Seeds of Scanning: Exploring the Effects of Datasets, Methods, and Metrics on IPv6 Internet Scanning

Grant Williams
Georgia Institute of Technology
Atlanta, United States
gwilliams319@gatech.edu

Paul Pearce
Georgia Institute of Technology
Atlanta, United States
pearce@gatech.edu

Abstract

Large-scale Internet scanning is a vital research tool. While IPv4 can be exhaustively probed, the size of IPv6 precludes complete enumeration, limiting large-scale measurement. Target Generation Algorithms (TGAs)—algorithms which ingest lists of prediscovered addresses ("seeds") and produce new addresses to scan—have begun bridging this IPv6 measurement gap. To date, there has been limited exploration of how changes in seed addresses, scanning methods, and dataset composition impact TGA-driven IPv6 host discovery.

In this work, we provide a roadmap for how to use TGAs for Internet-wide scanning by evaluating how changes to input datasets, preprocessing, liveness, alias detection, and metrics impact TGA performance. We also explore how choice of scan target—ICMP Echo, TCP80, TCP443, or UDP53—across both inputs and outputs, impact discovered addresses.

From this analysis, we provide guidance on how to properly preprocess a TGA input (seed) dataset and the importance of removing aliases; simple preprocessing at scan time can significantly improve network diversity and can increase discovered hosts by over 700% across combined approaches. We further compare TGA generation budgets, analyze discovered populations, and demonstrate the utility of running multiple TGAs together. Finally, we summarize recommendations for effective TGA use for Internet-wide IPv6 scanning.

CCS Concepts

• **Networks** → **Network measurement**.

Keywords

IPv6, Scanning, Network Measurement

ACM Reference Format:
Grant Williams and Paul Pearce. 2024. Seeds of Scanning: Exploring the Effects of Datasets, Methods, and Metrics on IPv6 Internet Scanning. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3646547.3688449

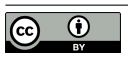
1 Introduction

Internet scanning is a vital research tool enabling studies ranging from understanding network topology [6], to measuring the impact of weather on network links [19, 35], to determining how oppressive regimes target political dissidents [30]. Driven by the development of ZMap [19], whole-Internet scanning is now ubiquitous. Unfortunately, our understanding is primarily limited to the IPv4 Internet, as ZMap and other tools leverage brute force address exploration. The tremendous size of IPv6—340 trillion trillion trillion addresses—precludes the use of IPv4 methods to scan IPv6.

Simultaneously, the adoption of IPv6 has grown significantly since the 2010s. Google reports that the total volume of IPv6 visitors to their services has increased from 0.25% in 2010 to 45% in 2024 [24]. The confluence of these two factors—the importance of Internet scanning and the size of the IPv6 address space—gave rise to a host of IPv6 address discovery mechanisms known as Target Generation Algorithms, or TGAs [11–13, 21, 25, 26, 29, 33, 43, 44, 49, 53–55]. TGAs work by taking a set of known IPv6 addresses as input and generating new, similar candidate addresses to probe. These mechanisms range from heuristic-based [21, 29, 33, 49, 54] to complex machine learning [11–13, 25, 51], and can be both online [25, 26, 44, 51] (adapting generation to real-time scan results) or offline [11–13, 21, 29, 33, 49, 53–55].

While numerous algorithms have been proposed, there exists no comprehensive study of how to effectively use TGAs, including: how input datasets impact discovered results, how to pre-process input data, whether TGA performance differs based on the success metric, and how different ports or protocols impact results. Worse still, various proposals use disjoint and frequently contradictory methods, with no comprehensive set of comparison metrics used across the community. Steger et al. [47] made initial steps towards dataset evaluation but more work is needed. Establishing effective TGA usage is critical to giving large-scale IPv6 Internet measurement the same sound and rigorous basis as IPv4 measurement.

In this work, we provide a foundational understanding of how to use TGAs to scan the IPv6 address space. We focus on answering a series of Research Questions (RQs) centering around what data to train TGAs on, how to pre-process that data, how various methodological decisions impact TGA output, how decisions across port and protocol impact performance, and how to utilize multiple TGAs to yield the best outcome based on application goal. Across our evaluation we explore two distinct metrics for TGA success: total discovered responsive IPv6 addresses ("hits"), and network diversity of responsive addresses (via autonomous systems, or "ASes"). We conduct our study over a diverse set of 8 popular, high performing TGAs: DET [44], 6Sense [51], 6Tree [29], 6Scan [26], 6Hit [25], 6Graph [54], 6Gen [33], and Entropy/IP [21]. These TGAs range from statistical to machine learning methods. We explore a wide range of input dataset sources across TGAs including traceroutes, DNS lookups, and previously discovered IPv6 addresses ("hitlists").



IMC '24, November 4–6, 2024, Madrid, Spain

ACM ISBN 979-8-4007-0592-2/24/11
https://doi.org/10.1145/3646547.3688449

We find that optimal TGA usage varies based on the desired use-case, but certain overall tradeoffs and principles apply. Different tools perform better in different types of scans, and different input datasets significantly alter results. We find that how you pre-process the datasets and conduct scans changes performance and optimal outcomes. Lastly, we find that optimization metrics also change which datasets and methods perform best. The best performing TGAs overall vary depending on metric, with some optimizing for network diversity and some for address discovery. All told, we distill these empirical findings into a set of best practices for TGA usage. Specifically, our contributions include:

- We summarize the distribution of TGA input datasets used in past studies.
- We explore how to clean and construct datasets for TGAs, finding the best seed dataset and construction method depends on the desired metric.
- We test and quantify seed address pitfalls involving the use of aliased seeds and demonstrate that generators with dealiased seeds can discover on average 1.70x more hits in 1.32x more ASes.
- We analyze the effects of different kinds of dealiasing for seeds, finding that just relying on published alias lists is not enough, with some TGAs generating over 19% of their budgets in aliased regions (compared to, on average, less than 0.5% with the addition of adaptive online approaches).
- We show that simple preprocessing of no-longer (or never) responsive addresses can increase discovered hits by 2.28x across 1.53x more ASes.
- We find tailoring input datasets to activity on a desired port/protocol improves discovery by 2.31x on average, but at the cost of network diversity for some generators.
- We show that scanning smaller datasets from specific sources, while not optimal for hit discovery, can help explore more diverse networks.
- We perform an end-to-end comparison of current TGAs using our best-discovered methods. We find that online models (including DET, and the generative component of 6Sense) perform best across many datasets and metrics, especially address diversity. However, we notice 6Tree (an early model and the basis of many approaches going forward) still outperforms many state-of-the-art models (including some online successors). No single generator performs best, and our results suggest that running multiple TGAs together can provide a more comprehensive viewpoint of the Internet.
- We provide a list of concrete recommendations for effective TGA usage going forward.

2 Related Work and Background

Understanding the effective use of TGAs intersects with prior work on IPv6 Internet scanning, IPv6 aliasing, existing IPv6 datasets, and prior analysis, which is discussed here.

2.1 Internet Scanning and IPv6

Since the introduction of ZMap [19] in 2013, large-scale Internet scanning has served as a cornerstone of network and security research, with tools like Censys [18] enumerating active ports and services on all devices on the IPv4 Internet. Shortly thereafter, work on IPv6-based tools began, starting with Ullrich et al.'s [49] pattern-based approach in 2015. Such approaches are referred to as Target Generation Algorithms (TGAs). TGAs discover new active IPv6 addresses by extracting patterns from known active addresses, dubbed "seeds." Seeds often originate from non-scanning sources such as traceroutes or topology data [5, 46], DNS domain lookups, and public datasets [23].

While Ullrich et al. manually created patterns from active addresses, TGAs quickly expanded to automate address generation. Entropy/IP (EIP) [21] efficiently generated addresses by extracting patterns in the entropy of seed address nybbles (hexadecimal digits). 6Gen [33] followed with a clustering approach for pattern discovery.

Generator efficiency quickly improved. 6Tree [29] marked the emergence of tree-based algorithms. 6Tree creates an address tree, splitting hierarchically on address nybbles from the higher granularity prefixes down. It then generates addresses by expanding variable nodes. 6Hit [25] followed as the first fully online model (adapting over time to scan results) by targeting active tree nodes with reinforcement learning and periodically recreating the tree. 6Scan [26], meanwhile, expands 6Tree to dynamically update which nodes to sample from by encoding node information in the packet payload to quickly update scan directions over time.

DET [44] further enhanced tree-based generation by updating 6Tree's splitting heuristic to an entropy-based approach, while periodically updating the tree with active addresses, making it an online model. 6Graph [54] alternatively expanded 6Tree offline, deploying an approach with similar splitting mechanisms to DET.

Finally, 6Sense [51] used an online adaptive Reinforcement Learning approach to find active regions. It hierarchically generated address sections separately from each other using a deep learning system, and dedicated a variable part of its scan budget to expanding AS coverage.

The aforementioned functional tools—6Sense, DET, 6Tree, 6Scan, 6Graph, 6Gen, 6Hit, and Entropy/IP— are the focus of our study. Other approaches exist, such as the deep learning models 6GAN [12], 6VecLM [13], and 6GCVAE [11], or the forest approach of 6Forest [53]. Prior work [47, 56] showed these other deep learning approaches could not efficiently generate addresses, and found orders of magnitude fewer active addresses than other methods. We also find these approaches are unable to scale to tens of millions of generated IPs (a requirement both for this study and Internet-scale usage), and thus we exclude them from our study.

2.2 IPv6 Aliasing

While discovering new active IPv6 addresses often corresponds to discovering new devices, sometimes vast ranges of addresses map to the same physical device. This phenomenon is called IPv6 aliasing [22, 28, 33]. A prefix is aliased when the entire IPv6 prefix is responsive and maps to a single device (or handful of devices). Aliasing complicates IPv6 scanning. A single aliased /64 prefix (a size commonly assigned to endhosts [48]) has 2^{64} active addresses—32 orders of magnitude (base 2) more than the entire IPv4 space. Thus aliasing can causes significant miscounting of distinct physical devices discovered in measurement scans.

TGAs often do not address aliases, with online models [25, 26, 44] particularly susceptible to falling into aliased regions. 6Gen [33] proposed an online dealiasing approach based on the principle that: in a large enough IPv6 prefix, if numerous random addresses are active, all addresses must be active in that prefix, and thus it is likely an alias. 6Gen sent randomized lower-32 bits to /96 prefixes and marked /96s as aliased if their probes returned active results. 6Sense [51] successfully deployed this approach to alias filtering in an online TGA. The IPv6 Hitlist [23] also publishes a list of verified aliased prefixes for offline dealiasing (filtering known aliases), but it is not complete, and misses previously undiscovered aliases a generator may find. However, many prior TGAs rely solely or partly on this list [25, 26, 29, 51, 53, 54].

2.3 Seed Datasets

Beyond deciding on a generation algorithm, IPv6 scanners must identify an input seed dataset and how to process it.

Dataset Sources. Early generative models [21, 33, 49, 55] used many seed collection methods. Commonly, the Rapid7 FDNS dataset [39] provided domains expandable to IPv6 addresses via AAAA lookups [21, 33, 55]. Since its introduction in 2016, the IPv6 Hitlist [23] has become a widely used dataset for IPv6 scanning, serving as seeds for most TGAs post-2016 [11–13, 25, 26, 29, 53, 54]. The IPv6 Hitlist provides a more diverse seed set than any one source. It contains addresses collected from many sources including DNS resolution of domains from Certificate Transparency Logs, zone files, the Rapid7 FDNS dataset [39], and domain toplists [31, 32, 50] and addresses from RIPE Atlas [46], Scamper [9, 15], Bitnodes [4], traceroutes of collected addresses [6], and active addresses from TGAs [12, 13, 21, 29, 33, 54, 56, 56]. Separately, AddrMiner [42, 43] also developed an IPv6 Hitlist based on output from AddrMiner (a TGA expanded from DET to focus on long-term measurement). Rye et al. [40] introduced a hitlist collected from NTP pools, but it is not publicly available.

Dataset Preprocessing. Seed addresses provide a basis for algorithm input, but those addresses can be preprocessed in various forms. Prior work takes inconsistent approaches to preprocessing, making direct comparison challenging. For example, some prior work inputs responsive addresses on the port they scan [26], while others use all addresses collected [29], and still others are unclear what preprocessing they use [12]. In addition, some pre-filter active addresses for aliases and many use the alias list published by the IPv6 Hitlist [12, 23, 47, 56]. The wide variety of approaches makes comparison between prior work challenging, as reported results vary significantly based on dataset and preprocessing.

2.4 Prior Dataset Analysis

Prior work exists exploring the impact of different datasets on Internet scanning [7, 25, 47, 53, 54] but a comprehensive analysis of TGA performance across datasets, methods, and metrics remains unexplored.

Steger et al. [47], the most relevant prior work, performed a comparison of TGA performance across subsets of the IPv6 Hitlist split by PeeringDB labels [16] (for classifying AS/organization type), building on prior IPv6 Hitlist exclusive comparisons [11, 13, 23, 25, 26, 29, 53, 54, 56]. While Steger et al.'s work is both important and apt to the goals of our work, it is not without limitations. Steger et al. evaluated different seed dataset categorizations on TGAs, but they did not evaluate port-specific seeds, preprocessing methods, large-scale scanning (they used primarily <10M budgets, varying per generator), and only addressed running with different Peering DB classifications of seeds. Further, dynamic TGAs only used ICMP scans, meaning they did not accurately adapt to multiple ports/protocols, and Steger et al. suggests their comparison results are biased by the disproportional presence of AS12322. While the primary goal of Steger et al. is to classify how a single dataset source, the IPv6 Hitlist, affects TGAs, we aim to expand further into fundamental questions on best practices for TGA use. Although the IPv6 Hitlist shares many data sources with those evaluated here (per Section 2.3), we show only moderate overlap of addresses between the Hitlist and our other collected sources in Section 5, suggesting either differences in collection/filtering of the Hitlist or significant temporal changes in datasets.

Beverly et al. [7] in 2018 performed a topology study of IPv6 datasets (including CAIDA DNS Names [14] and the Rapid7 FDNS [39] dataset), but since its publication in 2018, seed dataset sources and makeup changed significantly with the introduction of the IPv6 Hitlist. Additionally, Beverly et al. did not evaluate how these seeds affected generative performance across TGAs (comparing topology only to addresses generated with 6Gen).

Included	6Sense	DET	6Scan	6Hit	6Graph	6Tree	6Gen	EIP
All	-	-	-	-	-	-	✓	✓
No Dealiasing	-	-	-	-	-	-	✓	✓
Offline Dealiasing	✓	✓	✓	✓	✓	✓	-	-
Online Dealiasing	✓	-	-	-	-	-	-	-
Include Inactive	-	-	-	-	-	✓	✓	✓
Only Active	✓	✓	-	✓	✓	✓	-	-
Port Spec.	-	-	✓	-	-	-	-	-

Table 1: Overview of dataset construction and preprocessing methods by TGA. Some TGAs (including 6Hit [25] and 6Graph [54]), use the IPv6 Hitlist directly without re-verifying that addresses are still responsive (accounting for the large number of TGAs using offline dealiased active IPs). For our purposes, we shall consider this prior work to use offline dealiased active IPs.

3 Experimental Construction

We perform a series of experiments characterizing the optimal approaches to scan, dealias, preprocess, and analyze TGAs. We aim in this work to create a list of best practices for TGA usage (Section 10), based on our observations.

TGAs operate by expanding upon patterns present in their seed datasets. This makes them very dependent on their input. Seed datasets, how they are processed, how hits are evaluated, and how scans are conducted are all vital components of TGAs and IPv6 scanning; all need carefully controlled exploration to understand effective usage. If, for example, a seed dataset does not provide adequate coverage of specific network regions, those regions may be effectively missed in measurements. Unfortunately, prior work used an inconsistent set of methods and datasets when evaluating TGAs; Table 1 shows an overview of construction and preprocessing methods used in prior work, with substantial variation

Section	Dataset
RQ1.a	Full Dataset Offline Dealiased Dataset Online Dealiased Dataset **Dealiased:** Online+Offline Dealiased Dataset
RQ1.b	**All Active:** Dealiased - Unresponsive
RQ2	**Port-Specific:** All Active - Inactive per Port
RQ3	**Source-Specific** All Active ∪ Seed Source
RQ4	**All Active** comparing generators

Table 2: Overview of the primary datasets in each RQ. For RQ1.a we compare different dealiasing approaches. For RQ1.b-RQ3 we compare each row's dataset with the optimal dataset from the prior rows. For RQ4 we compare TGAs on the All Active dataset.

across methodologies. Note, we use the term "Active" in this context to refer to responsive addresses across ports and protocols

While many open questions exist related to effective use of TGAs, a large number relate to the variety of different seed data sources, preprocessing methods, and seed dataset permutations possible. In particular, in this work, we set out to answer 5 primary Research Questions (RQs) (with subquestions) characterizing TGA best practices:

- **RQ1:** How should we preprocess seed datasets for TGAs?
  - **RQ1.a:** How do aliases within the seed dataset and dealiasing methods impact TGA output?
  - **RQ1.b:** Does using previously active seeds improve or hurt TGA performance?
- **RQ2:** How does port or protocol (and active seeds on a specific port/protocol) impact TGA performance?
- **RQ3:** How do different seed data sources impact TGA performance?
- **RQ4:** What is the overlap in generator output? How do generators perform when used together?
- **RQ5:** What are the concrete recommendations and best practices for TGA usage?

To evaluate these research questions, we compare TGA output using different seed datasets. The seed dataset each research question compares is shown in Table 2. The experimental methodology is provided in Section 4.

4 Method

Driven by the questions posed in Section 3, we evaluate how TGA input impacts generator performance across metrics. Evaluating seed datasets depends on many factors, including scanning methodology and dealiasing. In this section we explain our experimental methodology, scanning methodology, and dealiasing approaches.

4.1 Experimental Methodology

To perform the experiments described in Section 3, we begin by collecting a seed dataset (Section 5) from many sources, guided by prior work. We evaluate dataset construction methods, pre-processing methods, and scanning methods, based on the RQs above, across 8 TGAs to quantify how input processing methods impact results. We select the following 8 TGAs due to their applicability to Internet-wide scanning and ability to consistently generate over 50M addresses: Entropy/IP [21], 6Gen [33], 6Tree [29], 6Hit [25], DET [44], 6Graph [54], 6Scan [26], and 6Sense [51]. We use optimized versions of 6Gen, 6Hit, and 6Tree from Hou et al. [26], due to their availability and performance in prior comparisons. We used the official (to the best of our knowledge) open-source versions of these TGAs [1–3, 17, 20]. We used default TGA parameters, except for 6Sense where we scale default parameters with the budget as described in its documentation.

We generate 50M addresses, with each TGA using each RQ's experimental construction, and scan them using ICMP ECHO, TCP80, TCP443, and UDP53 probes. We chose 50M because it was sufficiently large to capture longer-term trends, while not taking an infeasibly long time to generate across many TGAs and seed datasets. All TGAs successfully generated 50M addresses from each seed dataset. We explore protocols and ports beyond ICMP because ICMP is highly responsive on IPv6 in prior work [23], and may not correlate to application-level results relevant to interesting use cases. For online generators (adapting to scan results in real-time), we rerun generation for each port and protocol scanned to ensure a fair comparison. After generation, we scan and dealias addresses in alignment with prior work [47] using the scanner and dealiaser described in Section 4.2.

For ICMP, we do not count ICMP Destination Unreachable messages as "hits" in response to ICMP Echos, for consistency across comparisons. Similarly, we do not count TCP RST packets as hits for TCP80 and TCP443 scans. In both cases, these responses do not indicate whether devices are open on any of our evaluated ports/protocols, and inclusion of these responses is inconsistent across prior work. This required updating 6Scan and 6Hit's built-in scanner.

Metrics. Across experiments, we consistently evaluate two core metrics: **Hits** (dealiased active addresses discovered) and **Active ASes** (ASes with active addresses discovered by the TGA: representing network diversity). We take this approach as hits can be significantly influenced by one or two highly responsive, but not aliased, networks. AS diversity can correspond to broader "whole-Internet" scanning. We find in subsequent experiments, choice of metric matters, with different metrics pointing to different optimal input treatments. When evaluating dealiasing in RQ1.a, we also compare discovered aliased addresses. We note that defining and evaluating detailed metrics for large-scale Internet scanning is still an open problem requiring future work.

For ICMP evaluation, we filter addresses in AS12322 (known to cause problematic results in prior work [26, 44, 47, 51]). Steger et al. [47] showed a saturation of AS12322 addresses in TGA results with only variations in the 10-15 nybble range (and a fixed lower-64 bits of ::1). We scanned a random subset of 1M addresses on ICMP in this pattern and found 35.03% active. Given this pattern contains 16.7M addresses, this suggests 5.8M easily discoverable ICMP responsive addresses. Because our goal is to compare generator performance across TGAs across datasets, these addresses bias generation, since we can find them already using the given pattern. Thus, we filter these addresses from ICMP evaluation to obtain an unbiased picture of generator performance.

In RQ1 and RQ2, TGA metrics (hits, ASes, and aliases) are compared (where necessary) using a Performance Ratio between the *original* and *changed* datasets (designated as "*changed* vs. *original*"), defined as: $3 \times \frac{metric_{changed} - metric_{original}}{metric_{changed} + metric_{original}}$. Intuitively, if a change does not vary generator performance on a metric, the performance ratio is 0. If it doubles performance, it is 1.0, and if it halves performance, it is -1.0. While raw metrics can vary wildly in scale and magnitude across generators and ports/protocols, this performance ratio allows us to clearly compare generator performance across many situations.

Raw numbers of discovered hits and ASes for RQ1, RQ2, and RQ3 are included in Appendix E.

4.2 Scanning and Dealiasing

Prior IPv6 scanning tools exist that take lists of IPv6 addresses, emit packets, and check responses [6, 23]. Through the course of our study, we encountered challenges with many of these tools, such as missing or problematic blocklisting and lack of packet verification. We use Scanv6 [41], a Go-based scanner proposed in 6Sense [51] for conducting scans from generators without their own integrated scanners, as it solves these concerns. We combine all addresses generated between TGAs per dataset per port and scan those unique IPs together, for consistency and to minimize the times each address is probed. Scans were conducted continuously between March 11th, 2024 and May 10th, 2024.

Aliased regions in IPv6 can drastically alter perceived scan performance by inflating hitrates. Thus, to ensure consistency across scans and to preserve the ability to compare results, we must properly discern aliased regions from legitimately active regions in scan results. We use a two-tier dealiasing approach suggested by 6Sense [51]. First, we remove aliases appearing in known aliased prefixes recorded by the IPv6 Hitlist [23] in accordance with prior work [12, 23, 47, 56]. However, this does not catch never-before-seen aliases, particularly problematic for datasets not derived from the IPv6 Hitlist (and so not dealiased in the hitlist creation).

Thus, second, we deploy the dealiasing method proposed by 6Gen [33] (described previously in Section 2.2). We keep the prefix length at /96 (a /96 contains 4 billion addresses). For all active addresses, when we encounter a new /96 prefix, we generate 3 random addresses within that prefix (with 3 packet retries). If two or more of those random addresses are active, we call that /96 an alias and classify all addresses within that /96 as aliased. We remove all aliased addresses from our results for active addresses since our goal is to quantify entirely new devices.

5 Dataset Composition

We now provide an overview of our datasets and their composition. While multiple potential seed datasets exist [15, 23, 42, 43, 46], our study requires collecting updated data from many sources and understanding their distribution.

5.1 Sources

Consistent with prior work [21, 23, 33], we collect addresses from three main sources: domain names resolved via AAAA lookups, traceroute-based router topology datasets (RIPE Atlas [46], Scamper [15]), and pre-compiled hitlists [23, 42, 43]. Table 3 characterizes IP volume and other statistics across datasets and is discussed subsequently. We provide dates of address collection in Appendix B.

Domains. We collect domains from four sources: Certificate (CT) Logs hosted on Censys [10, 18], the Rapid7 FDNS dataset [39], CAIDA DNS Names [14] (overlapping with CAIDA's IPv6 Topology dataset [9]), and domain toplists (Cisco Umbrella [50], Tranco [36], SecRank [52], the Majestic Million [32], and Cloudflare Radar [37]). For Rapid7, we use an archival version collected in November 2021 (given recent licensing changes), and include 15M IPv6 addresses from archival AAAA lookups. We resolve all domains using ZDNS [27] to perform AAAA lookups to Google's Public DNS. Overall, we collected 37 million unique server addresses from domains. Appendix C provides more information about successful domain lookups from each source. Overall, Censys and Rapid7 provide the majority of domains and IPs.

Addresses: Routers, Traceroutes, and Hitlists. We collect router IP addresses from the IPv6 Topology Dataset [15] (based on traceroutes collected by Scamper [9]), and the RIPE Atlas dataset [46]. Gasser et al. [23] proposed the IPv6 Hitlist, combining potentially active addresses from many non-scanning sources [4, 5, 8, 9, 18, 31, 32, 34, 38, 45, 50] and some TGAs [12, 13, 21, 29, 33, 54, 56]. We use the list of active IPs provided by the IPv6 Hitlist. AddrMiner [42, 43] provides an alternative hitlist based on address generation using the AddrMiner generator [42, 43] for long-term generation.

5.2 Comparisons

We characterize how much each dataset contributes to our overall seed set. Figure 1 shows the overlap of seed addresses (left) and ASes (right) across datasets. The far right "Overlap" columns show the percentage of that dataset present in one or more other datasets collected (i.e. 40.7% of IPv6 addresses collected from Censys also appeared in one or more of the other 11 datasets). This allows us to examine how much each dataset uniquely contributes to the whole.

RIPE Atlas, Scamper, and the IPv6 Hitlist provide the most coverage of seeds and ASes (only 12.9% of Scamper overlaps with other datasets). Scamper and RIPE Atlas in particular cover nearly all ASes observed (with over 30K each), likely due to coverage being the purpose of Scamper, and large-scale measurement usages of Atlas nodes. AddrMiner provides many seeds, but also many aliases (with little overlap of other datasets). Domain sources provide a nontrivial number of IPs in ASes covered already by other sources (and are helpful for increasing depth in those ASes). Toplists provide some unique contributions and are worth incorporating, but their contribution varies based on the toplist. A high AS coverage can be achieved with just traceroute sources, while domains and hitlists help bolster the number of seed IPs in otherwise observed ASes.

5.3 Active Discovery

After we collect the full set of seed addresses, we proceed to scan and dealias the dataset as described in Section 4.2 on four ports and protocols: ICMP, TCP80, TCP443, and UDP53. Table 3 shows the results of this scanning and dealiasing across all data sources. In addition, we characterize how each dataset contributes to the responsive IPs in Figure 2.

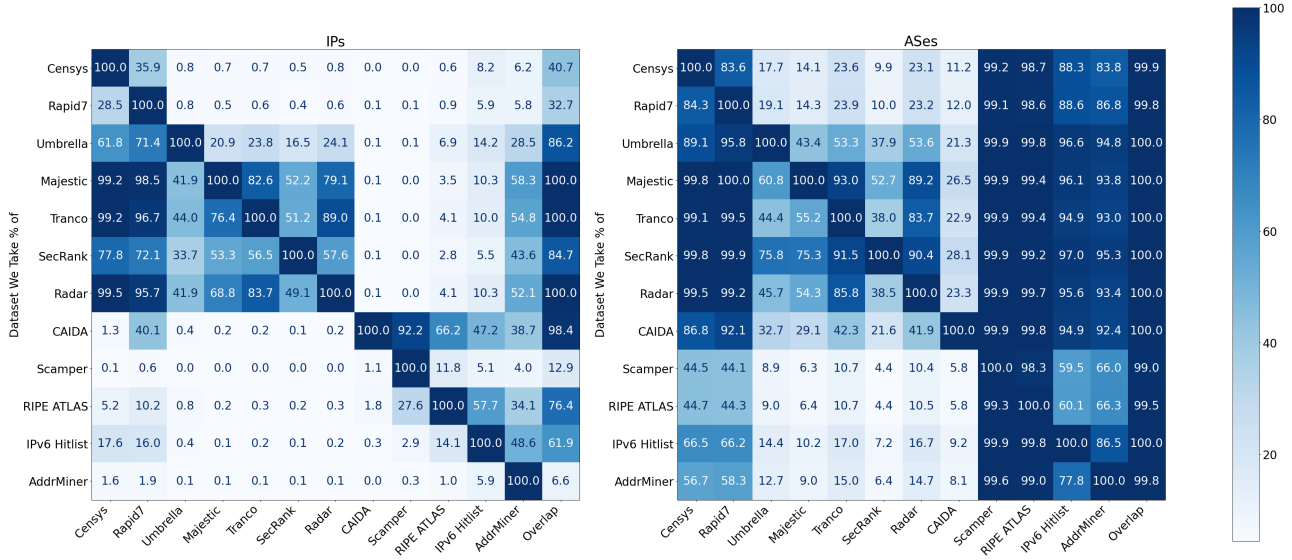


Figure 1: Seed source percent overlap by IP and AS. Domain-based sources tend to overlap, and comprise a majority of the overlap. Scamper covers almost all ASes collected, without significant IP coverage.

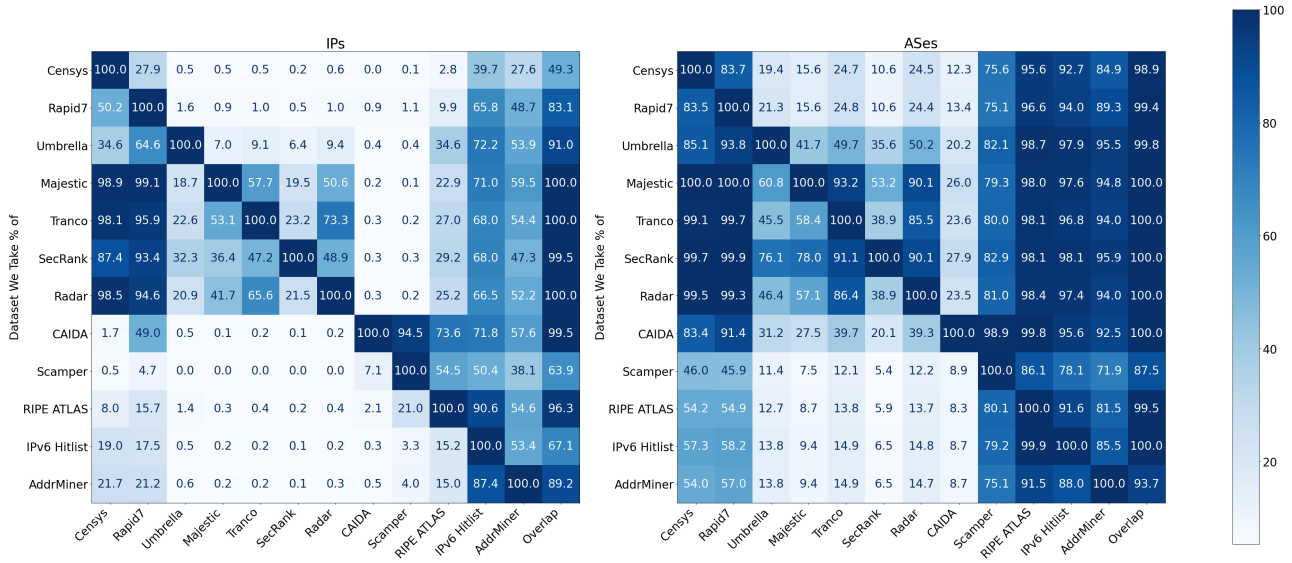


Figure 2: Seed source percent overlap of responsive IPv6 addresses by IP and AS. Similar distributions exist to the full dataset, although the IPv6 Hitlist and AddrMiner have higher AS overlap with Scamper and RIPE Atlas.

Hit contributions were similar by percentage between responsive and unresponsive addresses (from Figure 1), while AS contributions were nearly identical. Censys, the IPv6 Hitlist, and Scamper uniquely contributed a large part of their responsive addresses (although Scamper provided fewer responsive addresses overall compared to other non-toplist sources). As expected, the IPv6 Hitlist is a good single source of IPs (providing more responsive IPv6 addresses than any other single source at 7.6M), but no single other source is fully covered by the hitlist. Meanwhile traceroute-based sources still tended to discover the highest number of responsive ASes, similar to the unresponsive case.

6 RQ1: How should we preprocess seed datasets for TGAs?

While seed datasets form a key part of any TGA, little consensus exists on how to preprocess a seed dataset, and how this preprocessing affects generative performance. To answer this, we examine two important preprocessing steps (proposed in prior work [23, 47, 56])

Source	Pop.	Unique	ASes	Dealiased	ICMP	TCP80	TCP443	UDP53	Active	Active ASes
Censys CT	D	19,446,042	13,950	7,482,129	3,537,844	802,522	851,344	122,667	3,654,876	11,050
Rapid7	D	24,537,629	13,840	6,930,413	1,936,549	1,109,582	1,007,271	163,391	2,028,611	11,079
Umbrella	D	261,717	2,764	59,039	44,136	42,195	44,829	2,532	49,927	2,517
Majestic	D	130,751	1,973	21,646	16,829	17,663	17,345	2,188	18,519	1,724
Tranco	D	141,325	3,321	24,509	18,005	18,804	18,490	3,228	20,145	2,751
SecRank	D	127,963	1,381	13,065	8,437	8,811	7,934	524	9,909	1,176
Radar	D	150,319	3,239	27,374	20,189	21,067	20,862	2,945	22,516	2,722
CAIDA DNS	D	59,348	1,800	56,318	36,988	648	813	1,267	37,006	1,631
All Domains	D	37,103,077	16,305	12,162,665	4,554,353	1,322,314	1,265,818	189,370	4,700,354	13,096
Scamper	R	5,194,955	31,122	2,414,558	491,727	14,806	5,545	4,595	492,506	18,132
RIPE Atlas	R	2,214,546	30,787	2,113,404	1,250,095	290,168	287,974	73,822	1,278,586	19,501
All Routers	R	6,797,649	31,326	3,930,353	1,473,690	300,964	289,624	75,055	1,502,764	22,020
IPv6 Hitlist	Both	9,063,317	23,104	8,993,074	7,473,465	1,643,307	1,466,003	236,279	7,619,875	17,878
AddrMiner	Both	74,348,374	20,610	10,378,135	4,640,430	1,086,799	929,572	181,223	4,659,058	17,363
All Hitlists	Both	79,009,285	23,104	14,979,467	8,058,355	1,743,400	1,560,122	284,993	8,208,355	19,962
All Sources	Both	118,729,345	31,389	27,179,296	10,783,974	2,254,886	2,083,836	367,917	10,999,613	23,613
All ASes	Both	31,389	-	23,613	23,360	11,762	11,047	8,776	23,613	-

Table 3: Full summary of all seed data sources. The unique population as well as activeness across ports and protocols are shown. "D" denotes domains, and "R" denotes routers.

for TGA seed datasets and their effects on generation: dealiasing (RQ1.a), and using unresponsive seeds (RQ1.b). We begin with a baseline of our full collected seed dataset of 118.7M addresses, and refine as we make conclusions on dataset preparsing best practices.

6.1 RQ1.a: How do aliases within the seed dataset and dealiasing methods impact TGA output?

While dealiasing TGA *outputs* (Section 2.2) is necessary to understand the utility of discovered IPv6 addresses, prior work has varied in how they handle aliases. It remains unclear how dealiasing *input* seeds affects generation. It is possible aliases may impact critical addressing pattern information or degrade performance if aliases are clustered within certain patterns or regions.

To evaluate the effects of input dealiasing, we generate 50M IPs with the 8 TGAs on our full input dataset across multiple ports. Then, we compare these results to the TGA output on a dealiased version of the input dataset (using the online+offline dealiasing method described in Section 2.2). The Performance Ratios of hits, ASes, and aliases are shown in Figure 3. As expected, generated aliases are orders of magnitude lower using the dealiased dataset. With DET for instance, the dealiased dataset on ICMP contains only 74K aliased IPs vs. 33M from the full dataset. Hits and ASes tend to universally increase with dealiasing. Two exceptions appear: 6Sense (likely because it includes dealiasing already in generation) and EIP. EIP finds few addresses already and over 99% of discovered TCP443 addresses are from a single Amazon prefix, 2600:9000:2000::/48, suggesting EIP either stumbled on a lucky pattern, or this region contains aliasing not detectable by the dealiasing methods deployed. 99% of the 495K addresses EIP generates from only offline dealiased seeds also appear in this prefix, and it is common in TCP80 EIP results but not in experiments with online dealiased seeds, suggesting rate-limiting here may interfere with on-the-fly dealiasing. The large portion of the probe budget no longer expended on aliases accounts for the increased performance in most generators. This result implies that patterns generators exploit correlate strongly to where aliases exist.

Model	D_{All}	$D_{offline}$	D_{online}	D_{joint}
6Sense	94,178	21,439	17,478	12,819
DET	33,103,213	5,546,423	199,354	74,469
6Tree	5,126,493	331,685	451,825	21,001
6Scan	4,735,290	326,434	402,911	17,881
6Graph	40,093,456	9,517,183	153,312	16,067
6Hit	10,635,015	108,409	1,122,723	43,691
6Gen	1,613,403	212,601	400,396	149,001
EIP	31,386,566	5,232,157	1,441,175	1,495,913

Table 4: Aliased addresses discovered by each TGA on a 50M ICMP scan of seeds with: no dealiasing (D_{All}), only offline dealiasing ($D_{offline}$), only online dealiasing (D_{online}), and both online and offline dealiasing (D_{joint}).

Offline vs. Online Dealiasing. While it is important to understand the effects of dealiasing on seed datasets, what dealiasing actually entails is inconsistent. Prior work tends to deploy one of the approaches discussed in Section 2.2 (offline filtering of IPv6 Hitlist aliases, or 6Gen's online approach). We use both to dealias our seed dataset (32.4M with offline, and 27.3M with online dealiasing), but recognize the need to evaluate how each affects TGA alias discovery.

Table 4 shows the number of aliased addresses discovered by the 8 TGAs using: no dealiasing, only offline dealiasing, only online dealiasing, and both. We restrict our discussion to just ICMP, as we observed similar trends across the other ports and protocols

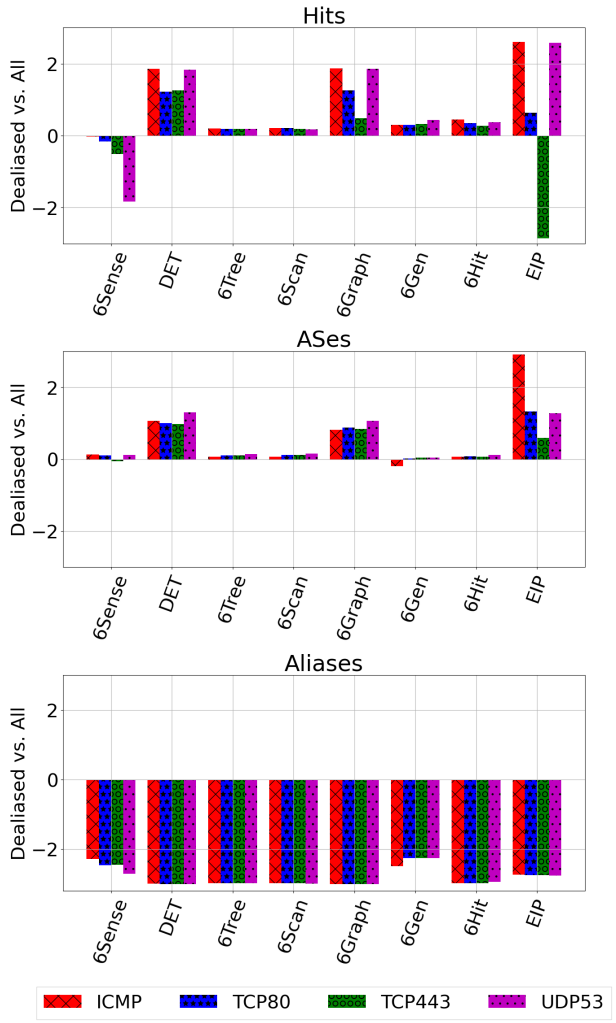


Figure 3: Performance Ratio of Dealiased seeds Hits, ASes, and Aliases to Hits, ASes, and Aliases from the Full dataset.

evaluated. Overall, we observe magnitudes of discovered aliases decrease as aliasing becomes more specific (to the right of the table). For DET, 6Graph, and Entropy/IP, using online dealiasing leads to millions fewer discovered aliases versus offline, demonstrating the need for online dealiasing. Notably, the improvement is not all one-directional: 6Tree, 6Scan, and 6Gen all saw small increases (>200K) in aliases with only online dealiasing compared to offline, and 6Hit notably found 1M more aliases. This suggests there are either aliases that do not follow the statistical pattern of fully responsive /96s as required by the online dealiasing algorithm or rate limiting can cause some inconsistency in online alias discovery. Overall, the almost universally lower aliases discovered when using a joint approach suggests both methods miss some aliases. In addition, since offline dealiasing can drastically shorten preprocessing times (86.1M dealiased addresses from the full dataset were filtered by both methods, but online dealiasing requires sending up to 747M packets to dealias these IPs), this suggests using both is preferable.

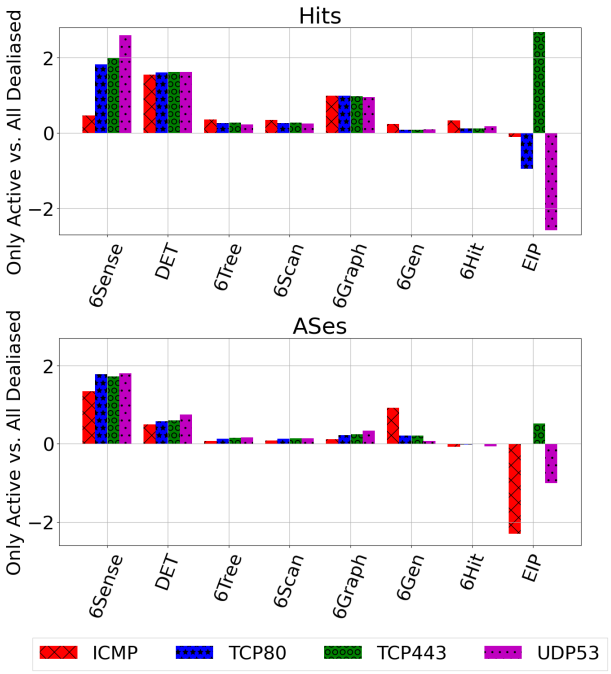


Figure 4: Performance Ratio of Only Active seeds Hits and ASes to Hits and ASes from Active and Inactive seeds.

Overall, these results suggest future work is necessary for optimal dealiasing design.

RQ1.a Takeaway: TGAs should dealias seed datasets as a crucial step in seed dataset preprocessing. Using online approaches (like 6Gen's) is crucial to avoiding aliased regions missed by offline approaches based on alias lists. We suggest dealiasing using a joint online+offline approach. We select the dealiased dataset to refine further in RQ1.b.

6.2 RQ1.b: Does using previously active seeds improve or hurt TGA performance?

Seed datasets contain many IPv6 addresses that do not respond on any port/protocol, or no longer respond to the port/protocol they originally replied to. As these addresses were observed in use (even if not responsive) within at least one data source, they may still prove useful for generators. Their existence in seed datasets provides evidence for active addresses in those regions (e.g., that they appear in traceroutes or AAAA lookups), and they may indicate addressing patterns that are fruitful for generation. However, a risk exists that these no longer active addresses will mislead generators towards regions that no longer contain responsive addresses, or whose addresses are firewalled or blocked.

Thus, while some prior work includes these addresses as input to TGAs [21, 33], many restrict to only addresses observed to be responsive on some port or protocol [26, 51]. Complicating matters, some of this prior work [25, 29, 54] uses responsive addresses directly obtained from the IPv6 Hitlist [23] (supposedly only active IPs) without verifying whether these addresses are still active when performing generation. Per Table 3, only 84% of the IPv6 Hitlist

responds on one of the common ports and protocols evaluated here, meaning 1.4M unresponsive addresses (for our purposes) exist in these supposedly responsive seeds (potentially due to address churn, as suggested in prior work [56]).

Given the inconsistency in how unresponsive addresses are included in TGA seeds, it is vital to understand their effect. We run our 8 TGAs with only the responsive addresses (10,999,613 IPs) and compare to the full dealiased dataset (27,179,296 IPs). Figure 4 shows the Performance Ratio for TGA hits and ASes.

With few exceptions, most generators perform better on hits and ASes when seeds only include responsive IPs. DET finds more than double the hits and 1.5x the ASes. A possible explanation may be that too many seed addresses creates too large of a space to mine for patterns, meaning generators cannot focus on active regions. Many of the inactive regions may no longer be in use or be configured to block or filter incoming packets (such as routers found via Scamper [9]). While this seems intuitive for hits, one may expect active ASes to decrease since the responsive IPs contain only 23,613 ASes compared to 29,477 in the combined dealiased responsive and unresponsive set. TGAs are likely similarly misled on ASes by the number of ASes without any active (or at least discoverable) addresses. Notably when including unresponsive IPs, EIP and 6Hit find more ASes in some experiments, and EIP finds significantly more ICMP and TCP80 hits. This may point to the formerly active addresses revealing address assignment structure those methods can exploit. However, the overall poor performance of those methods negates any benefits of adding these addresses to TGA datasets.

RQ1.b Takeaway: By restricting seeds to only use addresses shown to be responsive on some port or protocol, we can improve TGA performance across generators, ports, and protocols for both hits and active ASes. The improvement is likely because unresponsive addresses mislead generators. We select the responsive seeds to refine further in RQ2.

7 RQ2: How does port or protocol (and active seeds on a specific port/protocol) impact TGA performance?

We so far observed that generator behavior across datasets varies based on the port and protocol scanned. Up until now, our exploration only considered responsive addresses and not what port/protocol they responded to. Section 6.2 showed refining datasets to just active addresses can improve generative performance, so it makes sense to go a step further and consider how refinement to port-specific responsive input datasets affects generator performance (i.e., if training models on just ICMP addresses yields better ICMP results than training models on all responsive addresses).

We here refine to datasets responsive on each port/protocol scanned (e.g., when scanning ICMP, using ICMP active addresses; for TCP80, using TCP80 active addresses, etc). Such exploration follows from Section 6.2, where generating from unresponsive seeds decreased overall performance. Not being active on the port or protocol scanned draws a direct analog. Similarly, refining seeds to just those active on a desired port should increase hits, as it removes regions lacking the desired services on the desired port/protocol.

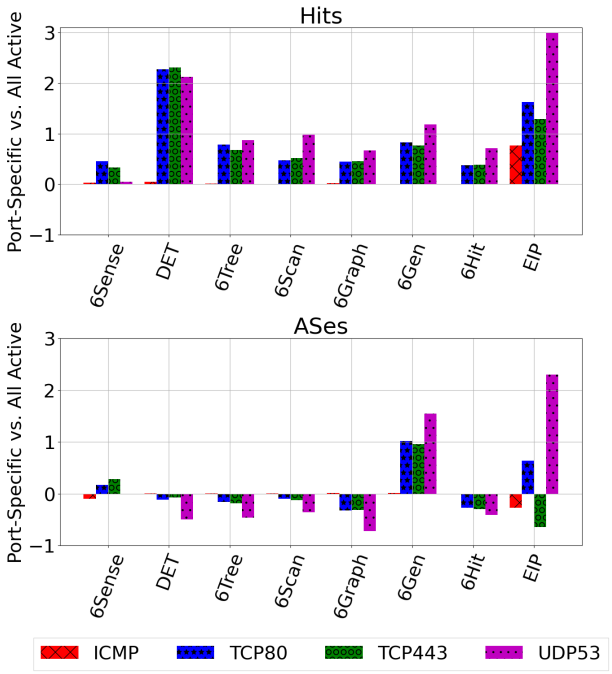


Figure 5: Performance Ratio of port-specific dataset Hits and ASes to All Active Hits and ASes. Limiting input datasets to the target port/protocol improves application protocol (TCP/UDP) hits across models and scan types, has minimal impact on ICMP scans, but typically decreases AS diversity.

While removing inactive addresses did increase active ASes, it is unclear how such observations hold across protocol and port.

We generate and scan 50M addresses with each TGA using each port/protocol: ICMP, TCP80, TCP443, and UDP53 on its port-specific dataset. Figure 5 shows the Performance Ratios of Hits and ASes to the all-responsive dataset.

We find the best approach to dataset composition and scanning varies based on how we evaluate. While generators uniformly find more hits on their port-specific dataset, some sacrifice active ASes to do so.

One explanation is that port-specific datasets lose seed ASes. ICMP shows the least difference of all datasets, discovering on average only 1.09x more hits than the ICMP active dataset, and only 1.03x fewer active ASes. This likely occurs because the majority of the All Active seeds are active on ICMP already (10.7M out of 10.9M), meaning the dataset changes little. Conversely, TCP80, TCP443, and UDP53 show higher hits, typically just below 2x (with the exception of DET which is 5-7x across ports) while typically seeing an overall decrease in active ASes. 6Sense's algorithmic focus on active AS discovery likely led to its negligible differences.

These results suggest a tradeoff: port-specific active datasets can increase hits but may become too specific to discover some active ASes. This makes sense intuitively since the port-specific active dataset is much smaller and includes far fewer ASes (11,762 in TCP80 vs. 23,613 in the All Active dataset). The significance of this tradeoff differs per TGA. For DET (finding 2.9M TCP80 active

addresses vs. 408K on the All Active seeds) it makes sense to accept the AS decrease (3,810 from 4,109) for the sheer number of new hits. A model like 6Graph, only having a 1.35x increase on TCP80 in hits but a 0.81x decrease in active ASes, holds less incentive.

It is important to note that, the All Active dataset is not every responsive address on all 65536 ports, but only active addresses on ICMP, TCP80, TCP443, and UDP53. The ports and protocols a particular experiment may be interested in may vary (with the All Active dataset varying in composition). However, across TGAs, on average 98% of the ASes missed in the port-specific dataset existed in the ICMP dataset and 11% appeared only there. While the exact ports and protocols involved may vary, ICMP responsive addresses make sense to include in seeds when aiming for address diversity.

RQ2 Takeaway: Using scan-target-specific active seeds significantly increases application layer hits discovered by TGAs (in some cases by more than 7x), but it can also decrease discovered ASes (specifically from the ICMP active dataset), leading to a tradeoff between metrics that should be weighed on a per-use-case and per-TGA basis.

To further explore port-specific datasets, we characterize what *other* ports and protocols are discoverable based on these single-port seed datasets in Appendix D.

8 RQ3: How do different seed data sources impact TGA performance?

We thus far discussed how preprocessing and port/protocol can significantly alter TGA performance. However, we have yet to address diversity inherent within the dataset itself. As we've observed, each dataset source (from domain toplists to traceroute topologies) contributes differently to the overall makeup of the seeds (traceroutes providing more ASes, compared to the high hits given by domains and hitlists).

We seek here to quantify the effects each of these seed sources have on TGA output. Two specific questions interest us: *1. Does scanning with these smaller subpopulations provide benefits compared to a larger scale scan across the same number of addresses? Specifically, can smaller subpopulations help find more network diversity?*, and *2. What types of addresses are generated from each seed dataset? Is there a difference in discovered population across seed sources?* This has special relevance to IPv6 scanning, because the large scan space makes targeting scans towards the kinds of hosts/organizations a scanner is interested in crucial.

To evaluate these questions, we run each TGA on responsive seeds from each seed dataset source for 50M IPs. We use the All Active seeds (instead of port-specific seeds), due to concern over the very low population of some dataset sources active on some ports/protocols (such as CAIDA DNS on TCP80/TCP443 or Scamper on UDP53 per Table 3). Our focus on understanding network diversity also makes the All Active dataset appealing based on our understanding of its network diversity properties in RQ2.

Subpopulation Scanning. Table 5 shows TGA results across datasets on ICMP. We compare the combined 50M outputs from all datasets (600M total) per TGA to results from running each TGA with a 600M budget.

We discover that, while large-scale scans using the All Active dataset do tend to find more than twice as many unique hits (likely due to duplicates generated between the smaller datasets), source-specific datasets tend to excel at network diversity, finding over 1000 more ASes with some TGAs. Notably, 6Tree and 6Scan are an exception to this rule, perhaps due to exploration inherent in their algorithms (since they each share a similar formulation). In all, this suggests some utility for running TGAs on smaller sets, and suggests future work is warranted on tailoring seed datasets towards discovering specific populations on the Internet.

	Hits		ASes	
	Combined	600M	Combined	600M
6Sense	31,129,215	86,832,921	16,585	15,337
DET	42,678,773	67,490,924	17,891	14,284
6Tree	8,535,454	13,117,211	12,215	12,878
6Scan	8,689,246	16,787,500	12,319	12,833
6Graph	5,983,188	8,502,276	14,799	13,220
6Gen	7,716,020	14,095,195	9,340	3,617
6Hit	5,478,957	8,195,023	10,629	10,528
EIP	69,550	38,922	3,849	1,037

Table 5: Combined ICMP Hits and ASes across the twelve 50M source experiments compared to All Active output on ICMP using a 600M budget.

AS Characterization. Although hitrates and AS numbers can show the *number* of addresses found, we are also interested in the *kinds* of addresses discovered, especially by each specific seed dataset. In Table 6, we look at the top 3 ASes and total ASes on the combined discovered active addresses (from all 8 TGAs) for each port and protocol using seeds from each source. We manually classify AS organization types. Cloud and hosting providers like Cloudflare, OVH (a French hosting provider), and Huawei were common in addresses discovered from domain seeds. Secrank gave mostly ISPs in China, likely due to Secrank's China-heavy focus [52]. The total number of ASes discovered across each dataset scaled with dataset size as expected. Scamper found fewer TCP80, TCP443, and UDP53 ASes than other traceroute sources.

Meanwhile, datasets deriving from traceroutes or the hitlist found high concentrations of IPs in a mix of sources from ISPs like Claro and Hughes (a satellite provider) to CDNs like Fastly and Cloudflare. Results varied by port and protocol, with UDP53 dominated by Incapsula (a cybersecurity firm and hosting provider), while Cloudflare appeared heavily in ICMP results and OVH appeared in both TCP80 and TCP443.

Notably, ISPs were scattered geographically across datasets. Locations ranged from Nepal to Mexico, from the Netherlands to Japan. The strong geographic distribution suggests TGAs can discover a wide population across the Internet.

RQ3 Takeaway: Each individual source of seed addresses contributes a unique vantage point on the Internet. These unique vantage points can be used by TGAs to discover more network diversity by generating on smaller source-specific datasets. But running TGAs on individual seed sources will decrease hits compared to a single TGA run with a larger budget. We observe discovered AS distributions vary across seeds depending on seed type and port. Each seed dataset tends to show consistency of ASes found between ports, while ICMP and UDP53 tend to show consistency in the ASes

	ICMP				TCP80				TCP443				UDP53			
	1st	2nd	3rd	Total	1st	2nd	3rd	Total	1st	2nd	3rd	Total	1st	2nd	3rd	Total
Censys	50% ⊙	9% ⋈	8% ⋊	7.7K	43% ◎	5% ⊖	5% ⊛	4K	9% ◎	6% ⊖	5% ⊛	3.7K	11% □	9% ★	8% ⊛	2.4K
Rapid7	45% ⊙	11% ⋈	11% ⋊	7.8K	9% ▲	7% ◁	7% ⊖	3.9K	9% ◁	9% ▲	7% ⊖	3.7K	93% ⁕	0.7% □	0.7% ★	2.2K
Umbrella	50% ⊟	47% ♣	1% △	2.1K	12% ◇	12% ★	10% ⊛	1.1K	15% ⊕	11% ★	9% ≀	1.1K	49% ★	8% ⊛	8% ♭	404
Majestic	21% ⊛	17% ≀	8% □	1.4K	97% ⊠	0.5% ⊛	0.3% □	798	97% ⊠	0.5% ⊛	0.3% □	783	30% ★	12% ⊛	10% □	374
Tranco	22% ⊛	17% ≀	12% □	2.8K	18% ⊛	14% □	10% ★	1.2K	18% ⊛	14% □	11% ★	1.1K	34% ★	13% ⊛	12% □	664
Secrank	80% ⊞	5% ◇	2% ≀	1.1K	35% ⊠	35% ⊞	26% ◎	541	41% ⊞	40% ⊠	17% ◎	527	59% ★	9% ⊛	6% ♣	246
Radar	21% ⊛	16% ≀	15% ∽	1.7K	18% ⊛	11% □	11% ★	856	17% ⊛	12% ★	11% ≀	825	37% ★	14% ⊛	10% □	460
Caida	17% ★	10% ▶	8% ∅	3.1K	46% ⊗	11% ⧫	7% ⊵	634	25% ⊗	10% ⊵	4% ⊛	569	19% ⊵	8% ▼	6% Ⓢ	411
Scamper	16% ●	16% ⋊	5% ⋈	15K	86% ○	4% ★	2% ▹	3.2K	89% ○	3% ★	2% ⊗	2.7K	66% ★	12% ♠	4% ℧	1.9K
RIPE	35% ⊙	15% ●	5% ♡	15K	32% ⊡	20% ★	8% ◎	5.7K	16% ★	13% ⊡	7% ◁	5.2K	80% ★	2% ⊘	2% ⊘	3.4K
Hitlist	27% ⊙	8% ◀	7% ●	13K	33% ⊘	11% ★	5% ▽	5.1K	16% ⊘	13% ★	7% ◎	4.6K	66% ★	3% ⋓	3% ♯	2.4K
AddrMiner	27% ⊙	11% ●	4% ⋈	12K	24% ★	23% ×	4% ⊖	4.7K	24% ★	23% ×	4% ⊗	4.3K	93% ★	3% ⁕	0.6% ∞	2.8K

Table 6: Top 3 ASes and total ASes discovered by each Dataset. The top organizations by AS (manually classified) include ISPs/Mobile carriers: Claro (●), Vodafone (⋈), PenTeleData (★), satellite provider Hughes Network Systems (⊡), China Unicom Guangdong (⊟), China Mobile (♣), Sky (⋊), Comteco (⋓), Comcast (∅), Vivacom (℧), Hurricane (⊵), Telkomnet ID (♯), DishNet NP (▹), Mega Cable MX (♡), KPN NL (▶), SoftBank JP (◀), ChongQing Broadcast and TV Broadband (♠), PJMnet (⧫), HeiLongJiang Mobile (⊞), and Hebei Mobile (⊠); Cloud/Hosting/CDNs: Akamai (⊘), OVH (⊛), Huawei (◇), Cloudflare (⊙), DigitalOcean (≀), Confiared (◁), Netactuate (▼), xTom (⊗), tdyun.com (◎), PrivateSystems (♭), Amazon (△), Performive (∽), Fastly (○), Azure (⊕), and Hostinger (□); and Others: Incapsula (★), MysticalKitten (▲), Trex (×), Apple (∞), SmartNet (▽), Brazilian Municipal Government (Ⓢ) and CERNET2 (⁕).

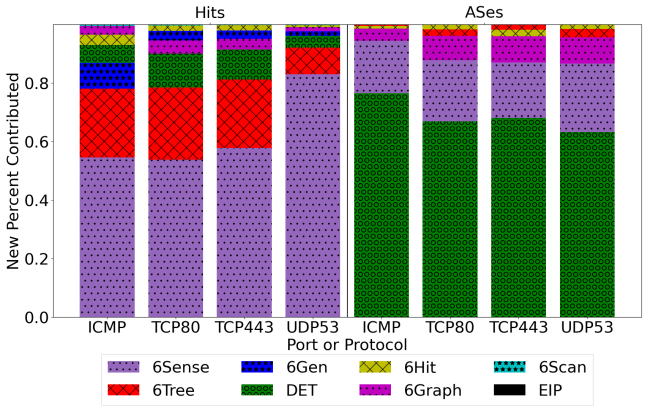


Figure 6: Cumulative total of unique addresses and ASes contributed by each generator, ordered by most unique contributions, broken out by protocol/port. Subsequent generators show only their new contributions, accounting for prior generators. For hits, top contributors are 6Sense, 6Tree, then DET, and 6Gen. For ASes, DET contributes most, followed by 6Sense and 6Graph.

found across datasets (ICMP focusing on Claro, Cloudflare, and Vodafone; UDP53 focusing on OVH and Incapsula).

9 RQ4: What is the overlap in generator output? How do generators perform when used together?

Across all our experiments we observe significant variation between generators, datasets, and ports. A likely and existing best practice [22, 56] is to run multiple generators on a seed dataset and utilize the combined hits and ASes for the given use case. However, thus far we have not explored overlap between generator outputs. Prior work suggests tree-based algorithms (specifically 6Tree and 6Graph) show significant overlap of discovered addresses [56], but we'd like to understand more broadly how all evaluated generators perform together. To do so, we examine the output of all generators across our All Active dataset.

Figure 6 shows the unique cumulative contribution of each generator, ordered by unique contribution. i.e., if one only used the top performing TGA on hits, 6Sense, one would find almost 60% of hits; adding 6Tree would take it to 80% of hits, etc. Notably, for hits, we discover that 6Gen contributes a non-trivial number of unique hits (especially on ICMP), and combinations of 6Sense, DET, 6Gen, 6Graph, and 6Tree tend to discover most addresses. Similarly, DET+6Sense+6Graph cover the vast majority of uniquely discovered active ASes. Notably, 6Scan shows almost no contribution, likely due to its algorithmic similarity to 6Tree, leading it to contribute little to the overall hits or ASes in combination, despite performing competitively on its own.

RQ4 Takeaway: Using multiple generators increases the total yield of both hits and ASes, but a small number of generators yield a supermajority of coverage. The specific generator yielding the best coverage varies first by hit vs AS metric, and second by port/protocol. This suggests that combining multiple TGAs may be a useful approach for generation.

10 RQ5: What are the concrete recommendations and best practices for TGA usage?

Results and observations from experiments lead us to make the following operational recommendations for TGA usage:

- **Dealiasing:** Operators should dealias seed datasets used as input to TGAs, and ensure they use both offline and online dealiasing.

- **Unresponsive Addresses:** We recommend pre-scanning and removing unresponsive seeds.
- **Port-Specific:** Restricting seed datasets to port-specific responsive addresses increases discovered addresses. However, to obtain broader AS and network coverage, we recommend including addresses active on other ports/protocols (especially ICMP).
- **Ports:** It is important to evaluate TGAs across multiple ports and protocols as topology differences can lead to different TGAs performing differently per scan type.
- **Generators:** Across datasets, 6Sense and 6Tree consistently perform best on hits, with DET varying significantly per dataset, but occasionally outperforming others. DET tends to perform best on ASes, with 6Tree, 6Graph, and 6Sense performing comparatively. We recommend using multiple TGAs to optimize for multiple metrics.
- **Combining Generators:** While most prior work evaluates which TGA performs "best", we suggest running multiple TGAs when scanning IPv6 to reach a more representative proportion of the Internet.

11 Concluding Discussion

This work explores best practices for TGA use, including preprocessing, input construction, scanning, and combining TGAs effectively. Broadly, prior work existed in isolation without controlled experiments to understand the best use of these systems. Our work provides a foundational basis on which TGAs can be utilized, evaluated, and compared. Looking towards future work, we distill key lessons learned.

Input datasets have significant impact. We have shown that the data source and type of address utilized significantly impacts TGA output. Future work is warranted exploring new data sources and developing new TGAs specifically engineered to use different data sources.

Dealiasing and checking for activity at scan time is critical. TGA performance degrades significantly when aliases or inactive addresses are introduced. Future work should implement pre-processing before conducting evaluations. Similarly, dealiasing addresses is critical to yielding comparable, correct results. We observed that not all dealiasing is equal, with online dealiasing approaches preferable. However, even current online dealiasing approaches are not perfect, and future work is needed to determine the optimal approach to removing aliases.

Metrics matter. We have shown that the best answer to most questions studied depended on how success was evaluated. It is critical that further work focus on developing reliable and apt metrics for IPv6 Internet scanning.

12 Acknowledgements

The authors thank the anonymous reviewers and shepherd for their thoughtful and productive feedback and guidance during the review process. This work was supported by a Georgia Tech Research Institute Independent Research and Development grant and NSF CNS award 2319315.

References

[1] 6Graph. 2021. https://github.com/Lab-ANT/6Graph.
[2] 6Scan. 2023. https://github.com/hbn1987/6Scan.
[3] 6Sense. 2024. https://github.com/IPv6-Security/6Sense.
[4] Addy Yeow. Bitnodes API. 2024. https://bitnodes.earn.com/.
[5] Ark IPv6 Topology Dataset. 2023. https://catalog.caida.org/dataset/ipv6_allpref_topology. Accessed: 2023-5-26.
[6] Robert Beverly. 2016. Yarrp'ing the Internet: Randomized high-speed active topology discovery. In *Proceedings of the 2016 Internet Measurement Conference*. 413–420.
[7] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P. Rohrer. 2018. In the IP of the beholder: Strategies for active IPv6 topology discovery. In *Proceedings of the Internet Measurement Conference 2018*. 308–321.
[8] Zach Bloomquist. 2023. TLDR 2 - A Continuously Updated Historical TLD Records Archive. https://github.com/flotwig/TLDR-2.
[9] CAIDA. 2023. scamper. https://catalog.caida.org/software/scamper.
[10] Censys. 2023. Censys Internet Scanning Intro. https://support.censys.io/hc/en-us/articles/360059603231-Censys-Internet-Scanning-Intro.
[11] Tianyu Cui, Gaopeng Gou, and Gang Xiong. 2020. 6GCVAE: Gated Convolutional Variational Autoencoder for IPv6 Target Generation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 609–622.
[12] 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 INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.
[13] Tianyu Cui, Gang Xiong, Gaopeng Gou, Junzheng Shi, and Wei Xia. 2021. 6VecLM: Language modeling in vector space for IPv6 target generation. In *ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV*. Springer, 192–207.
[14] The CAIDA UCSD IPv6 DNS Names Dataset. 2023. https://www.caida.org/catalog/datasets/ipv6_dnsnames_dataset/.
[15] The CAIDA UCSD IPv6 Topology Dataset. 2023. https://www.caida.org/catalog/datasets/ipv6_allpref_topology_dataset/.
[16] Peering DB. 2023. Peering DB. https://catalog.caida.org/dataset/peeringdb..
[17] DET. 2023. https://github.com/sixiangdeweicao/DET.
[18] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. Association for Computing Machinery, New York, NY, USA, 542–553. https://doi.org/10.1145/2810103.2813703
[19] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide scanning and its security applications. In *22nd USENIX Security Symposium (USENIX Security 13)*. 605–620.
[20] Entropy/IP. 2018. https://github.com/akamai/entropy-ip.git.
[21] 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.
[22] 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 *Proceedings of the Internet Measurement Conference 2018*. 364–378.
[23] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. In *Proc. of 8th Int. Workshop on Traffic Monitoring and Analysis*. Louvain-la-Neuve, Belgium.
[24] Google. 2023. IPv6 Statistics. https://www.google.com/intl/en/ipv6/statistics.html.
[25] 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 INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.
[26] Bingnan Hou, Zhiping Cai, Kui Wu, Tao Yang, and Tongqing Zhou. 2023. 6Scan: A High-Efficiency Dynamic Internet-Wide IPv6 Scanner With Regional Encoding. *IEEE/ACM Transactions on Networking* (2023), 1–16. https://doi.org/10.1109/TNET.2023.3233953
[27] Liz Izhikevich, Gautam Akiwate, Briana Berger, Spencer Drakontaidis, Anna Ascheman, Paul Pearce, David Adrian, and Zakir Durumeric. 2022. ZDNS: A Fast DNS Toolkit for Internet Measurement. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 33–43. https://doi.org/10.1145/3517745.3561434
[28] Ken Keys. 2010. Internet-scale IP alias resolution techniques. *ACM SIGCOMM Computer Communication Review* 40, 1 (2010), 50–55.
[29] 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 (May 2019), 31–46. https://doi.org/10.1016/j.comnet.2019.03.010
[30] William R Marczak, John Scott-Railton, Morgan Marquis-Boire, and Vern Paxson. 2014. When governments hack opponents: A look at actors and technology. In *23rd USENIX Security Symposium (USENIX Security 14)*. 511–525.

[31] Alexa Top 1 Million. 2023. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.
[32] Majestic Million. 2023. https://majestic.com/reports/majestic-million.
[33] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. 2017. 6Gen - Target generation for internet-wide IPv6 scanning. In *Proceedings of the 2017 Internet Measurement Conference*. ACM, London United Kingdom, 242–253. https://doi.org/10.1145/3131365.3131405
[34] RIPE NCC. 2023. IPMap. https://ipmap.ripe.net/.
[35] Ramakrishna Padmanabhan, Aaron Schulman, Dave Levin, and Neil Spring. 2019. Residential links under the weather. In *Proceedings of the ACM Special Interest Group on Data Communication*. 145–158.
[36] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2018. Tranco: A research-oriented top sites ranking hardened against manipulation. *Proceedings 2019 Network and Distributed System Security Symposium* (2018).
[37] Radar 2023. Cloudflare Radar. https://radar.cloudflare.com/.
[38] Rapid7. 2013. Project Sonar. https://www.rapid7.com/research/project-sonar/.
[39] Rapid7 FNDNS 2023. Rapid7 Forward DNS. https://opendata.rapid7.com/sonar.fdns_v2/.
[40] Erik Rye and Dave Levin. 2023. IPv6 hitlists at scale: Be careful what you wish for. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 904–916.
[41] scanv6. 2024. https://github.com/IPv6-Security/scanv6.
[42] Guanglei Song, Lin He, Zhiliang Wang, Jiahai Yang, Tao Jin, Jieling Liu, and Guo Li. 2020. Towards the construction of global IPv6 hitlist and efficient probing of IPv6 address space. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.
[43] Guanglei Song, Jiahai Yang, Lin He, Zhiliang Wang, Guo Li, Chenxin Duan, Yaozhong Liu, and Zhongxiang Sun. 2022. {AddrMiner}: A Comprehensive Global Active {IPv6} Address Discovery System. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*. 309–326.
[44] Guanglei Song, Jiahai Yang, Zhiliang Wang, Lin He, Jinlei Lin, Long Pan, Chenxin Duan, and Xiaowen Quan. 2022. DET: Enabling Efficient Probing of IPv6 Active Addresses. *IEEE/ACM Transactions on Networking* (2022), 1–15. https://doi.org/10.1109/TNET.2022.3145040
[45] Spamhaus. 2023. The Spamhaus Project. https://www.spamhaus.org.
[46] RIPE NCC Staff. 2015. RIPE ATLAS: A global internet measurement network. *Internet Protocol Journal* 18, 3 (2015), 2–26.
[47] Lion Steger, Liming Kuang, Johannes Zirngibl, Georg Carle, and Oliver Gasser. 2023. Target acquired? evaluating target generation algorithms for ipv6. In *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA)*. Naples, Italy.
[48] G. Huston T. Narten and L. Roberts. 2011. *IPv6 Address Assignment to End Sites*. RFC 6177. https://www.rfc-editor.org/rfc/rfc6177.html
[49] Johanna Ullrich, Peter Kieseberg, Katharina Krombholz, and Edgar Weippl. 2015. On Reconnaissance with IPv6: A Pattern-Based Scanning Approach. In *2015 10th International Conference on Availability, Reliability and Security*. IEEE, Toulouse, France, 186–192. https://doi.org/10.1109/ARES.2015.48
[50] Umbrella 2023. Cisco Umbrella Popularity List. http://s3-us-west-1.amazonaws.com/umbrella-static/index.html.
[51] Grant Williams, Mert Erdemir, Amanda Hsu, Shraddha Bhat, Abhishek Bhaskar, Paul Pearce, and Frank Li. 2024. 6Sense: Internet-Wide IPv6 Scanning and its Security Applications. In *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA. https://www.usenix.org/conference/usenixsecurity24/presentation/williams
[52] Qinge Xie, Shujun Tang, Xiaofeng Zheng, Qingran Lin, Baojun Liu, Haixin Duan, and Frank Li. 2022. Building an Open, Robust, and Stable Voting-Based Domain Top List. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 625–642. https://www.usenix.org/conference/usenixsecurity22/presentation/xie
[53] 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 INFOCOM 2022 - IEEE Conference on Computer Communications*. 1679–1688. https://doi.org/10.1109/INFOCOM48880.2022.9796925
[54] Tao Yang, Bingnan Hou, Zhiping Cai, Kui Wu, Tongqing Zhou, and Chengyu Wang. 2022. 6Graph: A graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning. *Computer Networks* 203 (2022), 108666.
[55] Gang Zheng, Xinzhong Xu, and Chao Wang. 2020. An Effective Target Address Generation Method for IPv6 Address Scan. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. IEEE, 73–77.
[56] Johannes Zirngibl, Lion Steger, Patrick Sattler, Oliver Gasser, and Georg Carle. 2022. Rusty Clusters? Dusting an IPv6 Research Foundation. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 15 pages. https://doi.org/10.1145/3517745.3561440

Source	Collected	Source	Collected	Source	Collected
Censys CT	12-11-2023	Tranco	11-30-2023	Scamper	12-07-2023
Rapid7	11-26-2021	SecRank	11-30-2023	RIPE Atlas	12-11-2023
Umbrella	12-01-2023	Radar	12-04-2023	IPv6 Hitlist	12-06-2023
Majestic	12-12-2023	CAIDA DNS	11-30-2023	AddrMiner	12-12-2023

Table 7: Date of dataset collection.

Source	Domains	AAAAs	Unique IPv6 IPs
Censys Certs	2,517,952,172	117,503,681	19,446,042
Rapid7 FDNS	1,931,094,237	97,487,730	9,278,627
CAIDA DNS Names	1,004,287	57,197	59,348
Cisco Umbrella	1,000,000	229,207	261,717
Majestic Million	1,000,000	285,110	130,751
Tranco	1,000,000	278,461	141,325
SecRank	999,505	113,809	127,963
Cloudflare Radar	1,000,011	284,459	150,319

Table 8: Domain dataset volume breakdown. In addition to the 9.2M IPs discovered with DNS lookups, the Rapid7 dataset contained 15.2M IPs from archival lookups in November 2021 not included in this table.

A Ethics

Our work complies with all applicable ethical standards of our home institution. Further, we follow scanning best practices and ethical guidelines for Internet scanning proposed by ZMap [19]. We maintain an opt-out page on all scanning devices and IP addresses, respond promptly, and honor all requests to allow targets to easily opt out of scanning. We note here that prior work involving IPv6 scanning tools, specifically 6Scan [26], did not include blocklisting capability, and online models using this scanner (specifically 6Scan and 6Hit) required we add blocklisting to 6Scan. Further, we randomize scan order and *significantly* rate-limit all scans to ten thousand packets per second to ensure that even in the presence of aliases we do not inadvertently cause network load issues.

B Experiment Dates

Table 7 provides the dates each dataset was collected. All datasets were collected between November 30th, 2023 and December 12th, 2023, except Rapid7 (where we only had access to an archival version from 2021). Domains were combined across data sources and uniquely resolved between December 22nd, 2023 and January 1st, 2024. Scans of ICMP, TCP80, TCP443, and UDP53 on the combined dataset (across sources) were conducted between February 29th, 2024 and March 11th, 2024.

C Domain Collection

Table 8 provides the total domains, domain lookups that returned AAAA responses, and IPs from each DNS-based dataset source in Section 5. Censys CT Logs and Rapid7 are the largest suppliers of domains and, by extension, IP addresses, although Rapid7 domains return fewer total IPs. Meanwhile, toplists had high AAAA response rates and IP discovery rates for their size, potentially because they focused on well used domains (more likely to have associated addresses).

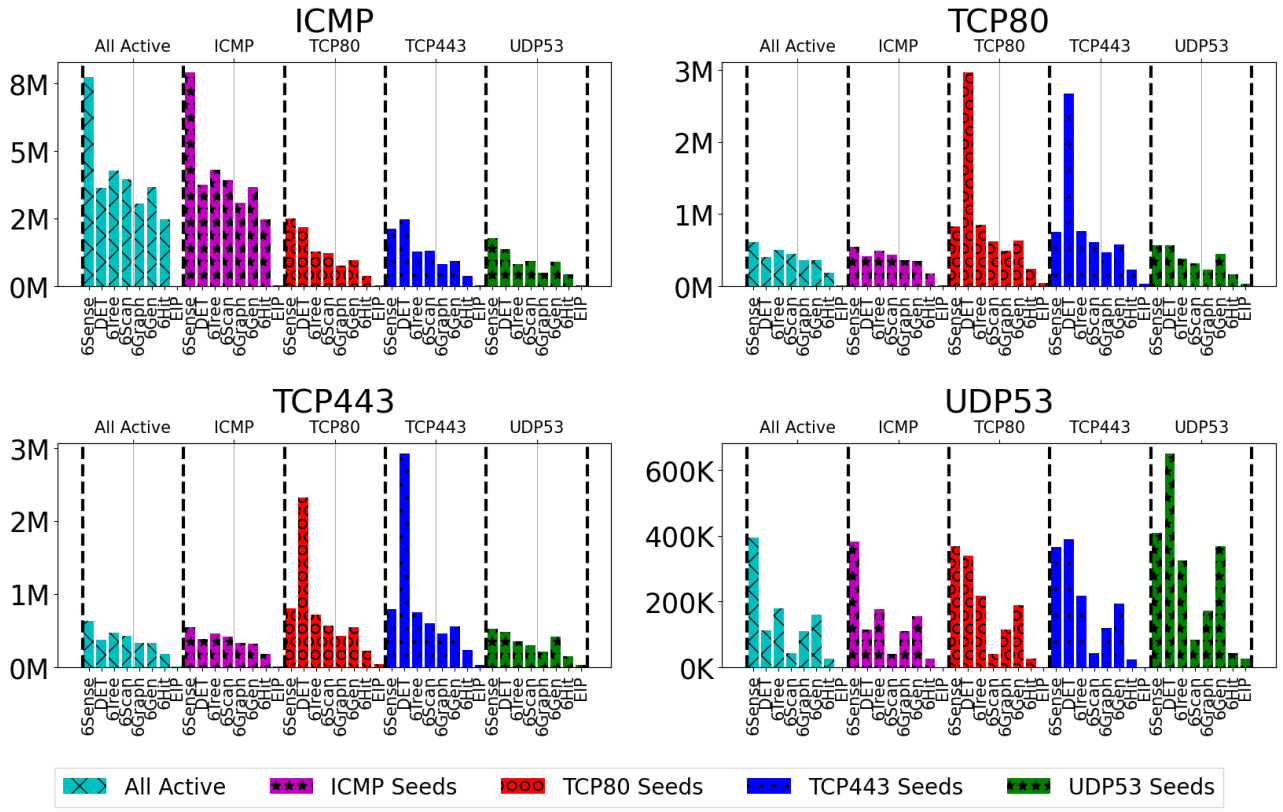


Figure 7: Active addresses for each port/protocol scanned, split by port/protocol input dataset.

D What addresses can TGAs discover using seeds active on a *different* port/protocol than scanning with?

In addition to the analysis of port-specific seeds, we seek to quantify the utility of scanning various ports or protocols based on the output of a model trained against a different port/protocol's active input dataset. In other words, we seek to understand the effectiveness of scanning for TCP443 when we generate off of ICMP-active addresses (for example). This helps quantify the types of devices and accessible services discoverable with TGAs from these regions.

We look at each seed dataset active on ICMP, TCP80, TCP443, and UDP53 on each of our four ports/protocols. Figure 7 shows sub-figures representing the scan results for each of our 4 protocols/ports, sub-divided by input dataset.

We find the utility of this scanning approach varies. For TCP (top-right, bottom-left) and UDP scans (bottom-right), there is little commonality across datasets, including for ICMP. For ICMP scans (top-left), ICMP and All Active perform roughly the same. Interestingly, for ICMP the yield of TCP and UDP input datasets are, while worse, in the same order of magnitude as directly training and scanning on those datasets/protocols. This indicates a significant number of discovered hosts on those protocols respond to pings.

This leads to some interesting trends in evaluating TGAs. DET tends to perform inordinately well on smaller, more specific datasets (likely due to its online component being able to hone into active regions more quickly with a smaller dataset). While 6Sense is highly variable on ICMP hits across datasets, it is constant per-port across other datasets (only slightly decreasing responsiveness to non-port-specific input). This trend is common across generators: TGAs tend to perform similarly across their non-port-specific datasets on any specific port (though this may be an artifact of the choice of ports, since TCP80 and TCP443 would be expected, and are observed, to have similar populations).

Takeaway: ICMP primarily matches the All Active dataset on all ports. TGAs on most ports perform best with the port-specific dataset, but tend to show little variation across other non-port-specific datasets.

E Experiment Raw Numbers

For reference, we provide raw numbers for Hits and ASes found from each TGA on each dataset in RQs 1.a, 1.b, 2, and 3. For RQs 1 and 2, Table 9 has ICMP results, Table 10 has TCP80 results, Table 11 has TCP443 results, and Table 12 has UDP53 results. Raw numbers for RQ3 ICMP are in Table 13. Raw numbers for RQ3 TCP80, TCP443, and UDP53 are in Table 14 and Table 15.

ICMP									
	Dataset	**6Sense**	**DET**	**6Tree**	**6Scan**	**6Graph**	**6Gen**	**6Hit**	**EIP**
Hits	All	5,736,672	271,400	2,936,804	2,713,244	355,263	2,552,740	1,452,851	996
	Offline Dealiased	5,737,832	994,479	3,336,451	3,105,493	1,321,170	3,049,401	1,928,240	3,642
	Online Dealiased	5,575,516	1,144,787	3,319,592	3,118,443	1,527,214	3,084,314	1,963,049	9,822
	Active-Inactive	5,642,761	1,151,329	3,358,278	3,127,628	1,527,138	3,116,638	1,968,050	14,290
	All Active	7,726,582	3,627,670	4,268,384	3,958,222	3,041,654	3,658,612	2,472,054	13,350
	ICMP	7,897,442	3,743,219	4,290,726	3,938,637	3,083,903	3,671,463	2,479,509	22,481
	TCP80	2,498,285	2,187,884	1,288,495	1,231,900	750,886	964,006	378,163	39,515
	TCP443	2,136,817	2,460,849	1,274,445	1,302,250	803,581	935,690	383,396	30,066
	UDP53	1,773,098	1,374,447	815,629	943,345	491,883	908,311	436,491	31,252
ASes	All	4,445	4,473	8,356	8,207	6,326	1,769	6,906	39
	Offline Dealiased	4,777	8,841	8,748	8,600	10,556	1,644	7,478	1,368
	Online Dealiased	4,862	9,368	8,731	8,604	11,050	1,601	7,520	2,246
	Active-Inactive	4,862	9,390	8,790	8,640	11,080	1,563	7,227	2,933
	All Active	12,803	13,147	9,250	9,145	12,038	2,961	6,880	388
	ICMP	12,005	13,184	9,293	9,174	12,114	2,984	6,866	323
	TCP80	6,960	6,082	3,591	3,612	4,349	1,081	2,852	80
	TCP443	6,401	5,598	3,350	3,374	4,069	1,057	2,700	33
	UDP53	5,009	3,459	1,949	1,906	2,690	1,477	2,005	390

Table 9: Raw Numbers for ICMP Experiments in RQ1-RQ2.

TCP80									
	Dataset	**6Sense**	**DET**	**6Tree**	**6Scan**	**6Graph**	**6Gen**	**6Hit**	**EIP**
Hits	All	163,858	51,803	370,343	329,162	74,811	275,917	135,276	14,311
	Offline Dealiased	153,477	192,708	418,941	378,458	301,871	326,738	180,679	32,928
	Online Dealiased	160,217	122,862	417,177	375,820	183,201	353,161	172,342	21,266
	Active-Inactive	146,720	122,780	420,298	377,557	183,072	337,288	171,306	21,978
	All Active	605,590	408,508	501,618	451,725	363,028	356,331	185,339	11,424
	ICMP	546,801	413,687	486,278	439,835	362,477	347,157	179,166	13,651
	TCP80	824,549	2,960,647	854,775	622,558	489,487	626,920	239,123	38,533
	TCP443	755,872	2,671,464	768,358	610,653	465,409	580,902	226,867	28,804
	UDP53	565,143	569,695	381,059	321,299	224,385	445,038	158,381	28,510
ASes	All	772	1,381	2,633	2,382	1,835	414	2,412	5
	Offline Dealiased	846	2,589	2,827	2,610	3,169	440	2,576	14
	Online Dealiased	843	2,755	2,816	2,607	3,352	423	2,571	14
	Active-Inactive	830	2,772	2,839	2,586	3,377	420	2,548	13
	All Active	3,255	4,109	3,107	2,823	3,925	482	2,520	13
	ICMP	2,820	3,987	2,997	2,787	3,765	475	2,449	18
	TCP80	3,642	3,810	2,790	2,640	3,167	974	2,103	20
	TCP443	3,291	3,488	2,535	2,396	2,868	927	1,903	12
	UDP53	1,856	1,455	1,126	1,099	1,331	804	1,073	53

Table 10: Raw Numbers for TCP80 Experiments in RQ1-RQ2.

TCP443									
	Dataset	**6Sense**	**DET**	**6Tree**	**6Scan**	**6Graph**	**6Gen**	**6Hit**	**EIP**
Hits	All	178,315	45,845	350,265	311,924	123,006	252,193	142,136	25,396
	Offline Dealiased	111,696	169,947	392,590	352,287	521,407	306,295	186,428	495,148
	Online Dealiased	100,776	112,713	391,835	352,258	170,735	331,269	169,295	449
	Active-Inactive	126,038	112,086	395,229	353,443	170,591	313,778	170,445	647
	All Active	634,417	377,852	475,671	423,637	336,522	332,371	184,256	11,421
	ICMP	547,493	383,080	457,459	413,780	335,068	324,773	176,071	13,584
	TCP80	802,541	2,319,295	715,002	563,768	423,496	550,155	225,433	38,407
	TCP443	791,786	2,924,154	751,468	599,700	456,677	559,871	238,804	28,696
	UDP53	524,879	484,553	352,848	294,167	208,870	416,827	144,963	28,150
ASes	All	760	1,265	2,484	2,222	1,721	414	2,230	8
	Offline Dealiased	814	2,327	2,658	2,424	2,860	438	2,364	15
	Online Dealiased	819	2,454	2,647	2,404	3,029	419	2,371	15
	Active-Inactive	737	2,492	2,670	2,420	3,066	427	2,348	12
	All Active	2,719	3,727	2,949	2,659	3,618	493	2,335	17
	ICMP	2,426	3,601	2,818	2,623	3,454	484	2,256	15
	TCP80	3,085	3,434	2,522	2,373	2,797	932	1,890	15
	TCP443	3,288	3,558	2,599	2,439	2,935	955	1,916	11
	UDP53	1,622	1,276	1,047	1,018	1,186	749	994	38

Table 11: Raw Numbers for TCP443 Experiments in RQ1-RQ2.

UDP53									
	Dataset	**6Sense**	**DET**	**6Tree**	**6Scan**	**6Graph**	**6Gen**	**6Hit**	**EIP**
Hits	All	119,634	8,028	134,465	31,239	13,359	112,035	18,306	11
	Offline Dealiased	36,405	28,859	149,872	35,745	49,134	151,397	23,087	68
	Online Dealiased	36,021	33,264	150,950	36,179	56,586	160,125	23,188	121
	Active-Inactive	28,694	33,086	152,372	35,161	56,613	149,768	23,566	159
	All Active	395,266	111,160	177,803	41,747	109,464	159,425	26,703	11
	ICMP	382,198	114,378	176,338	40,720	110,314	153,873	25,493	56
	TCP80	367,302	340,249	216,593	41,132	114,771	188,557	25,330	137
	TCP443	365,623	389,887	217,024	42,405	118,541	193,755	24,349	169
	UDP53	408,291	649,471	324,171	82,468	172,159	368,108	43,267	26,748
ASes	All	396	526	1,286	877	788	233	1,028	4
	Offline Dealiased	432	1,215	1,415	969	1,582	246	1,120	9
	Online Dealiased	416	1,315	1,409	969	1,677	226	1,112	12
	Active-Inactive	428	1,343	1,421	974	1,668	241	1,114	10
	All Active	1,726	2,244	1,584	1,074	2,096	254	1,070	5
	ICMP	1,553	2,242	1,584	1,096	2,050	271	1,064	9
	TCP80	1,668	1,788	1,237	987	1,290	385	801	12
	TCP443	1,444	1,723	1,175	974	1,246	368	783	6
	UDP53	1,707	1,602	1,160	843	1,282	796	811	38

Table 12: Raw Numbers for UDP53 Experiments in RQ1-RQ2.

ICMP RQ3									
	Dataset	**6Sense**	**DET**	**6Tree**	**6Scan**	**6Graph**	**6Gen**	**6Hit**	**EIP**
Hits	AddrMiner	6,959,532	3,464,180	2,175,059	2,698,640	1,456,791	1,685,543	1,187,411	7,871
	Caida	142,302	92,423	95,776	106,979	76,894	110,057	100,218	17,409
	Censys	7,771,049	2,805,713	2,115,544	2,158,615	1,083,707	1,867,506	564,504	4,130
	Radar	37,882	259,590	208,028	241,334	103,820	205,433	213,069	2,412
	Hitlist	7,663,345	4,416,337	3,668,631	3,630,623	2,490,217	3,239,285	2,230,777	3,219
	Majestic	142,422	216,624	223,893	222,974	101,307	262,208	221,183	796
	Rapid7	6,190,040	1,765,119	1,647,727	1,806,818	757,912	978,142	362,582	19,649
	Ripe	5,312,611	3,805,408	1,181,734	1,285,658	708,736	872,269	662,052	4,320
	Scamper	5,841,448	3,439,817	3,016,633	3,004,835	1,560,735	2,398,161	1,363,372	10,593
	Secrank	223,054	3,092,061	315,628	328,666	101,111	268,962	267,170	1,664
	Tranco	162,894	175,992	220,543	246,667	100,443	238,727	225,940	3,698
	Umbrella	154,256	26,138,917	466,501	301,636	154,570	222,562	470,214	3,922
	600M	86,832,921	67,490,924	13,117,211	16,787,500	8,502,276	14,095,195	8,195,023	38,922
ASes	AddrMiner	9,141	8,370	6,726	6,795	7,299	1,405	5,076	354
	Caida	1,040	886	604	616	1,064	1,550	504	359
	Censys	5,916	5,262	2,556	2,688	3,104	631	2,141	57
	Radar	58	1,104	373	369	683	510	340	275
	Hitlist	9,978	9,408	6,754	6,811	7,849	1,037	5,359	164
	Majestic	1,124	796	288	281	454	270	273	134
	Rapid7	6,356	5,051	3,142	3,205	3,609	1,026	2,659	89
	Ripe	11,514	11,003	7,963	7,975	9,066	3,651	6,987	1,690
	Scamper	10,792	9,575	6,512	6,528	8,149	6,133	5,340	2,235
	Secrank	765	520	146	150	251	226	136	180
	Tranco	1,697	1,177	351	345	713	741	333	402
	Umbrella	1,648	1,021	600	608	821	489	594	180
	600M	15,337	14,284	12,878	12,833	13,220	3,617	10,528	1037

Table 13: Raw Numbers for Source Specific ICMP Experiments in RQ3.

RQ3 Hits: TCP80, TCP443, UDP53									
	Dataset	**6Sense**	**DET**	**6Tree**	**6Scan**	**6Graph**	**6Gen**	**6Hit**	**EIP**
TCP80	AddrMiner	620,763	828,528	404,038	381,302	250,004	262,577	147,769	1,112
	Caida	11,546	1,652	536	544	1,028	1,672	380	63
	Censys	631,377	286,369	234,587	194,870	128,135	179,353	87,444	323
	Radar	25,181	46,541	112,705	107,667	45,732	102,675	79,002	815
	Hitlist	1,455,118	452,908	293,882	276,723	210,520	227,316	117,215	55
	Majestic	60,085	6,518,041	118,663	103,614	51,550	136,474	66,193	131
	Rapid7	349,283	419,781	204,003	157,717	128,542	152,875	79,106	18,259
	Ripe	1,180,084	373,688	177,052	155,953	129,303	97,152	105,012	199
	Scamper	3,603,328	72,777	132,518	95,002	82,113	117,563	25,477	98
	Secrank	287,811	8,522,351	130,809	111,151	40,367	111,041	55,278	459
	Tranco	67,642	43,214	116,722	107,623	46,956	117,158	71,435	1,613
	Umbrella	53,171	79,586	72,486	58,048	34,601	65,478	54,209	2,374
TCP443	AddrMiner	617,919	801,102	378,305	364,875	232,654	249,816	142,952	1,230
	Caida	7,893	1,807	841	818	1,238	1,753	633	120
	Censys	443,341	270,423	225,278	186,774	124,090	170,492	91,315	151
	Radar	16,596	42,777	103,523	99,848	42,211	96,280	73,242	830
	Hitlist	1,199,601	411,909	268,507	255,861	193,313	202,586	121,863	58
	Majestic	54,414	6,213,079	110,510	96,285	47,489	124,593	63,104	115
	Rapid7	353,218	386,194	195,365	152,802	123,796	144,450	81,367	18,250
	Ripe	686,599	350,078	160,867	137,085	109,799	92,834	108,053	149
	Scamper	3,747,248	33,416	35,741	33,818	19,129	31,131	14,230	71
	Secrank	49,728	7,745,195	100,061	84,502	31,452	82,749	42,512	357
	Tranco	59,328	39,018	107,319	99,898	42,189	106,673	66,973	1,272
	Umbrella	55,037	79,012	93,937	74,170	38,441	79,168	71,053	869
UDP53	AddrMiner	3,505,113	146,901	169,310	40,050	90,669	152,654	20,061	100
	Caida	1,786	580	374	201	518	721	111	31
	Censys	50,520	35,532	47,859	20,328	25,652	32,775	7,099	8
	Radar	3,484	18,254	37,012	9,844	7,275	30,079	6,856	141
	Hitlist	268,604	51,937	63,666	33,443	39,556	53,313	15,988	20
	Majestic	16,703	17,650	35,890	11,766	14,933	47,347	6,342	36
	Rapid7	1,326,079	52,914	37,055	16,428	23,711	25,195	7,474	27
	Ripe	334,117	75,991	49,172	15,105	29,561	33,003	13,110	137
	Scamper	192,877	10,701	15,561	9,038	7,783	12,908	3,759	100
	Secrank	7,866	6,836	24,918	3,271	3,863	6,064	1,228	26
	Tranco	17,063	17,259	35,946	11,371	9,631	31,147	7,243	384
	Umbrella	17,178	10,341	22,987	1,841	6,442	22,048	1,495	25

Table 14: Raw Hits for Source Specific TCP80, TCP443, and UDP53 Experiments in RQ3.

RQ3 ASes: TCP80, TCP443, UDP53									
	Dataset	**6Sense**	**DET**	**6Tree**	**6Scan**	**6Graph**	**6Gen**	**6Hit**	**EIP**
TCP80	AddrMiner	2,892	2,996	2,716	2,641	2,740	652	1,933	29
	Caida	291	211	103	103	151	225	71	38
	Censys	2,447	2,906	1,800	1,780	1,958	534	1,365	12
	Radar	21	657	283	262	414	267	244	38
	Hitlist	3,054	3,300	2,829	2,683	2,924	541	2,062	10
	Majestic	477	493	262	243	345	232	234	21
	Rapid7	2,539	2,663	1,894	1,776	2,033	668	1,417	18
	Ripe	3,525	3,904	3,176	3,128	3,260	1,540	2,542	164
	Scamper	2,243	1,100	1,014	984	936	978	879	83
	Secrank	328	346	130	126	188	148	108	29
	Tranco	709	625	293	272	411	333	267	57
	Umbrella	705	698	395	386	528	332	364	29
TCP443	AddrMiner	2,542	2,728	2,510	2,423	2,498	647	1,834	26
	Caida	290	178	94	93	137	186	70	29
	Censys	2,201	2,767	1,813	1,788	1,932	563	1,373	10
	Radar	19	636	278	257	398	250	244	37
	Hitlist	2,676	3,116	2,693	2,545	2,744	546	1,914	14
	Majestic	473	498	255	234	335	231	232	17
	Rapid7	2,383	2,585	1,883	1,751	1,973	695	1,415	14
	Ripe	3,093	3,636	2,966	2,894	2,995	1,485	2,353	118
	Scamper	1,872	818	867	845	724	781	739	60
	Secrank	329	340	126	122	177	143	112	25
	Tranco	657	614	289	269	400	311	257	45
	Umbrella	729	734	420	414	557	346	394	31
UDP53	AddrMiner	1,534	1,695	1,401	1,053	1,391	306	799	21
	Caida	192	143	73	54	100	138	38	20
	Censys	1,450	1,449	768	630	806	271	475	6
	Cloudflare	9	325	136	109	173	159	85	33
	Hitlist	472	1,758	1,376	955	1,467	279	759	12
	Majestic	233	192	103	91	128	105	82	13
	Rapid7	1,383	1,472	867	688	896	295	527	11
	Ripe	1,870	2,258	1,614	1,204	1,694	725	999	108
	Scamper	1,359	540	544	297	506	535	271	86
	Secrank	184	106	46	31	68	57	26	14
	Tranco	401	306	143	112	185	193	97	34
	Umbrella	245	198	96	60	116	93	59	14

Table 15: Raw ASes for Source Specific TCP80, TCP443, and UDP53 Experiments in RQ3.