Who Knocks at the IPv6 Door?
Detecting IPv6 Scanning

Kensuke Fukuda
National Institute of Informatics/Sokendai

John Heidemann
USC/Information Sciences Institute

ABSTRACT

DNS backscatter detects internet-wide activity by looking for common reverse DNS lookups at authoritative DNS servers that are high in the DNS hierarchy. Both DNS backscatter and monitoring unused address space (darknets or network telescopes) can detect scanning in IPv4, but with IPv6's vastly larger address space, darknets become much less effective. This paper shows how to adapt DNS backscatter to IPv6. IPv6 requires new classification rules, but these reveal large network services, from cloud providers and CDNs to specific services such as NTP and mail. DNS backscatter also identifies router interfaces suggesting traceroute-based topology studies. We identify 16 scanners per week from DNS backscatter using observations from the B-root DNS server, with confirmation from backbone traffic observations or blacklists. After eliminating benign services, we classify another 95 originators in DNS backscatter as potential abuse. Our work also confirms that IPv6 appears to be less carefully monitored than IPv4.

CCS CONCEPTS

• **Networks** → **Network measurement**; **Network security**;

KEYWORDS

IPv6, Scanning, DNS backscatter

ACM Reference Format:
Kensuke Fukuda and John Heidemann. 2018. Who Knocks at the IPv6 Door? Detecting IPv6 Scanning. In *Proceedings of 2018 Internet Measurement Conference (IMC '18).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3278532.3278553

1 INTRODUCTION

Active network scanning is a popular approach to study the Internet topology [7], the network edge [19], with applications including identification of Internet-of-Things devices [25], security risks [11] and mechanisms [10], and network reliability [26]. With multiple IPv4 scanning tools freely available today [11, 18, 21], IPv4 scanning is something anyone can do, and many groups are doing it [9, 23].

As IPv6 use grows, so does interest in carrying out and detecting IPv6 scanning. Yet IPv6 scanning is much more difficult, because the much larger address space size (2^{128} instead of only 2^{32}) makes brute-force enumeration impossible. Optimizations to search the IPv6 address space are an open area of research [1, 2, 12, 13, 16, 17, 24]. Yet little is known about scanning in IPv6 today.

One thing we know about IPv6 is that it is new, and one consequence of this novelty is that often it is not as carefully secured as IPv4, as shown by Czyz et al. [8]. Their study was made with dual-stack IPv4 and IPv6 computers, but our study includes additional sources allowing broader IPv6 detection.

DNS backscatter detects network-wide events by watching for frequent, common reverse DNS names [14]. Although developed for IPv4, its sensitivity depends on traffic triggered by network wide events (not address space size), so DNS backscatter holds promise for IPv6.

The goal of this paper is to adapt DNS backscatter to IPv6 and to use it to understand IPv6 scanning and security. Our first contribution is to show how DNS backscatter can be adapted from IPv4 to IPv6, taking care in filtering out network services that also cause DNS backscatter (§2). Although similar to IPv4, we use a different (simpler) classification to accommodate lower amounts of backscatter in IPv6. Second, we use IPv6 DNS backscatter to confirm that IPv6 security policies are weaker than IPv4 (§3), reexamining prior evaluation of dual-stack hosts [8]with probe measurements across all IPv6 hosts. Our final contribution (§4) is to study what IPv6 DNS backscatter finds over six months of data observed at B-root DNS server. We detect a number of major internet services (cloud providers, NTP operators), and lookups of router interfaces due to traceroute-driven topology studies. More importantly, we find 16 active IPv6 scanners per week, seen in DNS backscatter and confirmed in backbone packet traces or blacklists. We also find another 95 potential abuse cases not seen in traces or darknets, suggesting the importance of backscatter for IPv6. Our observations suggest that scanning rates are growing slowly over six months.

2 DNS BACKSCATTER AS IPV6 SENSOR

We first review how DNS backscatter works from prior work, then describe how we adapt it to IPv6 and classify originators.

2.1 Background on DNS Backscatter

We first summarize DNS backscatter from prior work [14].

Consider a network scanner (the *originator*) who sends probes to a number of hosts (the *targets*) in the IPv6 Internet. Firewalls on or in front of some of these targets investigate probe packets, looking up the reverse DNS name of the probe's source IP address. This DNS query is done by the recursive resolver (the *querier*) for the firewall, and is ultimately handled by the authoritative server (the *authority*) that is responsible for the originator's reverse address in ip6.arpa. Depending on caching, this query may also be seen at other authorities higher in the DNS hierarchy.

DNS backscatter is the process of observing originators (IPv6 addresses) that occur frequently from many queriers. These are

IMC '18, October 31-November 2, 2018, Boston, MA, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5619-0/18/10…$15.00
https://doi.org/10.1145/3278532.3278553

backscatter detections that represent some kind of widespread network activity from the originator. DNS backscatter is attenuated by caching, and the degree of attenuation depends on where in the hierarchy the authority is. Thus, although attenuation makes it difficult to quantify the size of scans, in principle, many network-wide events can be observed in reverse queries at a root DNS server.

Many types of events trigger DNS backscatter; some of them are benign, such as CDNs, cloud providers or large services (Google, Microsoft, Facebook), and NTP servers. Others are malicious or potentially malicious, including network scanners and spammers.

2.2 DNS Backscatter in IPv6

We describe the procedure to characterize network-wide events using authoritative DNS server logs.

We first extract reverse IPv6 address queriers from the DNS logs and group queriers per originator. We discard querier-originator pairs where all queriers and the originator belong to the same Autonomous System (AS) because such activities are local to that AS and not network-wide activity. We aggregate data over some duration d, then report cases where there are more than a detection threshold q queriers in that period. We use 7 days for d and 5 distinct queriers for q, values chosen based on our comparison of observations to ground truth (see also below). The result is a list of significant queriers per originator in d days.

Finally, we apply classification, as described in §2.3, to originators in the list. The result is network services (mostly benign) and a few that are potentially abuses. Finally, we check potential abuse (originator IP addresses that do not match any of our benign classes) to DNS-based black lists (spam and scan) and other ground truth data of anomalous activities to confirm, as described in §4.1.

Our classification procedure and the parameters for duration and threshold all differ from IPv4 because DNS backscatter is less frequent in IPv6 than in IPv4. The IPv6 duration and threshold (d of 7 days and q of 5 queriers) are both laxer than IPv4, where $d = 1$ and $q = 20$ [14]. In preliminary investigations using the IPv4 parameters we did not detect any ground truth scans (Table 5). This absence in IPv4 likely results from fewer present targets, and less logging per target. Thus for IPv6 we adopt larger d and smaller q. Since one target can trigger multiple queriers (due to multiple recursive resolvers), we tune q to capture network events related to more than one target. Should future IPv6 responses grow (due to greater logging per target, or perhaps due to improved scanning heuristics), it may be possible to use paramters and ML techniques as we used for IPv4.

2.3 Originator Classification in IPv6

We next describe our heuristics to classify originators. Originator are assigned to the first class they match.

major service Big application servers, including Facebook, Google, Microsoft, Yahoo. Determined by AS numbers.

cdn CDN infrastructure, including Akamai, Cloudflare, Edgecast, CDN77, Fastly. Determined by AS number or name suffix.

dns nameservers like ns.example.com. Determined by keywords in name: cns, dns, ns cache, resolv, name. We also rely on root.zone file for authoritative servers. Additionally, we find other dns servers by sending DNS queries to originators.

ntp NTP servers like ntp.example.com. Determined by keywords (ntp, time) in name, and by crawling IP addresses in pool.ntp.org (4.8k IPs).

mail mail servers like mail.example.com. Determined by keywords in name: mail, mx, smtp, post, correo, poczta, send, lists, newsletter, spam, zimbra, mta, pop, imap.

web web servers, determined by keyword (www) in name www.example.com.

tor tor servers, as appear in https://www.dan.me.uk/torlist/ (1.2k IPs).

other service Other application servers, e.g., push services, VPN services. Determined by name suffix.

iface router interfaces. Determined by interface or location in name (like ge0-lon-2.example.com), or by presence in the publicly available IPv6 topology data provided by CAIDA [4].

near-iface router interfaces inferred by following conditions: (1) all queriers belong to the same AS name, and (2) the originator's AS provides transit to querier's AS. These queriers are doing many traceroutes traversing a common link, and these are inferred to be interfaces are near the traceroute source. (If DNS confirmed them as interfaces they would be just "iface", but they either lack reverse DNS or it is not recognizable.)

qhost quasi-hosts—inferred to be edge devices seen in several ISPs, where the originator has no reverse name and all queriers are end-hosts in one AS (i.e., /64 randomized IPs or automatically assigned names like home-1-2-3-4.example.com. We believe these represent some software running on customer-provided equipment.

tunnel IP addresses for IPv4/v6 tunneling: Teredo [20] (2001::/32) and 6to4 [5] (2002::/16).

scan Confirmed scanners, as determined by appearance in blacklists: https://www.abuseipdb.com or https://access.watch, or in backbone traffic data (see also §4.1).

spam Confirmed spammers, as determined by appearance in either of DNSBLs: sbl.spamhaus.org, all.s5h.net, dnsbl.beetjevreemd.nl.

Different from our prior work on DNS backscatter in IPv4 [14], we directly infer the class of originator instead of using machine learning (ML) techniques. We shift away from ML because the number of queriers is much smaller, so the dataset is too small for effective classification with ML. However, our IPv6 rules include discriminative features similar to those we used in IPv4's ML-based classification, such as keywords, geolocation diversity and similarity of querier's IPs shown in near-iface and qhost. In fact, non-matched originators are queried from queriers spread in multiple ASes by definition.

We evaluate this classification in §4. As with prior work, some rules are forgeable. For example, rules that use domain names will misclassify if scanning is done from mail.example.com. As IPv6 use increases, more backscatter will allow use of more robust rules and potentially machine learning, as we used for IPv4 [14].

3 REACTIVITY OF IPV6 HOSTS TO SCANS

We first use controlled experiments to show that IPv6 hosts react less frequently to scanning than IPv4 hosts.

Label	# addrs	Description
Alexa	10k	Alexa 1M; servers
rDNS	1.4M	Reverse DNS
P2P	40k	P2P Bittorrent; clients

Table 1: IPv4/IPv6 hitlists

3.1 Methodology

To understand DNS backscatter as a sensor, we first consider how often IPv6 hosts react to scanning compared to IPv4 hosts.

To answer this question we scan IPv6 ourselves and observe the response. Following prior work in IPv6 scanning [8, 16], we harvest IPv4 and IPv6 hitlists from three sources, as listed in Table 1: *Alexa*: we resolve Alexa 1M domains and pick up domains that have both IPv4 and IPv6 addresses. *rDNS*: we scan the IPv4 reverse DNS map and list all names that also have IPv6 addresses (following [16]). *P2P*: we crawl IPv4 and IPv6 addresses in a DHT-based BitTorrent network for a month. We expect Alexa to represent servers, P2P clients, and rDNS to have both. Alexa and rDNS hosts are names that bind to both v4 and v6, but for P2P we do not have pairs of addresses. We crawl many more IPv4 addresses than have IPv6 in P2P. We normalize the sizes of the two sets by randomly sampling IPv4 addresses from the set to match the number of IPv6 addresses.

We set up an IPv4 (with ZMap) and IPv6 (with a custom scanner) network scanner. The scanner sends a packet to each target IP, then records corresponding reply packets. It probes multiple application ports (ICMP echo, HTTP, ssh, DNS, and NTP). We also prepare a local authoritative DNS server for monitoring queriers querying reverse lookups of the scanner's IP address. The TTL of the PTR record is set to 1 second at the authority to minimize caching effects[1]. For IPv6, we embed target IPv6 information to the source IP address of the scanner, allowing us to track correspondence between the target IP we scan and any DNS backscatter triggered by that scan. (Backscatter is sent from the querier, the recursive resolver of the target, so without this embedding we must guess the target.)

For IPv4, there is only one source IPv4 address for the scanner and thus we cannot directly pair replies to requests. Instead, we count total replies over the 24 hours following a scan. Our data from IPv6 confirms that this period will cover 99% of DNS backscatter that will be generated. We also exclude resolvers that appear in our DNS logs in weeks before our experiments as background noise. These include shodan.io, he.net, and Google's crawlers.

3.2 Comparing Backscatter: IPv4 and IPv6

We now compare IPv4 and IPv6 response to scanning using the methodology we just described. DNS backscatter is caused by reverse DNS queries from the target or middleboxes, typically due to security policies that investigate or log traffic. Comparing DNS backscatter between v4 and v6 will therefore highlight any differences in security policy. Prior work has shown IPv6 security is often more lax [8], so our study will reevaluate that result.

Figure 1 shows the amount of DNS backscatter that results from scans using each of our three target lists (Alexa, P2P, and rDNS). Colors and labels indicate particular target lists, while squares show the IPv6 version and circles the IPv4 version. For reference, we also provide observations for scans of random IPv4 addresses (data from Figure 4 from [14]) and a projected fit along the diagonal.

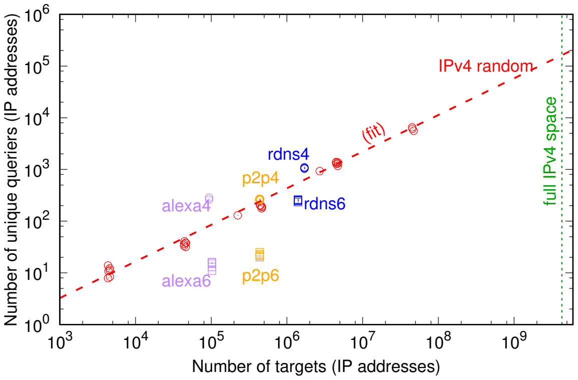

Figure 1: DNS backscatter sensitivity

Comparing the Alexa and rDNS datasets, we see that the IPv4 version of each target list produces about 10× more DNS backscatter than IPv6. This confirms that *IPv6 appears to be less heavily logged than IPv4*. An alternative hypothesis is that our target lists are unusual in some way, but, if anything, Alexa4 and rDNS4 are more heavily monitored than random IPv4 addresses, since they are above the dashed line fitting DNS backscatter resulting from random probing.

Finally, the P2P6 dataset is even more below the IPv4 baseline (for its size) than Alexa6 or rDNS6. While Alexa6 and rDNS6 generally represent servers, P2P6 represents clients, so one possible explanation is that clients are even less monitored in IPv6 than servers, perhaps due to very wide use of ephemeral IPv6 addresses.

3.3 Applications and Backscatter in IPv6

To better understand how prevelant monitoring is in IPv6 and how that affects DNS backscatter, we next look at scans to specific application ports. We evaluate applications in two steps: we establish a baseline response rate. We then compare DNS backscatter that results from scans on different ports.

For both experiments we scan targets from the rDNS hitlist (our largest list), then evaluate how often we see an expected reply (for example, an ICMP echo reply in response to an echo request), an unexpected reply (for example, ICMP destination unreachable), or lack of reply.

Direct scans: Table 2 shows the results of direct scans of five different application ports. As expected, the fraction of replies varies by application, with the most replies from ICMP (62.9%) and fewest from NTP (4.7%). These results are consistent with prior application scans (for example, [8]), although they show that our target lists have slightly higher response rate than random scanning.

Our IPv4 reply rate is also about the same as the v6 rate.

Backscatter: Having established that our target list is typical, we next consider what DNS backscatter triggered by these scans shows. Our goal is to understand what DNS backscatter sees of our scans, so we can evaluate what backscatter shows of *other* scanners.

[1] We do not have enough knowledge on the distribution of originator's TTLs in the wild, however, our controlled experiment setting expects to yield the highest number of queriers.

type	**icmp6** (ping)		**tcp22** (ssh)		**tcp80** (web)		**udp53** (DNS)		**udp123** (NTP)	
queries				…	1476509	100%	…			
expected reply	928953	62.9%	410421	27.8%	661182	44.8%	69965	4.7%	140893	9.5%
other reply	145264	9.8%	205446	13.9%	201627	13.7%	672171	45.5%	371044	25.1%
no reply	402292	27.2%	860642	58.3%	613700	41.5%	734373	49.4%	964572	65.3%
expected v4 reply	-	57.8%	-	30.0%	-	35.4%	-	6.3%	-	5.9%

Table 2: Scan results overview (rDNS): expected reply is the number of expecting unique replies, e.g., ICMP echo reply for ICMP echo. Other reply is unexpected one, e.g., ICMP destination unreach., and no reply is lack of reply.

	icmp6 (ping)		**tcp22** (ssh)		**tcp80** (web)		**udp53** (DNS)		**udp123** (NTP)	
v6 backscatter	1809	(0.12%)	774	(0.05%)	1020	(0.07%)	653	(0.04%)	746	(0.05%)
w/expected reply	1371	75.8% (0.09%)	365	47.2% (0.03%)	597	58.5% (0.04%)	137	21.0% (0.01%)	134	18.0% (0.01%)
w/other reply	44	2.4% (0.002%)	94	12.1% (0.006%)	87	8.5% (0.006%)	265	40.6% (0.02%)	183	24.5% (0.01%)
w/no reply	394	21.8% (0.03%)	315	40.7% (0.02%)	336	32.9% (0.02%)	251	38.4% (0.02%)	429	57.5% (0.03%)
v4 backscatter	4478	(0.30%)	2731	(0.18%)	3094	(0.21%)	3961	(0.27%)	4045	(0.27%)

Table 3: DNS backscatter and application behavior (rDNS).

The relationship between DNS backscatter and applications is not obvious, because DNS backscatter typically results from logging, and the choice to log (or not) depends on perceptions of protocol sensitivity.

Table 3 shows DNS backscatter detections and their yield, how many replies are seen relative to different types of replies (expected, other, or no reply). Yield is small, varying from 0.12% for ICMP echo (icmp6) to 0.04% for DNS (udp53), consistent with limited monitoring in IPv6, compared to monitoring DNS backscatter in IPv4 (0.2-0.3%). The amount of DNS backscatter depends on the protocol. For common protocols like icmp6 and web, we see more DNS backscatter from IP addresses that give the expected reply (for example, icmp6 shows 0.09% yield for IP addresses that return an echo reply). For less common protocols like DNS and NTP, DNS backscatter is more common for hosts that do not reply to the protocol, suggesting organizations that are logging traffic to closed ports.

4 FINDING SCANNING ACTIVITY IN IPV6 DNS BACKSCATTER

Having established DNS backscatter for IPv6, we next examine how much scanning we see in six months of data observed at B-Root DNS.

4.1 Datasets

We use B-Root DNS to find DNS backscatter, then confirm scanners against data from the MAWI backbone and the NII darknet.

DNS backscatter: We extract all reverse DNS for IPv6 as seen at B-Root from July to December 2017. Original data is full capture, but with occasional packet loss during very busy periods. We use both UDP and TCP queries. We see 31M unique querier-originator pairs, 435k unique queriers, and 29M unique IPv6 originators over this time.

Backbone traffic: To confirm scanners we use MAWI traffic traces [6] that are captured at a transit link of AS2500 (WIDE) from June 2017 to March 2018. Data is a sample taken for 15 minutes at 2pm JST each day. We extract IPv6 packets from the mixed unanonymized v4 and v6 trace. We see about 7M IPv6 packets in each day's sample.

Followed by a heurestic classifier [22] for MAWI data, we define a network scanner as a source IPv6 address that (1) has five or more destination IPs, (2) all going to a common destination port, (3) with, on average, fewer than ten packets per destination IP, and (4) the entropy of packet length is smaller than 0.1. The last criterion helps distinguish network scans from DNS resolvers because DNS resolvers query a wide variety of QNAMEs. These criteria are conservative to reduce false positives.

Darknet traffic: We also use darknet data to confirm scanners. Darknets are network address blocks that are routed, but that have no hosts in them, so traffic that arrives is likely not benign (instead it is scanning, DoS reflection, misconfiguration, etc.). We operate a /37 IPv6 darknet from June 2017 to March 2018. We announce it with a different AS (AS2907; SINET) than the backbone measurements to avoid measurement overlap. We capture 15k packets from 106 source IPs in this period.

4.2 Backscatter Detection

We next look at what DNS backscatter sees in IPv6 over six months: services, routers, and potential abuse. Table 4 gives the mean number of each group that appears per week over all six months.

We first show that DNS backscatter detects a variety of services: large service and cloud providers (Facebook, Google, and Microsoft) are prominent, as are CDNs. This result suggests services may consider increasing use of reverse DNS names in IPv6.

Well known services account for about 12% of DNS backscatter. Reverse name checks are part of validation of services such as NTP and SMTP.

	Count	**%**
Category	(mean/week)	**total**
Services:		
Content Provider	**4722**	**70.24**
Facebook	3653	54.34
Google	727	10.82
Microsoft	329	4.89
Yahoo	13	0.19
CDN	**286**	**4.25**
Well-known service	**815**	**12.12**
DNS	337	5.01
NTP	414	6.16
mail (SMTP)	42	0.62
web (HTTP)	22	0.33
Minor service	**268**	**3.99**
other services	83	1.23
qhost	185	2.75
Routers:		
Router	**288**	**4.28**
iface	256	3.81
near-iface	32	0.48
Tunnel	**216**	**3.21**
Teredo/6to4	207	3.08
tor	9	0.12
Potential Abuse:		
Abuse	**128**	**1.90**
spam	17	0.25
scan	16	0.24
unknown (potential abuse)	95	1.41
Total	**6723**	**100.00**

Table 4: Weekly average number of originators in each class for six month DNS backscatter data. (Indented values sum to their boldface parent.)

We also see a large number of routers and tunnel interfaces. We believe those interfaces appear as a result of traceroutes from topology studies. Traceroutes will look up the reverse names of each router hop, and carrying out traceroutes everywhere will look up the names of first few hops many, many times (even with caching); our near-iface definition captures this abundance. (This observation was confirmed by operators of a major ISP.) Tunnels and VPNs seem to often do reverse queries, presumably during setup.

Finally, the smallest but most important category is potential abuse. We see 17 spammers, 16 scanners, and 95 events that are consistent with scanning, on average per week. We discuss these cases in detail next.

4.3 Confirming Scanners

We next discuss 7 scanners we see in backbone and darknet data.

Completeness: We first compare DNS backscatter against backbone and darknet data. Backscatter provides wide-angle view that can see globally, but it only sees large events. Backbone and darknet data are both narrowly focused, seeing only events that traverse the backbone segment or send traffic to the darknet, but potentially more sensitive at detecting small scans.

First, we find four scanners in *both* DNS backscatter and MAWI backbone data: scanners (a) through (d) in Table 5. Only scanner (a) appears in darknet data.

Scanner (a) probes TCP port 80. It appears in MAWI on six days, but the intensity of DNS backscatter is not high (Parenthetic number in DNS BS indicates the number of weeks the originator appears at least once). Scanners (b) to (d) appear two times in DNS backscatter and also two days in MAWI. These results provide confirmation that DNS backscatter does see actual scanners.

This result also shows the limited effectiveness of darknets for IPv6: they can only see a tiny fraction of the vast IPv6 space, making DNS backscatter and traffic observation more important techniques in IPv6. Only scanner (a) appears in the darknet, MAWI, and DNS backscatter. Some of CAIDA's Archipelago measurements [3] appear only in the darknet.

Second, we see that DNS backscatter misses three of the scanners we see in MAWI (scanners e, f, and g). DNS backscatter only detects big events that generate many reverse queries, and these scanners are fairly brief (1 or 2 days seen in MAWI). In addition, scanners (e) through (g) target only a narrow range of IP blocks (i.e., a single /48), so DNS backscatter from many locations is unlikely. Thus, these scanners show that DNS backscatter will miss small scans.

Third, we see that there are 95 unknown (potential abuse) detections seen in backscatter data only. We suggest that these are potential scanners missed in MAWI and our darknet. In fact, in discussions with a researcher doing scanning we confirmed that we see scanner IPs in backscatter although not in MAWI, probably because their scans do not include enough hosts at the vantage point in the sampling time window (15 minutes per day) [15].

Scan types: A natural question is to ask what hitlists these known, detected scanners employ. Carefully checking target IP addresses of the scanners, we find three typical patterns. First, *rand IID*, IPs consisting of /64 prefix + small and random right most nibble in IID such as scanning 2001:db8:1::10, then 2001:db8:ff::10. For *rDNS*, IPs are those with reverse name registered in reverse DNS. Finally, *Gen* suggests use of a target generation algorithm. The hitlist of scanner (a) appears to use a target generation algorithm. This scanner originates from address space used by Murdock et al. [24] developers of one such algorithm; they confirmed that we detected their scanning. Scanners (b) and (c) are rand IID, but since they lack traffic in the darknet, we guess that they probe specific routed prefixes as seeds. On the other hand, scanners (d) through (g) rely on reverse names (rDNS), similar to our probes. In summary, we confirm that the detected scanners employ multiple types of hitlists.

Temporal correlation: To better understand the nature of IPv6 scanning we next investigate the temporal behavior of scanners (a) through (d) in both DNS backscatter and MAWI traffic. Figure 2 shows our six months of observations for each of these scanners. Each "x" is a detection in MAWI, and the bars show the number of queriers seen in DNS backscatter.

This comparison confirms that DNS backscatter successfully detects network-wide scans, since most scans seen in MAWI result in DNS backscatter. Queries for other isolated DNS backscatter suggest a possibility of network scans targeting other networks, or

	IP	MAWI			Backscatter	Dark	ASN	info
		#days	port	scan type	#weeks	#weeks		
(a)	2001:48e0:205:2::/64	6	TCP80	Gen	1 (5)	1	40498	New Mexico Lambda Rail
(b)	2a02:418:6a04:178::/64	2	ICMP	rand IID	2 (4)	0	29691	Nine, CH
(c)	2a02:c207:3001:8709::/64	2	TCP80	rand IID	2 (2)	0	51167	Contabo, DE
(d)	2a03:f80:40:46::/64	2	ICMP	rDNS	2 (3)	0	5541	ADNET-Telecom, RO
(e)	2405:4800:103:2::/64	2	ICMP	rDNS	0 (4)	0	18403	FPT-AS-AP, VN
(f)	2a03:4000:6:e12f::/64	1	ICMP	rDNS	0 (0)	0	197540	NETCUP-GmbH, DE
(g)	2800:a4:c1f:6f01::/64	1	ICMP	rDNS	0 (0)	0	6057	ANTEL, UY

Table 5: Observed IPv6 scanners in MAWI; /64 of IPs are anonymized. Scan types consist of Gen: target generation, rand IID: random and small right most nibble, and rDNS: reverse name registered.

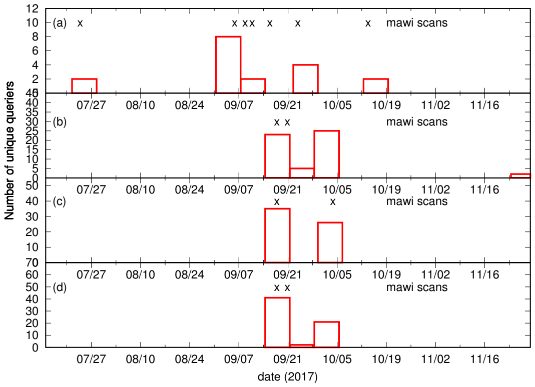

Figure 2: MAWI scans and DNS backscatter

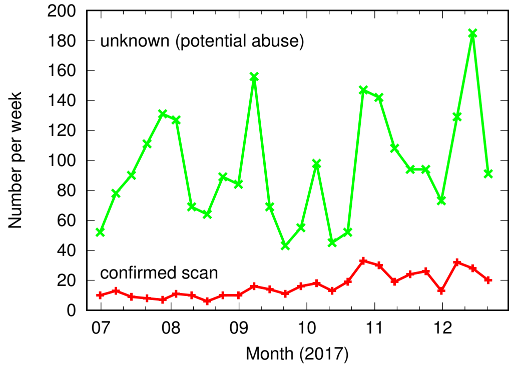

Figure 3: Number of scans and unknown (potential abuse) over time

scanning that does not occur in the brief fraction of the day our MAWI dataset provides.

4.4 Abuse Over Time

Finally, Figure 3 examines the trend in potential abuse originators over time. We see considerable variation in the unknown (potential abuse) category, and although the trend is slightly upward, it is very noisy.

However, we see a consistent, slow increase in confirmed scanners over time. Confirmed scanners increase from 8 originators in July to 28 in December. We are cautious in interpreting this trend, since three factors are relevant: greater scanning, greater classification of scanning, and generally greater use of IPv6. We do see this increase outpaces the general increase in all DNS backscatter over that time, which went from about 5000 to 8000 IPs over this period. However, the 3× increase in scanning is larger than the 60% increase in all DNS backscatter. Our count of confirmed scanners is based on seeing them some other source (MAWI or darknet or blacklist), so it is possible that we are just better at *confirming* scanners. However, we cautiously suggest that IPv6 scanning seems to be increasing over time.

5 CONCLUDING REMARKS

This paper adapts DNS backscatter to IPv6. DNS backscatter is able to detects many cloud providers, CDNs and services in IPv6, in addition to finding 16 confirmed scanners and about 95 unknown (potential abuses) per week. We also show that IPv6 scanning activity is increasing, and confirm that IPv6 is less closely monitored than IPv4. We believe DNS backscatter will be an important tool in IPv6, since approaches such as darknets are much less effective with IPv6's huge address space compared to IPv4.

ACKNOWLEDGMENTS

We thank Yuri Pradkin for B-Root data collection. We thank our shepherd Alan Mislove, the IMC reviewers, Yoshinobu Matsuzaki, Kenjiro Cho, Romain Fontugne, and Johan Mazel comments on this paper. We thank the researchers on IPv6 scanning who discussed their systems with us [15, 24].

Kensuke Fukuda's work is partially funded by JSPS KAKENHI Grant Number 18H03237.

John Heidemann's work was supported by the Office of the Director of National Intelligence (ODNI) and the Intelligence Advanced Research Projects Activity (IARPA) via the Air Force Research Laboratory (AFRL) contract number FA8750-16-C-0112, and the Department of Homeland Security (DHS) via AFRL agreements number FA8750-17-2-0280 and FA8750-17-2-0096. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, DHS, AFRL, or the U.S. Government.

REFERENCES

[1] Steve Bellovin, Bill Cheswick, and Angeleos Keromytis. 2006. Worm propagation strategies in an IPv6 Internet. *;Login:* 31, 1 (Feb. 2006), 70–76. https://www.cs.columbia.edu/~smb/papers/v6worms.pdf

[2] Kevin Borgolte, Shuang Hao, Tobias Fiebig, and Giovanni Vigna. 2018. Enumerating Active IPv6 Hosts for Large-scale Security Scans via DNSSEC-signed Reverse Zones. In *Proceedings of the 39th IEEE Symposium on Security and Privacy*. IEEE, San Francisco, CA, USA, 438–452. https://kevin.borgolte.me/files/pdf/sp2018-dnssec-ipv6.pdf

[3] CAIDA. 2009. Archipelago (Ark) Measurement Infrastructure. http://www.caida.org/projects/ark/. (2009).

[4] CAIDA. 2017. The CAIDA UCSD IPv6 Topology Dataset. http://www.caida.org/data/active/ipv6_allpref_topology_dataset.xml. (2017).

[5] Brian E. Carpenter and Keith Moore. 2001. Connection of IPv6 Domains via IPv4 Clouds. RFC3056. (2001).

[6] Kenjiro Cho, Koshiro Mitsuya, and Akira Kato. 2000. Traffic Data Repository at the WIDE Project. In *USENIX 2000 Annual Technical Conference: FREENIX Track*. USENIX, 263–270.

[7] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. 2009. Internet Mapping: from Art to Science. In *Proceedings of the IEEE Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH)*. IEEE, Alexandria, VA, USA, 205–211. https://doi.org/10.1109/CATCH.2009.38

[8] Jakub Czyz, Matthew Luckie, Mark Allman, and Michael Bailey. 2016. Don't Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy. In *Proceedings of the ISOC Network and Distributed System Security Symposium*. The Internet Society, San Diego, CA, USA. http://mdbailey.ece.illinois.edu/publications/ndss16_ipv6.pdf

[9] Zakir Durumeric, Michael Bailey, and J. Alex Halderman. 2014. An Internet-Wide View of Internet-Wide scanning. In *Proceedings of the 23rd USENIX Security Symposium*. USENIX, San Diego, CA, 65–78.

[10] Zakir Durumeric, James Kasten, Michael Bailey, and J. Alex Halderman. 2013. Analysis of the HTTPS Certificate Ecosystem. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Barcelona, Spain. http://conferences.sigcomm.org/imc/2013/papers/imc257-durumericAemb.pdf

[11] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *Proceedings of the USENIX Security Symposium*. USENIX, Washington, DC, USA, 605–620. https://www.usenix.org/system/files/conference/usenixsecurity13/sec13-paper_durumeric.pdf

[12] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Krugel, and Giovanni Vigna. 2017. Something From Nothing (There): Collecting Global IPv6 Datasets From DNS. In *Proceedings of the Passive and Active Measurement Conference*. Springer, Sydney, Australia, 30–46.

[13] Pawel Foremski, David Plonka, and Arthur Berger. 2016. Entropy/IP: Uncovering Structure in IPv6 Addresses. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Santa Monica, CA, USA. https://doi.org/10.1145/2987443.2987445

[14] Kensuke Fukuda, John Heidemann, and Abdul Qadeer. 2017. Detecting Malicious Activity with DNS Backscatter Over Time. *ACM/IEEE Transactions on Networking* 25, 5 (Aug. 2017), 3203–3218. https://doi.org/10.1109/TNET.2017.2724506

[15] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Boston, Mass., USA, to appear.

[16] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. In *Proceedings of the IFIP International Workshop on Traffic Monitoring and Analysis*. IFIP, Louvain La Neuve, Belgium, 1–7.

[17] Fernando Gont and Tim Chown. 2016. Network Reconnaissance in IPv6 Networks. RFC7707. (2016).

[18] Robert Graham, Paul McMillan, and Dan Tentler. 2014. Mass Scanning the Internet. Presentation at Defcon 22. (Aug. 2014). https://defcon.org/images/defcon-22/dc-22-presentations/Graham-McMillan-Tentler/DEFCON-22-Graham-McMillan-Tentler-Masscaning-the-Internet.pdf

[19] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. 2008. Census and Survey of the Visible Internet. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Vouliagmeni, Greece, 169–182. https://doi.org/10.1145/1452520.1452542

[20] Christian Huitema. 2006. Teredo: Tunneling IPv6 over UDP. RFC4380. (2006).

[21] Gordon Lyon. 1997. nmap. computer software at http://insecure.org/nmap/. (Sept. 1997). http://insecure.org/nmap/

[22] Johan Mazel, Romain Fontugne, and Kensuke Fukuda. 2014. A Taxonomy of Anomalies in Backbone Network Traffic. In *TRAC'14*. IEEE, Nicosia, Cyprus, 30–36. https://doi.org/10.1109/IWCMC.2014.6906328

[23] Johan Mazel, Romain Fontugne, and Kensuke Fukuda. 2017. Profiling Internet Scanners: Spatiotemporal Structures and Measurement Ethics. In *Proceedings of the IEEE International Conference on Traffic Monitoring and Analysis*. IFIP, Dublin, Ireland, 9. https://doi.org/10.23919/TMA.2017.8002909

[24] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. 2017. Target Generation for Internet-wide IPv6 Scanning. In *Proceedings of the ACM Internet Measurement Conference*. ACM, San Diego, CA, USA, 242–253. https://doi.org/10.1145/3131365.3131405

[25] Robert O'Harrow, Jr. 2012. Cyber search engine Shodan exposes industrial control systems to new risks. *The Washington Post* (June 3 2012). http://www.washingtonpost.com/investigations/cyber-search-engine-exposes-vulnerabilities/2012/06/03/gJQAIK9KCV_story.html

[26] Lin Quan, John Heidemann, and Yuri Pradkin. 2012. Trinocular: Understanding Internet Reliability Through Adaptive Probing. In *Proceedings of the ACM SIGCOMM Conference*. ACM, Hong Kong, China, 255–266.