Still on Target? An Evaluation of IPv6 Target Generation Algorithms

Lion Steger, Liming Kuang, Johannes Zirngibl, Georg Carle, Oliver Gasser

***Abstract*—Internet measurements are a crucial foundation of IPv6-related research. Due to the infeasibility of full address space scans for IPv6 however, those measurements rely on collections of reliably responsive, unbiased addresses, as provided *e.g.*, by the *IPv6 Hitlist* service. Although used for various use cases, the hitlist provides an unfiltered list of responsive addresses, the hosts behind which can come from a range of different networks and devices, such as web servers, customer-premises equipment (CPE) devices, and Internet infrastructure.**

In this paper, we demonstrate the importance of tailoring hitlists in accordance with the research goal in question. By using PeeringDB we classify hitlist addresses into six different network categories, uncovering that 42% of hitlist addresses are in ISP networks. Moreover, we show the different behavior of those addresses depending on their respective category, *e.g.*, ISP addresses exhibiting a relatively low lifetime. Furthermore, we analyze different Target Generation Algorithms (TGAs), which are used to increase the coverage of IPv6 measurements by generating new responsive targets for scans. We use seed sets, *e.g.*, based on the categorized Hitlist. We evaluate the performance of TGAs under various conditions and find generated addresses to show vastly differing responsiveness levels for different TGAs. Furthermore, we evaluate of algorithm run times and differences between multiple TGA runs.

***Index Terms*—Internet measurements, IPv6, IPv6 scanning, machine learning, target generation**

I. Introduction

THE adoption of IPv6 is continuously increasing, with on average 40% of all Google users connecting via IPv6 in March 2023 [1]. Due to the sheer size and sparse population of the IPv6 address space, exhaustive scans such as in IPv4 [2], [3] are infeasible in the IPv6 Internet. Therefore, Internet measurements targeting IPv6 hosts rely on up-to-date collections of responsive addresses, often known as *Hitlists*. Moreover, the success of these measurements heavily depends on the quality of their input, reliable targets, and high coverage of the active IPv6 Internet. While use cases for such hitlists can vary greatly, hitlists are usually a collection of addresses belonging to different types of devices, such as routers, web servers, or customer-premises equipment (CPE) devices, treated as a homogeneous set. This is very inefficient for many measurement studies, as these targets can be expected to be found in completely different network types. For example, a study on self-hosted video platforms would mainly target educational and company networks, while a study on web content will target vastly different networks, such as Content Delivery Networks (CDNs) and hosting providers. These studies could profit from a categorization of hitlist addresses, as this could allow more focused scans resulting in a reduced scanning overhead and lower load on the network.

The most popular and commonly used *IPv6 Hitlist* by Gasser *et al.* [4], [5] combines IPv6 addresses from different sources and performs regular scans to ensure reliable responsiveness. However, little is known about the current and historic composition of the *IPv6 Hitlist*, namely which categories of addresses it contains and whether there is a bias towards CPE devices, routers, or servers. This makes the use of the hitlist unnecessarily difficult and inefficient for many measurement studies. We address this problem by analyzing the different network categories represented in the data provided by the hitlist service and showing how the categorization of the contained addresses improves the hitlists' usability.

In addition to hitlists, different approaches exist to increase IPv6 address coverage, *e.g.*, by generating new targets. This is often achieved through so-called Target Generation Algorithms (TGAs), which employ different methods such as machine learning [6], [7] and other pattern recognition techniques [8], [9]. Similar to hitlists, little is known about characteristics of TGAs with respect to input from different categories, whether they exhibit biases towards specific address categories, or whether their results can be improved given more specific input. Therefore, existing TGAs could benefit from categorizing their input, enabling them to spend their algorithmic and scanning budget on application-tailored target generation.

In this extended paper based on our previous work [10], we perform an in-depth analysis of the *IPv6 Hitlist* as well as TGAs by categorizing IPv6 addresses. This research enables fellow researchers to make better use of the *IPv6 Hitlist* and TGAs. Our contributions in this work are:

1. ***IPv6 Hitlist* address categorization:** We analyze the *IPv6 Hitlist* by Gasser *et al.* [4] with respect to IP address categories. We show that it includes addresses from a variety of categories, *e.g.*, Internet Service Provider (ISP) and Network Service Provider (NSP) in the input but also the set of responsive addresses, finding a general bias towards ISP networks with $42\,\%$ of responsive addresses.

This work was partially funded by the German Federal Ministry of Education and Research under the projects PRIMEnet (16KIS1370), 6G-life (16KISK002) and 6G-ANNA (16KISK107). The authors would like to thank the author of 6Sense, Grant Williams for their assistance in the technical setup of 6Sense.

Lion Steger and Georg Carle are with the Chair of Network Architectures and Services, Technical University of Munich, 85748 Garching, Germany. (E-Mail: {steger, carle}@net.in.tum.de)

Liming Kuang is a student at the TUM School of Computation, Information and Technology, Technical University of Munich, 85748 Garching, Germany. (E-Mail: liming.kuang@tum.de)

Johannes Zirngibl is with the Department Internet Architecture, Max Planck Institute for Informatics, 66123 Saarbrücken, Germany. (E-Mail: jzirngib@mpi-inf.mpg.de)

Oliver Gasser is with IPinfo Inc., Seattle, WA 98136, USA. (E-Mail: oliver@ipinfo.io)

2. **Characterization of address categories:** We evaluate whether addresses from differing categories exhibit different behavior over time. We show that addresses from educational and content serving networks are more stable with a median of over 200 days uptime, while ISP addresses are often only responsive during a single scan. ISP and NSP addresses almost exclusively respond to ICMP, with less than $10\,\%$ response rate to any other protocol.
3. **Effectiveness analysis of TGAs:** We evaluate the effectiveness of 13 TGAs to identify previously unknown addresses in experiments two years apart. Furthermore, we analyze whether categorized input leads to a change in behavior for TGAs, finding stark contrasts in metrics such as number of generated and responsive addresses and responses to different protocols. For example, output generated from the ISP category has up to $50\,\%$ responsiveness, however almost exclusively to ICMP with below $10\,\%$ for any other protocol, whereas CDN addresses can generate $65\,\%$ responsiveness to HTTP. In this extended version, we add a total of 11 new seed sets. We analyze randomly sampled (10k, 100k, and 1M), AS-sampled (10, 100, 1k per AS) and protocol-filtered (ICMP, TCP/80, TCP/443, UDP/443, and UDP/53) addresses. We find that the effect of sampling on TGA output is not consistent across algorithms (see Section VI-B).
4. **Robustness analysis:** We perform multiple TGA runs on the same seed data and compare the results to identify TGAs that generate more robust output. Our analysis shows that robustness varies between algorithms both regarding generated addresses and represented ASes (see Section VI-B).
5. **Stability analysis:** We analyze the temporal stability of the addresses generated in the first generation run. We observe a constant decline of responsive addresses with roughly 25% of addresses being responsive today, while the amount of represented ASes remains relatively stable (see Section VI-A).
6. **Data and Code:** We publish our adaptations to the used TGAs, generated and responsive addresses, analysis scripts and tools used throughout this work, as well as an ongoing categorization of the *IPv6 Hitlist* addresses [11]. In order for users of the *IPv6 Hitlist* to benefit from our findings, we update the service, include the newly discovered addresses and provide categorized statistics and data to the established service.

II. Background

We introduce TGAs, the *IPv6 Hitlist* service [4], [5] and used data to categorize IP addresses.

A. Target Generation Algorithms

Discovery of responsive targets for IPv6 scans is an important task since full address space scans are infeasible. Besides hitlists, combining targets from existing sources, *e.g.*, resolution of domain names, public sources and traceroutes, a variety of so-called Target Generation Algorithms (TGAs) were developed. TGAs take a completely different approach to this problem. They try to identify patterns within existing collections of responsive addresses called the *seed data set*, and generate new targets which are likely to be responsive, called a *candidate set*. These addresses can be used as input for scans and tested for their responsiveness. Some of these algorithms also implement their own dynamic scanning mechanisms, which allows them to adapt their search strategy based on intermediate scanning results, and achieve a higher response rate. Table I provides an overview about algorithms we evaluated and used in 2023 and 2025. These were all the algorithms found in related work which provided publicly accessible source code.

Table I: List of target generation algorithms with publicly available code used in this study. We conducted measurements in 2023 (i) and in 2025 (ii). Based on insights from the measurements in 2023 and new publications, the set of TGAs for the second measurement differs.

Year	Authors	Name	Scanning	Ref	(i)	(ii)
2016	Foremski et al.	Entropy/IP	Static	[9]	x	
2018	Foremski et al.	eip-generator	Static	[4]		x
2019	Liu et al.	6Tree	Dynamic	[12]	x	x
2020	Song et al.	DET	Dynamic	[13]	x	x
2020	Cui et al.	6GCVAE	Static	[6]	x	
2021	Cui et al.	6VecLM	Static	[14]	x	
2021	Cui et al.	6GAN	Static	[7]	x	
2021	Hou et al.	6Hit	Dynamic	[15]	x	x
2022	Yang et al.	6Graph	Static	[8]	x	x
2022	Yang et al.	6Forest	Static	[16]	x	x
2022	Song et al	AddrMiner-S	Dynamic	[16]		x
2023	Hou et al.	6Scan	Dynamic	[17]	x	x
2024	Williams et al.	6Sense	Dynamic	[18]		x

B. IPv6 Hitlist Service

The *IPv6 Hitlist* service from Gasser *et al.* [4] was started in 2018 and is maintained since. It collects IPv6 addresses from different sources and conducts scans for ICMP, TCP/80 (HTTP) and TCP/443 (HTTPS), UDP/53 (DNS) and UDP/443 (QUIC) on a regular basis. It was updated in 2022 by Zirngibl *et al.* [5] to improve the quality of the service. Their hitlist holds over 1.09 billion unique IPv6 addresses. Before scanning, they apply different filters, including a blocklist used to ensure opt-out possibilities for networks and ethical scanning. After this, addresses from *aliased* prefixes are being marked and removed.

Gasser *et al.* [4] described aliased prefixes as subnets for which every contained address is mapped to and responded to by one single host, *e.g.*, through the `IP_FREEBIND` feature of Linux. Zirngibl *et al.* [5] showed that some of these prefixes are only fully responsive and used by multiple hosts. However, each of these prefixes (mostly /64) is infeasible to scan by itself and introduces massive biases of the hitlist. Therefore, the *IPv6 Hitlist* service runs a detection and filters the prefixes. The list of addresses after all filters is then scanned with probes for different protocols, of which $25\,\mathrm{M}$ were responsive to at least one protocol as of May 2025.

C. Categorization

PeeringDB, run by a community of network-operators, collects information about peerings and interconnections of net-

works around the world. Alongside this information, network operators can assign a category to their Autonomous System (AS) from twelve categories, including *Content* (Content Delivery Network, short CDN), *Cable/DSL/ISP* (Internet Service Provider, short ISP), *Educational/Research* (Universities, Research Institutes, short EDU), *NSP* (Network Service Provider, Transit networks) and *Non-Profit* (Non-Profit Organizations, short ORG), which are the five categories relevant in this work. Alongside PeeringDB, there was an AS classification by CAIDA, but it was discontinued in 2021 [19]. Furthermore, Ziv *et al.* [20] proposed ASdb in 2021, a system utilizing machine learning approaches to categorize networks at AS level with high accuracy. We did however not consider it in our project since their data only goes back to 2021 while the historic data from Gasser *et al.* [4] starts at 2018.

III. Related Work

The unpredictability of active addresses in the vast IPv6 address space leaves a lot of room for innovative discovery approaches, making the field of TGAs very interesting within IPv6 research. Discovery strategies were already described in RFC7707 [21] based on drafts dating back to 2012. In 2015, Ullrich *et al.* [22] were among the first to publish on this topic. They propose an algorithm which iterates through different patterns of a training set, selecting sub-patterns with the highest amount of matching addresses. New addresses are generated from combining the undetermined bits of the patterns, outperforming the strategies laid out in RFC7707. In 2016, Foremski *et al.* [9] presented Entropy/IP, while Murdock *et al.* [23] presented 6Gen in 2017. The latter identifies dense regions in the input seeds and grows each input address into an independent cluster based on Hamming distance. All addresses which are inside the clusters and do not belong to the input seeds are regarded as candidate addresses. The authors claim that 6Gen outperforms Entropy/IP by a factor of 1-8 for identical input data sets. These early results already show large differences between TGAs and a detailed comparison including different input sets is required. More recently, several new TGAs have been published. These include dynamic TGAs such as 6Sense [24] as well as algorithms based on different machine learning methods such as 6GAI [25], which uses a Generative Adverserial Network (GAN), 6MCBLM [26], which uses a multi-scale Convolutional Neural Network (CNN), 6Diffusion [27], which uses a diffusion model, and 6Vision [28], which applies image encoding methods. Other TGAs employ pattern mining methods, such as 6Community [29], which uses a community discovery algorithm to filter the address space, and 6Subpattern [30], which introduces the concept and analysis of subpatterns in address allocation patterns. The field of IPv6 topology discovery also saw different new publications presenting new algorithms, including Sweeper [31], TreesTrace [32], 6Search [33] and 6Former [34].

The remaining TGAs collected for this work (see Table I) follow similar approaches. They extract structural information from IPv6 seed sets and apply different methodologies to improve the quality of generated addresses. They report different response rates which are hardly comparable. In 2022, Zirngibl *et al.* [5] applied four TGAs during their improvement of the *IPv6 Hitlist*. They find that 6Graph and 6Tree generate the highest number of responsive addresses but do not evaluate algorithms in more detail and different input scenarios.

Williams *et al.* [18] presented a study comparing eight different TGAs in 2024. They use different seed data sets, such as addresses responsive to specific protocols, different public data sets and de-aliased data sets, analyzing the impact of the seeds. Their results show that online de-aliasing decreases the generation of addresses from aliased prefixes, which underpins our de-aliasing approach. Using addresses responsive to specific protocols as seed can lead to more responsive candidates in fewer ASes, decreasing the diversity of the output. They lastly find that DET, 6Sense, and 6Tree outperform other models, while no single generator performs best across all data sets and metrics. This aligns with our results presented in Section VI-B.

Rye *et al.* [35] took a slightly different approach when introducing *edgy* in 2020, focusing on the efficient discovery of the IPv6 periphery, *i.e.*, not servers or clients, but last hop routers. With *edgy* they were able to discover more than $64\,\mathrm{M}$ active last hop router addresses. One year later, Li *et al.* [36] describe a similar approach, discovering more than $50\,\mathrm{M}$ last hop router addresses through tracerouting non-existent IPv6 addresses in known or suspected customer subnets of ISPs. Lastly, Beverly *et al.* [37] presented their work focussing on IPv6 topology discovery. They develop and analyze strategies to collect new interfaces by efficient TTL-limited probing of target address sets, finding $1.3\,\mathrm{M}$ new router interfaces from their single vantage point.

IV. Data Sources and Target Generation

In the following, we describe our data sources and the approach to evaluate the collected TGAs introduced in Section II.

A. Target Generation Methodology

Figure 1 shows our pipeline to test TGAs on the different input files. In our study we run and evaluate the 13 TGAs listed in Table I. We ran a first experiment in 2023 with 10 algorithms. They run on a machine with an NVIDIA GeForce RTX 2080 GPU, a 24-core Intel Xeon Silver 4214 CPU and $256\,\mathrm{GB}$ of RAM.

For our second experiment in 2025, we selected an updated set of TGAs for evaluation based on findings from our first run and related work. We added three new algorithms. In turn, we omitted the four algorithms we deemed most inefficient in the first experiment to reduce the runtime of our second experiment. We adapted our hardware setup to the requirements of the new algorithms, specifically 6Sense, which requires a machine to have scanning capabilities as well as a GPU. The machine used in this experiment features an AMD EPYC 7542 32-core processor, 1TB of RAM, and a Nvidia GeForce 2080 Super GPU with 8GB of VRAM.

We run the static TGAs without any modifications apart from input files or output hyper-parameters. We modify the hyper-parameters number of epochs for 6GAN and the generation budget for 6GCVAE. To run algorithms in a feasible timeframe, we set the total number of epochs of 6GAN to ten and run

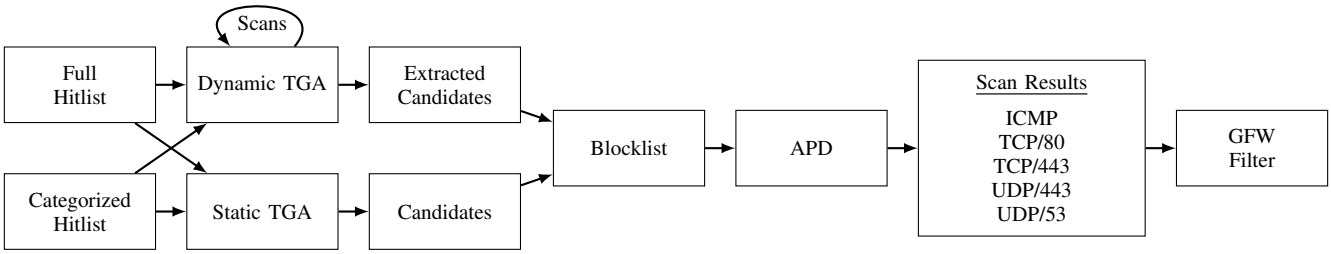

Figure 1: Pipeline to analyze TGAs (see Table I) and their performance within different IP address categories.

6VecLM with only the first of the predefined temperature hyperparameters. 6GAN offers multiple modes for seed classification, of which we choose the *Entropy Clustering* method since the authors report the highest number of generated addresses for this method, which is the metric we optimize for. Since 6GAN and 6VecLM define the amount of input that is processed via hard-coded values and Entropy/IP is not intended for input greater than $100\,\mathrm{k}$ addresses, we randomize the input data set with a static, reproducible seed. Whenever possible we set the output budget to $10\,\mathrm{M}$, since we wanted to keep the size of the candidate sets at a similar scale to the input, *i.e.*, the hitlist. We implement the approach described by the authors of 6Graph to generate candidates from the dense regions identified by their algorithm. For 6Forest, this process is not fully described or implemented. Therefore, we follow the same procedure as for 6Graph, generating distance-based targets and additionally generating full combinations if the number of wildcards, *i.e.*, free dimensions, is smaller than four. For 6Sense and the configured budget of $10\,\mathrm{M}$, Williams *et al.* [18] recommend changing the *ppi* parameter from $1\,\mathrm{M}$ to $100\,\mathrm{k}$ and the *allocation_gradient_threshold* parameter from 0.005 to 0.05, which we used.

Before running the algorithms with dynamic scanning capabilities, we modify them *(i)* to conform to our scanning parameters and *(ii)* to output not only the addresses responsive to their scans but also the addresses which they probed, *i.e.*, the addresses which they consider to be in the *candidate set*. We do this because the integrated scanning mechanisms of the algorithms only scan with ICMP probes, while we want to apply our own scanning mechanisms with a variety of protocol probes. Furthermore, in order to compare the response rate of both dynamic and static algorithms, we have to scan the candidate sets for both instead of only the results of the dynamic algorithms. In order to achieve consistency with the static algorithms, we run each dynamic TGA with the same $10\,\mathrm{M}$ scanning budget.

After the successful generation of all candidate sets, we combine them into one target file. As a first step, the target file is stripped of duplicates and filtered by applying a blocklist, which we actively maintain in order to adhere to ethical scanning guidelines (see Section IV-D). Next, we conduct aliased prefix detection, as described by Gasser *et al.* [4] and additionally use the known aliased prefixes from the *IPv6 Hitlist* service as a filter. The remaining, non-aliased addresses are then used as input to the ZMapv6 port scanning tool[1]. We scan from a single vantage point in Munich, Germany, connected to the German National Research and Education Network. As a last step, the responses to the UDP/53 (DNS) probes are passed through a GFW filter, which we describe in Section IV-C.

Two iterations of target generation were conducted on *2025-02-26* and *2025-03-11*, while the combined set of candidates was scanned on *2025-04-11* on the same machine. The first generation run is evaluated throughout the following sections, while the second run is conducted to analyze the robustness of the algorithms, see Section VI-B. We largely follow the same scanning procedure as in the original experiments described in Section IV-A. The only difference to the original methodology lies in the aliased prefix detection process, where we modified the threshold of responses that are needed per prefix for it to be counted as aliased. We lowered the original number of 16 to 10 responses per prefix to account for the large number of addresses in one single AS (AS13335, Cloudflare) that were scanned during the process. While the scanned addresses are shuffled in the beginning of the scanning process, the number of probes sent to the AS within a short amount of time is still high enough to potentially trigger rate limiting. We set the new threshold of 10, as suggested as the optimal value by Erdemir *et al.* [38].

During the analysis stage, we take the following steps: First, all duplicates and overlaps with the respective seed set are removed for all candidate sets. Then the filtered candidate sets and filtered scan file are matched to identify the responsive portions of the candidate sets. This provides the final result for each TGA.

B. Seed Sets

We use different seed sets to evaluate their effect on TGAs. We used six different seed sets for our initial experiments in 2023, and extended our set based on further insights and related work for experiments in 2025.

In 2023, we use the full list of responsive addresses from the *IPv6 Hitlist* service from March 3, 2023 as input for TGAs for experiments. Furthermore, we divide the input into filtered versions, *i.e.*, the addresses inside the active hitlist filtered based on the PeeringDB categories *Content*, *ISP*, *NSP*, *Non-Profit*, and *Educational*. While PeeringDB has more network categories, we excluded all categories with less than $5\,\%$ representation in the hitlist, additionally including the two categories *Educational* and *Non-Profit* to test the algorithms on smaller seed sets.

In 2025, we again started with an up-to-date, full set of responsive addresses from the IPv6 Hitlist for our second

[1]https://github.com/tumi8/zmap

Table II: Properties of the seed data set.

	#ASN	#Prefix	#IP	#1 Cat.
Full	21.76k	75.78k	43.28M	NSP (35%)
CDN	1.19k	8.16k	8.96M	CDN (100%)
EDU	777	1.83k	134.72k	EDU (100%)
ISP	5.72k	20.22k	9.88M	ISP (100%)
NSP	2.31k	13.32k	14.99M	NSP (100%)
ORG	347	732	22.33k	ORG (100%)
DS10k	718	2.37k	10.00k	NSP (34%)
DS100k	2.69k	7.87k	100.00k	NSP (34%)
DS1M	7.92k	22.18k	1.00M	NSP (35%)
AS10	21.76k	34.26k	145.75k	Others (46%)
AS100	21.76k	48.19k	638.48k	Others (38%)
AS1k	21.76k	60.55k	1.89M	ISP (36%)
ICMP	21.23k	67.57k	24.04M	CDN (36%)
TCP80	11.71k	26.25k	2.32M	CDN (50%)
TCP443	10.64k	24.32k	1.79M	CDN (60%)
UDP53	7.96k	15.92k	426.66k	Others (41%)
UDP443	3.33k	7.30k	310.06k	CDN (56%)

experiments. However, the second experiment also features an extended selection of seed data sets, shown in Table II. We added three sets of addresses that were uniformly sampled from the full data set, ranging from 10k to 1M addresses, as well as three sets of addresses sampled from ASes. For the latter data set, the full data was randomized and 10 to 1k addresses were selected from each AS represented in the data. Lastly, we selected five data sets which are filtered based on the responsiveness to probes on specific ports common for a respective protocol. We select the same ports that are used in our experiments (ICMP ping, TCP/80, TCP/443, UDP/53, UDP/443).

C. GFW Filtering

As described in Figure 1, the responses to our DNS probes are post-processed with a *GFW Filter*. Our probes contain `AAAA` queries for `www.google.com` and frequently receive responses with addresses from the *Teredo* prefix in their answer section. These responses do not originate from legitimate hosts, but are instead likely injected by the Great Firewall of China (GFW) for reasons of censorship [5]. `google.com` is on the list of censored domains in China [39] and no web service for `www.google.com` is reachable at the returned addresses. Following the description of Zirngibl *et al.* [5], we filter all responses containing addresses from the *Teredo* prefix in their answer section and do not count them as responsive in our work.

In both our scans in 2023, however, we see a substantial change in the format of the injections, as we receive responses containing addresses from Facebook's network in their answer section. This change in behavior indicates that the GFW tries to adapt the IPv6 injections to the format of their IPv4 injections, which contain addresses from a fixed pool of IPv4 addresses, including addresses from Facebook and Twitter [40]. Until 2023, every returned Teredo address encoded a corresponding address from the IPv4 pool in the last 32 bits of the address. Starting in 2023, a separate pool of IPv6 addresses from similar networks such as Facebook is being returned. We argue that, for these scans and for probes querying `www.google.com`, filtering out responses containing addresses from Facebook is sufficient, as our scans show similar response rates to DNS probes as the *IPv6 Hitlist* service. To the best of our knowledge, this new behavior of the GFW has not been documented before. DNS scans targeting IPv6 addresses need to take this behavior into account and adjust the filtering pipeline accordingly. Since we, however, expect that the GFW can change its behavior in unpredictable ways, we chose not to adapt the filter of the *IPv6 Hitlist* service to this new type of injection, and instead changed the domain name which is used in the regular query probes as of 2023. The new domain name is not censored by the GFW and does not trigger any injections, which we expect to remain the same in the future. We argue that this yields more stable and usable results for researchers using the *IPv6 Hitlist*. We furthermore verified that this change in domain name does not influence the regular responses through the historic data of the *IPv6 Hitlist*, which has implemented the same change in March 2024.

D. Ethical Considerations

During this work, we strictly follow ethical considerations for scanning as described in [41], [42]. We limit the rate of all scans, apply a blocklist and filter aliased prefixes based on our own detection but also the list of published aliased prefixes by Gasser *et al.* [4]. We evaluated dynamic TGAs whether they adhere to our scan limits and executed them in an environment where we can monitor their behavior and apply our own blocklist. We inform about our scans based on reverse DNS, a website hosted on the scanning machine and in WHOIS. We respond to all opt-out requests and add address ranges to our internal blocklist.

V. Hitlist Categorization

We analyze the *IPv6 Hitlist* composition with regard to different network categories. We use the complete historic data from the *IPv6 Hitlist* service [4] from July 1, 2018 until March 3, 2023. We analyze the historic data to gain insights into its categorical composition over time and the responsiveness and stability within each category. To map addresses to the AS announcing the respective prefix, we use historic Border Gateway Protocol (BGP) Route Views data [43] for one route collector from each scan date. These mappings to ASes are further used to identify the respective category based on historic PeeringDB data [44].

First, we analyze the different network categories of the *IPv6 Hitlist*'s input and responsive addresses. The network categories represented in the *IPv6 Hitlist* show different prevalence and behavior. Figure 2 shows the distribution of addresses across categories in the full hitlist input as well as its responsive part. The responsive addresses as well as the full hitlist are dominated by ISP and CDN addresses, with almost $50\,\%$ combined.

Next, we analyze the responsiveness in more detail, by looking at different probe protocols and network categories. Figure 3 shows how many protocol-specific responses the latest scan receives per category, relative to the total number of IP addresses per category which responded to at least one protocol probe. Addresses belonging to CDNs have the highest relative number of responses to HTTP and HTTPS probes, with a

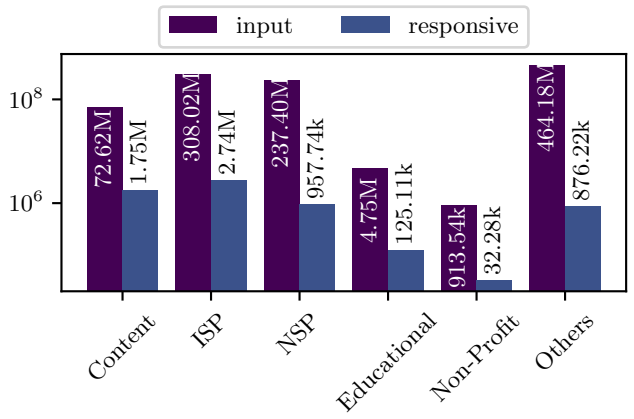

Figure 2: Prevalence of different categories in the *IPv6 Hitlist* on March 3, 2023. Note the logarithmic y-axis.

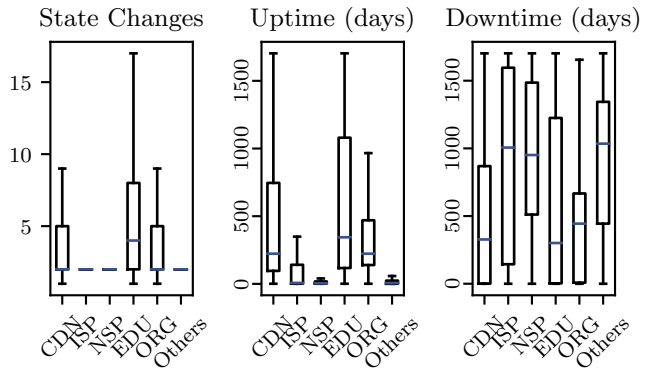

Figure 4: Stability of responsive IPv6 addresses of the *IPv6 Hitlist* per category. Data from scans is used from July 2018 to March 2023, while addresses newly discovered from November 2022 onwards are excluded to reduce the impact of newer scans.

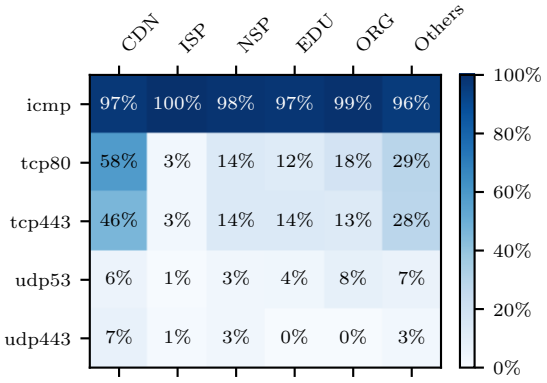

Figure 3: Responses to the different protocols per category within the *IPv6 Hitlist* on March 3, 2023.

low, but still comparatively large number of QUIC responses. This is expected, since web hosting via HTTP/S is one of the primary functions of CDNs, which are also among the first to deploy QUIC at scale [45]. ISP addresses, on the other hand, show almost no response to any protocol other than ICMP.

The high total share of ISP addresses in the hitlist, together with the low response rate to any protocol other than ICMP shows the importance of categorizing hitlists before using them as input for application specific scans, as a large part of scanning traffic can be avoided by carefully selecting target addresses from specific categories.

To better understand the stability of addresses within the *IPv6 Hitlist*, we analyze the categories represented in the hitlist over time using three *IP stability* metrics. First, the number of *state changes*, *i.e.*, the number of times an IP was added to or removed from the responsive part of the hitlist. This can be seen as a lower-bound for the times an IP address changes between online and offline. Second and third, we look at the summed up number of uptimes and downtimes of each address, starting when an IP address is first added to the hitlist, and ending at the time of analysis. These three metrics combined make up the *IP stability* of an address over time. A stable IP address has a small number of state changes, high uptime, and low downtime, the opposite being true for an unstable IP address. For this analysis, in order to avoid analyzing IP addresses with not enough historic data, we exclude all addresses added to the hitlist within the last 100 days.

Figure 4 shows that the different network categories exhibit very distinctive behavior in IP stability. Most addresses from the categories ISP, NSP, and Others have exactly one state change, but median uptimes of less than a week, meaning that they are included once in the responsive hitlist for seven days and never again afterwards. ISP networks often use prefix rotation to avoid tracking of their clients and enhance their privacy [46], [47], which means that devices like home routers often change IP address. Including them in hitlists leads to an increase in unstable targets, which is underlined in the results of this analysis. This also applies to NSP networks, which offer similar services and partly to IP addresses in the Others category. In contrast to this, addresses in CDN, Educational, and Non-Profit networks have much higher uptimes, as addresses hosting content have to be available reliably. The higher number of state changes in these networks can be due to maintenance periods or changes in ownership of the respective servers.

Key Takeaway: For longitudinal measurement studies which focus on protocols other than ICMP, addresses from categories such as NSP and ISP should be used with care, as they have only very limited periods of responsiveness and generally respond less to protocols other than ICMP, compared to addresses from Content Delivery, Educational or Non-Profit Organization networks.

VI. Target Generation Algorithms

We evaluate the performance and value of TGAs in two different experiments in 2023 and 2025. The initial experiment provided us with first insight into the performance of TGAs and their reaction to different seed sets. Furthermore, we added the addresses to the IPv6 Hitlist to allow an evaluation of the

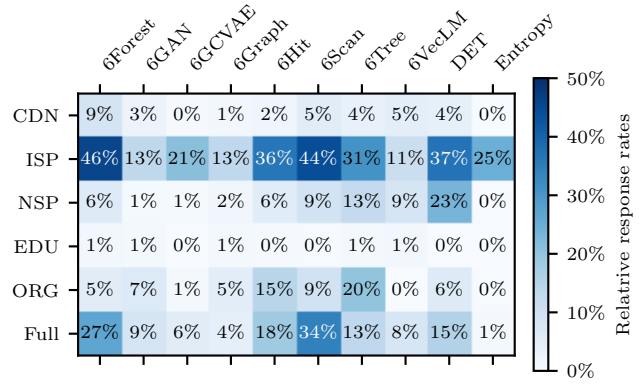

Figure 5: Relative response rate for every candidate set generated by the TGAs with different categorized input sets as well as the full hitlist as input in 2023.

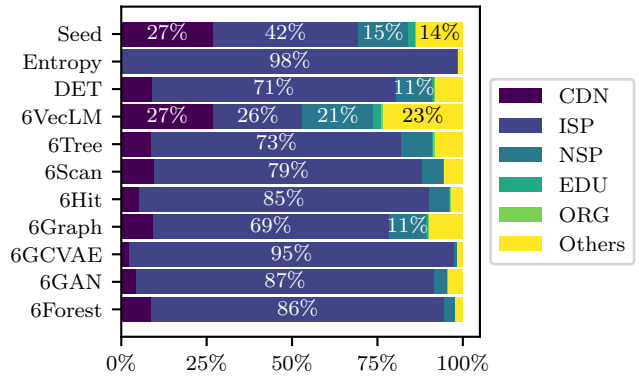

Figure 6: Category distribution of responsive addresses from each Target Generation Algorithm applied on the full set of responsive addresses from the *IPv6 Hitlist* in 2023.

temporal stability of candidates. Based on these insights, we extended our methodology and executed a second experiment with additional seed sets and an adapted set of TGAs.

A. Initial TGA Experiment

Our initial experiment was driven by the diverse composition of the *IPv6 Hitlist* with respect to covered network categories and the differences in responsiveness and IP stability for these categories. We evaluated the generated candidate sets of 10 TGAs in two ZMap scans on March 17 and March 23, 2023 as described in Section IV-A. The scans combined the generation based on the full set of responsive addresses as well as the categorized input sets. The second scan was conducted due to an error in the first one, through which the candidate sets of the dynamic algorithms were not included. For this second scan, including only the candidate sets from the dynamic algorithms, Aliased Prefix Detection (APD) was replaced with a filter for known aliased prefixes from the *IPv6 Hitlist* service. We merge the results from both scans and present the results in the following, together with the different metrics used for the evaluation.

Generation rate and candidate set size. The various algorithms generate vastly different numbers of candidate addresses. Moreover, the generated addresses are also highly dependent on the used seed set. As elaborated in Figure 2, the prevalence of categories in the hitlist and therefore the size of the categorized seed sets vary. Therefore, we compare the *generation rate* of the algorithms, *i.e.*, the size of the candidate set relative to the seed set size. Algorithms like 6VecLM and 6GAN have relatively low generation rates, which is due to the fact that these algorithms limit the amount of processed input to a hard coded value (see Section IV). Algorithms such as 6Graph can generate more than 100 M candidate addresses, which is 1638 % of the size of the seed sets. With the Non-Profit seed set, 6VecLM is unable to generate a candidate set, as the seed set is smaller than the predefined input size, which we could not successfully modify. For the exact sizes and generation rates, see Appendix Table VII.

Response rates. Internet measurement studies are not only dependent on a scanning budget, but also strive to avoid unnecessary probes which are unlikely to trigger responses. Therefore, it is important to analyze the response rate, *i.e.*, the portion of addresses which responds to at least one protocol, for the different candidate sets. As can be seen in Figure 5, a larger candidate set does not lead to a higher response rate. Instead, response rates are more strongly linked with the input set category as well as the difference between dynamic and static algorithms. Dynamic algorithms, due to their ability to adapt their generation strategy based on the results of their scans, have among the top response rates for all categories, up to 45 % for some. On the other hand, static algorithms rarely show response rates over 15 %, with 6Forest being one of the few exceptions. Using ISP addresses as input shows the best response rates for almost all algorithms, even better than with uncategorized input. Candidate addresses generated from educational networks, on the contrary, have the lowest response rate at hardly over 1 % for any algorithm.

Category distribution in responsive addresses. While all TGAs receive the same input, not only do their candidate sets vary greatly in size, but also in the distribution of represented network categories. Figure 6 shows the category distributions in the candidate sets generated by the algorithms and the seed set when using the full hitlist as input. Most algorithms show a strong bias towards ISP addresses, which are also present in the seed data set, although at a much lower percentage. Especially the relatively small percentage of generated CDN addresses is in stark contrast to the ratios of the seed set. When using categorized input, all but two algorithms generate 95–100 % of their addresses in the same category as the input. The only exceptions are 6GCVAE and Entropy/IP, which generate up to 62 % and 13 % from other categories for some inputs, respectively.

AS origin distributions. While categorization on an AS level via PeeringDB already gives us some information of the origin of the contained addresses, the exact AS distributions still hold some more information. The cumulative AS distribution of the candidate sets generated from the full hitlist are shown in Figure 7. Most candidate sets generated by the TGAs are more

Table III: Amount of candidate (cand.) and responsive (resp.) addresses generated by the algorithms when using different categories as well as the full hitlist as seed data set.

	6Forest		6GAN		6GCVAE		6Graph		6Hit		6Scan		6Tree		6VecLM		DET		Entropy	
	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.
Content	2M	174k	487k	13k	3M	14k	35M	443k	10M	231k	9M	491k	11M	417k	78k	4k	9M	361k	6M	8k
ISP	3M	2M	410k	55k	845k	179k	25M	3M	8M	3M	8M	4M	11M	3M	18k	2k	8M	3M	6M	1M
NSP	2M	128k	521k	4k	3M	15k	31M	527k	10M	552k	9M	884k	9M	1M	66k	6k	2M	382k	6M	16k
Educational	1M	19k	316k	3k	700k	585	2M	22k	24M	100k	10M	38k	11M	107k	84k	1k	1M	745	4M	3k
Non-Profit	711k	39k	125k	9k	284k	3k	296k	15k	20M	3M	10M	946k	8M	2M	0	0	6M	356k	4M	14k
Full	2M	494k	486k	41k	2M	111k	106M	5M	18M	3M	6M	2M	35M	5M	49k	4k	8M	1M	6M	59k

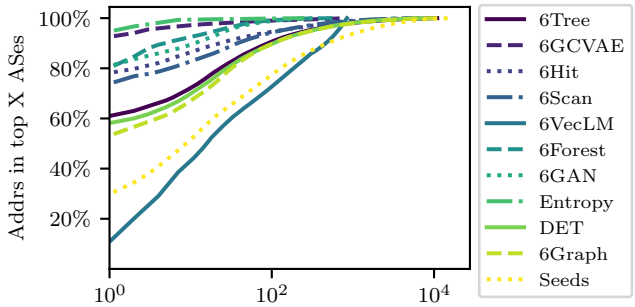

Figure 7: Cumulative AS distribution of the responsive candidate sets generated by the algorithms using the full hitlist as input in 2023. Note the logarithmic x-axis.

biased towards single ASes: The majority of TGAs contain 50–95 % addresses from a single AS, whereas the top ten ASes of the seed set cover only around 50 % of their addresses. The most popular AS for all but one candidate sets is AS12322 (FREE SAS, an ISP network from France). The only exception to this bias is the candidate set of 6VecLM, which contains only five addresses from AS12322 and is even more evenly distributed among ASes than the seed data set. While AS12322 is also the AS with the highest share in the seed data set, it covers only 30 % of it. Looking at the structure of the addresses from this AS responding to ICMP, it is visible that over 99 % of them have the host part set to `::1`. They are all within the same /39 prefix and only differ in the 10 to 15 nibble of the address. This is a very clear structure which has easy to detect patterns, ideal for discovery by TGAs. Addresses from this AS were first added to the hitlist via CT logs and the Bitnodes data set [4] and their share drastically increased with the first usage of TGAs in the 2022 paper [5]. The high percentage of addresses from AS12322 in most candidate sets also explains their bias towards the ISP network category. This further stresses the need to filter certain categories for use cases where addresses from the respective categories should not be targeted. Furthermore, addresses from specific ASes should also be filtered, as their inclusion in the seed set can introduce biases towards those networks far beyond their presence in the seed set.

Number of covered ASes. Next to the distribution of the ASes contained in the candidate sets, their absolute number is equally relevant. TGAs should generate new targets which represent the active part of the IPv6 Internet, which cannot be achieved if only a few ASes is covered in their candidate sets. Most candidate sets cover substantially fewer total ASes than the respective seed sets, especially when using the full hitlist as input. Only in very specific circumstances, when the seed set already contains very few ASes (such as for Educational or NSP), some candidate sets cover more ASes. Very low coverage rates compared to the seed set means that algorithms discover ASes from very specific origins which cannot represent the IPv6 Internet. Even when combined, the candidate sets of all algorithms only cover 75 % of the ASes in the seed set. While the combined candidate sets include 684 ASes which have not been covered by the seed set, 4875 ASes from the seed set are not included. The exact number of newly covered ASes can be seen in Table VII.

Ratio of aliased prefix. Aliased prefixes, as defined in Section II, are excluded in our scans as they do not add any valuable information, but instead introduce a bias to the results. It is therefore an important measure of quality for a candidate set to contain few addresses from aliased prefixes. As described in Section IV-A, we conducted APD ourselves and with the aliased prefixes published by the *IPv6 Hitlist* service. We compared the unfiltered sets with non-aliased versions and found that most algorithms have a negligible rate of addresses from aliased prefixes, thereby not impacting the algorithms' candidate set quality when filtered. Two exceptions are 6GCVAE and Entropy/IP, which generate up to almost 50 % aliased addresses for some categorized input as well as the full input. This decreases the usable size of their candidate sets substantially, which should be kept in mind before scanning. The exact rate of aliased prefixes can be found in Appendix Table VII. Although the rate of aliased prefixes in most candidate sets were relatively low, which means that only little unnecessary scanning overhead would be introduced, we still stress the need for APD.

Protocol responses. Depending on the use case for the generated addresses, it can be crucial to discover targets with a high response rate to a certain protocol. Figure 8 shows the response rate to the different protocols per candidate set. All candidate sets have the highest response rate to ICMP probes, which is to be expected due to the prevalence in the seed set. Moreover, unlike in IPv4, ICMP in IPv6 can not simply be fully blocked due to its important functionality in stateless address autoconfiguration [48]. Responses to other protocols are much less frequent for all candidate sets. Especially the response rate for HTTP and HTTPS is very similar to the share of non-ISP addresses in the responsive portion of the candidate

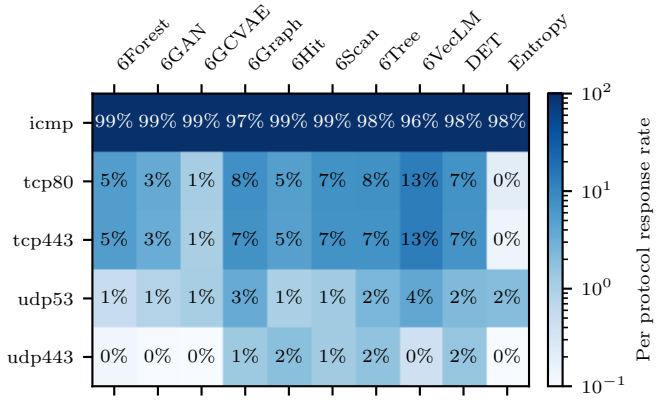

Figure 8: Response rates to the different protocols per algorithm generated on full hitlist input in 2023. Note the color map log scale.

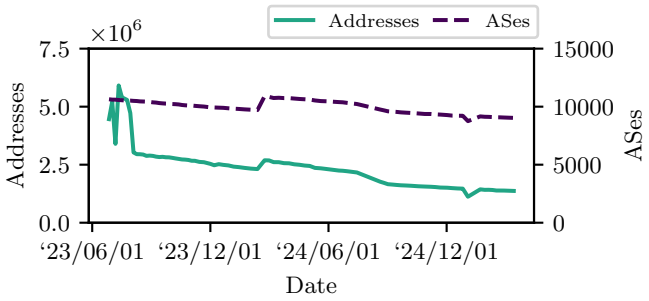

Figure 9: Responsiveness of addresses added to the *IPv6 Hitlist* service over time. In total 9.1 M new addresses were added on June 27, 2023

sets. Looking at Figure 6, we can see that the responses to the candidate set of 6VecLM have the lowest share of ISP addresses and the highest number of responses to HTTP and HTTPS. Entropy/IP and 6GCVAE on the other hand, have more than 95 % ISP addresses in their respective responses and the lowest share of protocol responses other than ICMP. The per-protocol response rate for the candidate sets generated with categorized inputs show a very strong correlation with the per-category response rates of the hitlist, see Figure 3. With CDN input addresses, all candidate sets receive between 30 and 65 % HTTP and HTTPS responses, whereas with ISP input, no candidate set generates more than 3 % response rate for any protocol besides ICMP.

Temporal Stability of Responsive Candidates. We added generated addresses to the *IPv6 Hitlist* service on June 27, 2023 with the publication of the initial paper [10]. We filtered addresses from AS12322 as to not introduce a bias to the service and short-lived addresses. In total, 9.1 M addresses from 10.6 k ASes were added. This allows us to analyze the development of those addresses over time until March 2025 as shown in Figure 9. The observed period covers 82 iterations of scanning conducted by the *IPv6 Hitlist* service.

Initially, 5.9 M addresses were responsive to at least one protocol as part of the service. However, 2.9 M addresses were only responsive during the initial two months. The majority of these addresses were part of AS29432 (TREX). Only two addresses originated by this AS remained responsive afterwards. The remaining 3.0 M million addresses remain relatively stable with a constant decline over time. At the end of our observation period, 1.4 M addresses are still responsive. The number of ASes originating these addresses decreases from 10.6 k to 9.0 k. Thus, while the overall number of addresses decreases over time and new generations should be triggered, the overall coverage of ASes remains good.

There is fluctuation over time. Throughout the 20-month period, 7.5 M out of the added 9.1 M addresses were responsive at least once. 530.0 k addresses are responsive in all scans. The average responsiveness per IP address is 27 scans. However, removing the outlier AS29432, increases the average to 52 scans. In contrast to the general stability per category as seen in Figure 4, there is no correlation of the stability of generated addresses with their respective category based on PeeringDB.

Key Takeaway: Choice of algorithm and seed dataset influences many properties of the generated candidate addresses and their responsiveness. This includes the total amount of generated addresses, the total amount and percentage of responsive addresses, the distribution of autonomous systems and aliased prefixes represented in the addresses and the response rate to specific protocol probes. If scanning budget is limited, certain algorithms or inputs should be avoided; the output can alternatively be sampled. The choice is different *e.g.*, if the total amount of responses is of higher priority. Addresses from certain categories and especially ASes contain patterns which are easier to recognize than others, leading to a higher representation of these categories and ASes in the generated addresses. Seed datasets need to be carefully filtered to avoid this bias. Independently of the seed data, TGAs struggle to achieve a high coverage of ASes, which is why additional sources should be considered if a high AS coverage is desired. This also raises the questions if current TGAs adequately address the need for a balanced candidate set across many ASes. Lastly, the inclusion of TGA-generated addresses into long-term measurement campaigns is proven feasible, as a considerable amount of addresses from almost 85 % of ASes were still responsive at the end of our observation period regardless of their category.

B. Extended TGA Analysis

Based on the insight of our first experiments, the used seed sets and historical development, we improved our evaluation of TGAs and focused on the detailed behavior of TGAs in different scenarios. Therefore, we ran a second experiment with an adapted list of TGAs, further seed sets and a focus on additional metrics.

We create 17 seed sets based on the IPv6 Hitlist and previous insights. Table II highlights the properties and diversity of these data sets. The number of IP addresses is highest for the *ICMP*

data set, as most responsive targets from the full data set respond to ICMP. The number of IP addresses is relatively low for *AS-sampled* data sets, indicating that the full data set is not uniformly distributed. This is further observable when looking at the numbers of ASes and prefixes included in the *randomly sampled* data sets. AS and prefix diversity show a strong decrease with all seed selection mechanisms except for AS-based sampling. There, $100\,\%$ of the AS coverage and almost $50\,\%$ of prefix coverage can be achieved with $7\,\%$ and $0.5\,\%$ of IP addresses for AS100 and AS10 respectively. Lastly, the most frequent PeeringDB category and the respective percentage of IP addresses can be observed. While the full data set (and therefore the randomly sampled data sets) features CDN addresses most prominently and the categorized data sets feature their respective category, other data sets differ. When sampling based on ASes, first ISP networks, then Other network categories become most frequent. This stems from the fact that while most IP addresses in the full data set come from CDN and ISP networks, the distribution of ASes is different. Most ($11\,\mathrm{k}$) ASes fall into the categories which we merge into the category "Others", while ISP ($5.7\,\mathrm{k}$), NSP ($2.3\,\mathrm{k}$) and CDN ($1.2\,\mathrm{k}$) are less prominent. Therefore, when sampling based on ASes, the distribution of addresses into the categories gets closer to the distribution of ASes. For the protocol-filtered data sets, CDN addresses are most prominent in all except for UDP/53. While this is expected for TCP/80, TCP/443 and UDP/443, which usually serve CDN-common protocols and ICMP responsiveness is shared by all categories, UDP/53 usually serves DNS and is dominated by networks from smaller categories.

Runtimes. Table IV shows the runtimes of the different algorithms, depending on the seed data set. 6Forest had to be interrupted manually for some seed data sets, indicated by *DNF*. 6Forest, 6Graph and 6Sense did not produce any candidates for some seed data sets, signalled by *ERR*. In the case of 6Forest and 6Graph, no error was reported, while we could link the errors of 6Sense to the insufficient VRAM size of our GPUs. While the latter is a limitation caused by hardware, we try to enable comparability by treating the algorithms as a black box and following the instructions and suggestions by the authors, thereby demonstrating possible limitations to other researchers.

The runtime of the algorithms mainly depends on the input size. All but three algorithms take longest for the full input data set due to it being the largest data set. One exception is 6Scan, which is the only algorithm to strictly adhere to the set budget of $10\,\mathrm{M}$. The generation time seems to be mainly bound by the online scanning, which depends only on the stable number of candidates. As 6Sense could not be run on the full data set, only the runtime of the successful runs depends on the size of the seed data. Lastly, 6Forest finishes generation in so little time that a correlation with the size of the seed data is not observable.

Candidate sets. Table V shows the size of the candidate sets generated by the different algorithms depending on the different input data sets. The sizes vary greatly depending on the algorithm as well as on the input. Algorithms such as 6Graph and 6Tree can generate more than 2 billion and $500\,\mathrm{M}$ candidates respectively, while other candidate sets contain only several thousand addresses. The size of the candidate set depends mostly on the size of the seed data set. It can be observed that using the *Full* seed data set leads to the largest (or among-largest within a small margin of error) candidate data set for all algorithms. Other large data sets like *ICMP* or the sampled data sets of different sizes show a similar pattern among most algorithms. There are also notable exceptions to this rule, such as 6Forest generating only very few candidates from the *ISP*, *ICMP* or *DS100k* data sets or 6Tree and DET generating more candidates from the *AS10* data set than from *AS100*. Dynamic algorithms with a scanning budget option (which we set to $10\,\mathrm{M}$ if applicable) show that they use this budget independently of the seed data size. This applies to most dynamic algorithms; 6Scan and 6Sense use close to or more than $90\,\%$ of their budget and strictly adhere to the limit, 6Hit often exceeds the limit, and AddrMiner-S and DET do not always use all scanning budget. It should be noted here that AddrMiner-S and DET use ZMap for scanning and share large portions of their code base, as do 6Scan and 6Hit, which implement their own scanning mechanism.

Robustness. When using TGAs, it is important to have an estimate of how consistent the generation of candidates are between multiple generation runs. We chose the term *robustness* for this metric and ran the algorithms twice under the same conditions to measure it. We used the exact same seed data sets and parameters for both rounds, the only difference between the runs lies in the two-week time gap, which influences the results of the online scanning mechanisms of the dynamic TGAs. We compare both candidate sets by computing the overlap between the generated IP addresses and the corresponding ASes. Table VI shows the relative overlap per algorithm and seed data set, where the number of overlapping elements is compared to the total number of elements in both sets. For algorithms such as AddrMiner-S, DET, 6Sense and eip-generator, the overlap in IP addresses is low, while the overlap in ASes is high. This indicates that the algorithms follow similar strategies between runs on an AS-level, but introduce more variability in the lower address bytes. For 6Sense this difference coincides with the algorithm function, as the upper and lower 64 bits of the address are generated independently. Algorithms such as 6Tree, 6Scan and 6Hit on the other hand show strong overlap in both IP addresses and ASes. As they are dynamic algorithms, which adapt their scanning strategies to the results of their scans, this indicates that they either mostly probe targets which are stable for at least two weeks or do not strongly adjust their scanning to unresponsive addresses. Lastly, 6Forest and 6Graph show strongly varying output in both IP addresses and ASes. Even though both runs were executed in the same environment, using the same configuration and same input, the algorithms do not deterministically terminate and produce results. Table VI further shows that published algorithms are not necessarily stable. We argue that this further demonstrates the need to select TGAs based on different criteria such as robustness. We suggest that future research on TGAs considers these insights and evaluates new algorithms in respect to their usability and robustness.

Table IV: Runtime of the different algorithms depending on input data set in 2025. *ERR* indicates that the algorithm did not produce any candidates, while *DNF* indicates that the algorithm had to be manually stopped, while still producing results. A log-scale color map is applied for better readability.

	6Forest	6Graph	6Hit	6Scan	6Sense	6Tree	AddrMiner-S	DET	eip-gen
Full	00:00:18	05:15:49	28:40:58	02:16:46	ERR	41:35:36	10:54:23	10:37:15	00:59:51
CDN	00:00:19	01:37:30	19:37:54	02:11:40	06:29:14	05:45:59	05:30:34	02:23:33	00:12:08
EDU	00:00:04	ERR	05:24:51	02:14:27	01:39:38	01:55:49	00:30:42	01:04:37	00:00:47
ISP	00:00:18	01:15:14	05:20:38	02:11:40	ERR	08:39:04	01:04:22	02:41:48	00:10:47
NSP	DNF	00:03:14	02:16:38	02:15:18	09:21:58	16:42:08	01:26:03	03:57:56	00:20:58
ORG	ERR	00:00:18	03:55:16	02:15:06	01:33:39	01:53:31	00:03:45	02:37:38	00:06:49
DS10k	ERR	ERR	04:36:41	02:14:30	01:36:23	02:18:14	03:27:21	01:01:47	00:03:49
DS100k	00:00:13	00:00:42	02:07:36	02:14:38	1:50:00	04:15:54	03:29:41	00:57:27	00:07:34
DS1M	00:00:05	00:07:27	02:01:12	02:14:38	03:42:05	01:13:39	03:26:15	00:55:11	00:08:39
AS10	ERR	00:04:25	01:55:18	02:14:43	01:46:42	01:45:39	00:04:10	00:53:19	00:02:14
AS100	00:00:12	00:07:20	02:15:25	02:14:30	02:14:16	02:34:59	00:07:58	00:54:00	00:03:33
AS1k	ERR	00:18:30	02:10:57	02:14:02	03:23:17	01:42:13	00:21:17	01:03:20	00:03:13
ICMP	00:00:14	ERR	27:50:06	02:13:58	ERR	19:05:54	08:20:20	05:44:48	00:27:31
TCP80	00:00:12	00:02:20	02:05:23	02:14:17	03:50:47	01:51:09	01:35:09	01:20:57	00:03:17
TCP443	DNF	00:01:46	02:04:56	02:14:05	03:40:05	01:15:50	01:36:34	01:13:30	00:03:06
UDP53	00:00:10	ERR	02:07:06	02:14:18	01:57:59	01:44:05	01:55:42	01:09:57	00:01:37
UDP443	DNF	00:03:13	02:09:08	02:14:09	01:43:08	02:42:54	04:02:30	01:19:49	00:04:46

Table V: Size of candidate sets and the responsive subsets in 2025. Asterisks denote that the generation failed. For 6Sense, this was due to insufficient VRAM size, while 6Forest and 6Graph were unable to run on the respective input. The largest candidate set per algorithm is highlighted with bold script. A log-scale color map is applied for better readability.

	6Forest		6Graph		6Hit		6Scan		6Sense		6Tree		AddrMiner-S		DET		eip-gen	
	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.	cand.	resp.
Full	2M	111k	**2B**	39M	**128M**	14M	10M	4M	0*	0*	**547M**	30M	10M	2M	**10M**	617k	**7M**	13k
CDN	20k	2k	404M	18M	88M	591k	10M	972k	10M	4M	67M	3M	9M	247k	10M	390k	2M	10k
EDU	223k	13k	0*	0*	25M	731k	10M	118k	10M	3M	11M	181k	836k	52k	2M	1M	2M	10k
ISP	3k	913	470M	22M	24M	11M	10M	7M	0*	0*	113M	18M	232k	101k	10M	2M	2M	3k
NSP	**2M**	248k	2M	106k	10M	2M	10M	3M	10M	3M	225M	7M	572k	121k	10M	341k	5M	293
ORG	0*	0*	313k	32k	18M	28k	**10M**	29k	10M	3M	11M	37k	129k	8k	5M	4k	4M	10k
DS10k	0*	0*	0*	0*	21M	2M	10M	4M	**10M**	4M	9M	2M	10M	2M	8M	513k	6M	4k
DS100k	2M	248k	5M	522k	10M	3M	10M	5M	10M	3M	6M	1M	**10M**	1M	6M	553k	7M	14k
DS1M	152k	10k	44M	4M	10M	3M	10M	4M	10M	4M	16M	2M	10M	3M	10M	1M	7M	14k
AS10	0*	0*	2M	118k	10M	646k	10M	321k	10M	3M	11M	2M	193k	7k	6M	394k	6M	3k
AS100	2M	248k	44M	3M	11M	950k	10M	1M	10M	3M	10M	2M	282k	10k	2M	484k	7M	2k
AS1k	0*	0*	38M	3M	10M	1M	10M	2M	10M	3M	22M	3M	675k	26k	10M	2M	5M	2k
ICMP	83k	6k	0*	0*	124M	14M	10M	4M	0*	0*	241M	29M	10M	2M	10M	1M	4M	725
TCP80	2M	248k	1M	102k	10M	1M	10M	2M	10M	3M	24M	3M	2M	199k	10M	1M	2M	6k
TCP443	2M	248k	1M	96k	10M	1M	10M	2M	10M	3M	16M	3M	2M	228k	10M	2M	2M	9k
UDP53	2M	220k	0*	0*	11M	1M	10M	1M	10M	3M	12M	661k	3M	180k	386k	17k	4M	9k
UDP443	2M	248k	1M	47k	11M	2M	10M	918k	10M	3M	8M	594k	6M	238k	5M	138k	6M	25k

Table VI: Overlap of candidate sets between first and second generation run from our 2025 evaluation. One asterisk indicates that neither generation run yielded any results, two asterisks indicate that the first run failed, three asterisks indicate that the second run failed.

	6Forest		6Graph		6Hit		6Scan		6Sense		6Tree		AddrMiner-S		DET		eip-gen	
	IP	AS	IP	AS	IP	AS	IP	AS	IP	AS	IP	AS	IP	AS	IP	AS	IP	AS
Full	100%	100%	100%	100%	100%	100%	100%	100%	*	*	100%	100%	14%	92%	1%	72%	0%	74%
CDN	0%	0%	44%	48%	100%	100%	100%	100%	14%	66%	100%	100%	17%	95%	8%	94%	0%	60%
EDU	16%	1%	**	**	100%	100%	91%	97%	44%	86%	46%	99%	37%	99%	0%	34%	1%	62%
ISP	0%	0%	100%	100%	100%	100%	100%	100%	*	*	100%	100%	18%	99%	5%	83%	0%	85%
NSP	0%	10%	0%	13%	100%	100%	91%	100%	23%	86%	100%	100%	27%	98%	2%	86%	0%	55%
ORG	**	**	100%	100%	100%	99%	96%	99%	50%	93%	99%	99%	23%	98%	7%	30%	4%	82%
DS10k	**	**	**	**	100%	100%	100%	96%	48%	98%	96%	100%	23%	68%	39%	91%	0%	83%
DS100k	1%	23%	100%	100%	88%	99%	98%	99%	24%	98%	38%	100%	20%	60%	31%	77%	0%	66%
DS1M	0%	0%	41%	47%	98%	100%	99%	100%	27%	99%	100%	100%	22%	93%	47%	99%	0%	80%
AS10	**	**	27%	63%	98%	100%	98%	100%	28%	100%	83%	100%	11%	84%	55%	93%	0%	86%
AS100	99%	99%	50%	97%	99%	100%	98%	100%	27%	100%	91%	100%	11%	62%	36%	95%	0%	85%
AS1k	**	**	59%	65%	100%	100%	92%	98%	33%	100%	100%	100%	19%	90%	30%	94%	0%	85%
ICMP	0%	4%	*	*	100%	100%	100%	100%	*	*	100%	100%	17%	84%	2%	78%	0%	76%
TCP80	***	***	***	***	99%	100%	88%	96%	25%	99%	100%	100%	27%	98%	26%	94%	0%	82%
TCP443	***	***	***	***	99%	100%	99%	100%	26%	99%	100%	100%	29%	94%	44%	98%	0%	56%
UDP53	0%	18%	*	*	98%	100%	98%	100%	24%	99%	45%	100%	38%	93%	0%	26%	0%	66%
UDP443	***	***	***	***	100%	100%	94%	98%	33%	99%	20%	100%	42%	98%	1%	89%	1%	75%

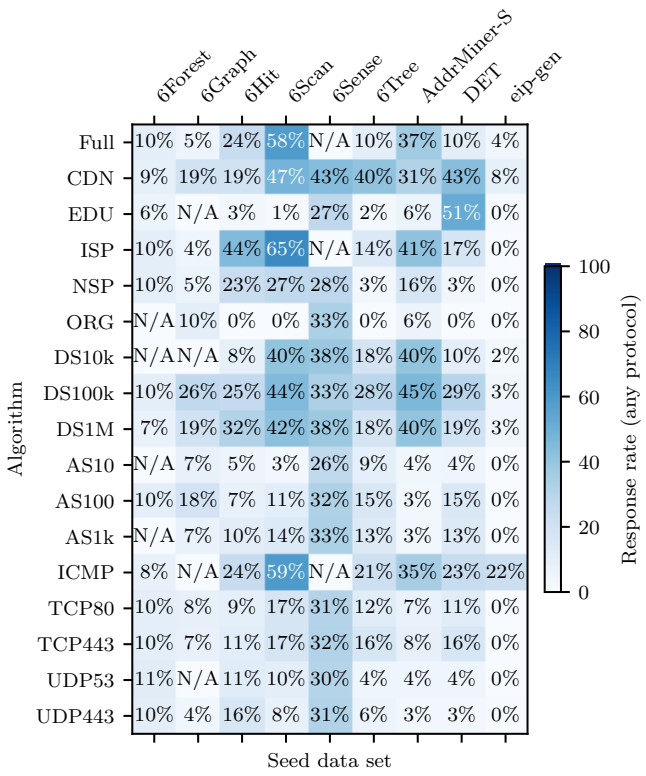

Figure 10: Relative hit rates per algorithm and input seed set from our 2025 evaluation.

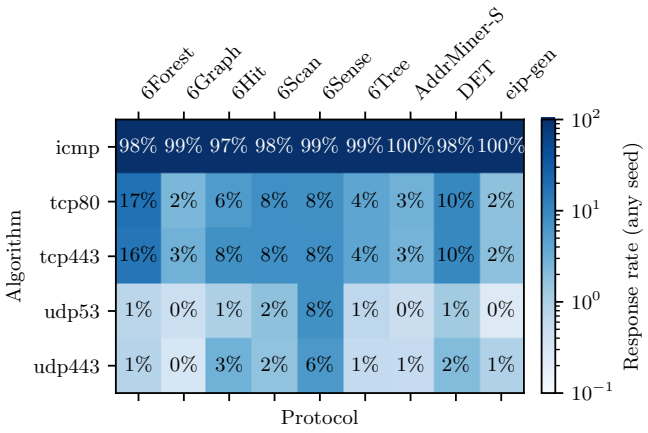

Figure 11: Protocol-dependent hit rate compared to the total number of responsive IP addresses from our 2025 evaluation.

Aliased prefix ratio. Before scanning the combined generated target data sets, we filter out the addresses from aliased prefixes, as described in Section IV. We calculated the number of addresses per candidate set that is filtered out by this step in order to quantify the importance of this step depending on algorithm and input. Most algorithms generate a negligible percentage of addresses from aliased prefixes, except 6Graph and 6Tree, which generate up to 75 % and 20 % addresses from aliased prefixes respectively when using the *Full* seed data set. As the candidate set of 6Graph includes more than two billion addresses, this demonstrates the effectiveness and importance of this filter step to reduce many unnecessary scanning probes. The ratio of addresses from aliased prefixes might however be even higher, as described in Section IV.

Scan results. Table V shows the number of responsive addresses depending on the algorithm and seed data set from a scan of the first target set. The total number of responses ranges from 39 M for 6Graph running on the full data set to below one thousand for responses from small data sets such as the set generated from *ISP* addresses by 6Forest. This means that, e.g., 6Graph, can generate a 90 % increase in address coverage from the full data set. The impact of sampling, both random and AS-based, is not consistent among algorithms in terms of the responsive generated addresses. While the total number of addresses can be an important metric for TGA performance when scanning costs are not the bottleneck, scenarios with lower scanning budgets require TGAs with a high *hit rate*. Figure 10 shows this hit rate, i.e., the ratio of responses (any protocol) to probed addresses. Hit rates also vary between algorithms as well as seed data sets, from close to 0 % to up to 65 %. Similarly to our first experiment, using the full data set as well as ISP addresses as input leads to high hit rates. In contrast, CDN addresses also lead to high hit rates for most algorithms. The impact of sampling is not consistent on all algorithms and can increase as well as decrease the hit rate depending on the algorithm. Only the *AS10* data set leads to consistently low hit rates for all algorithms, indicating that ten addresses per AS is too little data for the algorithms.

Protocol responses. Depending on the use case of the generated responsive addresses, the type of probe which the address responded to, can be important. Figure 11 shows the ratio of addresses responsive to a certain type of probe compared to all responsive addresses generated by an algorithm across all seed data sets. As observed in our previous experiments, more than 90 % of all addresses generated from all algorithms respond to ICMP probes. However, response rates to any other protocol vary between algorithms. All other algorithms share a second-highest response rate to TCP probes on port 443 and 80, the standard ports for HTTP/S, indicating a varying ability between algorithms in discovering web deployments. UDP port 443 and 53, mostly used for HTTP/3 or DNS deployments respectively, respond least frequently to our probes across all algorithms, matching the results from our previous experiments.

When considering the responsiveness of addresses generated from port-specific seed data sets, we observe similar results as Williams *et al.* [18]. For TCP/80 and TCP/443 and most algorithms, these addresses show an up to 36 times increase in responsiveness to the respective protocol probes compared to the combined results. For UDP/53 and UDP/443, the increase can reach a factor of several hundred and several thousands for eip-generator. The exception here is ICMP, which already has a very high response rate for the combined data. Training TGAs on addresses responsive on TCP/80, TCP/443 and UDP/443 yields higher responsiveness on the respective other protocols as well, indicating the ability to predict addresses hosting web services of different kinds. UDP/443 as seed data yields the highest responsiveness. Using UDP/53 as seed data leads to

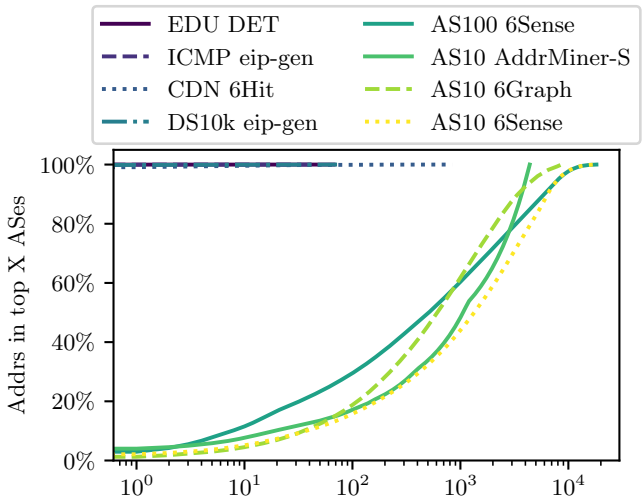

Figure 12: AS distribution of responsive candidates with highest and lowest entropy from our 2025 evaluation. The entropy is calculated as Shannon entropy over the distribution of ASes per set.

the highest increase in UDP/53 response rates by far, also increasing the response rate in TCP/80 and TCP/443, but not for UDP/443.

Autonomous System distribution. The distribution of ASes represented in the candidate sets, and especially the responsive subsets is crucial to measure in order to estimate the scan traffic needed to reach a certain AS coverage. AS coverage can be an important metric to optimize in scans, depending on the use case, while scan traffic should always be minimized. Figure 12 shows the sets of responsive candidates with the highest and lowest entropy to show the most and least evenly distributed sets in terms of represented ASes. The graph highlights the very strong difference in distribution depending on input and algorithm. The data sets with the lowest entropy all contain more than 95 % of addresses from a single AS. The seed data sets *EDU*, *CDN* and *DS10k* also contain less than 1 k ASes, which is only a small portion of the full data set, which limits the ASes TGAs can generate new addresses from. While the ICMP data set contain a larger number of ASes, two ASes appear to contain addresses patterns which are easy for TGAs to recognize, as they are the top two most common ASes in all candidate sets using this seed set. We argue that the most frequent AS, AS13335 (Cloudflare) is found most frequently due to the problems in aliased prefix detection (see Section IV), while the second most common AS, AS12332 (Free SAS), uses an address allocation strategy which is easy to understand for TGAs. Upon manual inspection, we find that the TGAs discover responsive, incremental addresses in AS12332, which are easy to predict. Using AS-sampled seeds as input consistently leads to the most evenly distributed responsive candidates across multiple algorithms. This indicates that an even distribution of ASes in the seed data set can lead to a more even distribution in the resulting responsive candidates.

Key Takeaway: The observations of the initial TGA experiment regarding influences of seed data sets and algorithms on different output properties remain true for the extended set of TGAs. Additionally, the size of the seed data should be taken in consideration when runtime of the algorithms is important. The seed data set and its AS distribution also influences the AS distribution of the responsive candidates. For an even distribution, AS-sampled seeds are recommended. Furthermore, algorithms vary in their ability to generate addresses responsive to protocol other than ICMP. Using a protocol specific seed data set leads to the highest responsiveness to the respective protocol probes. This further demonstrate the need to choose both algorithm and seed data set depending on the use case. Independently of the seed data, some algorithms produce considerably different output between two executions under identical conditions and configuration, which influences usability of the algorithms.

VII. Discussion and Conclusion

In this work, we have highlighted the dependency of IPv6 measurements on their targets. We have shown that address collections such as the *IPv6 Hitlist* service contain multiple types of networks, including ISPs, CDNs, and NSPs, with different behavior. While addresses from CDN networks respond to HTTP and HTTPS at a rate of around 50 % and are responsive for a median time span of more than 200 days, ISP addresses are mostly only available for a single scan and only respond to ICMP for 97 % of the addresses.

Furthermore, we evaluated the behavior of different Target Generation Algorithms, using the full and categorized versions of the hitlist as input. We demonstrated that the input has a strong influence on various metrics, such as the number of generated and responsive addresses, protocol responses, and addresses origin. All but one candidate sets generated from uncategorized input show a very strong bias towards ISP networks, which in turn have a strong bias towards single ASes and generally have a response rate below 10 % for any protocol other than ICMP. Output from categorized seed sets consists of addresses from the respective input category, exhibiting behavior similar to the addresses from the respective categories in the hitlist. However, we learned that most TGAs are complex tools and the majority of published tool chains are trained and optimized on a specific input. While we tried to adapt the algorithms and parameters to suit our use cases and scenarios, we were not able to reach published rates of responsive addresses. Furthermore, algorithms with dynamic scanning capabilities are not suited for all use cases, as the adherence to scanning rates, blocklists and detection of aliased prefixes cannot be achieved without modifications. Our work provides a detailed comparison under different circumstances to allow for a selection of suitable TGAs and a more focused analysis and optimization in the future. As an example, a scan application which requires large numbers of targets and does not have tight restraints on scan budgets, should opt for an algorithm such as 6Graph or 6Tree, as they generate the largest candidate sets. Scenarios, on the other hand, which

dictate efficient scanning with a high response rate and do not require modifications to the candidate set before scanning, are best suited for dynamic algorithms, as they reach the highest response rates.

Our extended analysis of further TGAs and seed data sets aims to catch up with the rapidly evolving research field of TGAs. We highlight the vast differences of algorithm and seed data set choice on the amount of generated candidates, their responsiveness on different protocol, their distribution over different ASes as well as the run time and differences in the results of multiple generation runs. This further complements the data based on which an informed choice for TGA and seed data set can be made.

Future IPv6 Internet measurements are encouraged to use our findings to increase the efficiency of their scans by removing unnecessary scanning overhead and generating targets better suited for their use case. Researchers conducting IPv6 measurements should keep in mind that the current hitlist shows a bias towards ISP addresses. These addresses are only short-lived and should therefore not be used for long-term studies. A proper selection of scan specific targets from the hitlist and a proper application of TGAs on specific seed sets can however improve future scans and reduce unnecessary probing.

REFERENCES

[1] Google, *Google IPv6 Statistics*, https://www.google.com/intl/en/ipv6/statistics.html.

[2] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *USENIX Security Symposium*, 2013.

[3] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, "Zippier ZMap: Internet-Wide Scanning at 10 Gbps," in *WOOT*, 2014.

[4] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczyński, S. D. Strowes, L. Hendriks, and G. Carle, "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in *Proc. ACM Int. Measurement Conference (IMC)*, 2018. DOI: 10.1145/3278532.3278564.

[5] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle, "Rusty Clusters? Dusting an IPv6 Research Foundation," in *Proc. ACM Int. Measurement Conference (IMC)*, 2022. DOI: 10.1145/3517745.3561440.

[6] T. Cui, G. Gou, and G. Xiong, "6GCVAE: Gated Convolutional Variational Autoencoder for IPv6 Target Generation," in *Advances in Knowledge Discovery and Data Mining*, 2020. DOI: 10.1007/978-3-030-47426-3_47.

[7] T. Cui, G. Gou, G. Xiong, C. Liu, P. Fu, and Z. Li, "6GAN: IPv6 Multi-Pattern Target Generation via Generative Adversarial Nets with Reinforcement Learning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2021. DOI: 10.1109/infocom42981.2021.9488912.

[8] T. Yang, B. Hou, Z. Cai, K. Wu, T. Zhou, and C. Wang, "6Graph: A graph-theoretic approach to address pattern mining for Internet-wide IPv6 scanning," *Computer Networks*, vol. 203, Feb. 2022. DOI: 10.1016/j.comnet.2021.108666.

[9] P. Foremski, D. Plonka, and A. Berger, "Entropy/IP: Uncovering Structure in IPv6 Addresses," in *Proc. ACM Int. Measurement Conference (IMC)*, 2016. DOI: 10.1145/2987443.2987445.

[10] L. Steger, L. Kuang, J. Zirngibl, G. Carle, and O. Gasser, "Target acquired? evaluating target generation algorithms for ipv6," in *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA)*, 2023.

[11] L. Steger, L. Kuang, J. Zirngibl, G. Carle, and O. Gasser, *Data and analysis at tum university library*, Dataset, 2023. DOI: 10.14459/2023mp1709953.

[12] Z. Liu, Y. Xiong, X. Liu, W. Xie, and P. Zhu, "6Tree: Efficient dynamic discovery of active addresses in the IPv6 address space," *Computer Networks*, vol. 155, May 2019. DOI: 10.1016/j.comnet.2019.03.010.

[13] G. Song, J. Yang, Z. Wang, L. He, J. Lin, L. Pan, C. Duan, and X. Quan, "DET: Enabling Efficient Probing of IPv6 Active Addresses," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, Aug. 2022. DOI: 10.1109/tnet.2022.3145040.

[14] T. Cui, G. Xiong, G. Gou, J. Shi, and W. Xia, "6VecLM: Language Modeling in Vector Space for IPv6 Target Generation," in *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track*, 2021. DOI: 10.1007/978-3-030-67667-4_12.

[15] B. Hou, Z. Cai, K. Wu, J. Su, and Y. Xiong, "6Hit: A Reinforcement Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2021. DOI: 10.1109/infocom42981.2021.9488794.

[16] T. Yang, Z. Cai, B. Hou, and T. Zhou, "6Forest: An Ensemble Learning-based Approach to Target Generation for Internet-wide IPv6 Scanning," in *Proc. IEEE Int. Conference on Computer Communications (INFOCOM)*, 2022. DOI: 10.1109/infocom48880.2022.9796925.

[17] B. Hou, Z. Cai, K. Wu, T. Yang, and T. Zhou, "6Scan: A High-Efficiency Dynamic Internet-Wide IPv6 Scanner With Regional Encoding," *IEEE/ACM Transactions on Networking*, 2023. DOI: 10.1109/tnet.2023.3233953.

[18] G. Williams and P. Pearce, "Seeds of scanning: Exploring the effects of datasets, methods, and metrics on ipv6 internet scanning," in *Proceedings of the 2024 ACM on Internet Measurement Conference*, 2024. DOI: 10.1145/3646547.3688449.

[19] *AS Classification*, https://catalog.caida.org/dataset/as_classification.

[20] M. Ziv, L. Izhikevich, K. Ruth, K. Izhikevich, and Z. Durumeric, "ASdb," in *Proc. ACM Int. Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487853.

[21] F. Gont and T. Chown, *Network Reconnaissance in IPv6 Networks*, RFC 7707, Mar. 2016. DOI: 10.17487/RFC7707. [Online]. Available: https://www.rfc-editor.org/info/rfc7707.

[22] J. Ullrich, P. Kieseberg, K. Krombholz, and E. Weippl, "On Reconnaissance with IPv6: A Pattern-Based Scanning Approach," in *10th International Conference on Availability, Reliability and Security*, 2015. DOI: 10.1109/ares.2015.48.

[23] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, "Target generation for Internet-wide IPv6 scanning," in *Proc. ACM Int. Measurement Conference (IMC)*, 2017. DOI: 10.1145/3131365.3131405.

[24] G. Williams, M. Erdemir, A. Hsu, S. Bhat, A. Bhaskar, F. Li, and P. Pearce, "6sense: Internet-Wide IPv6 scanning and its security applications," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024.

[25] L. Jiao, Y. Zhu, W. Zhang, L. Zhao, Y. Zhou, and Q. Liu, "6gai: Active ipv6 address generation via adversarial training with leaked information," in *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2024. DOI: 10.1109/cscwd61410.2024.10580769.

[26] L. Zhang and S. Fu, "6mcblm: Multi-scale cnn and bilstm-attention hybrid model for ipv6 target generation," in *2024 4th International Conference on Neural Networks, Information and Communication (NNICE)*, 2024. DOI: 10.1109/nnice61279.2024.10498956.

[27] N. He, D. Li, and X. Huang, *6diffusion: Ipv6 target generation using a diffusion model with global-local attention mechanisms for internet-wide ipv6 scanning*, 2024. DOI: 10.48550/ARXIV.2412.19243.

[28] W. Zhang, G. Song, L. He, J. Lin, S. Wu, Z. Wang, C. Li, and J. Yang, "6vision: Image-encoding-based ipv6 target generation in few-seed scenarios," in *2024 IEEE 32nd International Conference on Network Protocols (ICNP)*, 2024. DOI: 10.1109/icnp61940.2024.10858550.

[29] X. Chen, W. Shi, J. Liu, M. Hou, and Y. Li, "6community: An active ipv6 address detection method based on community discovery algorithm," in *Proceedings of the 2023 2nd International Conference on Algorithms, Data Mining, and Information Technology*, 2023. DOI: 10.1145/3625403.3625426.

[30] C. Liu, R. Li, F. Yuan, S. Ding, Y. Liu, and X. Luo, "6subpattern: Target generation based on subpattern analysis for internet-wide ipv6 scanning," *IEEE Transactions on Network and Service Management*, vol. 21, no. 4, Aug. 2024. DOI: 10.1109/tnsm.2024.3400864.

[31] T. Yang, B. Hou, Y. Yang, and Z. Cai, "Sweeping the ipv6 internet: High-efficiency router interface discovery with weighted sampling," *IEEE/ACM Transactions on Networking*, 2024. DOI: 10.1109/tnet.2024.3479420.

[32] T. Yang and Z. Cai, "Efficient ipv6 router interface discovery," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, 2024. DOI: 10.1109/infocom52122.2024.10621168.

[33] N. Liu, C. Jia, B. Hou, C. Hou, and Z. Cai, "6search: A reinforcement learning-based traceroute approach for efficient IPv6 topology

discovery," *Computer Networks*, Aug. 2023. DOI: 10.1016/j.comnet.2023.109987.

[34] Q. Liu and X. Li, "6former: Transformer-based ipv6 address generation," in *2023 IEEE Symposium on Computers and Communications (ISCC)*, 2023. DOI: 10.1109/iscc58397.2023.10218311.

[35] E. C. Rye and R. Beverly, "Discovering the IPv6 Network Periphery," in *Passive and Active Measurement*, 2020. DOI: 10.1007/978-3-030-44081-7_1.

[36] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, "Fast IPv6 Network Periphery Discovery and Security Implications," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2021. DOI: 10.1109/dsn48987.2021.00025.

[37] R. Beverly, R. Durairajan, D. Plonka, and J. P. Rohrer, "In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery," in *Proc. ACM Int. Measurement Conference (IMC)*, 2018. DOI: 10.1145/3278532.3278559.

[38] M. Erdemir, F. Li, and P. Pearce, "Understanding ipv6 aliases and detection methods," in *Passive and Active Measurement*. Springer Nature Switzerland, 2025, ISBN: 9783031859601. DOI: 10.1007/978-3-031-85960-1_3.

[39] N. Hoang, A. Niaki, J. Dalek, J. Knockel, P. Lin, B. Marczak, M. Crete-Nishihata, P. Gill, and M. Polychronakis, "How Great is the Great Firewall? Measuring China's DNS Censorship," in *30th USENIX Security Symposium*, 2021.

[40] Anonymous, A. A. Niaki, N. P. Hoang, P. Gill, and A. Houmansadr, "Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior," in *10th USENIX Workshop on Free and Open Communications on the Internet (FOCI 20)*, 2020.

[41] D. Dittrich, E. Kenneally, *et al.*, "The Menlo Report: Ethical principles guiding information and communication technology research," *US Department of Homeland Security*, 2012.

[42] C. Partridge and M. Allman, "Addressing Ethical Considerations in Network Measurement Papers," *Communications of the ACM*, vol. 59, no. 10, Oct. 2016.

[43] "University of Oregon Route Views Project." (), [Online]. Available: http://www.routeviews.org/routeviews/.

[44] *PeeringDB*, https://catalog.caida.org/dataset/peeringdb.

[45] J. Zirngibl, P. Buschmann, P. Sattler, B. Jaeger, J. Aulbach, and G. Carle, "It's over 9000: analyzing early QUIC deployments with the standardization on the horizon," in *Proc. ACM Int. Measurement Conference (IMC)*, 2021. DOI: 10.1145/3487552.3487826.

[46] E. Rye, R. Beverly, and K. C. Claffy, "Follow the Scent: Defeating IPv6 Prefix Rotation Privacy," in *Proceedings of the 21st ACM Internet Measurement Conference*, 2021. DOI: 10.1145/3487552.3487829.

[47] S. J. Saidi, O. Gasser, and G. Smaragdakis, "One Bad Apple Can Spoil Your IPv6 Privacy," *ACM SIGCOMM Computer Communication Review*, vol. 52, 2 Jun. 2022. DOI: 10.1145/3544912.3544915.

[48] S. Thomson, T. Narten, and T. Jinmei, *IPv6 Stateless Address Autoconfiguration*, RFC 4862 (Draft Standard), RFC, RFC Editor, Sep. 2007. DOI: 10.17487/RFC4862. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4862.txt.

Lion Steger is pursuing a PhD degree at the Technical University of Munich, Germany, at the Chair of Network Architectures and Services, advised by Prof. Dr.-Ing. Georg Carle. His research focuses on large scale Internet measurements with a special interest in IPv6, target generation algorithms, topology, privacy and security. He received his M.Sc. and B.Sc. in Informatics from TUM in 2022 and 2020, respectively, with focus topics ranging from IPv6 Internet measurements to censorship and honeypots.

Liming Kuang is currently a Ph.D. student at the Professorship of Photogrammetry and Remote Sensing at Technical University of Munich (TUM), in collaboration with Huawei Noah's Ark Lab London. His research centers on active 3D reconstruction, world action models, and vision-language models/agents (VLM/VLA). He is advised by Prof. Dr. Benjamin Busam. Before that, he received a M.Sc. in Informatics at TUM's Chair for Computer Aided Medical Procedures (CAMP), supervised by Prof. Dr. Nassir Navab. He received his B.Sc. in Informatics from TUM, supervised by Prof. Dr.-Ing. Georg Carle.

Johannes Zirngibl received the Master of Science degree in computer science (2019) and the Ph.D. degree in computer science (2025) from the Technical University of Munich. He is a postdoctoral research at the Max Planck Institute for Informatics and focuses on understanding the Internet as a large-scale, evolving system through empirical measurement and data-driven analysis. He conducts research on the evaluation of the behavior of new protocols, e.g., QUIC, due to their increasing complexity, and their behavior within the network in addition to existing protocols.

Georg Carle holds the chair on Network Architectures and Services at Technical University of Munich (TUM). He studied electrical engineering at University of Stuttgart, with studies abroad at Brunel University, London, and ENST Paris (now Telecom ParisTech). He received his Ph.D. at University of Karlsruhe (now KIT), and was scientist at Institut Eurecom, Sophia Antipolis, France, at the Fraunhofer Institute for Open Communication Systems FOKUS, Berlin, and professor at University of Tübingen, Germany. He conducts research on networked systems and security.

Oliver Gasser is Head of Research at IPinfo, where he leads strategic research initiatives, fosters collaboration with academic and industry partners, and drives innovation in data and product development. He also actively engages with the network operator community to bridge research and operational practice. He holds a Ph.D. from the Technical University of Munich, where his work focused on network security through large-scale Internet measurements, with emphasis on IPv6, TLS, and network attacks. Following his doctoral studies, he joined the

Max Planck Institute for Informatics, where he led the Internet Security Measurements research group. In 2024, Oliver joined IPinfo and established its research group from the ground up. Under his leadership, the team advances research in the fields of Internet measurement, IP geolocation, and protocol deployment.

Table VII: Appendix: Overview of different metrics for the algorithm per categorized input.

		6Forest	6GAN	6GCVAE	6Graph	6Hit	6Scan	6Tree	6VecLM	DET	Entropy
Number of candidate addresses	Content	1.94 M	486.59 k	3.28 M	34.85 M	10.02 M	8.94 M	11.08 M	77.82 k	8.56 M	5.87 M
	NSP	2.09 M	521.17 k	2.60 M	30.95 M	9.71 M	9.48 M	9.16 M	66.36 k	1.63 M	5.52 M
	Educational	1.44 M	316.24 k	699.80 k	1.98 M	24.02 M	9.88 M	11.05 M	83.54 k	1.06 M	4.48 M
	Non-Profit	710.61 k	125.30 k	284.16 k	295.57 k	19.57 M	9.97 M	7.89 M	0	5.87 M	3.79 M
	ISP	3.33 M	409.52 k	845.24 k	24.57 M	8.37 M	7.91 M	11.17 M	18.24 k	7.79 M	5.98 M
	Full	1.84 M	486.22 k	2.00 M	106.12 M	17.92 M	5.87 M	35.39 M	48.55 k	8.30 M	5.63 M
Generation factor	Content	1.11	0.28	1.88	19.97	5.74	5.13	6.35	0.04	4.90	3.36
	NSP	2.18	0.54	2.72	32.29	10.13	9.89	9.55	0.07	1.70	5.76
	Educational	11.48	2.53	5.60	15.82	192.22	79.09	88.38	0.67	8.48	35.83
	Non-Profit	22.01	3.88	8.80	9.15	605.98	308.69	244.39	0.00	181.82	117.42
	ISP	1.21	0.15	0.31	8.91	3.03	2.87	4.05	0.01	2.82	2.17
	Full	0.28	0.08	0.31	16.38	2.77	0.91	5.46	0.01	1.28	0.87
Number of responsive addresses	Content	173.90 k	13.42 k	13.97 k	443.44 k	230.62 k	491.22 k	416.75 k	3.87 k	360.54 k	7.62 k
	NSP	128.27 k	3.66 k	15.31 k	527.19 k	552.35 k	884.09 k	1.15 M	5.68 k	381.85 k	15.86 k
	Educational	19.37 k	2.62 k	585	22.04 k	99.82 k	37.73 k	106.52 k	1.23 k	745	2.59 k
	Non-Profit	38.91 k	8.50 k	2.61 k	15.12 k	2.86 M	946.16 k	1.58 M	0	355.65 k	13.66 k
	ISP	1.53 M	55.22 k	179.00 k	3.27 M	3.03 M	3.50 M	3.45 M	2.06 k	2.89 M	1.49 M
	Full	494.21 k	41.36 k	111.48 k	4.74 M	3.31 M	2.01 M	4.71 M	3.81 k	1.28 M	59.25 k
Relative response rate	Content	8.96 %	2.76 %	0.43 %	1.27 %	2.30 %	5.49 %	3.76 %	4.97 %	4.21 %	0.13 %
	NSP	6.15 %	0.70 %	0.59 %	1.70 %	5.69 %	9.33 %	12.54 %	8.56 %	23.46 %	0.29 %
	Educational	1.35 %	0.83 %	0.08 %	1.11 %	0.42 %	0.38 %	0.96 %	1.47 %	0.07 %	0.06 %
	Non-Profit	5.48 %	6.79 %	0.92 %	5.12 %	14.64 %	9.49 %	19.99 %	0 %	6.06 %	0.36 %
	ISP	45.88 %	13.48 %	21.18 %	13.29 %	36.21 %	44.28 %	30.89 %	11.29 %	37.07 %	24.97 %
	Full	26.85 %	8.51 %	5.58 %	4.46 %	18.46 %	34.31 %	13.32 %	7.84 %	15.46 %	1.05 %
Aliased prefix ratio	Content	1.75 %	6.14 %	35.14 %	0.21 %	0.38 %	0.15 %	0.12 %	0.37 %	0.31 %	40.94 %
	NSP	1.29 %	11.18 %	34.53 %	0.14 %	0.21 %	0.18 %	1.43 %	0.11 %	1.84 %	44.21 %
	Educational	0.66 %	14.92 %	17.81 %	0.73 %	4.26 %	0.44 %	0.07 %	0.08 %	0.63 %	50.06 %
	Non-Profit	1.78 %	2.22 %	13.89 %	2.53 %	0.02 %	0.13 %	21.02 %	0 %	11.39 %	41.67 %
	ISP	0.68 %	4.12 %	31.47 %	0.28 %	0.17 %	0.09 %	0.04 %	2.82 %	0.63 %	22.13 %
	Full	2.26 %	12.77 %	42.98 %	0.30 %	0.28 %	0.21 %	0.08 %	0.71 %	0.99 %	43.30 %
Candidate ASes	Content	1.03 k	1.22 k	5.14 k	4.03 k	969	884	1.08 k	675	2.64 k	3.13 k
	NSP	2.59 k	2.98 k	5.21 k	7.16 k	1.87 k	1.74 k	2.00 k	1.34 k	1.31 k	6.32 k
	Educational	819	1.33 k	980	2.36 k	493	473	627	572	469	2.50 k
	Non-Profit	430	377	215	1.50 k	263	264	323	0	147	3.29 k
	ISP	2.10 k	1.77 k	3.86 k	10.08 k	3.26 k	2.82 k	4.25 k	1.47 k	10.02 k	6.06 k
	Full	4.39 k	4.04 k	6.19 k	20.99 k	10.22 k	10.43 k	16.82 k	3.87 k	19.65 k	7.64 k
Responsive ASes	Content	186	121	1.24 k	1.04 k	668	654	803	207	1.05 k	62
	NSP	466	97	1.23 k	2.06 k	1.40 k	1.41 k	1.59 k	557	357	292
	Educational	230	101	250	538	334	354	429	145	57	212
	Non-Profit	205	62	60	311	179	194	222	0	44	576
	ISP	240	169	881	3.65 k	2.13 k	1.76 k	3.15 k	322	3.82 k	220
	Full	618	336	814	10.97 k	6.52 k	5.52 k	10.94 k	844	7.16 k	252
Coverage of seed ASes	Content	17.77 %	11.56 %	118.43 %	99.14 %	63.80 %	62.46 %	76.70 %	19.77 %	100.76 %	5.92 %
	NSP	23.57 %	4.91 %	62.11 %	104.05 %	70.71 %	71.52 %	80.58 %	28.17 %	18.06 %	14.77 %
	Educational	38.40 %	16.86 %	41.74 %	89.82 %	55.76 %	59.10 %	71.62 %	24.21 %	9.52 %	35.39 %
	Non-Profit	65.92 %	19.94 %	19.29 %	100.00 %	57.56 %	62.38 %	71.38 %	0 %	14.15 %	185.21 %
	ISP	5.66 %	3.99 %	20.79 %	86.10 %	50.31 %	41.58 %	74.28 %	7.60 %	90.09 %	5.19 %
	Full	3.63 %	1.97 %	4.78 %	64.48 %	38.34 %	32.42 %	64.31 %	4.96 %	42.10 %	1.48 %
Number of newly covered ASes	Content	27	8	1.14 k	268	20	14	23	0	296	44
	NSP	52	32	1.05 k	531	66	49	25	0	70	255
	Educational	32	15	231	128	12	24	15	0	10	191
	Non-Profit	65	7	57	109	15	18	10	0	13	555
	ISP	18	6	688	639	26	13	10	0	1.18 k	161
	Full	55	3	100	359	24	6	16	0	286	50