of NTP server vantage points, we were optimistic that EUI-64 IPv6 address prevalence would be low. Unfortunately, this was not the case.

Our dataset contains 238,281,703 EUI-64 IPv6 addresses: 3% of our corpus and more than the total number of *all* IPv6 addresses reported in our comparison datasets (see Table 1). Moreover, we can be certain that these are not randomly-generated addresses that *appear* to be EUI-64 due to `0xFF 0xFE` in the fourth and fifth bytes of the IID. The probability that a randomly-generated IID matches those bytes is 2^{-16}. Therefore, we would expect $\frac{7,914,066,999}{65,536}$ randomly-generated apparent EUI-64 IPv6 addresses, which is less than 121,000.

Among the EUI-64 IPv6 addresses present in our corpus, we find 171,611,786 unique embedded MAC addresses. MAC addresses can appear in multiple different IPv6 addresses for several reasons. First, when a device that uses EUI-64 addresses changes networks (e.g., due to mobility or a prefix rotation), its address will change but its EUI-64 will remain the same. Second, some device manufacturers have reused MAC address space, or assign common patterns (e.g., `00:00:00:00:00:00`), which then appear in EUI-64 IPv6 addresses for multiple devices.

To better understand the types of devices using EUI-64 addresses in our corpus, we first resolve the OUI from the MAC address extracted from the EUI-64 IID to the manufacturer listed in the IEEE's OUI database. We do this by removing the `0xFF 0xFE` from bytes four and five of the IID, and then inverting the Universal/Local bit of the resulting MAC address (the second-least significant bit of the first byte) if it is set. Table 2 contains the 10 most frequently observed manufacturers.

Surprisingly, the most common OUI we observe (126,789,603 MAC addresses, or 73.9% of all observed MACs in our dataset) is "Unlisted"—that is, they could not be resolved to a manufacturer at



(a) Lifetime of observed EUI-64 IIDs during the seven months of our study (x-axis logscale).

(b) CCDF of EUI-64 IIDs depicting the number of /64s each EUI-64 IID appears in.

Figure 6: EUI-64 IIDs permit long-term tracking of devices as they transition between network prefixes.

all. These are not merely random addresses; recall that we would expect fewer than 121,000 random IIDs to look like EUI-64. Moreover, manually inspecting the "Unlisted" addresses, we see significant MAC address counts in OUIs that are not in the IEEE OUI database. For instance, the most common "Unlisted" OUI is `F0:02:20`, with 52,218 distinct MACs embedded in EUI-64 addresses. The frequency with which this OUI appears in EUI-64 addresses makes it unlikely that these MAC addresses appear through a random process. On the other hand, 42,901 OUIs we classify as "Unlisted" appear in only one MAC address embedded in an EUI-64 address; we believe these are the randomly generated IIDs that appear to be EUI-64.

Included among the other nine most common manufacturers are makers of popular mobile, smart home, and IoT devices.

5.2 Tracking EUI-64 IIDs

Here, we evaluate the extent to which the EUI-64 IIDs in our dataset could be potentially used for tracking users.

Figure 6(a) depicts a CDF of the lifetime of all EUI-64 IIDs over the seven months of our study. While this generally tracks with all IIDs (see Figure 2(b)) EUI-64 IIDs are less likely to be observed only once (~55% compared to 60–70%) and exhibit the same long, fat tail that low-entropy IIDs do. This is due to the fact that when devices using EUI-64 change networks, they continue to use the same EUI-64 IID.

The lengthy observation window we observe for many EUI-64 IIDs demonstrates that *a passive adversary can leverage a large, longitudinal hitlist to track an unsuspecting user's device.* Because many service providers frequently "rotate" prefixes delegated to customers, often on timescales on the order of days or weeks, the ability to track a device by EUI-64 passively offers a major advantage over active techniques.

Figure 6(b) displays a CCDF of the number of /64s each EUI-64 in our corpus appears in. While most EUI-64 IIDs appear in only one /64 prefix, many appear in dozens, hundreds, or even thousands of /64s during the seven months of our study.

To determine which of the EUI-64 IIDs could be trackable users, we apply the following heuristics-based approach. We compute, for each EUI-64 IID: (1) The number of ASes it appears in; if more than 1, then we call it "high," otherwise "low." (2) The number of countries it appears in; if more than 1, then we call it "high," otherwise "low." (3) The number of transitions between different /64s it makes; if more than 10, then we call it "high," otherwise "low." If a device never changes its /64, then we consider it not trackable; of the 171,611,786 EUI-64 MAC addresses we observed, 14,943,429 (8.7%) of them appear in at least two /64s.

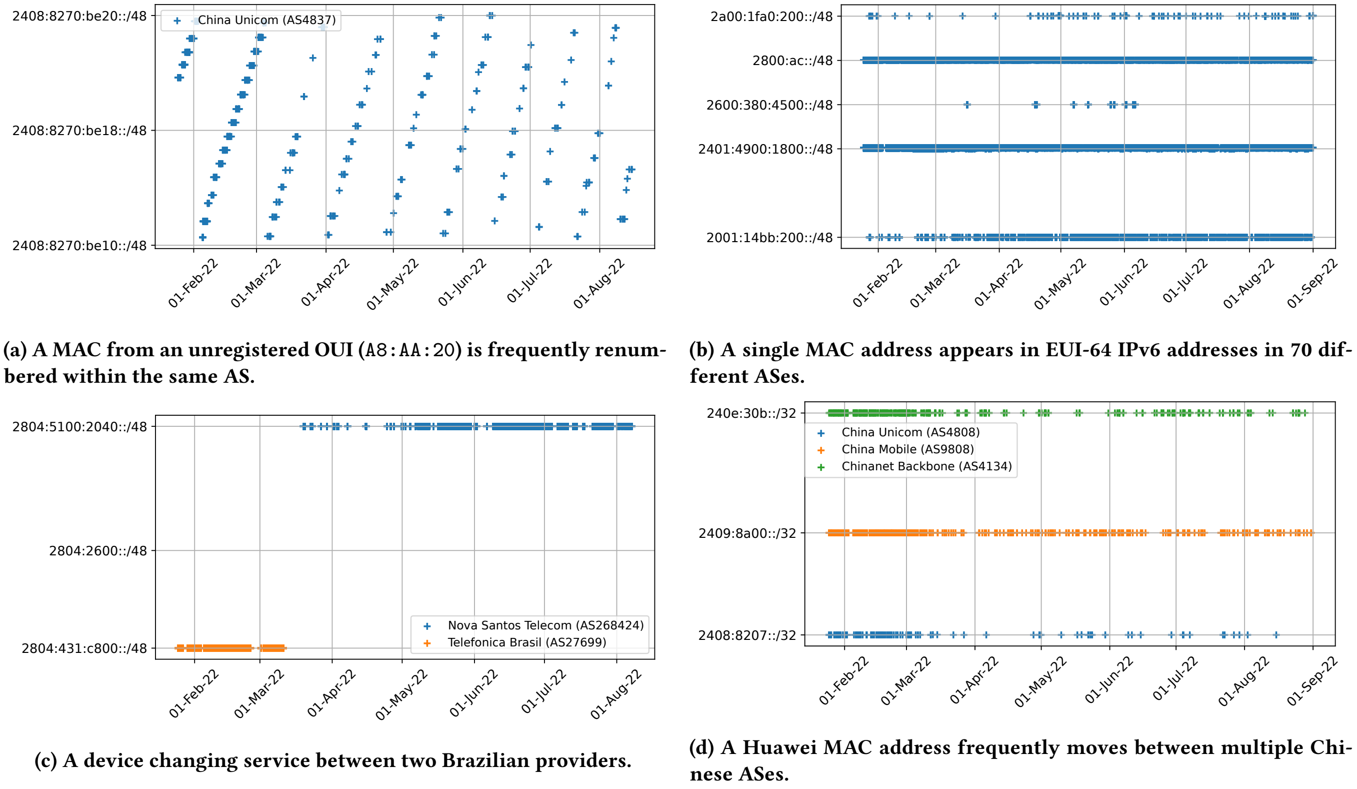
Using this heuristic, we classify EUI-64 IIDs into five categories as to the likely explanation for their re-occurrence:

Mostly static hosts The most common classification (12,853,055 of 14,943,429, or 86%) is EUI-64 IID that are labelled "low" across all three categories. These IIDs stay within the same AS and country throughout our observation period of them, and if they change /64s, they do so relatively rarely.

Likely prefix reassignment The second-most common classification (1,215,400 or 8%) appear in only one AS and one country, but /64 transitions is "high." One potential cause of this is service providers periodically reassigning new delegated prefixes to their customers Because this behavior is often based on provider policy, we observe it occurring more frequently in some providers than others. Figure 7(a) displays an exemplar of this behavior.

Likely MAC reuse In some instances (2,320 or 0.01%), we detect a single EUI-64 IID in a large number of ASes and countries, accompanied by a high number of transitions between /64. In these cases, we believe that are observing instances of MAC address reuse by a manufacturer, and that we are detecting several devices within different networks simultaneously. Figure 7(b) depicts a MAC address from EUI-64 IIDs that appear in a "high" number of countries and ASes.

Changing providers We observe 5% of devices in multiple ASes within the same country that transition a "low" number of times between /64s. This behavior could arise from a static IoT or CPE



(a) A MAC from an unregistered OUI (`A8:AA:20`) is frequently renumbered within the same AS.

(b) A single MAC address appears in EUI-64 IPv6 addresses in 70 different ASes.

(c) A device changing service between two Brazilian providers.

(d) A Huawei MAC address frequently moves between multiple Chinese ASes.

Figure 7: A variety of EUI-64 IID tracking situations arise through the use of EUI-64 IPv6 addresses.

device changing service providers. Figure 7(c) displays an example of an EUI-64 IID seen transitioning between one AS to another. For the first month and a half of our study, the device appeared only in Telefonica Brasil's network. Mid-March and later, however, the device appeared only in Nova Santos Telecom.

Likely user movement Finally, 66,187 (0.44%) EUI-64 IIDs are observed in a high number of ASes within the same country that are also classified as "high" in /64 transitions. One possible cause of this behavior is a mobile device using EUI-64 addressing; as it transitions from a home WiFi network to a cellular network, it appears in multiple ASes and frequently changes /64s. Figure 7(d) depicts a Huawei MAC address that moves between three Chinese networks frequently over time. We believe that these are indicative of addresses that permit user tracking over time, and thus pose a risk to users' PII. While low in percentage, the raw number of potential user-tracking events this permits is large and concerning.

5.3 Geolocation

Recently, Rye and Beverly [63] described a technique for geolocating CPE routers by linking MAC addresses seen in EUI-64 IIDs with BSSIDs in publicly available wardriving datasets. Here, we apply their technique to all of our EUI-64 addresses. Whereas their initial analysis relied on active scanning and thus comprised mostly CPE routers, our dataset allows us to also consider IoT and mobile devices within customers' LANs (see Table 2).

We query geolocation databases (e.g. WiGLE [71] and Apple and Google's WiFi Location APIs [7, 29]) for WiFi BSSIDs in the same OUIs as the 171,611,786 MAC addresses we derive from EUI-64 IIDs. This produces 2,692,307 distinct WiFi BSSIDs with associated geolocation data. Next, we use our EUI-64 MAC addresses and wireless geolocation data to infer the offsets between wired and wireless MAC addresses in the same OUI. For each MAC address embedded in an EUI-64 IID, we compare it to each wireless BSSID in the same OUI from our dataset, recording the offsets between each pair of two identifiers. We then tally the most common positive and negative offsets from the wired MAC to a geolocated BSSID, and select the offset with the largest number of MAC-to-BSSID matches as the "correct" offset. In this manner, we generate wired-to-wireless offsets for 117 OUIs with at least 500 wired MAC-to-BSSID pairs. Finally, we used these offsets to determine how many of the EUI-64 MAC addresses we could match with WiFi BSSIDs and therefore geolocate.

All together, our methodology links 225,354 unique MAC addresses from our collected NTP dataset with geolocated WiFi BSSIDs. Although we do not have ground truth for the geolocations of these EUI-64 IPv6 addresses, we note that prior work validated the efficacy of this technique with a large US residential ISP [63].

Our geolocations resolve to 140 different countries. However, a large majority (174,155 or 75%) of the geolocated EUI-64 IPv6 addresses are from Germany. Mexico (7%), India (4%), France (3%), and Luxembourg (2%) round out the top five countries of the geolocated EUI-64 IPv6 addresses. The over-representation of Germany and neighboring countries is due to the preponderance of AVM GmbH, the maker of the popular Fritz!Box router, MAC addresses in our geolocated EUI-64 IPv6 address set. AVM MAC addresses are responsible for 180,727 (80%) of the geolocated EUI-64 IPv6 addresses. Prior communication with AVM product security personnel confirmed this geolocation vector exists, and Fritz!OS version 7.50 eliminated EUI-64 WAN addresses support in December 2022.

Due to the prevalence of client addresses in our passive NTP corpus, using Rye and Beverly's CPE router geolocation technique permits fewer EUI-64 IPv6 geolocations than did their original study [63]. That work specifically targeted CPE with active measurements. However, we demonstrate that our NTP dataset contains some number of CPE routers that visit the NTP Pool for time that are susceptible to this privacy attack. Further, it is entirely passive. The only defense from this form of geolocation and tracking is to sever the linkage between the MAC addresses that appear in an EUI-64 IPv6 address and the BSSIDs that the WiFi access points use. Due to the potential for device tracking detailed earlier in this section, we recommend the use of random IPv6 addresses.

6 CONCLUSIONS

In this work, we accumulated the largest hitlist of IPv6 addresses solely through publicly-available means, without the aid of a CDN or ISP. We collected 7.9 billion unique addresses from a distributed set of 27 NTP servers located in cloud providers around the world. In addition to containing orders of magnitude more live addresses than existing hitlists, our dataset differs from current lists in that it contains *different types of addresses.* The addresses we obtain are highly entropic and ephemeral. They often come from client devices, as the OUIs of the embedded MAC addresses in EUI-64 IPv6 addresses show. And, crucially, these addresses are almost entirely absent in state-of-the-art hitlists today, which biases these hitlists toward addresses that can easily be discovered with active measurements or the DNS, like infrastructure devices and servers.

The ability to capture massive numbers of active client devices raises ethical questions not previously raised by active measurements. Since many of these addresses are highly random and ephemeral, they are likely uniquely associated with a single device or individual at a specific moment in time. Because of this uniqueness, we believe that these addresses deserve a special level of care in handling. As such, we will release our dataset truncated to the /48 level.

Finally, we repeat a plea for manufacturers to discontinue the use of EUI-64 IPv6 addresses. Our results show that EUI-64 addressing is not uncommon among NTP clients, and is implemented by popular manufacturers of IoT and smart home products. The use of these addresses permits, in some cases, tracking of devices across networks, as well as fine-grained geolocation through correlation with wireless identifiers.

ACKNOWLEDGMENTS

We thank Ask Bjørn Hansen and Steven Sommars from the NTP Pool project for their support and feedback on data sharing considerations. We also thank Rob Beverly and Justin Rohrer for early feedback and infrastructure support, as well as the anonymous reviewers for their helpful comments. This work was supported in part by NSF grants CNS-1901325 and CNS-1943240.

REFERENCES

[1] 2023. IPv6 Hitlist Service. `https://ipv6hitlist.github.io/`.

[2] 2023. Prolific Academic. `https://www.prolific.co/`.

[3] 2023. The NTP Pool Project. `https://www.ntppool.org/en/`.

[4] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, et al. 2015. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. In *ACM Conference on Computer and Communications Security (CCS).*

[5] S Alexander and R Droms. 1997. DHCP Options and BOOTP Vendor Extensions. RFC 2132. `http://www.ietf.org/rfc/rfc2132.txt`

[6] Amazon. 2023. Mechanical Turk (MTurk). `https://www.mturk.com/`.

[7] Apple. 2023. Location Services and Privacy. `https://support.apple.com/en-us/HT207056`.

[8] Genevieve Bartlett, John Heidemann, and Christos Papadopoulos. 2007. Understanding Passive and Active Service Discovery. In *ACM Internet Measurement Conference (IMC)* (San Diego, California, USA) *(IMC '07).* `https://doi.org/10.1145/1298306.1298314`

[9] Robert Beverly. 2016. Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In *ACM Internet Measurement Conference (IMC).*

[10] Robert Beverly and Arthur Berger. 2015. Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure via Active Fingerprinting. In *Passive and Active Network Measurement Conference (PAM).*

[11] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P. Rohrer. 2018. In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery. In *ACM Internet Measurement Conference (IMC).*

[12] Kevin Bock, Abdulrahman Alaraj, Yair Fax, Kyle Hurley, Eric Wustrow, and Dave Levin. 2021. Weaponizing Middleboxes for TCP Reflected Amplification. In *USENIX Security Symposium.*

[13] Kevin Borgolte, Shuang Hao, Tobias Fiebig, and Giovanni Vigna. 2018. Enumerating Active IPv6 Hosts for Large-scale Security Scans via DNSSEC-signed Reverse Zones. In *IEEE Symposium on Security and Privacy.*

[14] CAIDA. 2019. The CAIDA UCSD IPv6 Routed /48 Topology Dataset. `https://www.caida.org/data/active/ipv6_routed_48_topology_dataset.xml`.

[15] Costin, Andrei and Zaddach, Jonas and Francillon, Aurélien and Balzarotti, Davide. 2014. A Large-Scale Analysis of the Security of Embedded Firmwares. In *USENIX Security Symposium.*

[16] Tianyu Cui, Gaopeng Gou, Gang Xiong, Chang Liu, Peipei Fu, and Zhen Li. 2021. 6GAN: IPv6 Multi-Pattern Target Generation via Generative Adversarial Nets with Reinforcement Learning. In *IEEE Conference on Computer Communications (INFOCOM).*

[17] Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey. 2014. Measuring IPv6 Adoption. *ACM SIGCOMM Computer Communication Review (CCR)* 44, 4 (Aug. 2014).

[18] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, et al. 2014. The Matter of Heartbleed. In *ACM Internet Measurement Conference (IMC).*

[19] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and its Security Applications. In *USENIX Security Symposium.*

[20] Asma Enayet and John Heidemann. 2022. Internet Outage Detection Using Passive Analysis. In *ACM Internet Measurement Conference (IMC).*

[21] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. 2017. Something From Nothing (There): Collecting Global IPv6 Datasets From DNS. In *Passive and Active Network Measurement Conference (PAM).*

[22] Pawel Foremski, David Plonka, and Arthur Berger. 2016. Entropy/IP: Uncovering Structure in IPv6 Addresses. In *ACM Internet Measurement Conference (IMC).*

[23] Kensuke Fukuda and John Heidemann. 2018. Who Knocks at the IPv6 Door? Detecting IPv6 Scanning. In *ACM Internet Measurement Conference (IMC).*

[24] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In *ACM Internet Measurement Conference (IMC).*

[25] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. *CoRR* abs/1607.05179 (2016). arXiv:1607.05179 `http://arxiv.org/abs/1607.05179`

[26] R Gayraud and B Lourdelet. 2010. Network Time Protocol (NTP) Server Option for DHCPv6. RFC 5908. `http://www.ietf.org/rfc/rfc5908.txt`

[27] Google Git. 2016. Android. `https://android.googlesource.com/platform/frameworks/base/+/d3f689bf14a05de735b5cc92dcf20e7226c78690%5E%21/core/res/res/values/config.xml`.

[28] F. Gont. 2014. A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC). RFC 7217 (Proposed Standard). https://doi.org/10.17487/RFC7217

[29] Google. 2023. Geolocation API. https://developers.google.com/maps/documentation/geolocation/overview.

[30] Hang Guo and John Heidemann. 2020. Detecting IoT Devices in the Internet. *IEEE/ACM Transactions on Networking* 28, 5 (Oct. 2020).

[31] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. 2008. Census and Survey of the Visible Internet. In *ACM Internet Measurement Conference (IMC)*.

[32] Bingnan Hou, Zhiping Cai, Kui Wu, Jinshu Su, and Yinqiao Xiong. 2021. 6Hit: A Reinforcement Learning-Based Approach to Target Generation for Internet-Wide IPv6 Scanning. In *IEEE Conference on Computer Communications (INFOCOM)*.

[33] Gokay Huz, Steven Bauer, KC Claffy, and Robert Beverly. 2015. Experience in Using MTurk for Network Measurement. In *ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data*.

[34] Young Hyun and k. claffy. 2022. Archipelago Measurement Infrastructure. http://www.caida.org/projects/ark/.

[35] John Kohl, Clifford Neuman, et al. 1993. *The Kerberos network authentication service (V5)*. Technical Report. RFC 1510, september.

[36] Frank Li and David Freeman. 2020. Towards A User-Level Understanding of IPv6 Behavior. In *ACM Internet Measurement Conference (IMC)*.

[37] Zhizhu Liu, Yinqiao Xiong, Xin Liu, Wei Xie, and Peidong Zhu. 2019. 6Tree: Efficient Dynamic Discovery of Active Addresses in the IPv6 Address Space. *Computer Networks* 155 (2019), 31–46.

[38] Matthew Luckie. 2010. Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet. In *ACM Internet Measurement Conference (IMC)*.

[39] Matthew Luckie and Robert Beverly. 2017. The Impact of Router Outages on the AS-level Internet. In *ACM SIGCOMM*.

[40] Aanchal Malhotra, Isaac E. Cohen, Erik Brakke, and Sharon Goldberg. 2016. Attacking the Network Time Protocol. *Network and Distributed System Security Symposium (NDSS)* (2016).

[41] Linda Markowsky and George Markowsky. 2015. Scanning for vulnerable devices in the Internet of Things. In *2015 IEEE 8th International conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, Vol. 1.

[42] MaxMind Inc. 2022. MaxMind GeoLite Databases. https://dev.maxmind.com/geoip/geoip2/geolite2/.

[43] Microsoft. 2021. How the Windows Time Service Works. https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/how-the-windows-time-service-works.

[44] D. Mills. 1985. Network Time Protocol (NTP). RFC 958. http://www.ietf.org/rfc/rfc958.txt

[45] M Morowczynski. 2012. Did your active directory domain time just jump to the year 2000. *Microsoft Server & Tools Blogs http://blogs. technet. com/b/askpfeplat/archive/2012/11/19/did-your-active-directory-domain-time-just-jump-to-the-year-2000. aspx* (2012).

[46] T. Mrugalski, M. Siodelski, B. Volz, A. Yourtchenko, M. Richardson, S. Jiang, T. Lemon, and T. Winters. 2018. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 8415 (Proposed Standard). https://doi.org/10.17487/RFC8415

[47] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. 2017. Target Generation for Internet-Wide IPv6 Scanning. In *ACM Internet Measurement Conference (IMC)*.

[48] Alexander Mylnikov. 2023. Geo-Location API Download Section. https://www.mylnikov.org/download.

[49] Dr. Thomas Narten and Dr. Susan Thomson. 1998. IPv6 Stateless Address Autoconfiguration. RFC 2462. https://doi.org/10.17487/RFC2462

[50] T. Narten and R. Draves. 2001. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 3041. http://www.ietf.org/rfc/rfc3041.txt

[51] openwifi.su. 2023. OpenWifi.su Dataset. http://openwifi.su/db/.

[52] Ramakrishna Padmanabhan, Amogh Dhamdhere, Emile Aben, kc claffy, and Neil Spring. 2016. Reasons Dynamic Addresses Change. In *ACM Internet Measurement Conference (IMC)*.

[53] Ramakrishna Padmanabhan, Patrick Owen, Aaron Schulman, and Neil Spring. 2015. Timeouts: Beware Surprisingly High Delay. In *ACM Internet Measurement Conference (IMC)*.

[54] Ramakrishna Padmanabhan, Aaron Schulman, Alberto Dainotti, Dave Levin, and Neil Spring. 2019. How to Find Correlated Internet Failures. In *Passive and Active Network Measurement Conference (PAM)*, David Choffnes and Marinho Barcellos (Eds.).

[55] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. 2017. Global Measurement of DNS Manipulation. In *USENIX Security Symposium*.

[56] David Plonka and Arthur Berger. 2015. Temporal and Spatial Classification of Active IPv6 Addresses. In *ACM Internet Measurement Conference (IMC)*.

[57] radiocells.org. 2023. OpenBMap Dataset. https://radiocells.org/.

[58] Philipp Richter, Oliver Gasser, and Arthur Berger. 2022. Illuminating Large-Scale IPv6 Scanning in the Internet. In *ACM Internet Measurement Conference (IMC)*.

[59] Philipp Richter, Ramakrishna Padmanabhan, Neil Spring, Arthur Berger, and David Clark. 2018. Advancing the art of internet edge outage detection. In *ACM Internet Measurement Conference (IMC)*.

[60] 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 *ACM Internet Measurement Conference (IMC)*.

[61] RIPE. 2017. Best Current Operational Practice for Operators: IPv6 Prefix Assignment for End-Users - Persistent vs Non-Persistent, and What Size to Choose. https://www.ripe.net/publications/docs/ripe-690.

[62] Justin P. Rohrer, Blake LaFever, and Robert Beverly. 2016. Empirical Study of Router IPv6 Interface Address Distributions. *IEEE Internet Computing* (Aug. 2016).

[63] Erik Rye and Robert Beverly. 2023. IPvSeeYou: Exploiting Leaked Identifiers in IPv6 for Street-Level Geolocation. In *IEEE Symposium on Security and Privacy*.

[64] Erik Rye, Robert Beverly, and kc claffy. 2021. Follow the Scent: Defeating IPv6 Prefix Rotation Privacy. In *ACM Internet Measurement Conference (IMC)*.

[65] Erik C Rye and Robert Beverly. 2020. Discovering the IPv6 Network Periphery. In *Passive and Active Network Measurement Conference (PAM)*.

[66] Said Jawad Saidi, Oliver Gasser, and Georgios Smaragdakis. 2022. One Bad Apple Can Spoil Your IPv6 Privacy. *ACM SIGCOMM Computer Communication Review* 52, 2 (2022).

[67] Aaron Schulman and Neil Spring. 2011. Pingin'in the rain. In *ACM Internet Measurement Conference (IMC)*.

[68] Lion Steger, Liming Kuang, Johannes Zirngibl, Georg Carle, and Oliver Gasser. 2023. Target Acquired? Evaluating Target Generation Algorithms for IPv6. In *Network Traffic Measurement and Analysis*.

[69] Stephen D Strowes. 2017. Bootstrapping active IPv6 measurement with IPv4 and public DNS. *arXiv preprint arXiv:1710.08536* (2017).

[70] tumi8. 2022. ZMapv6: Internet Scanner with IPv6 Capabilities. https://github.com/tumi8/zmap.

[71] WiGLE – All the Networks. Found by Everyone. 2023. WiGLE – All the Networks. Found by Everyone. https://wigle.net.

[72] Tao Yang, Zhiping Cai, Bingnan Hou, and Tongqing Zhou. 2022. 6Forest: An Ensemble Learning-Based Approach to Target Generation for Internet-Wide IPv6 Scanning. In *IEEE Conference on Computer Communications (INFOCOM)*.

[73] Sebastian Zander, Lachlan LH Andrew, and Grenville Armitage. 2014. Capturing Ghosts: Predicting the Used IPv4 Space by Inferring Unobserved Addresses. In *ACM Internet Measurement Conference (IMC)*.

[74] Liang Zhang, David Choffnes, Dave Levin, Tudor Dumitraş, Alan Mislove, Aaron Schulman, and Christo Wilson. 2014. Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed. In *ACM Internet Measurement Conference (IMC)*.

[75] Johannes Zirngibl, Lion Steger, Patrick Sattler, Oliver Gasser, and Georg Carle. 2022. Rusty Clusters? Dusting an IPv6 Research Foundation. In *ACM Internet Measurement Conference (IMC)*.

[76] Maya Ziv, Liz Izhikevich, Kimberly Ruth, Katherine Izhikevich, and Zakir Durumeric. 2021. ASdb: A System for Classifying Owners of Autonomous Systems. In *ACM Internet Measurement Conference (IMC)*.