Decoding IPv6 Scan Traffic: Attraction, Analysis, Protection

Paper #324, 13 pages body, 17 pages total

ABSTRACT

The advent of IPv6 has considerably increased the complexity of detecting Internet scanning activity and analyzing scan traffic. This paper addresses the challenges of capturing and analyzing IPv6 scanning traffic by introducing novel techniques, vantage points, and datasets. Our long-term analysis of IPv6 scanning trends reveals a broad and substantial increase in scanning activity over the past two years. We introduce new tools and vantage points that leverage proactive techniques to attract IPv6 scan traffic. Using data collected with our new methods, enriched with crowdsourced IPv6 abuse reports, we provide a multi-perspective analysis of IPv6 scanning that yields new insights into today's IPv6 scanning landscape and highlights the limitations of existing network security tools in the IPv6 context. Finally, we offer actionable recommendations to improve the effectiveness of abuse reports and blocklisting methods for IPv6 networks.

ACM Reference Format:

Paper #324, 13 pages body, 17 pages total. 2024. Decoding IPv6 Scan Traffic: Attraction, Analysis, Protection. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Internet scanning is a vital tool for researchers and malicious actors alike: Researchers employ scanning to better understand network dynamics [11, 16, 29] while malicious actors utilize it to develop a network's threat surface. Capturing and analyzing this scanning traffic, in turn, allows network operators and researchers to study scanner behavior and intent, e.g., exploitation of specific vulnerabilities, and consequently to build effective defenses against potential malicious traffic.

However, the advent of IPv6 made the Internet scanning ecosystem vastly more complex: Brute-force scanning of the entire address space is no longer possible due to the prohibitively large IPv6 address space. IPv6 scanners now have to pick and choose their targets in order to increase the probability of finding active addresses in a sparse address space. The fact that scanners pick and choose specific targets significantly complicates capturing scanning traffic in IPv6 networks, which has traditionally been done in IPv4 by leveraging *darknets*; inactive regions of the address space which do not emit network traffic. While such regions are readily available in the IPv6 space, they are of little help to capture IPv6 scans, since scanners are not incentivized to target regions of the address space that emit little to no network activity (IPv6 *darknets*) and hence do not *attract* any IPv6 scanning activity.

As a result of these challenges we face a severe lack of visibility into potentially malicious scanning activity in the IPv6 space. Inadequate understanding of IPv6 scanning behaviors hinders our ability to develop efficient methods for securing and protecting IPv6 networks. This is now more important than ever; as recent reports suggest that the rise in IPv6 adoption (Google reports about ≈40% of its connections over IPv6 as of May 2024 [22]), has gone hand-in-hand with increased threats to IPv6 networks [3, 6]. However, implementations of tools/techniques that are effective in protecting IPv4 networks *en masse* (*e.g.,* IP blocklists, crowd-sourced abuse reports *etc.*) are complicated by varying address assignment and allocation practices in IPv6 networks. Taken together, we now face a situation in which many existing approaches to secure networks are of limited use in IPv6 networks, compounded by a lack of visibility into potentially malicious IPv6 traffic, which would in turn be needed to develop new defenses.

Towards tackling this situation, in our work, we introduce new techniques, vantage points, and datasets that enhance our ability to attract, capture, and analyze IPv6 scan traffic. We make four major contributions:

Trending analysis of IPv6 scanning: We provide a long-term analysis of trends in IPv6 scanning activity, as seen from the hosts of a major Content Distribution Network. We find that IPv6 scan traffic has increased by two orders of magnitude over the past two years (2022 – 2024). We also find today, scans are carried out by a continuously increasing number of sources when compared to the activity visible just two years ago. Our findings indicate a broader uptake in IPv6 scanning activity, and an ever more urgent need for

Conference'17, July 2017, Washington, DC, USA

© 2024 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

https://doi.org/10.1145/nnnnnnn.nnnnnnn

protective measures to monitor IPv6 scanning traffic and consequently secure IPv6 networks.

Tools and methods to capture IPv6 scan traffic: We introduce tools leveraging new ways to attract unsolicited, and potentially malicious, IPv6 scanning traffic. We deploy a set of geographically distributed vantage points to show that unused regions of the IPv6 address space receive significantly less scanning traffic than their active counter-parts. We leverage a /32 IPv6 address space—dedicated to a regional ISP—to run controlled experiments in order to emit network "liveness" and attract IPv6 scanners. Our controlled experiments leverage passive attraction methods *e.g.,* BGP announcements, IPv6 hitlist registration *etc.* and proactive attraction methods *e.g.,* deploying complex high-interaction honeypots specifically built to interact with IPv6 scanners.

Multi-perspective analysis of scan traffic: Leveraging the data gathered using our new IPv6 honeypot and honeynets, as well as crowdsourced data on potentially malicious IPv6 scanning behavior, we provide an in-depth overview of contemporary IPv6 scanning behavior and strategies. We find that some of our controlled experiments lead to an increase of 3 orders of magnitude of increased unsolicited network traffic to our previously un-probed honeyprefixes. We also find that all all of our controlled experiments led to increased scanning traffic; however, most scanning sources are only attracted by a small subset of our experiments.

Implications for protecting IPv6 networks: Leveraging our findings, we evaluate the efficacy of current network security tools in the context of IPv6 networks and offer actionable insights to help network operators and researchers enhance their IPv6 security measures and develop innovative approaches for IPv6 whitelisting or blocklisting. We find that network specific address allocation context is key in increasing the efficacy of crowd-sourced abuse reports and IP blocklisting methods. Without it 1) abuse reports and blocklisting methods only capture a small fragment of highly-distributed scanning entities and 2) proposed blocklisting techniques are susceptible to causing collateral by over-blocking chunks of commercial cloud provider's address space.

We believe that our findings have relevance both for the research as well as for the network operator community. The identified increase in scan traffic are a warning call to network operators who are in need to ramping up monitoring and defenses against potential IPv6 attacks. Our proposed techniques can support such approaches. The paper is structured as follows: We provide background and related work in Section 2, and study scanning trends in Section 3. We introduce our new measurement tools in Section 4 and describe our experiments in Section 5. We present results in Section 6 and discuss implications for today's IPv6 network security in Section 7.

2 BACKGROUND

In this section, we provide an overview of the utility of darknets in capturing unsolicited network packets, IPv6 Internet scanning techniques and complexities introduced by IPv6 address allocation and assignment practices to network security tools/techniques.

2.1 Internet Darknets

Darknets are regions of the address space which do not emit any network traffic. Hence, most network traffic, in-bound towards the darknet, can be considered as unsolicited. This unsolicited network traffic is, for the most part, a result of Internet scanners indiscriminately probing large sweeps of the address space. As tools like ZMap [7] have made brute-force scanning of the IPv4 address space possible in minutes, darknets can serve as a practical and efficient method of understanding IPv4 scanner behaviors.

However, brute-force scanning of the significantly larger IPv6 address space is not currently feasible. Consequently, IPv6 Internet scanners have to pick and choose their targets to increase their probability of finding active addresses to probe. Hence, they have little incentive to probe regions of the IPv6 address space that are not "live" (do not emit network traffic). This posits that traditional darknets would not serve as efficient tools for capturing a representative amount of IPv6 scanning traffic. Therefore, building darknets specifically made to capture IPv6 scanning activity need to simulate network "liveness" in order to attract IPv6 scanners.

2.2 IPv6 Internet Scanning Techniques

Contrary to the straight-forward brute-force approach used by IPv4 scanners, scanning IPv6 networks can be broken down into 2 steps; 1) collecting active IPv6 addresses, 2) generating candidate scanning targets. During the IPv6 address collection stage, IPv6 scanners leverage sources of information that contain either active IPv6 addresses *e.g., AAAA* records of domains or hints of address space "liveness" *e.g.,* specific BGP announcements. After this step, IPv6 scanners either have exact addresses to scan or a narrower search space to discover previously unobserved IPv6 addresses. During the candidate generation step, IPv6 scanners utilize the data acquired in step 1 to generate (previously unobserved) IPv6 addresses that have a higher probability of being active than addresses chosen at random. This involves using machine learning algorithms to find semantic patterns in observed IPv6 addresses and generating candidate addresses with similar patterns. Combining these two steps allows IPv6 scanners to scan the IPv6 address space more efficiently than relying on random probing.

2.3 IP-based Blocklisting and Crowd-sourced Abuse Reports

Crowd-sourced abuse reports and IP based blocklists are used as practical tools to act as the first line of defense to proactively block potentially malicious traffic. They work by aggregating lists of offenders by their IP addresses and subsequently blocking incoming traffic from IP addresses on this list. However, the efficacy of these tools is dependent on a reliable IP to host mapping. Unlike IPv4 hosts—which usually get assigned a single /32 address—each IPv6 end-host—by best practice convention—is assigned a /64 subnet [24]. This allows the host to choose a public facing /128 IPv6 address from 2^{64} possible addresses by using one of many address assignment techniques *e.g.,* Stateless Address Auto-configuration (SLAAC) [40]. Hence, IPv6 block-lists will have to operate on a more coarse-grained granularity; an IPv6 subnet instead of an individual IP address. Furthermore, although best practice suggests assigning a /64 subnet to each host, this subnet boundary is not definitive in the real world. Previous research has found instances of network operators assigning both, more and less specific IPv6 subnets to a host [32, 33]. Lastly, IPv6 addresses can also be much more dynamic than their IPv4 counterpartsbecause of the exceedingly large number of available IPv6 addresses.

2.4 Related Work

The methods developed to better capture and understand internet scanning traffic has been evolving for the past few years. Ford et al. [17], Houston et al. [25] and Czyz et al. [15] conducted initial studies to capture IPv6 scanning traffic using darknets. However, they only found trace amounts of IPv6 scanning traffic despite varying sizes of darknets. Consequently, Fukuda et al. [18] leverage a different approach, leveraging DNS backscatter to identify scanning activity, and they were able to establish some evidence of wide spread IPv6 scanning traffic. Richter et al. leveraged a large-scale commercial CDN to passively collect unsolicited network packets incident on the CDN server's inward facing IPv6 addresses [35]. Their technique was able to uncover thousands of weekly scan events originating from dozens of different ASes. **Methods used by these works are reliant on datasets that our not readily available to researchers. In our work, we focus on developing a methodology that can be used in most IPv6 networks to better attract and capture IPv6 scanning traffic.**

Recently, however, methods for capturing IPv6 scanning traffic have focused on stimulating network activity in darknets to attract IPv6 scanners. Tanveer et al. utilize a previously unused /56 IPv6 prefix to run services that emit network activity with the aim of advertising "liveness" of their address space [38]. They found that these services lead to an increase in scanning traffic be several orders of magnitude. Zhao et al. leverage previously unstudied methods of attracting scanning traffic to better understand the IP addresss discovery process and detail behaviors of observed IPv6 scanning sources [41]. **Although the methods developed in these works are reproducible, they lack in scale and breadth of measurements. In our work, we utilize a significantly larger IPv6 prefixes which allow us to test and evaluate new attraction methods, combined with external data.**

3 MOTIVATION

As IPv6 adoption continues to rise globally, the urgency of developing reliable methods to measure changing threats and developing defenses against them escalates. Furthermore, recent reports suggest that IPv6 networks now face more threats than ever [3, 6]. However, there has yet to be a systematic study which establishes how the IPv6 scanning ecosystem has evolved in the past few years.

In this section, we present an analysis of the recent growth observed in the IPv6 scanning ecosystem. Findings from this section serve as a motivation for urgency in answering the research questions we explore in later sections.

3.1 Trends in IPv6 Scanning Traffic

Establishing trends in the IPv6 ecosystem requires that we not only have longitudinal measurements, but also a truly distributed vantage point to observe global trends. To this end, we collaborate with a major CDN and capture unsolicited IPv6 packets incident at a subset of the CDN's servers. We collect any unsolicited incoming packets destined to port numbers other than TCP/80 and TCP/443. Our data ranges from January 1, 2022 to January 1, 2024 and covers traffic logged at some 230,000 machines in over 700 ASes. We note that while the CDN's infrastructure continues to grow and evolve in terms of bandwidth and deployments, there was no significant change in the number of reachable IPv6 address blocks over our measurement window. Thus, trends in scanning activity reported here are not the result of increased visibility of the CDN machines.

Here, we define a scan as a source hitting at least 100 IPv6 addresses of the CDN and with a timeout, or maximum packet inter-arrival time, of 3,600 seconds. This is the same definition used in previous work [8] where it is shown that alternative, shorter timeout intervals have only minor impact on the set of detected scans. Note that in all likelihood the scanning actor is not specifically targeting the CDN, and hence the full scan is likely broader. In the following, we show scan sources as */128* scan sources, */64* scan sources, and */48* scan sources. Here, we first aggregate all traffic that shares the same prefix of given size, and then apply our scan

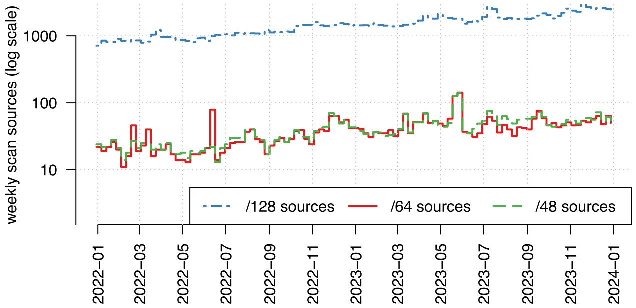


Figure 1: Weekly IPv6 scan sources hitting the CDN.

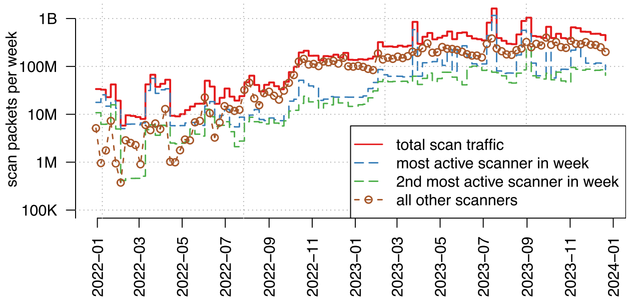


Figure 2: Weekly scan packets (/64 source aggregation).

detection method. By showing scan sources for different aggregation levels we account for the possibility of scanners to leverage random source addresses out of larger prefixes to potentially evade scan detection [8].

3.2 The Rise of IPv6 Scanning Activity

In Figure 1 we show the weekly number of detected IPv6 scan sources that hit the machines of the CDN. Over the course of the last two years we witness a remarkably steady rise in the number of weekly active IPv6 scanners: The number of active /128 scan sources more than doubled, from 1,000 in early 2022 to 2,400 at the end of 2023.

Regarding the actors initiating the scans, a more conservative measure is obtained by aggregating to /64s and /48s (the smallest BGP-routable entitiy in IPv6). As shown in Figure 1, the number of /64-aggregated sources and /48-aggregated sources are actually roughly equal and also steadily rise over our measurement period, increasing three times from fewer than 20 in early 2022 to between 50 and 70 in late 2023. The rise in scanning actors is also reflected in a growing number of ASes from which IPv6 scans are emitted. In early 2022, we witness about a dozen of ASes emitting scan traffic, which then increased to more than 30 ASes over two years (see Figure 11 in Appendix). ***Overall, our numbers show a remarkably clear trend of an increasing number of IPv6 scan sources and actors over the course of 2022 and 2023.***

Figure 2 shows on a weekly basis the number of packets associated with all detected scans (log-scale). Over the course of 2022 and 2023, we witness a massive increase in scanning traffic, seen on a per-packet basis. In early 2022 the weekly number of scan packets ranged between 10M and 60M and exceeded 1B packets in late 2023, an almost 100-fold increase in scan traffic. To study whether the traffic is dominated by just one or two sources, we show (see dashed lines) the weekly scan traffic generated by the top most active scan source and second-most-active scan source and also the aggregate of all other, less active, scan sources (see brown dashed line with dot). We notice that while in early 2022, scan traffic was often dominated by the most active scan source, this has changed over time—by late 2023 the majority of scan packets does not come from the topmost or second most active scan source, but from a broad range of scanning sources. ***Our empirical observations highlight that scanning traffic has increased by two orders of magnitude and also is no longer dominated by one or two actors, but is more broadly distributed.***

The growing diversity and intensity of Internet scanning traffic in IPv6 networks mandates that we capture as representative a set of IPv6 scanner behaviors as possible. Although a CDN vantage point is well-suited for capturing diversity in IPv6 scanning sources, data collected from it is not broadly accessible to researchers. To this end, we deploy our methods in a way that they are readily reproducible for future research; the detailed implementation is outlined in the next section.

4 NOVEL METHODS FOR CAPTURING UNSOLICITED IPV6 TRAFFIC

Previous works have shown that IPv6 prefixes which emit little to no network activity receive significantly less unsolicited network traffic when compared to their "live" counterparts [38, 41]. In this section, we outline methodologies to complement traditional darknets to be better suited for capturing IPv6 specific scanning behaviors.

Our approach leverages both passive as well as *proactive* methods to monitor unsolicited IPv6 traffic. The passive approach involves deploying a traditional darknet-based network telescope to capture traffic. In addition, we design and implement a novel *proactive telescope*, which not only reacts to incoming traffic but also stimulates Internet scanners that utilize various data sources to find probing targets.

4.1 Darknet Network Telescopes

Traditional darknet telescopes provide a good baseline of Internet-wide unsolicited traffic. We deploy network telescopes in two ways. First, the telescope monitors a dedicated network prefix, similar to [15]. This approach offers a static telescope size, providing stable longitudinal measurements.

Our second method is to capture traffic destined to unused space in a live network. The border router uses the internal routing table to forward ingress traffic sent to the unused subnets in the network. The size of the telescope may change over time due to the assignment of subnets.

4.2 Proactive Network Telescope

To setup our proactive telescope, we use the following approaches. First, we advertise prefixes from our address space to the Internet using multiple network protocols *e.g.,* BGP, DNS, and TLS. These prefixes are consequently listed on publicly available IP address/prefix stores *e.g.,* BGP update files, thereby signaling the presence of network activity. Although this technique has previously been used to attract IPv6 scanning traffic [38, 41], our proactive telescope IPv6 prefixes are significantly larger than those of previous studies—a /32 prefix compared to /64—which allows us to test scanning triggers for distinct subprefixes (*honeyprefixes*) that were not previously possible *e.g.,* BGP announcements. Furthermore, we advertise proactive prefixes using techniques which have not been utilized previously *e.g.,* TLS certificates.

Second, we engage with incoming IPv6 scanners using both low and high interaction honeypots. Previous works have shown the interaction with scanners elicits scanning behavior that is not observed by telescopes passively collecting scanning traffic [23]. However, such methods have only been studied in the context of IPv4 networks. Next, we outline the implementation of the methods we use in our proactive telescope to advertise "liveness."

BGP announcements: To indicate "liveness" using BGP announcements, we announce /48 prefixes—which we call *honeyprefixes*—from within the /32 covering prefix of our telescope. These honeyprefixes are chosen randomly from previously unused regions of the telescope's address space. We precisely choose the prefix length of 48 as it is the most specific routable IPv6 prefix [36]; /48 announcements significantly narrows the search space for IPv6 scanners (when compared to the covering /32 assignment) thereby increasing their chances of finding active addresses within the prefix.

Domain names: As most web services are accessed using URLs, the domain name ecosystem provides clues to find web services. We register new domain names in different top-level domains (TLDs) from domain name registrars and immediately deploy AAAA records pointing the root domain to a random address in the honeyprefixes. Therefore, scanners monitoring zone files can resolve the names for IP addresses in the honeyprefixes.

Subdomain names: In addition to the root names, network operators commonly use subdomain names for various services, such as `www`, `mail`, and `ns`, by convention. We selected a total of 374 names listed on at least three out of four popular subdomain name lists ([10, 12, 31, 34]). We deploy AAAA records to map each subdomain name to a randomly assigned IP address within a honeyprefix.

TLS Certificates: As most websites nowadays have adopted HTTPS, we issue TLS certificates for the root and subdomain names to mimic web services. As Certificate Transparency logs [2] expose the existence of subdomain names without allowing public access to our DNS zone file, the domains listed in these certificates can be easily discovered by scanners.

IPv6 hitlist: IPv6 hitlists [19, 37] compile lists of responsive targets based on public available datasets and active measurement results. Our proactive telescope setup could generically enroll some honeyprefixes' IP addresses into these hitlists. Further, we collaborate with a major hitlist provider to manually add random addresses within the honeyprefixes that are not expected to be automatically discovered by the hitlist.

4.3 Twinklenet: Our IPv6-native low-interaction honeypot

We design a lightweight low-interaction multi-protocol honeypot, namely *Twinklenet*, to respond to unsolicited incoming traffic to part of the telescope. Existing open-source honeypots (e.g., Amppot [26], T-pot [39], and Spoki [23]) neither natively support IPv6 nor multi-protocol IP aliasing (*i.e.,* handling packets to multiple destination IPs with a single interface). An alternative approach to enabling IP aliasing is to use Network Address Translation (NAT), but it cannot easily scale to handle address spaces in IPv6 (*e.g.,* , a /64 subnet).

Twinklenet supports IP aliasing for both IPv4 and IPv6 address spaces and responds to four popular protocols (Table 1). A single instance of Twinklenet can handle incoming packets toward multiple non-continuous subnets and addresses. Apart from responding to ICMPs, Twinklenet can bind to any TCP ports of any honeypot's IP addresses to accept incoming connections. It then terminates the connection using TCP FIN packets upon the completion of TCP handshake. As the sender usually sends the first data packet right after the handshake, Twinklenet can capture the first data packets sent by the scanner. Crafting responses for UDP-based protocols require parsing the queries. Twinklenet supports two popular UDP-based protocols: DNS and NTP. Instead of implementing the actual service that attackers may exploit for attacks, Twinklenet replies with an error message for each query to show the responsiveness to the sender.

We implement Twinklenet using Go utilizing BPF filters and PCAPGO [21] to capture and filter incoming packets to the telescope. The handling of outgoing packets differs

Table 1: Protocols and interactions supported by Twinklenet.

Protocols	Request	Response(s)
TCMP/ICMPv6	ICMP/ICMPv6 Echo request	ICMP/ICMPv6 Echo reply
TCP	TCP SYN to an open port	Complete three-way handshake and close the connection with FIN
	Other TCP packet to an open port	TCP RST
NTP (over UDP)	Any client NTP packet	NTP Kiss-of-Death packet (Reference Identifier=DENY)
DNS (over UDP)	Any DNS query	DNS respond with response code SERVFAIL

from normal services in two ways. First, the response packets use the destination addresses from the request packets rather than the outgoing interface's IP address. Second, the outgoing interface could be different from the one capturing incoming packets. Twinklenet leverages raw socket to bypass the system's routing table to achieve these two goals. We will make our Twinklenet source code and deployment instructions available to the research community.

4.4 High interaction honeypots

To investigate if scanners react differently to full-stack systems, we tackle challenges to incorporate existing high interaction honeypots in the proactive telescope. We employ T-Pot [39], which is a container-based framework to emulate multiple services and application protocols.

As a T-Pot instance can only bind to a single IPv4 address, we design a two-stage approach to enable IPv6 and IPv6 aliasing support (i.e., our honeypot responds to queries on an IP address within an entire IPv6 prefix) for the honeypot without modifying the source code (Fig. 10 in Appendix for details). We configure the internal routing table to redirect the traffic toward T-Pot's honeyprefix to a dedicated access router. The router maps all the traffic toward any addresses in the prefix to the first address of the honeyprefix (i.e., : : 1) and source ports-pair using Destination Network Address Translation (DNAT). The translated traffic then forward to a reverse proxy which conducts a static 6-to-4 translation to the IPv4 address listened by the T-Pot. The traffic then diverts the appropriate honeypot container by the protocols (TCP/UDP) and the destination ports. For example, traffic toward TCP ports 22 or 23 forwards to the Cowrie honeypot [14] container that emulates SSH and Telnet services.

Our setup stores the DNAT table records, which contains the time, original destination IP addresses and source ports, and allows us to recover the original destination address in the T-Pot logs. Furthermore, the access router mirrors the traffic to the packet capturer, which saves it in pcap format.

5 EXPERIMENTS AND DATASETS

In this section, we outline the implementation details of our controlled experiments in our proactive telescope. We also provide an overview of the datasets that we gather from these experiments for subsequent analysis.

5.1 Network Telescope Deployment

We deploy three geographically and topologically diverse IPv6 network telescopes (see Table 2)—one in a transit ISP and two in academic networks—to capture unsolicited network traffic to the unused address spaces. Over the course of our experimentation timeline, we capture over a billion unsolicited packets, from 2000 unique ASes targeting more than 150M unique destinations in all of our telescopes combined. A breakdown of the results is shown in Table 2.

NT-A is hosted in an ISP network with low IPv6 address space utilization. Of the /32 address space assigned by APNIC, ISP's equipment and its customers only use the initial five /48s. Apart from capturing unsolicited traffic, we collaborate with the ISP to modify the network configuration and deploy the proactive network telescope (§5.2).

NT-B [9] monitors incoming traffic to an unused /48 network dedicated for the purpose of running a network telescope since June 2022. We obtain 2-year data for our analysis.

NT-C is deployed in an academic network with a /32 assignment from ARIN. Similar to *NT-A*, it captures all the traffic sent to any unassigned subnets within the address space. The address space utilization of this network is higher than that of *NT-A*. Part of the address space is assigned to equipment and different departments in the university from the first /33 of the overall /32.

5.2 Proactive telescope deployment

We implement a proactive network telescope (§4.2) in the address space and infrastructure of *NT-A*. Fig. 3 shows *NT-A*'s infrastructure. The access router (AR) forwards ingress traffic to unused prefixes in ISP A to server Ⓐ, which captured the traffic in pcap format. The server also runs a BIRD Internet routing daemon to announce a total of 56 honeyprefixes randomly located in the lower /33 of *NT-A*'s /32 network (Fig. 12). *NT-A*'s network operator registered the prefixes on APNIC's Resource Public Key Infrastructure (RPKI) portal, such that the ISP's upstream providers would accept and propagate the new routes.

Table 2: Overview of scanning traffic/sources captured and destinations targeted in *NT-A*, *NT-B* and *NT-C*

Telescope	Address space	Location	Network type	Measurement start date	Unsolicited packets received	Unique traffic sources				Unique destinations targeted		
						/128	/64	/48	ASes	/128	/64	/48
NT-A	/32	Southern Asia	Transit ISP	07/23	654M	259k	190k	138k	1.9k	134M	3.1M	61.5k
NT-B	/48	Ireland	Research	01/23	300k	1.9k	367	354	60	100k	65.5k	1
NT-C	/32	United States	Academic	10/23	250M	57k	26k	24k	507	21M	14.9M	48.8k

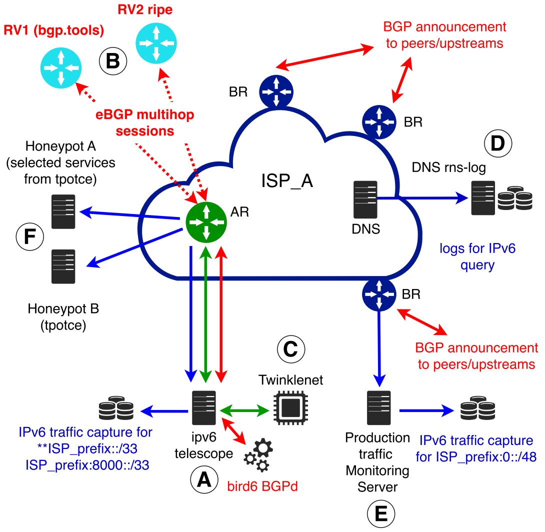

RV1 (bgp.tools)
RV2 ripe
B
BGP announcement to peers/upstreams
BR
BR
eBGP multihop sessions
Honeypot A (selected services from tpotce)
D
DNS rns-log
ISP_A
AR
DNS
F
logs for IPv6 query
Honeypot B (tpotce)
BR
BGP announcement to peers/upstreams
C
Twinklenet
IPv6 traffic capture for **ISP_prefix::/33 ISP_prefix:8000::/33
ipv6 telescope
A
bird6 BGPd
Production traffic Monitoring Server
E
IPv6 traffic capture for ISP_prefix:0::/48

Figure 3: Overview of IPv6 proactive telescope setup.

The Twinklenet low-interaction honeypot Ⓒ responds to traffic toward the honeyprefixes according to our configuration (Table 3). For high-interaction honeypots Ⓕ, we implement the two-stage address translation described in §4.4 and deploy dedicated servers for the two honeypot instances ($\mathcal{H}_{TPot1}$ and $\mathcal{H}_{TPot2}$) due to their high compute and memory resource requirements. To compare the traffic toward production networks, we mirror both incoming and outgoing traffic to a /48 subnet hosting ISP infrastructure (e.g., routers) at the border routers in each Point of Presence (PoP). As the subnet has no actual users, the traffic does not contain any personal information.

Honeyprefix configurations: We deploy 12 different types of honeyprefixes (see Table 3) to investigate how Internet scanners discover live hosts with public data sources and react to various types of network behavior. More specifically, we deploy services on random addresses in the honeyprefixes and attach different "clues" pointing to these addresses. As a /48 network consists of 2^{80} IP addresses, the probability of discovering these addresses in the honeyprefixes at random is very slim. Therefore, we can identify the data sources that scanners use to create their target lists.

IP aliasing: $\mathcal{H}_{Alias}$ and the two honeypots ($\mathcal{H}_{TPot1}$ and $\mathcal{H}_{TPot2}$) implement IPv6 aliasing using Twinklenet and the NAT router, respectively. All addresses in these prefixes respond to incoming ICMP echo requests.

ICMP responsiveness: We configure Twinklenet to respond to ICMP Echo requests for the first address ('::1') and two randomly selected addresses in non-aliased honeyprefixes ($\mathcal{H}_{RDNS}$, $\mathcal{H}_{TCP}$, and $\mathcal{H}_{UDP}$). One random address in $\mathcal{H}_{Combined}$ is also responsive to ICMP.

Domain and subdomain names: We purchase a total 9 domain names (6 `.com`s, 2 `.net`s, and 1 `.org`) from GoDaddy [4]. Shortly after the registration, we use the registrar-provided DNS server to set up the root (i.e., `@`) AAAA record for the corresponding DNS zones. Additionally, we deploy AAAA records of over 300 common subdomain names (§4.2) in four of the domain names. All records point to a randomly selected IP addresses in the honeyprefixes.

TLS certificates: Since our honeypots (see §4.3 and §4.4) do not have the functionality of an actual web server, we cannot use the `HTTP-01` challenge [28], which requires hosting a special file in a randomly generated location on the web server to validate our control over the domain names. Instead, we obtain TLS certificates using the `DNS-01` challenge [28]. More specifically, we implement a customized `certbot` plugin to support our domain registrar's APIs, enabling automatic insertion of `TXT` DNS records required by the challenge. We issue TLS certificates using Let's Encrypt for all the root domain names and only 50 subdomain names, due to Let's Encrypt's weekly certificate limit [27].

TCP/UDP open ports: Our honeypot deployment (Twinklenet and T-pot) reacts to incoming TCP and UDP traffic to specific IP addresses in the honeyprefixes. For the IP addresses we select to respond to ICMP in $\mathcal{H}_{TCP}$ and $\mathcal{H}_{UDP}$, we use Twinklenet to simulate popular services over TCP (web, and remote control) and UDP (DNS and NTP), respectively. We also enable web ports on the IP addresses pointed by AAAA records of domain/subdomain names in $\mathcal{H}_{Com}$, $\mathcal{H}_{Org/net}$, and $\mathcal{H}_{Combined}$.

We integrate multiple features in $\mathcal{H}_{Combined}$. The first address of $\mathcal{H}_{Combined}$ responds to all ICMP and TCP/UDP common ports. Also, we select four random addresses to respond to ports related web, remote control-related, DNS, and NTP services, respectively.

$\mathcal{H}_{TPot1}$ and $\mathcal{H}_{TPot2}$ respond on TCP/UDP ports corresponding to some of the most frequently targeted protocols *e.g.,* SSH, Telnet, DNS, SMTP *etc.*. Refer to E for a detailed breakdown of all responsive protocols and UDP/TCP ports.

IPv6 Hitlist: Some prefixes and IP addresses in honeyprefixes are listed by the IPv6 hitlist probing [19]. We find that the "aliased"/"non-aliased" prefix list included all three aliased ($\mathcal{H}_{Alias}$, $\mathcal{H}_{TPot1}$, and $\mathcal{H}_{TPot2}$) and five non-aliased ($\mathcal{H}_{RDNS}$, $\mathcal{H}_{TCP}$,$\mathcal{H}_{UDP}$, $\mathcal{H}_{Com}$, $\mathcal{H}_{Org/net}$, and $\mathcal{H}_{Combined}$) honeyprefixes, respectively. The hitlist also discovered some addresses with open ports on UDP port 53, and TCP port 80 and 443. Furthermore, we collaborate with the hitlist maintainers to manually add two IP addresses (one at the beginning of the address space, and one random in the honeyprefix) into each hitlist category. In total, we manually insert 40 addresses (20 per honeypot) across 10 hitlist categories.

Metadata and data processing: We map source IP addresses to Autonomous Systems (ASes) and countries using CAIDA's RouteView Prefix to AS mapping [13] and MaxMind geolocation database [30]. We utilize datasets collected on the same day as the packet timestamps, ensuring the timeliness of the mapping.

5.3 Crowd-sourced IPv6 Abuse Reports

For this, we leverage crowd-sourced abuse report data from AbuseIPDB [1]. AbuseIPDB is a web service where Internet users can report IP addresses that they observed engaging in some from of abusive behavior. We privately obtained a dataset of all reports concerning IPv6 addresses submitted to AbuseIPDB from January 1, 2023 to December 31, 2023. The dataset contains 805 368 abuse reports about 242 532 unique IPv6 addresses submitted by 1200 reporters. 222 858 (92%) of these IPv6 addresses received reports from a single user, with 194 209 (80%) of IPv6 addresses being reported only once during the observation period. Given the abundance of addresses with only a single reporter, we analyze if any of our findings differ when including only addresses reported by multiple users. We find that the main findings remain similar after filtering out single-reporter addresses (see Appendix B for more details). Thus, we perform the remainder of our analysis on all reports.

6 RESULTS

In this section, we present our analysis on the scanning traffic captured by our telescopes.

6.1 Characterization of Unsolicited Traffic

NT-A accounts for ~ 70% of all unsolicited traffic that we capture (x% toward *honeyprefixes*) from the most diverse set of sources. x% of all unsolicited /64 traffic sources in *NT-A* target at least one *honeyprefixes*. *NT-C* receives most of the remaining ~ 30% of the traffic albeit form a much less diverse set of sources. *NT-B* only receives a small fraction of the total traffic owing to its order of magnitude smaller address space.

Breakdown of scanner sources by telescope: Network operators often assign subnets with prefix lengths ranging from /48 to /64, rather than individual IP addresses (/128), to their users. Scanners can use a large number of unique source addresses within their address space to perform measurements and evade IP-based heavy hitter detection. As the sizes of the subnets assigned to scanners vary, we aggregate source IP addresses (/128) using three common prefix lengths (/48, /64, and /112) in our analyses.

We analyze the network types of the scanner sources mapped using ASdb [42] (Fig. 4). The type of network source is further broken down to the type of unsolicited network traffic they sent and the proportion of the total packets, unique destinations and unique sources they contribute to per telescope. During our analysis, we observe IP prefixes and ASes dedicated to conducting IPv6 Internet scanning for various purposes *e.g.,* Internet Measurement AS [5]. We assign 4 such network entities to the *Internet Scanner* category.

Hosting/cloud providers are responsible for the biggest chunk of unsolicited traffic sent toward our telescopes (> 50% of unsolicited network traffic in both *NT-A* and *NT-C* which receive > 99% of all the traffic). These providers have different breakdowns by traffic type on *NT-A* and *NT-C*; *Amazon AWS*/*Google Cloud Platform* are responsible for the most unsolicited network traffic in *NT-A*/*NT-C*, respectively.

The *Internet scanner* category dominates in the number of unique sources in our telescopes; 90% of all source /128 addresses targeted *NT-A* belong to this category, as these scanners use distributed IP addresses from a covering prefix as large as a /30. These sources also have a similar distribution of traffic type across all our network telescopes, mostly probing various popular TCP ports.

Cross telescope comparison: Fig. 5 shows an overlap of the sources aggregated by /64 that targeted each of our telescopes; The color in the figure represents the Jaccard Similarity ($\mathcal{JS}$) of the sources at each aggregation level where:

$$\mathcal{J}accard(NT_y^{agg},NT_x^{agg}) = \frac{|Sources_{agg}NT_y \cap Sources_{agg}NT_x|}{|Sources_{agg}NT_y \cup Sources_{agg}NT_x|}$$

The average Jaccard similarity across prefix aggregation levels of /32, /64 and /128, across all telescope combinations, is ~ 0.1. This shows that the sets of sources observed at different telescopes are highly distinct; the highest $\mathcal{JS}$ of 0.2 is observed between *NT-A* and *NT-C* at /32 aggregation.

Table 3: Configuration of Honeyprefixes.

Honeyprefixes	BGP	Aliased	ICMP	TCP	UDP	rDNS	Domain	Subdomain	IPv6 Hitlist ★
$\mathcal{H}_{Alias}$	1×/48	✓	●	✗	✗	✗	✗	✗	Aliased
$\mathcal{H}_{RDNS}$	1×/48	✗	◐	✗	✗	✓	✗	✗	NA
$\mathcal{H}_{TCP}$	✗‡	✗	◐	web, remote	✗	✗	✗	✗	✗
$\mathcal{H}_{UDP}$	1×/48	✗	◐	✗	53, 123	✗	✗	✗	NA, UDP53, ICMP
$\mathcal{H}_{Com}$	1×/48	✗	✗	web	✗	✗	2×`.com`	✗	NA, TCP80, TCP443
$\mathcal{H}_{Org/net}$	1×/48	✗	✗	web	✗	✗	1×`.org`, 1×`.net`	✓(only `.net`)	NA, TCP80, TCP443
$\mathcal{H}_{Combined}$	1×/48	✗	◐	web, remote	53, 123	✗	1×`.net`	✓	NA, TCP80, TCP443
$\mathcal{H}_{TPot1}$	1×/48	✓	●	See appendix D		✗	2×`.com`	1×`.com`	Aliased, Manual
$\mathcal{H}_{TPot2}$	1×/48	✓	●	See appendix D		✗	2×`.com`	1×`.com`	Aliased, Manual
$\mathcal{H}_{Specific}$	/49-/64	✗	✗	✗	✗	✗	✗	✗	✗
$\mathcal{H}_{eBGP}$	2×/48†	✗	✗	✗	✗	✗	✗	✗	✗
$\mathcal{H}_{Control}$	/48s	✗	✗	✗	✗	✗	✗	✗	✗

Note: ●/◐represent the entire subnet/specific addresses were responsive to ICMP, respectively.
‡: We configured BIRD to announce the prefix, but the announcement failed to reach the Internet due to a technical problem.
★: Aliased/NA represents the aliased/non-aliased prefixes list, respectively. ICMP, TCP80, TCP443, and UDP53 denote the hitlists that reported at least one IP in the subnet as responsive to the corresponding protocol.
web: 80, 443, 8080, 8443; remote: 22, 23, 2323, 3389. †: only to one collector in bgp.tool or RIPE.

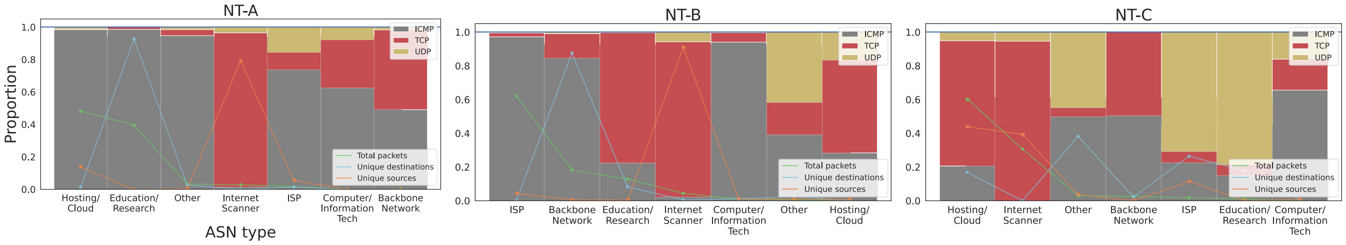


Figure 4: Breakdown of scanner sources by network types.

However, the small number of overlapping sources account for the majority of unsolicited network traffic and target the most unique destinations within our telescopes. Fig. 5 shows that those sources using larger prefixes increases the percentage of unsolicited network traffic sent by common sources. For example, if we compare the source IPs between *NT-A* and *NT-B* at a /128 aggregate, the common sources are responsible for 4.3% of unsolicited network traffic received by *NT-A*. However, this number increases to 96.3% when we aggregate by a larger prefix *i.e.,* /64. Common sources between *NT-A* and *NT-C* aggregated by /64 are responsible for 97.3%/ 99.2% of the unsolicited traffic received by *NT-A*/*NT-B*, respectively.

NT-A—our proactive telescope—has the smallest overlapping sources with other telescopes that target the most unique destinations. Fig. 5 shows that although overlapping sources, aggregated by /64, account for 96.8% of unsolicited traffic captured by the telescope, these sources only contribute to 45.1% of unique destinations targeted within *NT-A*. Hence, *NT-A* is able to attract scanner sources that probe a large portion of the telescope's address space; these sources that are not observed by *NT-B* or *NT-C*.

6.2 Scan Traffic Attraction by Controlled Experiments

This section will discuss the increase in scanning activity and sources we observe by running our controlled experiments.
Scan traffic attracted: The scanning activity trends observed in each of our honeyprefixes as a result of conducting our controlled experiments (Fig. 6). The line colors represent the controlled experiments run in each honeyprefixes. To increase the readability, we color honeyprefixes running similar controlled experiments with a common shade. The dotted lines represent when the BGP announcement for a certain honeyprefix was made. Markers represent additional triggers that we deployed to attract more scanning activity.

An increased in scanning activity is observed because of both, the initial BGP announcement and consequently by the additional triggers we deploy. For honeyprefixes that do not deploy additional triggers *e.g.,* BGP announcements (only),

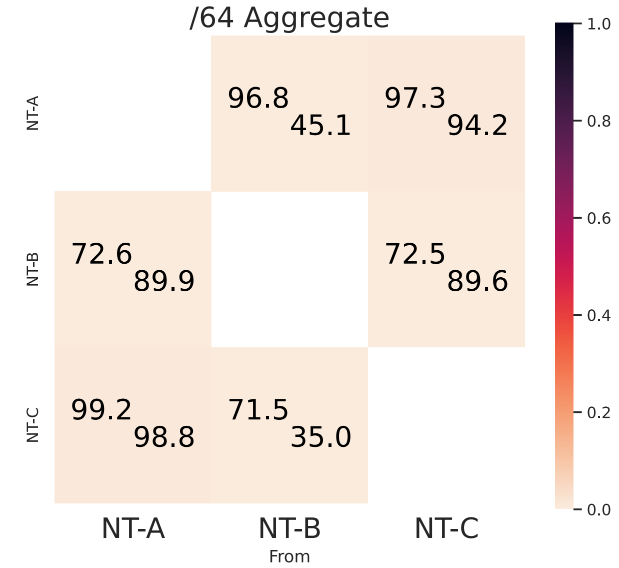


Figure 5: Heatmap of Jaccard similarity of sources at aggregation level of /64.

we notice that after an initial spike, unsolicited network traffic eventually subsides to a constant level. Furthermore, we observe that SSL certification registration trigger leads to an increase of 3 orders of magnitude in the high-interaction honeypot honeyprefix; unsolicited network traffic per day increased from 15000 packets to almost 1 million. Domain registration and IPv6 hitlist registration also also proceeded by spikes in scanning activity, albeit, not as intense.

To understand how scanners react to removal of a controlled experiment, we withdrew 2/3 honeyprefixes dedicated to the BGP announcements (only) experiment. Scanner's reaction to the withdrawal can be observed in Fig. fig. 6, which shows that scanning traffic in 2 of the withdrawn honeyprefixes almost instantly subside after 2024-02. This observation posits that IPv6 scanners regularly acquire new BGP update files to only scan actively announced IPv6 prefixes.

Scan sources attracted: Our honeyprefixes' characteristics triggered scanners to perform different types of measurements. Fig. 7 depicts the responsive features that scanners, aggregated by /48, probed using the labels we classified for each connection using the time and IP address they probed.

BGP announcement. As we show earlier in this section, making a BGP announcement is an essential step to attract scanners. Although $\mathcal{H}_{TCP}$ is reactive on popular TCP ports, we only observe opportunistic probing using ICMP and UDP from 7 /48s in 5 ASes to random addresses in the honeyprefix.

ICMP probing. Since the entire $\mathcal{H}_{TPot1}$, $\mathcal{H}_{TPot2}$, and $\mathcal{H}_{Alias}$ are responsive to ICMP, over 50k unique sources probe any address in these three honeyprefixes *solely* using ICMP pings. About 400 sources probe $\mathcal{H}_{UDP}$ and $\mathcal{H}_{RDNS}$ solely with ICMP, despite both honeyprefixes having only 3 addresses responding to ICMP. All these sources target only the first address of the subnet, without probing the other two active addresses in a random location of the honeyprefixes. Two sources discover the random address for which we enabled ICMP response in $\mathcal{H}_{Combined}$. They also scan UDP ports and other non-responsive parts of the subnet (magenta bars, "IU" and "IO", Fig. 7a).

Domain registration. A significant number of sources leveraged the root AAAA record of domain names to compile probing targets. 123/56 sources probe the IP in $\mathcal{H}_{Com}/\mathcal{H}_{Org/net}$ pointed to by the root records of the names in the honeyprefixes (light blue/dark red bars in Fig. 7 with "D" on the x-axis), respectively. However, 40.7% and 98.2% of the sources scan web-related ports that Twinklenet reacts to (x-axis with "DT", Fig. 7b). All these sources continued to probe after we issued the SSL certificate ("d"). Apart from domain names, all except one sources also probed targets in the hitlist ("H"). We capture vertical scanning activities to multiple TCP ports against the addresses. We start receiving traffic toward those IP addresses about 1 week after the domain registration. As there is a month-long gap between our domain registration (Sept. 19, 2023) and the addition to the IPv6 hitlist (Oct. 23, 2023), the sources likely learned the names (and the corresponding addresses) from zone files.

Subdomain names and TLS certificates. No sources are able to detect common subdomains without TLS certificates (*i.e.*, "s" always came with "S" in Fig. 7). Scanners quickly react to new TLS certificates. The first scanner from DigitalOcean arrives 7 seconds after the issuance of certificates. It is highly likely that Certificate Transparency logs [2] are the source of this information. Similar to the behavior triggered by domain registration, the scanning traffic targets web-related TCP ports ("Ss" and "T", Fig. 7).

IPv6 Hitlist. The manual addition of hitlist entries for $\mathcal{H}_{TPot1}$ and $\mathcal{H}_{Tpot2}$ enables us to isolate the effect of using hitlists. 115/111 sources probe the addresses in $\mathcal{H}TPot1/\mathcal{H}_{Tpot2}$ (and corresponding protocols) specified in the hitlists (Fig. 7b), respectively. These scanners also scan other open services in the honeypots (e.g., "HT", "HU", Fig. 7b) or send IMCP pings (e.g., "HI", Fig. 7b).

The addresses detected by the hitlist algorithm receive similar attention to the honeypots. For example, over 600 sources ping $\mathcal{H}_{UDP}$, which has one IP inserted into the ICMP list. The web services on the addresses pointed by the domain names are still popular scanning targets after the addition into the TCP80 and TCP443 hitlist. However, we cannot confirm if these scanners use the hitlist or other information sources to discover the IP addresses.

Key takeaways: We find that proactive telescopes are essential to capturing a more representative breadth of IPv6

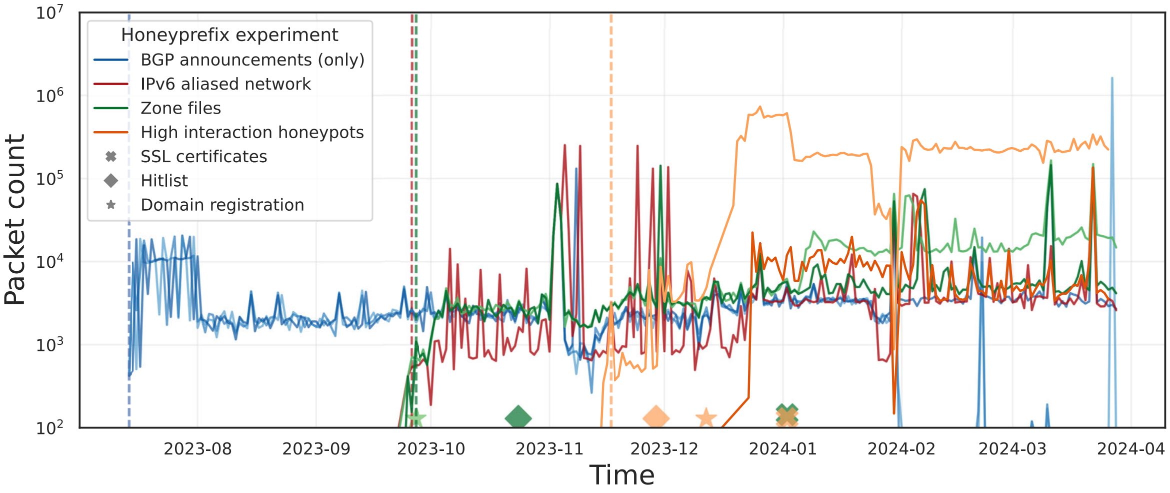


Figure 6: Scanning traffic trends in honeyprefixes.

scanner behavior. *NT-A* (our proactive telescope) was able to capture a wider-array of IPv6 scanner behaviors when compared to other network telescopes. We also find that some of our controlled experiments *e.g.,* SSL certificate registration, is extremely effective in attracting scanning sources.

7 EFFICACY OF NETWORK SECURITY TOOLS IN IPV6 NETWORKS

7.1 Analyzing IPv6 Security Incidents through Crowd-sourced Abuse Reports

Next, we investigate to what extent regular Internet hosts face security threats from IPv6 scanners.

The abuse reports contain a freetext comment field where reporters can describe the type of abuse. As many users submit their reports automatically via an API, these comments often include either log output from services or tools (e.g., web servers, firewalls, intrusion prevention systems) or a static string indicating why an IP address is being reported.

To better understand the reports, we classify them according to the type of activity described in the comment field. Based on common word patterns found in the comments, we manually construct regular expressions and keyword searches that group reports pertaining to similar incidents into categories. Using this method, we are able to classify 91% of all reports into 22 categories.[1]

Table 4 shows the top six categories for abuse reports by the number of users reporting a specific activity. The largest category—both in terms of the number of reporters and the number of IP addresses reported—pertains to attacks on WordPress servers (#1). It includes reports of addresses either crawling for WordPress-specific URLs (e.g., `/wp-login.php`) or trying to brute force a WordPress login. The second most-reported category comprises all remaining reports of IP addresses engaging in web crawling activities (#2), excluding those explicitly related to WordPress. We also see several users complaining about IP addresses connecting to their SSH (#3) and mail servers (#4). SSH incidents originate from only 2461 addresses. Similarly, many users report addresses for sending email or webform spam over IPv6 (#5), yet these reports only relate to 5637 IP addresses. Finally, a considerable number of IP addresses are reported for port scans (#6).

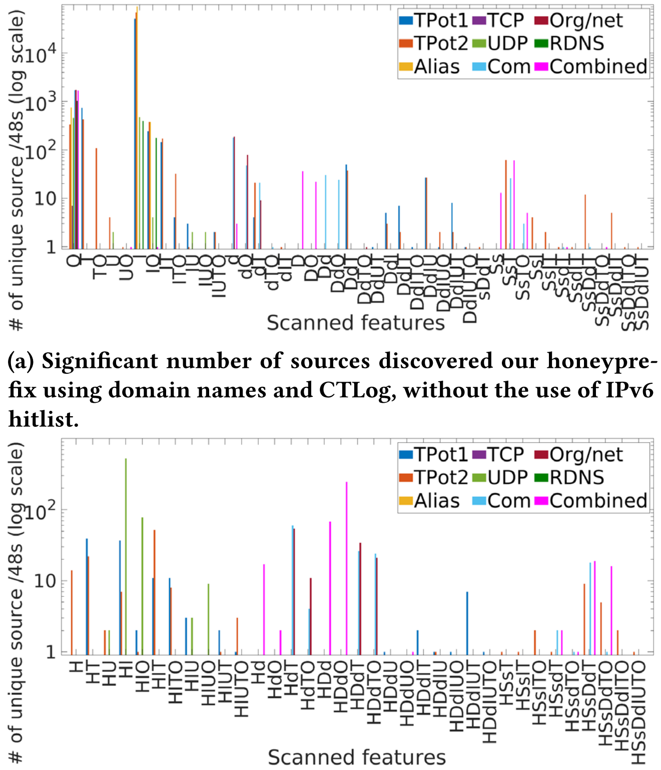
Table 4: AbuseIPDB: Top types of reported IPv6 activity.

	Activity	Reporters	Reports	Reported IP addresses			
				ASes	/48s	/64s	/128s
#1	WordPress	300	182 549	1474	26 256	45 280	68 847
#2	Web crawling	262	61 688	1276	12 315	17 053	27 349
#3	SSH	261	15 186	171	422	469	2461
#4	Mail	183	23 909	374	3073	5436	14 593
#5	Spam	141	12 757	349	1122	2116	5637
#6	Port scan	88	372 878	1758	8833	10 448	54 238

In the port scan category, we find that many reports carry information about the targeted ports in the comment field. Using regular expressions, we successfully extract destination ports from 44% of all port scan reports (submitted by 28 users), giving us the target ports of 6353 reported scanner addresses.

Table 5 shows the top four ports scanned according to AbuseIPDB users, ranked by the number of scan sources

[1] Note that 4% of reports have an empty comment field and can hence not be classified.



(a) Significant number of sources discovered our honeyprefix using domain names and CTLog, without the use of IPv6 hitlist.

(b) Manually inserted addresses in the TPots reveals that IPv6 hitlist was one of the information sources for finding targets.

Figure 7: The combination of tactics adopted by scanning traffic sources, grouped by /48. *I*: ICMP Responsive, *T*: TCP Open ports, *U*: UDP Open ports, *D*: Domain name, *d*: TLS certificate of domain name's root, *S*: Subdomain name, *s*: TLS certificate of subdomain names, *H*: IPv6 Hitlist, *O*: Others.

in various aggregations. Many IP addresses appear to be scanning for open FTP (21), SSH (22) and Telnet (23) ports. This observation corresponds to the findings of Richter *et al.* [35], who found these ports among the top four for IPv6 when ranking by the number of source /64s. In the larger aggregations, we further discover that HTTP (80) and IMAP (993) are popular targets; the same applies to port 33435, commonly used in UDP traceroutes.

We also perform a longitudonal analysis of the top categories and notice that the popularity of categoires remains fairly constant over time. (see Appendix C for an in-depth discussion of this.)

Coverage of abuse reports: To analyze the efficacy of crowd-sourced reports, we analyze the coverage of the IP addresses reported in AbuseIPDB. To this end, we calculate the overlap in IP addresses between the ones reported in AbuseIPDB and the ones that sent unsolicited network traffic to *NT-A* (which contains the most diverse sources of unsolicited network traffic that we received). We use the same source overlap metric as used in Section 6, i.e., the aggregate-specific Jaccard Similarity index. We find that although only 31k IP addresses overlap between the two sources, these IP addreses are responsible for 25% of all unsolicited network traffic we received at *NT-A*. Table 6 shows the source overlap metrics at different prefix aggregates.

Table 5: Top destination ports reported in AbuseIPDB.

Rank	**Ranking**					
	by #/128s		by #/64s		by #ASes	
	Port	#/128s↓	Port	#/64s↓	Port	#ASes↓
#1	21	2157	21	198	80	75
#2	22	2028	22	194	22	49
#3	23	1746	80	185	21	45
#4	8080	1666	23	155	33435	34
#5	3389	1579	993	153	993	34

We find the overlap metric quickly drops as aggregation levels become more specific than a /32. Although the insignificant overlap metric suggests little to no overlap between the two sources, a deeper analysis proves otherwise. As described in Section 6, the *Internet Scanner* category is responsible for the most amount of unique sources as they can use expansive covering prefixes to send unsolicited network traffic. Out of the 8 /32 prefixes scanners from this category used to scan *NT-A*, 7 were captured by AbuseIPDB. However, as *Internet Scanners* used vastly distributed /128 sources, the overlap metric for more specific prefixes was insignificant. As such distributed scanning sources might reduce AbuseIPDB's ability to proactively block unsolicited network traffic, we recommend aggregating reports from such sources into a covering prefix.

Once we aggregate IP sources from *Internet scanner* category into their covering prefix and focus on popular IP addresses in AbuseIPDB (IPs reported by at least 2 unique reporters), we find that the overlap metric jumps up 12x for sources aggregated at /64.

Proactivity of abuse reports: Next, we analyze how proactive crowd-sourced abuse reports are at blocking unsolicited network traffic. To this end, we compare the dates of when an overlapping source was first reported on AbuseIPDB and when it first appeared on *NT-A*. We find that 75% of the overlapping sources were either reported on the same day or earlier on AbuseIPDB. These source account for 100M unsolicited network packets on *NT-A* (15% of all unsolicited network traffic received). We also observe that 98.5% of IPs observed by the two data sources on the same day either originate from the *Internet Scanner* category or a scanning source operating from a cloud provider—which we observe

Table 6: Aggregated Jaccard Similarity between IP addresses in *NT-A* and AbuseIPDB.

Source IPs	**Aggregation**			
	/32	**/48**	**/64**	**/128**
All	0.31	0.02	0.01	0.05
Agg. IS / popular IPs	0.40	0.17	0.12	0.12

Table 7: Parameters for threshold-based source aggregation.

Level	Prefix length	Aggregate threshold	Blocked prefix lifetime
1	/128	5	1 hour
2	/112	10	50 mins
3	/64	15	45 mins
4	/56	20	30 mins
5	/48	–	15 mins

in all of our vantage points. This suggests that these sources were involved in a large scale global scanning campaign.

7.2 Blocklisting methods in IPv6 networks

Blocklisting has traditionally been deployed on an individual IP address level. Although this technique works for IPv4 networks – where a given host's IP address mobility is limited – the prefix allocation practices of IPv6 networks renders this technique unreliable. As discussed in Section 6, some internet scanners can employ IPv6 prefixes as large as a /30 dedicated to scanning IPv6 networks. Hence, implementing blocklisting solutions in IPv6 networks requires techniques specifically catered to such scanner behaviors.

Method: To the best of our knowledge, no current blocklisting techniques exist which take into account the prefix allocation practices of IPv6 networks. To this end, we implement a version of an approach [20] envisioned by Gont et al. that attempts to adapt blocklisting solutions to IPv6 networks. This technique is built on threshold-based source aggregation; instead of blocking individual /128 IPv6 addresses, this method aggregates multiple sources from a common IPv6 prefix when a certain threshold is met. Table 7 shows our choice of parameters – in way of a more cautious approach – for the proposed method suggested by Gont et al.

The parameters are interpreted in the following manner. 1) **Level:** defines the granularity of the blocked prefix in an increasingly coarser prefix, 2) **Prefix length:** is the size of the aggregated prefix blocked at its corresponding level, 3) **Aggregate threshold:** defines the number of prefixes observed at the current level before they can be aggregated to a prefix size of level (n+1) and 4) **Blocked prefix lifetime:** which is the maximum time a prefix length at a particular level should be blocked for.

Dataset: To examine the efficacy of this approach, we implement the method described above on a sample of 200M unsolicited network packets received at *NT-A*. We then extract the blocked prefixes and size of the blocklist to analyze how well this method fares against a diverse set of IPv6 scanner behaviors incident on *NT-A*. To evaluate cases of under/over blocking, we manually collect prefix sizes assigned to individual hosts for networks from where unsolicited network traffic is originated. For example, *Internet-Measurement.com* explicitly list IPv6 prefixes from where they send scanning traffic, cloud services providers like *Google Cloud Platform* state IPv6 prefixes assigned to each cloud instance *etc.*. All in all, we collect ground-truth IPv6 prefixes for sources that account for 95% of the 200M unsolicited packets we simulate the blocklisting method on.

Results: We find that this implementation produces cases of accurate, under and over-blocking of IPv6 prefixes. The block-listing method was able to accurately block a /64 prefix dedicated to *The Shadow Server Foundation* – which is used for IPv6 scanning – from within a /32 prefix assigned to *Hurricane Electric* which contained other sources of unsolicited network traffic in it as well. However, the cases of over/under blocking are more rampant.

Our implementation led to over-blocking of specifically IPv6 prefixes assigned to commercial cloud providers. We observe that although, *Google Cloud Platform (GCP)* and *Amazon EC2* assigns /96 and /80 IPv6 prefixes to their cloud instances, the blocklisting method ended up enlisting 1 /56 and 116 /64s from GCP and EC2 respectively. This was triggered by a coordinated scanning campaign – consisting of very similar scanning patterns – initiated from these cloud providers which leverage cloud instances spread around multiple countries. Blocking at larger prefix sizes than ones assigned to individual cloud instances can lead to collateral damage i.e., internet traffic from legitimate cloud instances being blocked. We also observed under-blocking of IPv6 prefixes belonging to sources of the *Internet Scanner* category. This method suggested that we block only 12 /112 prefixes from a large /32 IPv6 prefix dedicated to internet scanning. Furthermore, this method also failed to aggregate sources from *AlphaStrike* labs which used more than 180k unique source /128s to send scanning packets. As AlphaStrike distributed their scanning campaign evenly throughout their large /30 covering prefix – by targeting only 1 unique destination for each /128 source – it was able to evade even an aggregate based blocklisting solution.

8 CONCLUSION

In this work, we find that deploying a proactive telescope is essential to capturing a more representative breadth of IPv6 scanner behaviors. Capturing these behaviors is indispensable to examining the efficacy of our current implementations of tools developed to protect IPv6 networks. Throughout our analysis, we find that these tools are not sufficiently reliable for robust IPv6 network protection. We emphasize that developing methods to deterministically uncover the dynamic network-specific IPv6 address allocation practices is essential to moving this space forward. We will make our tools in the form of code and deployment instructions available to the research community.

REFERENCES

[1] AbuseIPDB. https://www.abuseipdb.com. Accessed: 2024-05-16.

[2] Certificate transparency. https://certificate.transparency.dev.

[3] Crowdsec's new cybersecurity majority report highlights the rise of ipv6 in cybercriminal activities | cybersecurity dive. https://www.cybersecuritydive.com/press-release/20230727-crowdsecs-new-cybersecurity-majority-report-highlights-the-rise-of-ipv6-in-2/. (Accessed on 05/16/2024).

[4] GoDaddy. https://www.godaddy.com.

[5] internet-measurement.com. https://internet-measurement.com/. (Accessed on 05/16/2024).

[6] Ipv6 threats are on the rise as adoption grows - sdxcentral. https://www.sdxcentral.com/articles/analysis/ipv6-threats-are-on-the-rise-as-adoption-grows/2023/08/. (Accessed on 05/16/2024).

[7] The ZMap Project. https://zmap.io/, 2024. Accessed 2024-05-15.

[8] Anonymized. Anonymized. In *ACM Internet Measurement Conference*.

[9] Anonymized. Anonymized for peer review.

[10] Assetnote. Commonspeak2. https://github.com/assetnote/commonspeak2-wordlists/, 2018.

[11] Shehar Bano, Philipp Richter, Mobin Javed, Srikanth Sundaresan, Zakir Durumeric, Steven J Murdoch, Richard Mortier, and Vern Paxson. Scanning the internet for liveness. *ACM SIGCOMM Computer Communication Review*, 48(2):2–9, 2018.

[12] bitquark. DNSpop. https://github.com/bitquark/dnspop, 2016.

[13] CAIDA. RouteViews Prefix to AS mappings. https://catalog.caida.org/dataset/routeviews_prefix2as.

[14] cowrie. Cowrie honeypot. https://github.com/cowrie/cowrie, 2023.

[15] Jakub Czyz, Kyle Lady, Sam G Miller, Michael Bailey, Michael Kallitsis, and Manish Karir. Understanding ipv6 internet background radiation. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 105–118, 2013.

[16] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. {ZMap}: Fast internet-wide scanning and its security applications. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 605–620, 2013.

[17] Matthew Ford, Jonathan Stevens, and John Ronan. Initial results from an ipv6 darknet13. In *International Conference on Internet Surveillance and Protection (ICISPâĂŹ06)*, pages 13–13. IEEE, 2006.

[18] Kensuke Fukuda and John Heidemann. Who Knocks at the IPv6 Door?: Detecting IPv6 Scanning. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, pages 231–237, New York, NY, USA, 2018. ACM.

[19] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, pages 364–378, New York, NY, USA, 2018. ACM.

[20] Fernando Gont and Guillermo Gont. Implications of IPv6 Addressing on Security Operations. Internet-Draft draft-ietf-opsec-ipv6-addressing-00, Internet Engineering Task Force, June 2023. Work in Progress.

[21] Google. pcapgo. https://pkg.go.dev/github.com/google/gopacket/pcapgo, 2020.

[22] Google. IPv6 Adoption. https://www.google.com/intl/en/ipv6/statistics.html, 2024. Accessed 2024-05-15.

[23] Raphael Hiesgen, Marcin Nawrocki, Alistair King, Alberto Dainotti, Thomas C. Schmidt, and Matthias W ahlisch. Spoki: Unveiling a New Wave of Scanners through a Reactive Network Telescope. In *Proceedings of USENIX Security Symposium*, 2022. Accessed: 2023-2-14.

[24] R. Hinden and S. Deering. IP Version 6 Addressing Architecture. Internet Requests for Comments, February 2006. http://www.rfc-editor.org/rfc/rfc4291.txt.

[25] Geoff Huston and Mirjam Kuhne. Background radiation in ipv6. *The ISP Column, APNIC*, 2010.

[26] Lukas Kramer, Johannes Krupp, Daisuke Makita, Tomomi Nishizoe, Takashi Koide, Katsunari Yoshioka, and Christian Rossow. AmpPot: Monitoring and Defending Amplification DDoS Attacks. In *Proceedings of the 18th International Symposium on Research in Attacks, Intrusions and Defenses*, 2015.

[27] Let's Encrypt. Rate limits. https://letsencrypt.org/docs/rate-limits/, October 2021.

[28] Let's Encrypt. Challenge types. https://letsencrypt.org/docs/challenge-types/, February 2023.

[29] Linda Markowsky and George Markowsky. Scanning for vulnerable devices in the internet of things. In *2015 IEEE 8th International conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, volume 1, pages 463–467. IEEE, 2015.

[30] MaxMind. MaxMind GeoIP® databases. https://www.maxmind.com/en/geoip-databases, 2024.

[31] Daniel Miessler. Combined subdomains. https://github.com/danielmiessler/SecLists/tree/master/Discovery/DNS, 2023.

[32] Ramakrishna Padmanabhan, John P Rula, Philipp Richter, Stephen D Strowes, and Alberto Dainotti. Dynamips: Analyzing address assignment practices in ipv4 and ipv6. In *Proceedings of the 16th international conference on emerging networking experiments and technologies*, pages 55–70, 2020.

[33] David Plonka and Arthur Berger. Temporal and spatial classification of active ipv6 addresses. In *Proceedings of the 2015 Internet Measurement Conference*, pages 509–522, 2015.

[34] rbsec. dnscan. https://github.com/rbsec/dnscan, 2022.

[35] Philipp Richter, Oliver Gasser, and Arthur Berger. Illuminating large-scale ipv6 scanning in the internet. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 410–418, 2022.

[36] Khwaja Zubair Sediqi, Lars Prehn, and Oliver Gasser. Hyper-Specific Prefixes: Gotta Enjoy the Little Things in Interdomain Routing. *ACM SIGCOMM Computer Communication Review*, 52, June 2022.

[37] Guanglei Song, Jiahai Yang, Lin He, Zhiliang Wang, Guo Li, Chenxin Duan, Yaozhong Liu, and Zhongxiang Sun. {AddrMiner}: A comprehensive global active {IPv6} address discovery system. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, pages 309–326, 2022.

[38] Hammas Bin Tanveer, Rachee Singh, Paul Pearce, and Rishab Nithyanand. Glowing in the dark: Uncovering {IPv6} address discovery and scanning strategies in the wild. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6221–6237, 2023.

[39] Deutsche Telekom Security GmbH. T-Pot - The all in one multi honeypot platform. https://github.com/telekom-security/tpotce.

[40] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. Internet Requests for Comments, September 2007. http://www.rfc-editor.org/rfc/rfc4862.txt.

[41] Liang Zhao, Satoru Kobayashi, and Kensuke Fukuda. Exploring theÂădiscovery process ofÂăfresh ipv6 prefixes: An analysis ofÂăscanning behavior inÂădarknet andÂăhoneynet. In *Passive and Active Measurement Conference*, pages 95–111, 2024.

[42] Maya Ziv, Liz Izhikevich, Kimberly Ruth, Katherine Izhikevich, and Zakir Durumeric. ASdb: a system for classifying owners of autonomous systems. In *Proc. ACM IMC*, 2021.

A ETHICS

This work does not raise ethical concerns.

B EXCLUDING SINGLE-REPORTER IP ADDRESSES FROM THE ABUSE REPORTS

Figure 8 depicts an empirical culmulative distribution of the number of reporters per IP address. To investigate if IP addresses reported by multiple users differ in their behavior, we rerun our analysis after removing the 92% of IP addresses with only a single reporter from our dataset. The resulting filtered dataset contains 483 129 abuse reports about 19 674 unique IPv6 addresses submitted by 995 reporters. For analysis of port scan reports (cf. Section 7.1), we extract destination ports form from 63% of all port scan reports in the filtered dataset (submitted by 28 users), giving us the target ports of 4053 reported scanner addresses.

Tables 8 and 9 correspond to Tables 4 and 5 in Section 7.1, respectively. While the top six categories (Table 8) remain the same, the number of addresses in each decrases and and four categories (SSH and Web crawling, Spam and Port scans) switch ranks. Regarding the top ports (Table 9) we once again observe that scanners focus on SSH (22), FTP (21), while HTTP (80) and UDP traceroutes (3345) are also popular in larger aggragations.

Table 8: AbuseIPDB: Top types of reported IPv6 activity, excluding IP addresses only reported by a single user.

Rank	Activity	#Reporters	#Reports	Reported IP addresses			
				ASes	/48s	/64s	/128s
#1	WordPress	274	121 168	745	3554	5921	9106
#2	SSH	227	14 327	58	122	126	1757
#3	Web crawling	220	35 924	490	2045	3630	6560
#4	Mail	128	7330	116	261	401	2638
#5	Port scan	80	246 469	955	2302	2655	6834
#6	Spam	79	7497	89	190	271	1691

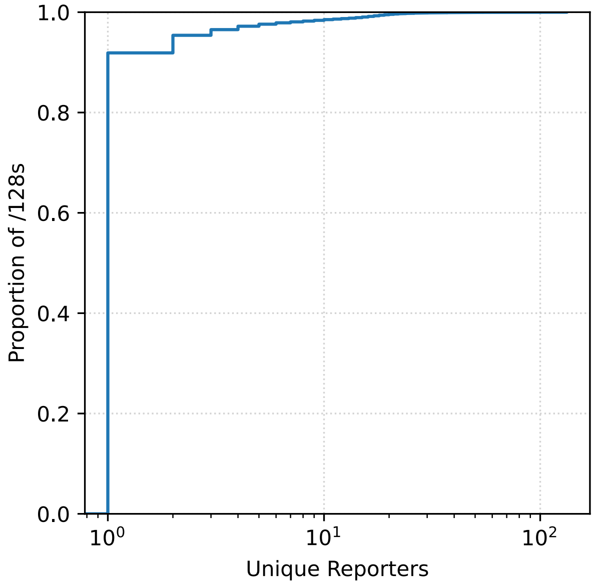


Figure 8: ECDF plot of the number of reporters per IP address.

Table 9: Top destination ports reported in AbuseIPDB, excluding IP addresses only reported by a single user.

Rank	Ranking					
	by #/128s		by #/64s		by #ASes	
	Port	#/128s↓	Port	#/64s↓	Port	#ASes↓
#1	21	1937	80	80	80	35
#2	22	1848	21	67	33435	25
#3	8080	1645	22	64	22	23
#4	23	1587	3389	53	443	23
#5	5900	1518	33435	50	21	21

C LONGITUDINAL ANALYSIS OF REPORT CATEGORIES

We perform a longitudinal analysis of the individual report categories to identify possible large-scale events affecting multiple users. Figure 9 shows the weekly number of users submitting reports for the top three categories.

In the WordPress and web crawling categories, we notice that—apart from some minor fluctuations—the number of reporters remains fairly constant over time. We observe similar curves for most of the other top six categories (not shown). One notable exception is a conspicuous rise in the number of users reporting SSH connection attempts during the third week of December 2023. During this event, users were mainly reporting two IP addresses from a well-known

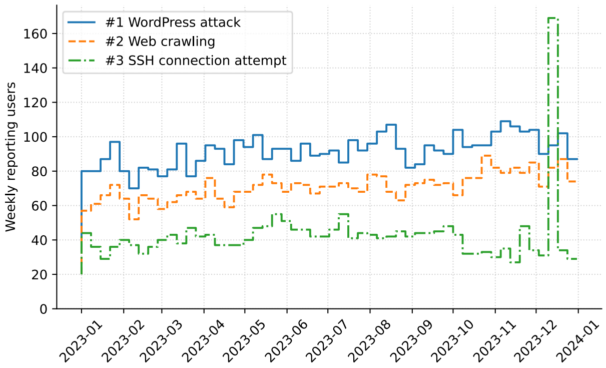


Figure 9: Weekly reporters on AbuseIPDB for the top three IPv6 categories. We observe a sharp increase in the number of users reporting SSH connection attempts during the third week of December 2023.

research institution. After reaching out to the organization in question, they confirmed to us that they had indeed been conducting active measurements involving SSH servers at the time.

D T-POT INFRASTRUCTURE

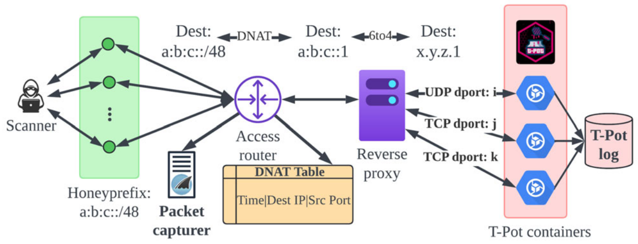


Figure 10: Overview of IPv6-enabled T-Pot infrastructure.

E BREAKDOWN OF SCANNER SOURCES IN CDN BY COUNTRY AND NETWORK TYPE

Also of interest is the country of origin and network type from which the scans are initiated. Table 10 shows the top 20 ASes, ordered by number of packets. For countries, United States and China dominate. Two of the ASes belong to cybersecurity companies. If the scanner has ill-intent, they could likely use cloud service providers or datacenters (though of course others would also be using these platforms) and these platforms are the most popular. Compared to an earlier study by Richter *et. al* [35], covering 15 months starting January 2021, the scan traffic reported here is much more dispersed. In Table 10 the top AS accounted for 18% of the packets across three /64's, whereas earlier, the top three /64's accounted for 87% of the packets.

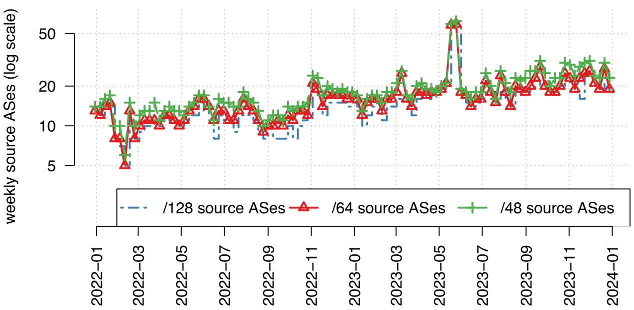


Figure 11: Weekly number of source ASes of the IPv6 scans.

			scan sources		
rank	**AS type**	**packets**	**/48s**	**/64s**	**/128s**
#1	Transit (global)	4.68B (17.6%)	1	3	2745
#2	Datacenter (CN)	4.08B (15.4%)	10	12	45
#3	Cybersecurity (US)	3.74B (14.1%)	7	7	367
#4	Datacenter (US)	3.17B (12.0%)	1	1	11
#5	Cloud (CN)	2.60B (9.8%)	15	17	310
#6	Cloud (CN)	2.42B (9.1%)	6	7	36
#7	Datacenter (CN)	1.72B (6.5%)	2	2	11
#8	Cloud (US/global)	899M (3.4%)	35	43	3312
#9	Cloud (US/global)	833M (3.1%)	4	4	53
#10	Datacenter (CN)	609M (2.3%)	1	1	4
#11	Cloud (US/global)	533M (2.0%)	12	12	2277
#12	Cloud (US/global)	392M (1.5%)	12	19	4475
#13	Cloud (US/global)	360M (1.4%)	22	22	41
#14	Cloud (US/global)	228M (0.9%)	7	7	21
#15	Cybersecurity (US)	91M (0.3%)	2	2	198
#16	Datacenter (CN)	44M (0.2%)	32	138	142
#17	Cloud (US)	28M (≤0.1%)	1	1	2
#18	University (CN)	20M (≤0.1%)	1	2	2
#19	Datacenter (CA)	14M (≤0.1%)	1	1	1
#20	Research (DE)	14M (≤0.1%)	1	1	1

Table 10: Top 20 source ASes by scan packets over the entire measurement window (packets shown for /64 source aggregation).

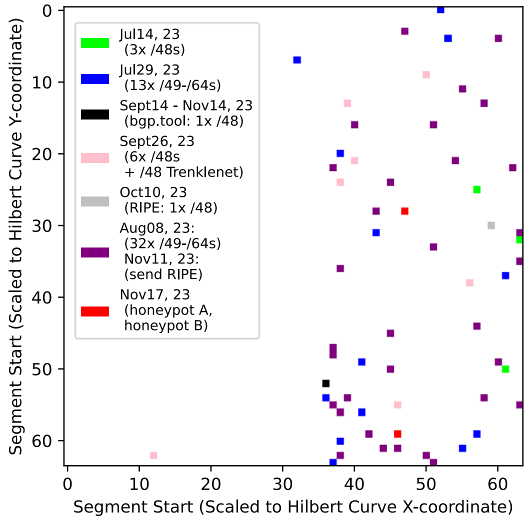


Figure 12: Location of honeyprefixes in the ISP A's network.

Table 11: Honeypot containers we deployed in our T-Pot instance and the corresponding ports listened. *@@echo: can you double check and add citation to individual pot?*

Honeypots	Protocol (Dest ports)	$\mathcal{H}_{TPot1}$	$\mathcal{H}_{TPot2}$
cowrie	TCP (22-23)	✓	✗
mailoney	TCP (25)	✓	✓
snare	TCP (80)	✓	✓
citrixhoneypot	TCP (443)	✓	✓
ciscoasa	UDP (5000), TCP(8443)	✓	✓
redishoneypot4	TCP (6379)	✓	✗
adbhoney	TCP (5555)	✓	✓
sentrypeer	UDP (5060)	✗	✓
dionaea	TCP (20-21, 42, 81, 135, 443, 445, 1433, 1723, 1883, 3306, 27017), UDP (69)	✓	✓
ddospot	UDP (19, 53, 123, 161, 1900)	✓	✓
conpot_kamstrup_382	TCP (1025, 50100)	✗	✓
elasticpot	TCP (9200)	✗	✓
dicompot	TCP (11112)	✗	✓