Where Have All the Firewalls Gone? Security Consequences of Residential IPv6 Transition

Erik Rye
Johns Hopkins University
Baltimore, MD, USA
rye@jhu.edu

Dave Levin
University of Maryland
College Park, MD, USA
dml@cs.umd.edu

Robert Beverly
San Diego State University
San Diego, CA, USA
rbeverly@sdsu.edu

Abstract

IPv4 NAT has limited the spread of IoT botnets considerably by default-denying bots' incoming connection requests to in-home devices unless the owner has explicitly allowed them. As the Internet transitions to majority IPv6, however, residential connections no longer require the use of NAT. This paper therefore asks: has the transition from IPv4 to IPv6 ultimately made residential networks more vulnerable to attack, thereby empowering the next generation of IPv6-based IoT botnets?

To answer this question, we introduce a large-scale IPv6 scanning methodology that, unlike those that rely on computationally intensive algorithms, can be run on low-resource devices common in IoT botnets. We use this methodology to perform the largest-scale measurement of IPv6 residential networks to date, and compare which devices are publicly accessible to comparable IPv4 networks. We received responses from 93,832,220 distinct IPv6 addresses, 14,901,892 of which are inside of residential networks (i.e., not the external-facing gateway IP). These residential network internal addresses span 4,412 ASes across 144 countries. These responses come from protocols commonly exploited by IoT botnets (including telnet and FTP), as well as protocols typically associated with end-user devices (including iPhone-Sync and IPP). Compared with IPv4, we reach 11× more HP printers over IPv6 than Shodan finds across the entire IPv4 Internet, and thousands of iPhones and smart lights whose services IPv4 NAT made unreachable by default. Collectively, our results show that NAT has indeed acted as the de facto firewall of the Internet, and the IPv4-to-IPv6 transition of residential networks is opening up new devices to attack. Finally, we discuss potential mitigations to prevent in-home network reachability as IPv6 adoption continues to grow.

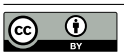
CCS Concepts

• **Security and privacy → Network security**.

Keywords

IPv6, network security, scanning

ACM Reference Format:
Erik Rye, Dave Levin, and Robert Beverly. 2026. Where Have All the Firewalls Gone? Security Consequences of Residential IPv6 Transition. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3830454.3846582



CCS '26, The Hague, Netherlands

ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3846582

1 Introduction

Residential networks are of paramount concern for Internet security. Users all over the world increasingly deploy vulnerable smart-home devices [25, 51]. IoT botnets have taken advantage of vulnerable in-home devices to launch unprecedentedly large attacks [12, 32], thereby threatening not only the device owners' own security and privacy, but everyone on the Internet.

To date, IoT botnets' growth has been limited by a small, unlikely defender: IPv4 Network Address Translation (NAT). Because IPv4 addresses are in limited supply, almost all IPv4 residential networks translate addresses at their gateway, between the home's single public IPv4 address and its many internal private addresses. One of NAT's accidental benefits is that it acts as a sort of firewall: it default-denies incoming connections unless a user has explicitly set up a port-forwarding rule to permit them. IoT botnets have therefore been limited in their efforts to scan devices on IPv4 residential networks, targeting almost exclusively gateway routers (which have a public IP address) and webcams (which commonly have NAT-bypassing port-forwarding rules). A litany of other vulnerable smart-home devices has been largely unreachable behind IPv4 NAT.

IPv6, conversely, has a virtually limitless number of public addresses, eliminating the need for NAT. As a result, when residential networks enable IPv6, they lose an accidental defense against IoT botnets.

There are two defensive mechanisms that *should* fill the inadvertent gap NAT leaves. First, home gateway devices should deploy *actual* firewalls. Second, IPv6's astronomically large address space (128 bits) is expected to render scanning ineffective. Some have theorized that these defenses are not present on the IPv6 Internet (see §2), but to date there have been no empirical studies, and moreover there are no established methods for effectively scanning residential IPv6 networks to perform such a study.

It therefore remains an open question whether the transition from IPv4 to IPv6 has ultimately made devices inside of residential networks more vulnerable to attack by IPv6-based IoT botnets.

We show in this paper that it is in fact possible to perform large-scale scans of IPv6 residential networks in a manner that low-resource devices common in IoT botnets can execute. We perform the largest measurement of IPv6 residential networks to date, and compare which devices are publicly accessible to comparable IPv4 networks. We present two key findings:

Effective, low-computation IPv6 scanning We introduce a scanning methodology that is computationally low-cost yet highly effective. Given a known-active IPv6 /48 prefix (such prefixes are publicly available), our technique simply scans the initial low-order addresses (`::1`, `::2`, etc.) within each /56 prefix. Though this is a narrower

portion of the IPv6 space than what compute-intensive target generation algorithms (§2) could potentially access, we show that it still allows us to reach over ten million in-home devices. Most importantly, it does so in a way that would be easily deployable on any low-resource IoT device.

Our technique thus shows that scanning large portions of the residential IPv6 Internet is within reach of IoT botnets.

An analysis of vulnerable services publicly accessible over IPv6 We applied our scanning methodology to perform the most comprehensive measurement study of residential IPv6 networks to date. We scanned for the presence of 30 port/protocol combinations, including many of those most commonly used by IoT botnets [12, 25], as well as the IPv6 iPhone-Sync protocol (to our knowledge, this is the first measurement of the public accessibility of iPhones). We find millions of publicly accessible, internal residential IPv6 hosts running potentially vulnerable services. We also compare our IPv6 findings to IPv4 through Shodan's scans of the *entire* IPv4 address space: for HP printers we reach an order of magnitude more devices over IPv6 than exist in all of IPv4, while for services such as iPhone-Sync and smart lighting—which IPv4 NAT renders unreachable altogether—IPv6 reachability is net-new exposure.

Collectively, our results show that IPv6 deployment in residential networks is opening up new devices to attack, empowering a new generation of IoT botnets, and ultimately putting the entire Internet at greater risk.

Contributions Our primary contributions are as follows:

- We introduce a scanning methodology that is significantly simpler than prior work, thereby demonstrating that even a low-resource IoT botnet could propagate across the IPv6 Internet (§3).
- We perform the most comprehensive scan of devices within IPv6 residential networks, measuring the public accessibility of 30 distinct port/protocol combinations commonly used by IoT botnets, and perform the first measurement of the reachability of Apple iOS devices.
- We compare reachability across IPv4 and IPv6: we reach 11× more HP printers than Shodan finds in all of IPv4, and iPhones, lighting systems, and IP cameras—unreachable through IPv4 NAT—by the hundreds to thousands (§4 and §5).
- We propose potential remediation based on our measurements and findings (§6).

To facilitate further study and to encourage greater adoption of defenses, we make our code and data available to the extent that is safe and ethical (Appendices A and B).

2 Background and Related Work

The core question of this paper is: Could IoT botnets today connect to millions of hosts within residential IPv6 networks? In this section, we review the IoT botnet threat model, give background on IPv6, and present prior work in IPv6 scanning.

2.1 Threat Model: IoT Botnets

Our threat model is meant to reflect the capabilities of today's IoT botnets that target devices primarily located within residential networks. The central goal of these adversarial bots is to connect to and exploit as many vulnerable devices as possible. To achieve this, the bots scan IP addresses.

How IoT botnets target IPv4 hosts To date, all IoT botnets of which we are aware have operated solely over IPv4. Specifically, they generate random IP addresses and attempt to connect to them over a set of ports corresponding to services against which the bot has exploits. This has proven effective over IPv4 [12]—because it comprises only 4B addresses—but enumerating the entire IPv6 address space would, at a 10,000 packets per second probing rate, require more than 10^{27} years. For IoT botnets to effectively scan IPv6, a new scanning methodology is necessary, which we provide.

Low-resource devices IoT botnets comprise primarily devices with low computation and memory resources, especially so-called "smart home" devices that have infamously poor security, such as default passwords and well-known CVEs [12, 25]. Resource constraints have no implications for how IoT botnets scan IPv4 today, but as we discuss later in this section, they preclude them from running the state of the art in IPv6 scanning (TGAs).

Lists of active prefixes We assume that the adversary can obtain a public dataset of IPv6 /48 prefixes that each have at least one active (though not necessarily reachable) host. These lists are readily available [9], relatively small when compressed, and could be easily disseminated on today's IoT botnet command-and-control infrastructure [25]. Starting with these prefixes, the adversary's goal is to discover full /128 IPv6 addresses belonging to active, reachable hosts within residential networks, as that is where vulnerable smart-home devices are likely to be.

We limit our adversary to knowing only active /48s because that is the most that we authors can ethically use (see Appendix B). In practice, an adversary could participate in running a popular service and scan back the full /128s of the clients that come back to them [31]. Even then, the adversary could *also* apply our techniques that derive /128s from known-active /48s. Thus, our attacker is strictly weaker than an attacker in practice could be, and yet we show that we are nonetheless able to connect to millions of devices inside of residential IPv6 networks.

2.2 Residential IPv6 Background

IPv6 is the successor to IPv4; it uses an impossible-to-enumerate 128-bit address space instead of IPv4's enumerable 32-bit space. This increase also affects how addresses are assigned, which we review here.

Residential networks Residential networks, broadly speaking, consist of a gateway router (CPE, or Customer Premises Equipment) that sits at the periphery of the network, and a set of devices located *inside* the network. Figure 1 provides a prototypical example of a residential network.

IPv6 address assignments in residential networks In IPv4, residential networks typically get a single public IPv4 address, and their gateways—running DHCP and NAT—assign a single private IPv4 address to each device within their network. Conversely, in IPv6, residential networks get an entire prefix from their ISP, ranging in practice from a /48 to a /64 [15]. Residential IPv6 gateways can then assign devices within their network a single IPv6 address (via DHCPv6) or allow devices to self-generate IPv6 addresses via Stateless Address Autoconfiguration (SLAAC) by providing them with a /64 (the hosts

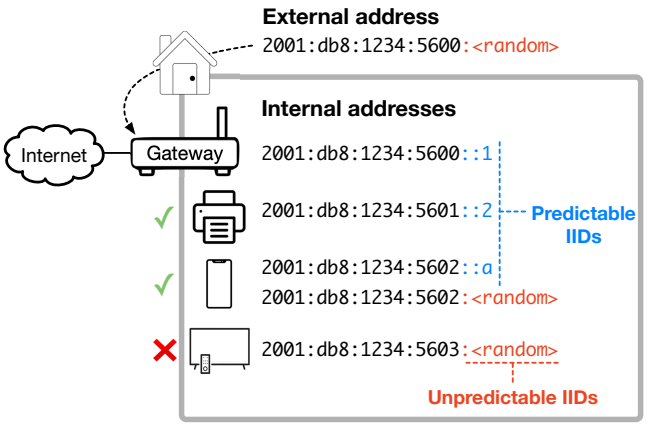


Figure 1: A prototypical residential network. Our methodology discovers devices with predictable IIDs, even if they also have unpredictable IIDs, like the phone in this example.

then choose their own lower 64 bits, known as an interface identifier (IID)). A subtle but relevant detail is that devices may receive a /128 address assignment via DHCPv6 *but also* auto-generate addresses via SLAAC. We are the first to report on the prevalence of this dual-use of DHCPv6, and we are the first to identify it as a security risk.

Unlike IPv4, in-home IPv6 addresses are almost always *publicly routable*, meaning that unsolicited traffic can be sent to residential LAN devices.

Unpredictable IIDs IIDs are the lower 64 bits of an IPv6 address. For routers and some public servers, network administrators commonly choose predictable and low-entropy IIDs (e.g., `::1`), as they are easier to remember and manually type [23]. Predictable addresses can make it possible to scan [23], track [47, 49], and even geolocate users' devices [44]. Therefore, a decades-long best practice in IPv6 is for clients to generate random IIDs and change them regularly, a process known as SLAAC Privacy Extensions or SLAAC-PE [38]. Similarly, RFC 7707 [23] recommends that, when DHCPv6 is used for assigning full /128s, it should choose random addresses from a large pool. We are, to the best of our knowledge, the first to show empirically that predictable addresses are *still* being assigned to devices, or assigned *in addition* to randomly generated addresses.

2.3 Scanning Residential IPv6 Networks

Unlike IPv4, IPv6's astronomically large address space makes it impossible to scan every address. Instead, IPv6 scanning techniques try to probe only addresses that are likely to respond. The central technical challenge behind IPv6 scanning is identifying IPv6 addresses that are likely to be in use.

IPv6 hitlists Several efforts have sought to catalog *IPv6 hitlists*: lists of IPv6 addresses or prefixes that have been observed to be in use. There are two noteworthy approaches: The IPv6 Hitlist [8, 22] is collected via *active* scans of the Internet that elicit responses from live IPv6 devices. Alternatively, Rye and Levin [47] described a *passive* approach: running servers in the NTP Pool—a volunteer-run group of NTP servers—and allowing IPv6 devices to come to them.

Rye and Levin compared these two approaches, finding that the IPv6 Hitlist comprises mostly routers and public (non-residential) servers with low-entropy IIDs, while the NTP-based hitlist can yield orders of magnitude more clients with high-entropy IIDs. However, even the NTP Pool-based hitlist is not comprehensive, as not all devices make use of the NTP Pool (Apple and Android devices, for instance, use their own NTP, non-NTP Pool time servers). As the NTP-based approach yields more clients, we adopt it to seed our scanning. We will demonstrate that our methods can discover devices that never come to NTP Pool servers, including Apple iPhones.

Target generation algorithms (TGAs) TGAs are algorithms that, when trained on IPv6 data (typically previously active addresses), generate new target IPv6 addresses to scan. While there have been many distinct TGAs introduced over the past decade [16–18, 21, 26, 27, 35, 52, 53, 57], they all share two features that are relevant to our work: *First*, none of them can guess clients' randomly generated IIDs any better than brute force. *Second*, they are all far too expensive in terms of computation and memory to be run on IoT devices. For example, 6Sense [57] requires 512GB RAM/48GB GPU-RAM. Today's TGAs are therefore not practical for IoT botnets to use for generating scanning targets. We present an effective, heuristics-based scanning methodology that is sufficiently low-cost for virtually any IoT device to run.

Prior studies of residential IPv6 gateways To the best of our knowledge, all prior scanning measurements of IPv6 residential networks have only studied the gateway routers sitting at the networks' periphery.

Olson et al. [39] studied ten gateway routers in a lab setting and analyzed their security settings and firewall implementations. They found that, contrary to RFC 3904's recommendations [28], most gateway routers do not enable a firewall by default. Both Olson et al. and RFC 3904 *hypothesized* that IPv4 NAT has acted as a de facto firewall, and that the transition to IPv6 could open new attacks. To the best of our knowledge, ours is the first work to demonstrate this empirically, at scale, in the wild.

Several studies have identified and elicited responses from residential gateways [19, 29, 34, 48]. These generally work by sending packets to IPv6 addresses that are within a residence's assigned network but are *not* in active use, thus resulting in an ICMPv6 Destination Unreachable message from the gateway, thereby revealing the gateway's address. These techniques enable scanning of the periphery of residential networks, but they are not effective at reaching devices inside the residential network.

Prior studies of devices inside residential IPv6 networks Devices *inside* of residential IPv6 networks are critical targets for IoT botnets. Yet, we are aware of no prior work that has scanned these devices at significant scale. The dearth of such studies is likely due to the difficulty in guessing randomly generated IIDs: there are no known techniques for effectively scanning devices inside residential networks at scale.

Rye and Levin [47] learned of many IPv6 addresses within residential networks, but they never scanned them, for various ethical reasons we describe and account for in Appendix B.

Some broader scanning studies that did not specifically target residential networks have *incidentally* detected hosts that may have

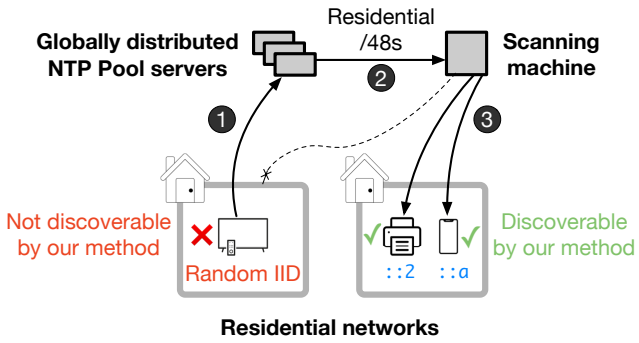


Figure 2: Overview of our scanning methodology. (1) We learn of an active /48 when a client visits one of our NTP Pool servers. (2) We extract only the residential /48s. (3) We scan the ::1–::a IIDs within each residential /56. This allows us to discover devices and networks even if they did not come to our NTP servers, like the network on the right in this example.

been inside residential networks. Williams et al. [57] used the 6Sense TGA to scan IPv6 addresses broadly (not just residential networks), and reported finding some devices commonly found in homes (e.g., printers). Klopsch et al. [31] ran NTP Pool servers and scanned back the addresses that came to them. This act has been prohibited by the NTP Pool maintainers [50] (Appendix B), and it misses the many such devices (e.g., iPhones) that never use the NTP Pool. Our methodology is able to partially discover these types of non-NTP Pool clients.

To the best of our knowledge, we are the first to scan devices inside residential IPv6 networks at scale: we elicited responses from millions of distinct devices from within residential networks. Moreover, our techniques can be run on resource-constrained devices commonly present in IoT botnets. Despite these constraints, and even though we limit our scans only to residential networks, our techniques discover e.g., over 128× more printers than 6Sense [57]. In short, we show for the first time that it is possible for IoT botnets to spread to the highly vulnerable devices that were long hidden behind NAT.

3 Scanning Methodology

In this section, we present our technique for scanning devices inside of residential IPv6 networks at scale. Figure 2 provides a high-level overview. We then present how we determine whether a responses came from the external interface of a gateway router or from and internal address. Finally, we discuss limitations of our technique.

3.1 Learning In-Use /48s via the NTP Pool

We learn active IPv6 addresses by running servers in the NTP Pool [10], following the methodology of Rye and Levin [47]. We ran 34 NTP servers in 25 countries (listed in Appendix C) between July 2024 and January 2026. The NTP Pool uses coarse IP geolocation data to direct NTP clients to a nearby NTP server. Thus, we obtain traffic from *outside* the 25 countries our servers are located in, as well. Collecting active IPv6 addresses by running NTP Pool servers was shown by Rye and Levin [47] to result in far more client addresses than other methods (§2).

Our seed data comprised 13,601,343 distinct /48s in total.

Why not just scan back the NTP Pool clients? At this point, it would be trivial to simply scan back the IPv6 addresses that came to our NTP Pool servers. However, prior to running our servers, we asked the NTP Pool operators what forms of data they were comfortable allowing us to collect, share, and act on.

The NTP Pool operators told us that it would *not* be acceptable to scan back the /128s due to their concerns that doing so would reduce user trust in the project. A decade ago, Shodan ran NTP Pool servers to scan back IPv6 addresses, which drew widespread condemnation from the community [50]. In response, the NTP Pool administrators removed them. Other recent work has sought to identify actors using the NTP Pool to learn IPv6 addresses with the aim of restricting their future participation in the NTP Pool [45].

However, the NTP Pool operators also told us that sharing the /48s publicly was acceptable (as with Rye and Levin [47]), and that any methodology that starts with *only* the /48s and derives a /128 to scan was also acceptable. We therefore discard the lower 80 bits altogether from every client IPv6 address that comes to our NTP Pool servers, saving only the /48s. The methodology we describe in the rest of this section starts with these /48 prefixes and derives /128s to scan.

Alternatives open to botnets Botnet operators likely would not adhere to the same ethical standards we do; they could run NTP Pool servers (or another popular service) and simply scan back the /128s that come to them. They could *also* perform our techniques, which start with the /48s and derive new /128s. In that sense, our results are a strict lower bound on the set of devices a botnet could potentially discover.

Alternatively, rather than run their own NTP Pool servers—which might raise suspicion—a botnet could simply download public datasets of active /48s [9]. This would effectively be equivalent to the methodology we present in this paper.

3.2 Narrowing to Residential /48s

Our study focuses on *residential* networks, but the NTP Pool data also contains many non-residential /48s, including infrastructure and mobile networks.

To focus on /48s allocated for residential networks, we keep only those that the MaxMind GeoIP2 Connection Type database [36] classifies as "cable/DSL" or "dialup." This reduced the candidate set to 8,799,019 /48s (65% of the original 13,601,343). Applying this filter provides a degree of confidence that the resulting /48 networks are assigned to residential or small business networks by access ISPs. We refer to these networks hereafter as "residential," with the caveat that MaxMind's cable/DSL category also covers small office/home office (SOHO) networks served by the same access ISPs.

3.3 Scanning Hosts within Residential /48s

Given residential /48s, we perform a series of active measurements to identify and then scan responsive addresses within those prefixes.

Address selection For a given /48, we iterate over each of its 256 /56s, and for each one we send ICMPv6 Echo Requests to its first ten IIDs (::1–::a). Unlike TGA-based IPv6 scanning techniques (§2), this simple heuristic requires no sophisticated computation and could run on virtually any IoT device.

To evaluate how many low-value IIDs to use, we randomly sampled 1,000 /48s, scanned the first 1,000 IIDs of each /56, and recorded which ones responded. Figure 9(b) in the Appendix shows that the vast majority (87.5%) of responsive addresses fell within the first ten.

The intuition behind scanning low-value IIDs is that it ought to capture IIDs that users set manually, as well as those given out by DHCPv6 servers contrary to the best practices of RFC 7707 [23]. As we will see, the prevalence of DHCPv6 makes this a surprisingly effective way of finding responsive devices—even devices that *also* assign themselves random IIDs.

Detecting responsiveness and aliases For each address we consider for scanning, we first test its responsiveness by sending an ICMPv6 Echo Request (ping). We do this using ZMap6 [55], the IPv6 port of ZMap [20]. If we do not receive any responses to this Echo Request, then it is less likely that there is an active device at that address (although it is possible that the device does not respond to ICMPv6 but does to other protocols). To limit the impact of our scans on networks, we do not subject those addresses to subsequent port scans. Botnets may not be so conservative, and would be able to probe hosts that are unresponsive to Echo Requests.

We also send an Echo Request to a random address in each of the /56s we probe. If we receive a response from this randomly chosen address, we conclude that the /56 network is *aliased*: i.e., that it responds to probes to *any* address. To avoid over-counting the number of reachable networks, we simply remove aliased networks from our further scanning and analyses altogether.

Protocol scans After identifying responsive, non-aliased addresses, we use ZGrab2 to scan them over 30 port/protocol combinations, including CWMP (router and VoIP management), HTTP, telnet, SSH, IPP (Internet Printing Protocol), and MSSQL. Descriptions of the port and protocol combinations we probed are given in Appendix D. We selected these particular protocols due to their general popularity as well as their being common targets of IoT botnets [25, 40].

To further evaluate the extent to which IPv6-based scans can better reach traditionally NATted devices, we also perform scans of the "iPhone-Sync" and Android Debug Bridge (ADB) protocols, which are used by iPhones and iPads to wirelessly sync with a computer over port 62078 [4] and to connect to Android devices, respectively. For the iPhone-Sync protocol, we wrote a custom ZGrab2 module that simply requests the iPhone's iOS version and then gracefully disconnects; it does not leak any PII. This module is publicly available (Appendix A). To the best of our knowledge, we are the first to measure the reachability of this protocol and the iOS versions deployed in the wild.

3.4 Differentiating Responses from Internal vs. External Addresses

Residential networks, like the example in Figure 1, consist of one or more *external* addresses belonging to the gateway router and a set of *internal* addresses belonging to both the gateway and, critically, devices inside the network. Prior work has established how to elicit responses from external addresses of residential IPv6 networks [19, 29, 34, 48]. The central goal of our work is to scan internal addresses, but it is possible that some of the responses to our probes were from external addresses. Thus, a critical part of our post-processing is to differentiate the two.

Responses from external addresses Every response to an ICMPv6 ZMap probe can be matched to the probe that elicited it. If any response comes from an address different from the destination to which we sent the probe, then we classify the responding address as an external address. For example, if the IPv6 address `2001:db8::928a:bac2` responded to the ICMPv6 Echo Request that we sent to an address within a target /56, then we classify that IP address as the external address for that /56.

Responses from these external addresses are typically ICMPv6 Destination Unreachable type error messages (e.g., Administratively Prohibited or Address Unreachable). In residential IPv6 networks, messages of these types generally originate from a gateway's external interface [46, 48] when the gateway cannot route the traffic to the destination, either because the destination address is not in use or its security policy does not permit it.

Responses from internal addresses If we receive an ICMPv6 Echo Reply from the address that we probed, we classify it as an *internal address*. For example, if `2001:db8::1` responds to our ICMPv6 ping with an ICMPv6 Echo Reply, we classify it as an internal address for that /56. In the event that the same address is in both sets, we remove those networks from further consideration, as we cannot differentiate between the internal IPv6 address space assigned to that home network, and the address assigned to the external interface of the gateway device. This affected 5,119,944 addresses (5.5% of all responses), which we exclude from the internal and external counts in §4.

3.5 Limitations

Methodological limitations Our techniques have several inherent limitations that prevent them from comprehensively scanning all residential IPv6 hosts:

First and most obviously, we only scan the first ten IIDs (`::1`–`::a`), which of course not all active devices make use of. We trade off comprehensiveness for feasibility of computation on resource-constrained devices, and show in §4, surprisingly, that it suffices to reach millions of devices inside of residential IPv6 networks.

Second, as discussed in §2, not all devices use the NTP Pool, so it is possible that we are missing entire active /48s from our initial set of prefixes to probe. Although Windows, iOS/macOS, and some versions of Android do not use the NTP Pool, many versions of Linux do, and many IoT devices use Linux. Note also that we do not need a device to come to our NTP Pool servers in order for us to potentially probe it; if *any* device within its /48 comes to our NTP Pool servers, then we can potentially scan it. Thus, despite this limitation, we are still able to discover millions of devices, including those that scanning back would not [31].

Third, IPv6 (like IPv4) can have highly dynamic addresses. Some ISPs reassign residential prefixes as frequently as every 24 hours [41, 42, 46]. This complicates measurement campaigns longer than the prefix rotation period, as probes sent to a network at one point in time may end up in a different customer's network at a later point in time. To minimize these effects, we tightly coupled address discovery using ZMap6 with the application-layer ZGrab2 scans we conducted.

Fourth, in some networks, the gateway may be assigned an address between `::1` and `::a` on its external interface. For gateways

that are configured this way, we will erroneously consider the responsive IPv6 address internal, rather than external. We eliminate instances in which this happens, as it confounds differentiation between the home network internal address space and the external address assigned to the gateway's WAN interface.

Self-imposed limitations In addition to the above limitations inherent to our methodology, we also imposed three constraints on ourselves to minimize the potential load on residential networks:

We scanned at the granularity of /56s, which many but not all ISPs delegate to residences; some use finer granularities (especially /60 or /64). In such cases, we fail to probe every residential network. One could apply our methodology at finer granularities, but with greater likelihood of re-scanning homes that have larger prefixes.

We scanned from four cloud vantage points, all of which are located in the United States. A lack of vantage point diversity in IPv4 has been shown to miss a small but significant fraction of HTTP(S) hosts (less than 10%) and a moderate fraction of SSH hosts (less than 20%) [56]. We are unaware of a similar study over IPv6, but we also expect that we are missing hosts we might have seen with additional vantage points, because the root causes identified over IPv4 (loss, blocking, and outages) likely translate over to IPv6 to some extent. Note, however, that the NTP Pool servers used to seed our study were geographically distributed across 25 countries (Appendix C).

Finally, our scans use ICMPv6 responsiveness as a prerequisite for subsequent scanning, but devices can be responsive to network services without being reachable via (or responsive to) pings. One could port-scan without first performing a responsiveness check, but that would increase the likelihood of scanning unused addresses.

Summary of data collection decisions

We believe that the heuristics and methodological decisions we make are reasonable, and that our results therefore constitute a lower bound on what a botnet might discover in probing IPv6. First, we seed our scanning from active NTP Pool /48s (13,601,343), so networks with no Pool client are never probed and the client addresses themselves are not probed. Second, we keep only the /48s MaxMind labels cable/DSL or dialup (8,799,019, 65%), which discards misclassified residential prefixes. Next, we enumerate every /56 of each /48, so homes delegated a /60 or /64 are probed at the wrong granularity or not at all, which skips potentially reachable devices in these networks. We probe ten low-order IIDs plus one random alias check per /56, so devices with predictable DHCPv6 addresses above `::a` are unprobed. We port-scan only the addresses that respond to ICMPv6 Echo Requests, which misses devices that may be unresponsive to ICMPv6 but do respond to port scans. Finally, we classify a reply from the probed address as internal and an error from any other address as external, dropping the 5,119,944 addresses that appear in both sets.

Implications for results All of the above heuristics, methodological decisions, and limitations cause our experiments to *underestimate* the number of reachable hosts within residential IPv6 networks. We do not anticipate that our limitations bias our results towards any particular networks or regions of the world, other than perhaps the geographical bias of our vantage points in the United States. Thus, we believe our results are representative (but a strict subset) of what an actual IoT botnet could achieve.

Table 1: Number of active residential /48s per country, as observed from our NTP Pool servers.

Rank	Country	/48s	%
1	JP	2,071,493	23.54
2	US	1,670,569	18.99
3	CN	1,231,894	14.00
4	DE	768,617	8.74
5	BR	756,514	8.60
Other (157 countries)		2,299,932	26.14
Total		8,799,019	100.00

It is not possible to evaluate precisely how much of an underestimate our results are. Rather than evaluate comprehensiveness, we view our results through a more pragmatic lens: whether or not it is possible for an IoT botnet to reach a significant number (millions) of hosts within residential networks.

4 Results

In this section, we describe the results of our IPv6 residential network measurement campaign.

4.1 Seed Data Coverage

Because we limit our scans to the /48s we learn from running our NTP Pool servers, the coverage, accuracy, and potential biases of our results ultimately derive from that data. We therefore begin this section by analyzing the representativeness of our seed dataset. To the best of our knowledge, this is the first analysis of strictly residential networks learned from NTP Pool servers [47].

Of the 13,601,343/48s observed from our 34 NTP servers in 25 countries, we categorized 8,799,019 (65%) as residential/SOHO networks (§3.2). These prefixes originate from a wide range of Autonomous Systems (ASes) and countries:

Coverage of countries Herwig et al. [25] observed that IoT botnets infect some countries far more than others. To reason about how the IPv4-to-IPv6 transition might impact the spread of botnets, it is important to have broad coverage across many countries.

To evaluate our dataset's country coverage, we aggregated the seed /48s by the country their AS is registered in, according to Team Cymru's ASN lookup service [5], which relies on `whois`. Table 1 shows the top 5 countries—which collectively constitute about 75% of the active residential /48s we observed. All of these have sizable IPv6 deployments [24]. Japan is the most common country, with slightly less than a quarter of all residential /48s. The US and China also contribute more than 10% of the /48s. However, we will see that, surprisingly, some of the countries with the most publicly accessible residential hosts are not in this top-5.

Coverage of ASes We aggregated the residential /48s by the AS announcing them using contemporaneous Routeviews [43] BGP RIB data from January 2026. 5,874 ASes are represented by at least one /48 in our data. The most common AS is KDDI (AS2516), a Japanese ISP, with about 20% of all residential /48s. An American, Chinese, German, and Norwegian ISP round out the top five.

Collectively, these coverage results indicate that our NTP Pool-derived data contains a geographically diverse set of candidate networks to probe. Moreover, the most highly represented countries align with some of the most highly infected countries of the Hajime IoT botnet [25]. We do not claim that this is necessarily representative of the *entire* IPv6 residential Internet. However, given that our scanning methodology is the first that would be feasible for common IoT botnet devices to perform, we believe that our results are representative of the coverage an IoT botnet could obtain.

4.2 Which Residential IPv6 Networks Respond?

Here, we evaluate how well our scanning methodology (§3.3) is able to discover responsive hosts inside of residential networks. Recall that our scanning methodology takes as input the 8.8M known-active residential /48s observed from our NTP Pool servers and first sends an ICMPv6 Echo Request (ping) to the first ten addresses (`::1`–`::a`) and one random IID within each /56. In total, this amounted to roughly 24B probes, sent at 10,000 packets per second. Recall further that we remove all aliased networks from consideration using responsiveness to the random IID probe as an indicator of aliasing (since it is extraordinarily unlikely that a randomly chosen address is assigned to a live host). We found only 58,888 /56 networks that were aliased, a small fraction of the 2,252,548,864 we probed (0.0026%).

In total, 93,832,220 unique addresses responded to our ICMPv6 probes, after excluding aliased networks. This includes both ICMPv6 Echo Replies as well as ICMPv6 error responses: both indicate a responsive host that can be further scanned. The responsive addresses span 2,414,088 /48s, 6,071 ASes, and 152 countries[1].

Internal vs. external responses We distinguish response addresses as being external (likely coming from the CPE) or internal (likely coming from a device within the residential network). If the response contains an ICMPv6 error from an address we did not probe, then we conclude the reply likely came from the home gateway's WAN-facing interface; we classify those as external responses. If it contains the address we probed for, then we classify it as an internal response.

The majority of the responses we received, 73.8M (78.7%), came from external IPv6 addresses. These addresses originate from 5,314 ASes and 143 countries; using Team Cymru's IP-to-ASN service [5], the most common ASes include Chinanet Backbone (AS4134) and China Mobile (AS9808), while the most common countries these responses originate from are China, Brazil, Germany, and Japan.

We received responses from 14.9M (15.9%) distinct *internal* IPv6 addresses. These addresses originated from 4,412 distinct ASes and 144 countries. The remaining 5,119,944 responsive addresses (5.5%) appeared in both sets and are excluded (§3.4). There is moderate overlap in the most common ASes and countries with external addresses—the most common ASes include Verizon (AS701), BT (AS2856) and Chinanet Backbone (AS4134). The most common countries include the US, the UK, Brazil, and China.

Which IIDs are most successful? We scanned the first ten IIDs (`::1`–`::a`) in each /56 that we probed. Among internal responses, `::1` is 31× more likely to respond than the second-most common IID, `::2`. Figure 9(a) in the Appendix displays the distribution of internal IPv6 response IIDs.

We speculate that some CPE devices self-assign `::1` to their LAN interface, leading it to be more common than other IIDs. To test this hypothesis, we computed the hop limit (IPv6's analog to IPv4's Time-to-Live (TTL)) distance from the responsive hosts to our measurement machine from the IPv6 response headers. More precisely, we assumed a starting hop limit of 64 if the received hop limit was less than 64, 128 if greater than or equal 64, and 255 if greater than or equal to 128, given that OSes typically set this value to a power of 2 or the maximum value of 255 when they create IPv6 packets. For example, if the response we received had a hop limit of 118, then we assume it started at 128 and was thus a distance of 10 hops away.

For the 5,960,191 residential networks where we received both an internal *and* an external response (many networks had only one or the other), we computed the difference in distances between them. A difference of zero indicates that the external and internal addresses correspond to interfaces on the same device; a positive difference indicates a device behind the home gateway.

Figure 3 shows the distribution of the differences in distance between internal and external responses from the same network. Indeed, we find that a difference of zero is the most common case, constituting 3,863,772 (65%) of the internal-external IPv6 pairs that responded to us. Another 997,436 (17%) of internal-external IPv6 pairs are at a distance of +1, indicating that the device assigned the internal IPv6 address is one hop behind the residential gateway. Larger differences in hop limits occur in 390,950 cases (7%), and may arise from path changes, load balancing [14], or an incorrect assumption about the starting hop limit. The remaining 708,033 pairs (12%) have a *negative* difference. Potential explanations for negative distances include topology changes or renumbering during our experiment, middleboxes that rewrite hop limits between the probed network and our vantage point [30], and nonstandard initial Hop Limits.

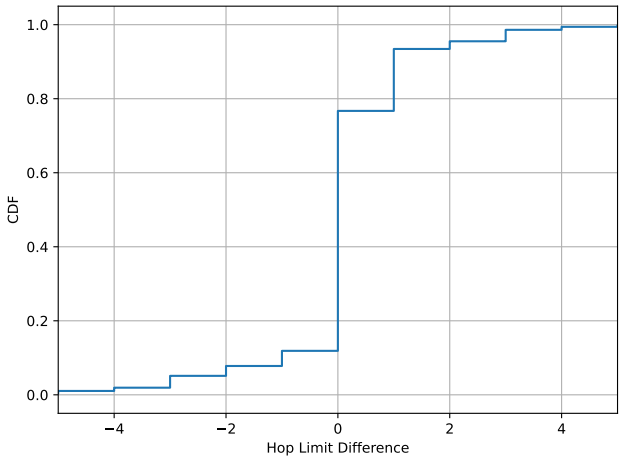


Figure 3: CDF of distance between internal and external hosts in the same /56 ($N = 5{,}960{,}191$ pairs).

[1]We use the term "countries" throughout, though we report ISO 3166-1 alpha-2 country codes, which include dependent territories.

Table 2: IPs responding to exactly k port/protocol combinations (cumulative % is $\leq k$), of 6,291,349.

Port/Protocols	IPs	Cumulative %
1	4,741,151	75.4%
2	886,665	89.5%
3	433,011	96.3%
4	164,398	98.9%
5	41,759	99.6%
6	21,289	100.0%
7	2,824	100.0%
8	173	100.0%
9	67	100.0%
10	10	100.0%
11	0	100.0%
12	2	100.0%

4.3 What Network Services Can Be Accessed?

Having established which residential IPv6 addresses are *responsive*, we next turn to evaluating which ones are running services that are *accessible*. The last stage of our scanning measurement scans the 93.8M addresses that responded to our ICMPv6 pings for 30 different port/protocol pairs (§3.3). Here, we report on who responded, to what protocols, and how this differs between external (CPE) and internal (LAN-side) responses.

Overall accessibility Of the 93,832,220 ICMPv6-responsive addresses, 6,291,349 (7%) responded to at least one of the protocols we probed. Table 2 shows how many distinct port/protocol combinations each of the responsive addresses responded to. 75.4% of them respond to only a single service, while 24.6% can be reached via two or more.

External vs. internal accessibility Internal IPv6 addresses were slightly more responsive to protocol scans, with 2,792,352 of 14,901,892 (18.7%) responsive, compared to external addresses' 3,453,701 of 73,810,384 (4.7%) responsiveness. This is a surprising result, given that 5× more external addresses responded to our ICMPv6 pings than did internal addresses. These numbers show that devices internal to a network are far more likely to be running an openly accessible network service.

One possible explanation for this disparity is that network-internal devices are—thanks to decades of NAT—tacitly assuming that any machine accessing them must be coming from the same LAN. Home gateway devices, on the other hand, have long understood the importance of restricting access to their network services only to LAN-side devices.

Service accessibility We next compare the specific networked services that can be reached on external and internal addresses. Figure 4 shows the number of responsive IPv6 addresses by protocol and response type. Internal addresses dominate protocols more likely to run on internal hosts, such as the Internet Printing Protocol (IPP) and database protocols like MSSQL; external addresses are more likely to be responsive to FTP and HTTP/80.

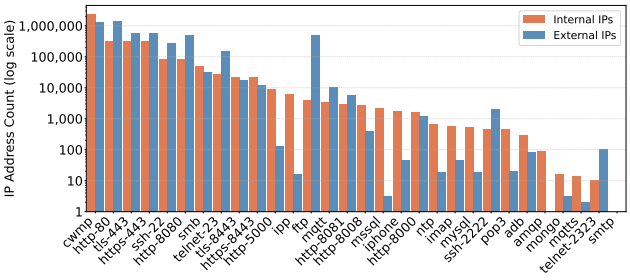


Figure 4: Overall number of responsive addresses by response type and protocol (y-axis logscale).

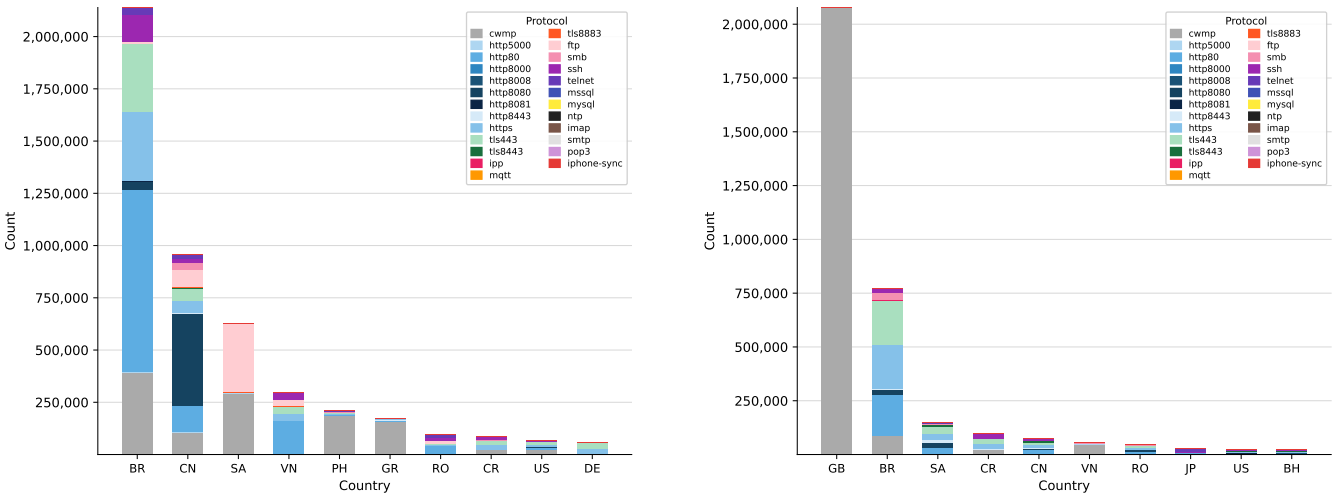
Figure 5 shows the top 10 countries with the most responsive external (left) and internal (right) addresses, broken down by the specific network services they responded to. We make two key observations:

First, internal and external accessibility can vary greatly by country. Note that the rank order of the top-10 (and even the set of countries comprising the top-10) is not the same across both plots.

Second, the set of services reachable via external addresses and internal addresses differs significantly. External addresses tend to host CWMP, web servers (HTTP and HTTPS), and FTP—all services that IoT botnets are well-known to attack [25]. Internal addresses, on the other hand, host a much greater diversity of services, including various database applications, IPP (printing), and the iPhone-Sync protocol. In Saudi Arabia (SA), for instance, external addresses tend to host CWMP and FTP. However, internal addresses host a range of HTTP ports, potentially indicating they host web pages to administer the gateway device on internal-facing IPv6 addresses and assume that it is restricted from the WAN. This is a troubling finding, as it indicates that there are many more potentially vulnerable services for future botnets to target.

What devices allow unsolicited traffic? Networks with both internal and external responses provide an opportunity to determine what types of CPE devices are allowing unsolicited traffic *into* residential LANs. Of 10,833,230 /56 networks with at least one ICMPv6-responsive internal address and one external address, 522,654 use EUI-64 external IPv6 addresses, embedding the MAC address of the WAN interface into the lower 64 bits of the IPv6 address. These MAC addresses, when resolved against the IEEE's Organizationally Unique Identifier (OUI) database [1], are concentrated in a relatively small number of router manufacturers. ZTE, eero, Taicang T&W Electronics, China Mobile, and Shenzhen Skyworth each have over 30k unique MAC addresses embedded in external IPv6 addresses; ZTE leads all manufacturers with over 126k unique MAC addresses.

The protocol scans of the external IPv6 addresses also provide additional context about which home router manufacturers permit unsolicited traffic into residential LANs. For instance, we find over 111k unique external IPv6 addresses that present Nokia-issued TLS certificates and have reachable internal IPv6 addresses. More than 50k external IPv6 addresses that permit internal network reachability present TLS certificates issued by Intelbras, a Brazilian network equipment manufacturer.



(a) Network services reachable on *external* addresses.

(b) Network services reachable on *internal* addresses.

Figure 5: The network services that are reachable on external IPv6 addresses differ significantly from those reachable on internal addresses.

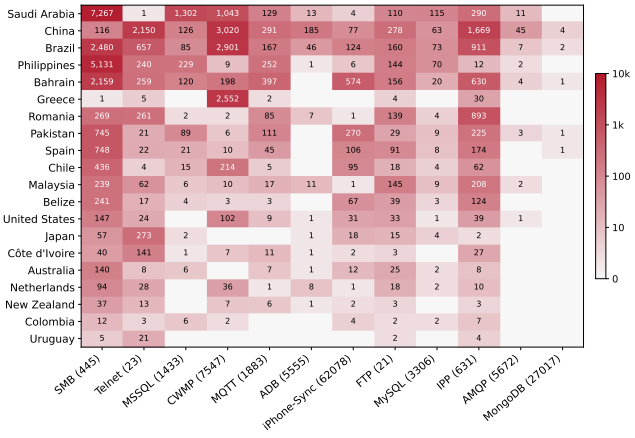


Figure 6: Unique internal IPv6 addresses that respond to a given protocol when their paired external address from the same network prefix does not.

4.4 Internal-Only Reachability

Finally, we look at networks in which both an internal and external address are responsive, and where the internal addresses have unique protocol reachability. That is, networks in which an internal address responds to ports that the external address *does not*.

Table 3 lists the protocols that are exposed only on the internal IPv6 address of a /56 but not the external IPv6 address; this scenario indicates that an attacker has increased access to in-home residential devices over IPv6. HTTP/80 is the most common port/protocol combination, with over 166k instances in which it is exposed on the internal address but not external. The file sharing protocol SMB is exposed in over 21k instances on the internal address but not the external address, potentially exposing private information were an attacker to connect to the exposed shared directories. Other protocols, such as SSH, may reveal information about the type of device or permit fingerprinting or tracking from the server reply.

Table 4 lists a number of SSH software versions we received from internal-only exposed SSH servers. We filtered these software versions for only those with *known high and critical CVEs*. Cumulatively, we find more than 10k devices running exploitable versions of SSH within residential networks, which make prime targets for botnets looking to move into residential networks over IPv6.

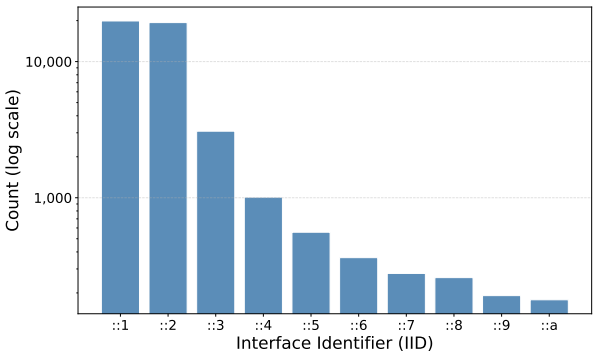
Finally, Figure 6 displays the number of internal-only reachable port/protocol combinations grouped by the WHOIS country the internal address is from. In Saudi Arabia, the country with the largest number of internal-only reachable ports, over 7,000 unique IPv6 addresses respond to SMB, as well as more than 1,000 to MSSQL and CWMP. In Greek residential networks we probe, by contrast, over 2,500 unique internal addresses have CWMP open when the external address does not.

5 Case Studies

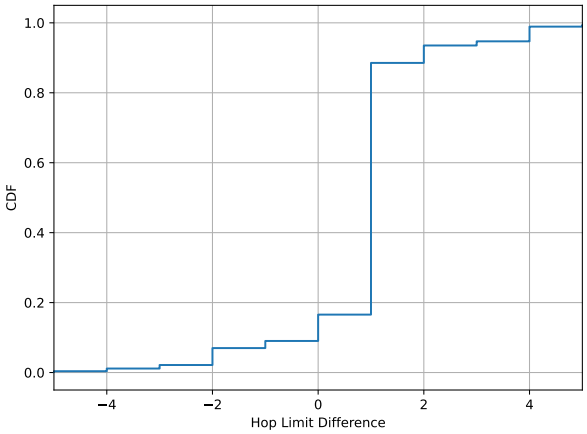
In this section, we examine in more detail case studies of devices that are definitively reachable on residential LANs. These case studies highlight the current operational vulnerability that CPE routers without IPv6 stateful firewalling present to botnets. In particular, we focus on printers, IP cameras, smart lighting systems, and Apple devices, as we can unambiguously identify them via their unique responses as non-CPE.

5.1 HP Printers

Hewlett-Packard (HP) printers commonly run an embedded web server in order to provide management and diagnostic capabilities. We take advantage of the fact that the HTTP responses from HP printers contain unique signatures. In general, HP printers include a `Server` header that follows the format "HP HTTP Server; <model



(a) Distribution of HP printer IIDs (y-axis logscale).



(b) Hop-limit distance of HP printers to their associated external address.

Figure 7: HP printer distributions of IPv6 IIDs and hop-limit distance to their gateway differ significantly from the overall population.

name>; Serial Number: <serial number>; Built: <build date> <build number>". By filtering for such responses within our HTTP ports 80 and 8080 and HTTPS 443 ZGrab2 results, we can both identify devices known to be in residential LANs and correlate unique devices across protocols and time via the serial number.

From 48,952 responsive printer addresses, we discover 45,129 unique devices after removing 3,823 duplicate serial numbers that occur at multiple IPv6 addresses. Some devices are discoverable at multiple addresses on the same LAN, such as the `::1` and `::4` addresses; we speculate that these devices may have both a wired and Wi-Fi connection. By contrast, Williams et al. discovered 351 HP printers (99% fewer) using their TGA technique [57].

Verifying printers are inside residential LANs Three pieces of evidence suggest that these printers are indeed inside residential networks, as we designed our methodology to discover. First, when both an HP printer (internal to the home LAN) and an external address are responsive for the same probed network, the distribution of their IPv6 IIDs in Figure 7(a) differs substantially from the overall internal address IID distribution (Figure 9(a) in the Appendix). Whereas the overall internal IID distribution is overwhelmingly concentrated at `::1` (94% of internal addresses), for HP printers, 19,337 (43%) have the IID `::2`, and 25,255 (56%) have an IID that is not `::1`.

Second, we examine how far, in hop-limit distance, HP printers sit from the external IPv6 address of the /56 we discover them in. Figure 7(b) shows that, of the 48,127 printer addresses with a responsive external address, nearly three-quarters (34,640, 72%) are at a hop-limit distance of +1 behind it. Contrast this with Figure 3, which shows the distance behind the external address for all ICMPv6-responsive internal addresses. Of all internal/external address pairs, the internal address is at a distance of 0 from the external address 65% of the time, while at a distance of +1 only in 17% of cases.

Third, the most common models we discover are low-cost desk printers, such as DeskJet series models; Table 7 in the Appendix lists the most common.

Comparison with IPv4

To compare our discovered HP printers in IPv6 with those discoverable in IPv4, we searched Shodan for the same "HP HTTP Server" server banner that we used to detect HP printers in IPv6. We start from 39,733 HP-printer banners returned by Shodan and discard probable honeypots in four passes. First, we drop any record Shodan itself tags as a honeypot. Second, we drop records whose HTTP "Server:" header exceeds 500 characters, a common honeypot tell where the response stitches together dozens of distinct server signatures (median header length is 3,017 chars for tagged honeypots vs. 148 for legitimate printers). Third, we drop records whose Server: header does not parse as the canonical HP format "HP HTTP Server; <model>; Serial Number: <serial>", since real HP firmware always emits this string. Fourth, we drop any record whose serial number is observed in more than one country, since real serials are globally unique and cross-country recurrence is the signature of a replayed honeypot banner—three serials alone (0123456789, TH73P4S0Q0063T, CN81HC6060) account for over 16,000 records spanning 22–51 countries each. After these filters we deduplicate by serial number, since one printer is often scanned on multiple HTTP ports (80, 443, 8080, 631), leaving 4,130 unique IPv4 HP printers.

The upper half of Table 5 lists the HP printers we discovered by AS and country (as determined using Team Cymru's AS lookup [54]). The lower half contrasts with scans of the entire IPv4 space conducted by Shodan [2]. We find significantly more (11×) total printers than Shodan does in the entire IPv4 space. Moreover, our discovery technique appears complementary to exhaustive IPv4 scans, as we find more HP printers in several individual countries than exhaustive IPv4 scanning does globally. Further, only 3 distinct serial numbers are found in both datasets, indicating that the overlap between the two scanning techniques is nearly disjoint.

Table 3: Protocols exposed on internal IPv6 addresses but not their paired external addresses, across all 10.8M same-network pairs. *Pairs* counts (internal, external) pairs with the service present only on the internal side; *Sole diff.* counts pairs where that protocol is the *only* asymmetric service; *Unique internal IPs* counts distinct internal addresses with that exposure.

Protocol (port)	Pairs	Sole diff.	Unique internal IPs
HTTP (80)	166,415	55,128	164,177
TLS/HTTPS (443)	138,144	21,746	136,665
HTTP (8080)	77,682	3,147	77,247
SMB (445)	21,158	13,888	21,044
SSH (22)	18,806	10,955	18,310
TLS (8443)	13,640	789	13,521
CWMP (7547)	10,514	8,328	10,336
HTTP (5000)	6,172	5,548	6,095
IPP (631)	5,567	809	5,527
Telnet (23)	4,634	1,809	4,515
HTTP (8008)	2,180	845	2,161
MSSQL (1433)	2,053	146	2,042
FTP (21)	1,942	127	1,864
MQTT (1883)	1,669	852	1,652
iPhone-Sync(62078)	1,512	1,493	1,489
HTTP (8000)	1,317	345	1,312
HTTP (8081)	1,159	512	1,134
NTP (123)	646	331	644
MySQL (3306)	436	62	433
IMAP (143)	389	12	345
POP3 (110)	321	0	282
ADB (5555)	277	215	277
SSH (2222)	118	43	105
AMQP (5672)	78	6	78
MongoDB (27017)	12	3	11
MQTTS (8883)	9	1	9
Telnet (2323)	3	2	3
Total (any)	**257,712**		

5.2 Apple Devices

Next, we examine the prevalence of Apple devices within residential networks using a custom ZGrab2 module we created. `lockdownd` is an Apple service that runs on devices such as iPhones, iPads, and Apple TVs [4, 6] to transfer data over a USB cable; optionally, users may opt in to enabling this service on TCP port 62078 to sync over their local networks (`nmap` lists this TCP port as "iphone-sync", terminology we borrow). While devices must be on the same iCloud account to transfer data, this process will provide the device's operating system version without authentication. We release this ZGrab2 module for other researchers (Appendix A).

We discovered 1,749 unique residential IPv6 addresses running this service. This is new exposure: iPhone-Sync is a LAN service that IPv4 NAT makes unreachable from the Internet unless a user deliberately forwards the port, and an Internet-wide IPv4 scan we conducted found only 850 such hosts. Figure 8 is a histogram of the iOS and tvOS versions we discovered through our iPhone-Sync ZGrab2 scans. Notably, there are several versions of iOS that are much more prevalent than others. iOS 26.2 was the latest release at the time of these scans; this is reflected in its dominance over other release versions. Several dozen iPhones were running the beta iOS version 26.3. Interestingly, we also note disproportionate numbers of iPhones running iOS versions 15.8, 16.7, and 18.7. We attribute these disproportionate version counts to the inability of certain iPhone models to be upgraded beyond a fixed iOS version. For instance, the iPhone 6s, SE (1st generation), and 7 cannot be upgraded to iOS 16 [3].

Table 4: SSH server versions on internal-only IPv6 addresses (internal responds to SSH; paired external does not), with known CVEs. Versions with < 100 addresses omitted. CVSS is v3 where scored, otherwise v2. CVE references: 1 CVE-2013-4421: DoS via memory exhaustion, CVSS 5.0 (v2); 2 CVE-2013-4434: user enumeration via timing, CVSS 5.0 (v2); 3 CVE-2016-7406: format string RCE via username or host, CVSS 9.8; 4 CVE-2016-7407: code execution via crafted key file in `dropbearconvert` (local tool), CVSS 9.8; 5 CVE-2016-7408: code execution via crafted `-m`/`-c` argument in `dbclient` (client), CVSS 8.8; 6 CVE-2016-7409: local memory disclosure via `-v` in DEBUG_TRACE builds, CVSS 5.5; 7 CVE-2018-15599: user enumeration, CVSS 5.3; 8 CVE-2023-48795: Terrapin prefix truncation attack, CVSS 5.9; 9 CVE-2024-6387: regreSSHion pre-auth RCE via SIGALRM race (glibc), CVSS 8.1.

Software	Version	IPs	Severity	CVEs
Dropbear	`dropbear_2012.55`	8,417	**Critical**	1–7
	`dropbear_2015.67`	464	**Critical**	3–7
	`dropbear_2014.63`	373	**Critical**	3–7
OpenSSH	`OpenSSH_9.2p1 (Debian)`	550	**High**	8–9
	`OpenSSH_8.9`	308	**High**	8–9
	`OpenSSH_8.9p1 (Ubuntu)`	172	**High**	8–9
	`OpenSSH_8.5`	149	**High**	8–9
	`OpenSSH_9.6p1 (Ubuntu)`	223	**High**	9
	`OpenSSH_9.6`	115	**High**	9

Table 5: HP printers discovered in IPv6 (our work) vs. all of IPv4 (Shodan), by AS.

Count	%	AS	CC
IPv6 (this work)			
7,465	16.5	STC (AS25019)	SA
6,165	13.7	V.tal (AS7738)	BR
6,022	13.3	V.tal (AS8167)	BR
5,878	13.0	RCS & RDS (AS8708)	RO
2,274	5.0	Chinanet (AS4134)	CN
17,325	38.4	637 others	
45,129	100	**Total**	
Shodan (all IPv4)			
1,147	27.8	Korea Telecom (AS4766)	KR
300	7.3	AT&T (AS7018)	US
124	3.0	SK Broadband (AS9318)	KR
83	2.0	Orange (AS3215)	FR
66	1.6	Deutsche Telekom (AS3320)	DE
2,410	58.4	932 others	
4,130	100	**Total**	

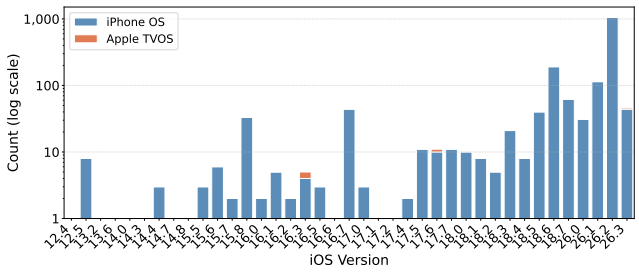


Figure 8: iOS and tvOS versions of 1,749 Apple devices.

Like the HP printers in §5.1, Apple devices are more likely to have IIDs other than `::1`. Some 324 (19%) Apple devices we discovered had the IID `::2` and 272 (16%) had `::3`, while 319 (18%) had `::1`.

5.3 IoT Devices

We discovered a wide variety of IoT devices during our measurement campaign.

Dahua IP camera systems We discovered 4,659 internal IPv6 addresses for Dahua Technology [7] IP cameras by filtering for a fingerprint unique to the HTML of these cameras' login pages ("Dahua" appears only in an image, so we filtered for the string `app-name=''cameraNewConfig''`). Like HP printers, these devices were present in many countries but were disproportionately located in a small number of them. Unlike HP printers, whose top networks span the Middle East, Latin America, and Europe, the majority of these devices were located in a single country.

Of the 4,659 Dahua camera systems we discovered, 3,196 (69%) were located in Brazil, according to Team Cymru's IP-to-ASN service. Another 913 (20%) were located in Romania. The remaining 550 (12%) were located in another 15 countries. Notably, a search for the HTML fingerprint we used to identify these devices returned no results on Shodan.

Nanoleaf smart lights Nanoleaf smart lighting systems [37] are another clear example of LAN-internal residential devices we discovered in our measurement campaign. In our measurements, we discovered 802 Nanoleaf devices in 24 countries across 35 ASes, as determined by Team Cymru. Like iPhone-Sync, this is a LAN service that IPv4 NAT makes unreachable unless deliberately forwarded.

These devices are recognizable due to their index page containing a link to upload new firmware. While the device indicates that the user must physically press a button on the lighting systems to effect the upgrade, Internet-accessible IoT devices that accept uploads from arbitrary hosts constitute a security risk that IPv4's NAT prevents. A majority of these devices (443, 55%) are found in Saudi Arabian networks, and 126 (16%) are found in Romania.

6 Mitigations

Mitigating the IPv6 exposure of devices targeted by botnets requires the assistance of multiple stakeholders.

Gateway vendors: Enable firewalls by default Although many residential gateway routers have firewalling capabilities [39], our empirical results show that millions of gateway routers do not turn them on. A stateful firewall that drops unsolicited inbound traffic provides the same protection that IPv4 NAT provided by accident. To the extent that manufacturers have eschewed it, we hope this work demonstrates that discovering devices inside their customers' networks is no longer a hypothetical attack, but is possible even in IPv6's vast address space.

ISPs: Require CPE vendor firewalling Many residential gateways are supplied to customers by their ISP. If ISPs require stateful IPv6 firewalling of the vendors whose equipment they sell or lease, this mitigation reaches users who never need to change a setting themselves. ISPs could also drop certain types of unsolicited inbound traffic to residential prefixes at their network boundaries. However, this mitigation has the potential to cause collateral damage for users who knowingly wish to expose services to the Internet.

Users: Enable stateful firewalling We recommend that technical users check the status of their gateway's IPV6 firewall through its administrative web console, if it has one. If firewalling options are exposed, users should ensure that they are enabled. While not all CPE devices expose this level of control, those that do offer their owners a mechanism to control their exposure to unsolicited inbound scanning.

Devices: Reject low-entropy leases We were able to connect to many devices (e.g., iPhones) despite the fact that they were following the best practices of generating random IIDs. This was because they were *also* accepting low-entropy IIDs, presumably from their local DHCPv6 server. A short-term mitigation for such devices would be to simply reject (or significantly restrict inbound connections to) any low-entropy IIDs if they already have a high-entropy IID. This mitigation would need to be implemented by operating systems. There is some precedent for doing so: Android does not support DHCPv6 address assignment at all and instead relies on SLAAC [11]. We reiterate the suggestion of RFC 7707 [23] for DHCPv6 servers to assign random addresses drawn from a large pool.

Operators: Monitor for IPv6 attacks Finally, operators should monitor IPv6 networks for attack traffic as they monitor IPv4 today. However, based on our experiences from actively scanning both IPv4 and IPv6 networks, it appears that operators may not be monitoring them equally. During our IPv4 scans, we received several opt-out requests and notifications that our vantage point had been infected by malware. This was not an intended goal of our measurements; on the contrary, we took steps to minimize how malicious our traffic might look and the impact that it would have on scanned networks or devices. Yet the same protocol scans that produced alerts over IPv4—some within four minutes of starting—produced not one over IPv6. This raises a question for future work: are operators monitoring their networks for malicious IPv6 traffic commensurately with how they monitor IPv4?

7 Conclusions

We showed that IPv6 networks can be an easy and effective addition to IoT botnets' arsenal. Using a publicly available dataset of active /48s and a straightforward scanning methodology that could run on virtually any device, we showed that we can reach millions of services on residential networks. We also demonstrated that, unlike with IPv4, we can reach *inside* of the networks—something that required sophisticated attacks or misconfigurations in IPv4 to accomplish [33].

Our results paint a grim picture of a looming threat for the Internet. At the time of writing, about half of the users reaching Google servers do so over IPv6 [24]. This change is—by design—largely transparent to residential Internet subscribers. In IPv4, NAT acted as an accidental but de facto firewall for residential customers. With NAT gone in IPv6, unwitting users must now rely on firewalls and ephemeral, randomized IIDs to protect their networks from being scanned. Unfortunately, our results show that randomized IIDs are of limited utility, as even devices that generate random IIDs (such as iPhones) still frequently obtain DHCPv6 leases, which often give low-order IIDs (e.g., `::1` and `::2`). And our results show that, in many cases, common CPE devices permit unsolicited, inbound scan traffic into users' homes, where future generations of botnets may find vulnerable devices ripe for exploitation.

To summarize our results simply: with their transition to IPv6, residential networks are now more vulnerable to IoT botnet scanning. We proposed new mitigations (and reiterated old ones) that users, service providers, and device manufacturers can implement in order to better protect their IPv6 home networks.

Acknowledgments

We thank Chris Eagle for early feedback. This research was partially supported by the NSF under grants CNS-2323193 and OAC-2613546. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF or the U.S. government.

References

[1] 2024. IEEE OUI database. http://standards-oui.ieee.org/oui.txt.
[2] 2024. Shodan Search Engine. https://shodan.io.
[3] 2025. iOS compatibility: Find out the latest version your iPhone can run. https://www.macworld.com/article/1811287/which-version-of-ios-can-my-iphone-run.html.
[4] 2025. Sync content between your Mac and iPhone, iPad, or iPod touch over Wi-Fi. https://support.apple.com/guide/mac-help/wi-fi-syncing-mchlada1d602/mac.
[5] 2025. Team Cymru IP to ASN Lookup v1.0. https://asn.cymru.com/.
[6] 2025. usbmuxd. https://man.archlinux.org/man/usbmuxd.8.en.
[7] 2026. Dahua Technology. https://www.dahuasecurity.com/.
[8] 2026. IPv6 Hitlist Service. https://ipv6hitlist.github.io/.
[9] 2026. The IPv6 Observatory. https://ipv6observatory.org/.
[10] 2026. The NTP Pool Project. https://www.ntppool.org/en/.
[11] Android Public Issue Tracker. 2026. Support for DHCPv6 (RFC 3315). https://issuetracker.google.com/issues/36949085?pli=1.
[12] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the Mirai Botnet. In *USENIX Security Symposium*.
[13] Michael Bailey, David Dittrich, Erin Kenneally, and Doug Maughan. 2012. The Menlo Report. In *IEEE Symposium on Security and Privacy*.
[14] Abhishek Bhaskar and Paul Pearce. 2024. Understanding Routing-Induced Censorship Changes Globally. In *ACM Conference on Computer and Communications Security (CCS)*.
[15] Tom Coffeen. 2024. Introducing DHCPv6 Prefix Delegation. https://blog.lacnic.net/en/introducing-dhcpv6-prefix-delegation/.
[16] Tianyu Cui, Gaopeng Gou, and Gang Xiong. 2020. 6GCVAE: Gated Convolutional Variational Autoencoder for IPv6 Target Generation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.
[17] Tianyu Cui, Gaopeng Gou, Gang Xiong, Chang Liu, Peipei Fu, and Zhen Li. 2021. 6GAN: IPv6 Multi-Pattern Target Generation via Generative Adversarial Nets with Reinforcement Learning. In *IEEE Conference on Computer Communications (INFOCOM)*.
[18] Tianyu Cui, Gang Xiong, Gaopeng Gou, Junzheng Shi, and Wei Xia. 2020. 6VecLM: Language Modeling in Vector Space for IPv6 Target Generation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.
[19] Jakub Czyz, Matthew Luckie, Mark Allman, Michael Bailey, et al. 2016. Don't forget to lock the back door! A characterization of IPv6 network security policy. In *Network and Distributed Systems Security (NDSS)*.
[20] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and its Security Applications. In *USENIX Security Symposium*.
[21] Pawel Foremski, David Plonka, and Arthur Berger. 2016. Entropy/IP: Uncovering Structure in IPv6 Addresses. In *ACM Internet Measurement Conference (IMC)*.
[22] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In *ACM Internet Measurement Conference (IMC)*.
[23] F. Gont and T. Chown. 2016. Network Reconnaissance in IPv6 Networks. RFC 7707 (Informational). http://www.ietf.org/rfc/rfc7707.txt
[24] Google. 2025. Google IPv6 Adoption Statistics. https://www.google.com/intl/en/ipv6/statistics.html.
[25] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, and Dave Levin. 2019. Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet. In *Network and Distributed System Security Symposium (NDSS)*.
[26] Bingnan Hou, Zhiping Cai, Kui Wu, Jinshu Su, and Yinqiao Xiong. 2021. 6Hit: A Reinforcement Learning-Based Approach to Target Generation for Internet-Wide IPv6 Scanning. In *IEEE Conference on Computer Communications (INFOCOM)*.
[27] Bingnan Hou, Zhiping Cai, Kui Wu, Tao Yang, and Tongqing Zhou. 2023. 6Scan: A High-Efficiency Dynamic Internet-Wide IPv6 Scanner With Regional Encoding. *Networking, IEEE/ACM Transactions on* (2023).
[28] C. Huitema, R. Austein, S. Satapati, and R. van der Pol. 2004. Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks. RFC 3904 (Informational). http://www.ietf.org/rfc/rfc3904.txt
[29] Young Hyun and k. claffy. 2022. Archipelago Measurement Infrastructure. http://www.caida.org/projects/ark/.
[30] Sebastian Kappes, Anja Feldmann, Tobias Fiebig, and Johannes Zirngibl. 2026. TTL Jumps: Unexpected TTL Rewrites Impacting Inferences from Traceroutes. In *ACM Internet Measurement Conference (IMC)*.
[31] Michael Klopsch, Constantin Sander, Klaus Wehrle, and Markus Dahlmanns. 2025. Time To Scan: Digging into NTP-based IPv6 Scanning. In *ACM Internet Measurement Conference (IMC)*.
[32] Brian Krebs. 2025. KrebsOnSecurity Hit With Near-Record 6.3 Tbps DDoS. https://krebsonsecurity.com/2025/05/krebsonsecurity-hit-with-near-record-6-3-tbps-ddos.
[33] Samy Kumar, Ben Seri, and Gregory Vishnipolsky. 2021. NAT Slipstreaming v2.0. https://samy.pl/slipstream/.
[34] Xiang Li, Baojun Liu, Xiaofeng Zheng, Haixin Duan, Qi Li, and Youjun Huang. 2021. Fast IPv6 Network Periphery Discovery and Security Implications. In *Dependable Systems and Networks (DSN)*.
[35] Zhizhu Liu, Yinqiao Xiong, Xin Liu, Wei Xie, and Peidong Zhu. 2019. 6Tree: Efficient Dynamic Discovery of Active Addresses in the IPv6 Address Space. *Computer Networks* (2019).
[36] MaxMind Inc. 2025. MaxMind GeoIP2 Connection Type. https://dev.maxmind.com/geoip/docs/databases/connection-type/.
[37] nanoleaf. 2025. smarter by design. https://nanoleaf.me/en-US/nlsmthm/2024/lighting-led/.
[38] T. Narten, R. Draves, and S. Krishnan. 2007. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941 (Draft Standard). doi:10.17487/RFC4941
[39] Karl Olson, Jack Wampler, Fan Shen, and Nolen Scaife. 2022. NATting Else Matters: Evaluating IPv6 Access Control Policies in Residential Networks. In *Passive and Active Network Measurement Conference (PAM)*.
[40] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yashioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. 2016. IoTPOT: A Novel Honeypot for Revealing Current IoT Threats. *Journal of Information Processing* 24, 3 (May 2016).
[41] Ramakrishna Padmanabhan, Amogh Dhamdhere, Emile Aben, kc claffy, and Neil Spring. 2016. Reasons Dynamic Addresses Change. In *ACM Internet Measurement Conference (IMC)*.
[42] Ramakrishna Padmanabhan, John P Rula, Philipp Richter, Stephen D Strowes, and Alberto Dainotti. 2020. DynamIPs: Analyzing Address Assignment Practices in IPv4 and IPv6. In *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*.
[43] Routeviews. 2025. University of Oregon Route Views Project. http://www.routeviews.org/routeviews/.
[44] Erik Rye and Robert Beverly. 2023. IPvSeeYou: Exploiting Leaked Identifiers in IPv6 for Street-Level Geolocation. In *IEEE Symposium on Security and Privacy*.
[45] Erik Rye and Robert Beverly. 2026. Cleaning the NTP Pool: Detecting and Mitigating NTP-Sourced IPv6 Scanning. In *ACM Conference on Computer and Communications Security (CCS)*.
[46] Erik Rye, Robert Beverly, and kc claffy. 2021. Follow the Scent: Defeating IPv6 Prefix Rotation Privacy. In *ACM Internet Measurement Conference (IMC)*.
[47] Erik Rye and Dave Levin. 2023. IPv6 Hitlists at Scale: Be Careful What You Wish For. In *ACM SIGCOMM*.
[48] Erik C Rye and Robert Beverly. 2020. Discovering the IPv6 Network Periphery. In *Passive and Active Network Measurement Conference (PAM)*. Springer.
[49] Said Jawad Saidi, Oliver Gasser, and Georgios Smaragdakis. 2022. One Bad Apple Can Spoil Your IPv6 Privacy. *ACM SIGCOMM Computer Communication Review*

52, 2 (2022).
[50] seclists.org. 2016. shodan.io actively infiltrating ntp.org IPv6 pools for scanning purposes. https://seclists.org/oss-sec/2016/q1/219.
[51] Satyajit Sinha. 2025. State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally. https://iot-analytics.com/number-connected-iot-devices.
[52] Guanglei Song, Jiahai Yang, Zhiliang Wang, Lin He, Jinlei Lin, Long Pan, Chenxin Duan, and Xiaowen Quan. 2022. DET: Enabling Efficient Probing of IPv6 Active Addresses. *Networking, IEEE/ACM Transactions on* (2022).
[53] Xikai Sun, Fan Dang, Zihao Yang, Xinqi Jin, Junhao Li, and Yunhao Liu. 2025. 6Loda: Pattern Filtering and Ensemble Learning for IPv6 Target Generation and Scanning. *IEEE Conference on Computer Communications (INFOCOM)* (2025).
[54] Team Cymru. 2025. IP to ASN Mapping. https://www.team-cymru.org/IP-ASN-mapping.html.
[55] tumi8. 2022. ZMapv6: Internet Scanner with IPv6 Capabilities. https://github.com/tumi8/zmap.
[56] Gerry Wan, Liz Izhikevich, David Adrian, Katsunari Yoshioka, Ralph Holz, Christian Rossow, and Zakir Durumeric. 2020. On the Origin of Scanning: The Impact of Location on Internet-Wide Scans. In *ACM Internet Measurement Conference (IMC)*.
[57] Grant Williams, Mert Erdemir, Amanda Hsu, Shraddha Bhat, Abhishek Bhaskar, Frank Li, and Paul Pearce. 2024. 6Sense: Internet-Wide IPv6 Scanning and its Security Applications. In *USENIX Security Symposium*.

A Open Science

We support the goals of open and reproducible science. Toward that end, our artifacts are available at https://github.com/Network-Security-Privacy-Research/Where-Have-All-The-Firewalls-Gone-Artifacts. This includes the active /48s we classified as residential (§3.2); our target-generation software for ZMap6 and our internal/external classification code (§3.3); and our custom iPhone-Sync and ADB ZGrab2 modules.

B Ethical Considerations

Our institution's IRB determined this work exempt and not human-subjects research. We treat that determination as necessary but not sufficient, and summarize here, following the Menlo Report's principles [13], the considerations that governed our data collection and handling.

Basis for proceeding Our probes are benign, standard protocol handshakes of the kind Internet measurement has relied on for over a decade [20], sent to addresses that are publicly routable by design of the ISPs that assign them. We weighed the cost of a small number of packets to each network against the status quo—millions of in-home devices reachable without their owners' knowledge, remediable only once demonstrated—and judged that the beneficence of identifying the vulnerability and providing concrete remediation (§6) outweighed the potential harms. We identified four potential harms: destabilizing or otherwise negatively affecting a scanned device, causing excessive or undue load on a network, collecting personal information, and collateral blocking of our provider's other tenants.

Minimizing harm We follow established best practices [20, 57]: randomized targets, rate limiting, an explanatory web page on every scanning host, and honored every opt-out request we received. We port-scanned only addresses that first responded to ICMPv6, probed only ten IIDs plus one alias check per /56, and dropped aliased networks outright (§3.3). We scanned using RFC-compliant, well-known protocols unlikely to negatively impact a device, and have disclosed our findings to a major router vendor, who acknowledged that some of its products are affected; disclosure to other manufacturers is ongoing. Our scanning platform does not explicitly prohibit scanning; we retained our IPv6 prefix for months after the campaign without scanning so that any block placed on it would not fall on a later tenant.

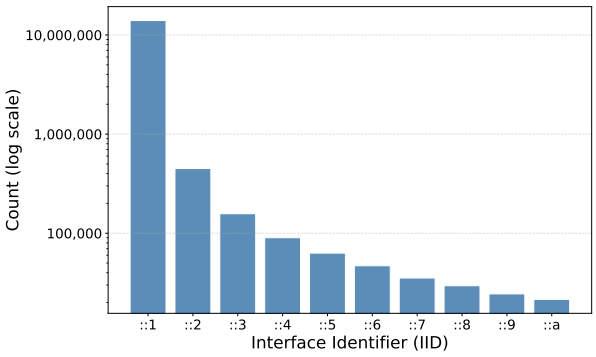
NTP Pool Clients that contact an NTP Pool server for time synchronization have not consented to be port scanned. As part of our stakeholder analysis we therefore consulted the Pool operators before running our servers, and were told explicitly that scanning back the /128s would violate the community's standards, but that /48s were acceptable because they provide sufficient anonymity: residential customers are typically allocated /56s, /60s, or /64s. We retained only /48s, as Rye and Levin did [47], and release only /48s. Active /48s are also obtainable with no Pool server at all, from BGP or public datasets, so our starting point models a realistic attacker. We note that botnets or their operators need not respect this prohibition and could obtain full /128s from NTP Pool servers directly.

Data We make minimal ZGrab2 requests and do not attempt authentication. We treat every protocol response as potentially sensitive. The raw responses we receive are stored on a machine only the authors can access and will not be released.

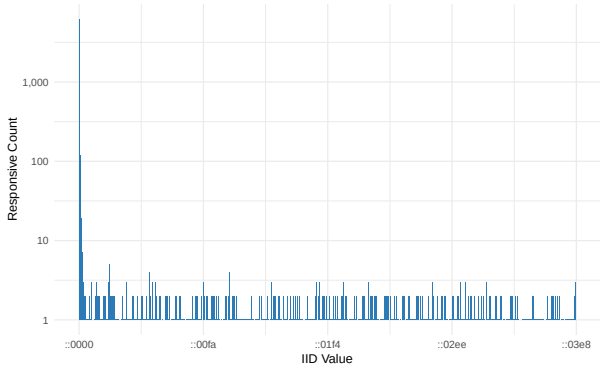
Justice The burden of our scans is small and falls uniformly on every network in our seed data. The benefit of this study accrues to the same population that bears it—residential IPv6 users, through vendor disclosure and the mitigations suggested in §6.

C NTP Pool Servers

We ran NTP Pool servers in the following countries: Australia, Brazil, Cyprus, Estonia, France, Germany, Hong Kong, Hungary, India, Israel, Japan, Kazakhstan, Norway, Poland, Russia, Serbia, Singapore, South Africa, South Korea, Spain, Türkiye, Ukraine, United Arab Emirates, United Kingdom, and the United States.



(a) ICMPv6-responsive internal hosts, `::1`–`::a`.



(b) First 1,000 IIDs of each /56 in 1,000 random /48s.

Figure 9: Responsive addresses by IID (y-axes logscale).

D Additional Tables and Figures

Table 6 provides a description of the port and protocol combinations we scanned using ZGrab2. Of note, some protocols (e.g., TLS on port 8883, which we probed as MQTTS) show up in some figures because the scanned device initiated that protocol.

Table 7 shows the counts of the most common printer models discovered on IPv6 internal addresses. These counts strongly indicate that we are indeed reaching devices internal to residential networks.

Table 6: Services and ports we probed during our port scans (see §3.3).

Protocol	Port(s)	Description
CWMP	7547	ISP management of CPE.
HTTPS	443, 8443	Primary and alternate port.
HTTP	80, 5000, 8000, 8008, 8080, 8081	Primary and alternate ports.
TLS	443, 8443	Bare TLS handshake.
iPhone-Sync	62078	iOS sync protocol.
IPP	631	Internet Printing Protocol.
MQTT	1883	Message queue protocol.
MQTTS	8883	MQTT over TLS.
AMQP	5672	Message queue protocol.
FTP	21	File transfer, command port.
Telnet	23, 2323	Remote access.
SSH	22, 2222	Secure shell.
SMB	445	Windows file sharing.
ADB	5555	Android Debug Bridge.
MongoDB	27017	MongoDB database.
MSSQL	1433	Microsoft SQL database.
MySQL	3306	MySQL database.
NTP	UDP/123	Network Time Protocol.
IMAP	143	Email retrieval.
POP3	110	Email retrieval.
SMTP	25	Mail transfer.

Table 7: Top 5 HP printer models on internal IPv6 addresses by count, deduplicated by serial number.

Model	Count	%
HP DeskJet 2700 series - 7FR22A	3,445	7.63
HP Ink Tank Wireless 410 series - Z4B55A	3,346	7.41
HP Smart Tank 580-590 series - 4A8D5A	3,330	7.38
HP Smart Tank 580-590 series - 1F3Y2A	3,083	6.83
HP DeskJet 2800 series - 6W7G2A	2,394	5.30
Others	29,531	65.44
Total	**45,129**	**100.00**

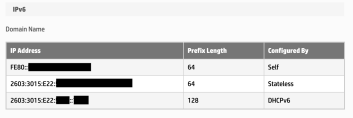


Figure 10: An HP printer status page. This device has both a DHCPv6-assigned address as well as a random, SLAAC-generated address. (Addresses anonymized for privacy.)

Figure 9(a) displays the distribution of internal IPv6 response IIDs. The IID `::1` is the most common, with 31 times more responsive internal addresses than `::2`, the next-most common.

Figure 9(b) displays the number of responsive IIDs observed from sending ICMPv6 Echo Requests to the first 1,000 addresses in each /56 of 1,000 randomly chosen /48s. This motivated the choice in §3.3 to probe only the first ten: 87.5% of responsive addresses fall there.

Figure 10 is an exemplar screenshot of the management web page of a consumer-grade HP printer we identified over IPv6. This printer's configuration page confirms that it has *both* DHCPv6 and SLAAC IPv6 addresses, in addition to a link-local IPv6 address. The DHCPv6 address, unlike the random SLAAC address, is easily discoverable by probing for addresses with only low IID bits set.