

Exploring the Exposure of IPv6 End-Host Networks

Hugo Hue
Georgia Institute of Technology
Atlanta, Georgia, USA
hugohue@gatech.edu

Abhishek Bhaskar
Georgia Institute of Technology
Atlanta, Georgia, USA
abhaskar@gatech.edu

Amanda Hsu
Georgia Institute of Technology
Atlanta, Georgia, USA
ahsu67@gatech.edu

Paul Pearce
University of California, Irvine
Irvine, California, USA
paul.pearce@uci.edu

Frank Li
Georgia Institute of Technology
Atlanta, Georgia, USA
frankli@gatech.edu

Abstract

Worldwide IPv6 adoption is pervasive, and continues to grow. IPv6 support is especially common for end-host deployments, with more than 45% of users connecting to Google’s services over IPv6. The vast size of the IPv6 address space eliminates the need for Network Address Translation (NAT), which is prevalent in IPv4 and has *implicit* security benefits. While an IPv4 client device (e.g., laptop, IoT device) using WiFi would typically be inaccessible to the public Internet due to the usage of NAT, such devices in IPv6 must rely on *explicit* security actions such as firewalling to avoid exposure. As a result, such devices may be publicly addressable and accessible over IPv6, but not IPv4. Although prior work has compared IPv4 and IPv6 security on dual-stack servers, similar comparisons for end-host networks remain largely unexplored.

This work seeks to understand the exposure (e.g., lack of firewall or NAT) and security posture of IPv6 end-host networks both in isolation and in comparison to similar IPv4 networks. To explore these questions, we collect and evaluate client IPv4 and IPv6 addresses observed by 14 NTP servers across 6 continents in the NTP Pool. We find that IPv6 end-host networks are significantly more open than IPv4, with 25% of explored IPv6 networks demonstrating a lack of *any* filtering, compared to only 5.6% in IPv4. We identify an array of consumer services exposed over IPv6 to the public Internet, including Android ADB, Apple Airplay, and Google Chromecast—all services not commonly seen in IPv4. We then perform traceroutes to the open networks to understand their topologies and the client’s residential routers. Ultimately, our findings signal the need for broad efforts in deploying network-level firewalling for IPv6 networks.

CCS Concepts

• **Security and privacy** → *Network security; Security protocols; Firewalls*; • **General and reference** → *Measurement; Empirical studies*.

Keywords

Network Security; Security Measurements

ACM Reference Format:

Hugo Hue, Abhishek Bhaskar, Amanda Hsu, Paul Pearce, and Frank Li. 2026. Exploring the Exposure of IPv6 End-Host Networks. 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/10.1145/3830454.3832565>

1 Introduction

As IPv4 address exhaustion has led to an increasing scarcity of addresses [15, 29], IPv6 has recently seen significant and widespread deployment [27, 35, 64]. IPv6 deployment is specifically common in end-host networks, with Google indicating more than 45% of users currently connect to their services over IPv6, up from 30% in 2020 [24]. Other platforms, such as Facebook [12] and Akamai [51], report similar growth in IPv6 traffic.

The widespread adoption of IPv6 has significant *practical* implications for network security. In a common residential end-host scenario, laptops, desktops, and Internet-of-Things (IoT) devices will share a single IPv4 address via the use of Network Address Translation (NAT) at the router, Wi-Fi access point, or similar [25, 41]. While not designed nor intended as one, NAT is an *implicit* security technology in this context; devices behind the NAT gateway are by-and-large inaccessible to the public Internet by virtue of them being not addressable—NAT is, essentially, a firewall.

However, due to the abundance of IPv6 addresses, where a single end-host can have *many* orders of magnitude more addresses than the entirety of IPv4, IPv6 deployments do not need nor typically utilize any address translation [66]. The same laptops, desktops, and IoT devices each have their own public IPv6 address that is routable and addressable by the Internet at large. Thus, without the use of *explicit* security technologies, such as firewalls at the network or device level, any open ports on these hosts will be exposed to the Internet. Unfortunately, prior work has shown that some popular consumer routers do not, by default, provide network-level firewalling of IPv6 hosts [52]. Meanwhile, despite the enormous address space of IPv6, prior work [34, 61] has shown that one can gather end-host IPv6 addresses at scale, such as through joining as a node in the NTP Pool [58], which is the default source for time synchronization in popular Linux distributions. Thus, an adversary can easily harvest end-host IPv6 addresses and exploit devices that are unintentionally exposed to the public, oftentimes with a poor security environment (e.g., having an outdated software version and lacking in access control, highlighted in prior work [34, 69]).



This work is licensed under a Creative Commons Attribution 4.0 International License. *CCS ’26, The Hague, Netherlands*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2871-6/2026/11

<https://doi.org/10.1145/3830454.3832565>

Goal. This work investigates the extent to which IPv6 end-host networks are exposed. We seek to understand if networks are firewalled or open, and the security implications of such openness. This study comprises two components: (1) an Internet-wide measurement of the network openness of IPv6 in comparison to IPv4; (2) an in-depth investigation of IPv6 open networks, including (a) network infrastructure, and (b) identifying misconfigured routers, which causes a network being open.

Approach. To understand these questions of end-host exposure, we collect millions of client IPv4 and IPv6 addresses and conduct comparative port scans in order to identify services. We collect client IP addresses through the NTP Pool [58] in both IPv6 and IPv4 by setting up NTP servers across 6 continents (Section 3). We focus on NTP clients because NTP is widely used by end-hosts (e.g., PCs, IoT devices, and smartphones) for time synchronization; in comparison, addresses from other sources such as IPv6 scanners or DNS tend to be servers [16, 61]. With this deployment method, we collect a median of 31M unique IPv6 addresses (across 8.9K ASes) and 38M IPv4 addresses (across 46K ASes) per day.

Leveraging this deployment, we build a pipeline that scans newly observed IP addresses in 24-hour windows (Section 4). We execute two types of scans (1) *extensive scans* and (2) *generic scans* that run simultaneously (Section 5). For each scan, we randomly sample the same amount of IP addresses to scan for IPv6 and IPv4. We balance the traffic generated by our measurements by scanning more ports and sampling fewer subnets in *extensive scans*, and scanning fewer ports across more subnets in *generic scans*. By performing port scanning to these hosts in both IPv6 and IPv4, we can evaluate the broad exposure of these end-host networks, comparatively.

To further understand IPv6 network exposure (Section 6), we identify five “Client-specific Services” that are: 1) client-specific and should not be, in expectation, exposed to the Internet, 2) are commonly seen in IPv6, and 3) are not commonly seen in IPv4. These services are Android ADB (TCP/5555), Chromecast (TCP/8443), Airplay (TCP/7000), iOS iTunes Sync (TCP/62078), and Netflix DIAL (TCP/9080). Focusing on networks containing hosts running these services along with their closed and filtered ports, we infer that a median of 73K /64s (IPv6 end-host subnets) from 2.7K ASes are open in IPv6 as opposed to 522 end-sites from 145 ASes in IPv4 per day. The open networks are largely from the ISPs (e.g., Free SAS (FR), Softbank BB Corp. (JP), and COMCAST-7922 (US) in IPv6 and HKT Limited (HK), Korea Telecom (KR), and Etisalat (AE) in IPv4).

To further explore the exposure of these networks, we conduct traceroutes to the hosts containing client-specific services (Section 6.2), and scan network-adjacent hosts behind the same routers as well as the routers themselves (Section 6.3). We are able to identify some router models and cross-reference them with the official documents that suggest these routers do not enable IPv6 firewalls by default or do not support IPv6 firewalls entirely (Section 6.4).

Finding summary. Our large-scale measurement study reveals an alarming state of exposed IPv6 end-host networks, which the community has arguably overlooked. This security issue purveying the IPv6 ecosystem is largely affected by two factors: (1) ISP routers given to end-users that lack a firewall by default, exposing end-host devices unintentionally (e.g., we found 14.9K exposed ADB devices each day are vulnerable to remote code execution and can

be taken over to form a new botnet in IPv6.); (2) Services (e.g., Netflix’s casting service) that are designed to be used within a local network are not prepared for external attacks (e.g., we discovered a new vulnerability on the Netflix mobile application). Given that adversaries could readily reproduce our attack, it is crucial to inform the community, including network operators, end-users, and application developers, about the emerging threat associated with adopting IPv6 and to enforce a firewall for IPv6.

Contributions. We present the first extensive measurement study exploring the security of IPv6 end-host networks compared to IPv4. Specifically, our contributions include:

- A security evaluation of identified unfiltered networks, identifying widespread exposure of client-specific services in IPv6 (e.g., 14.9K vulnerable ADB devices and 4.8K vulnerable Netflix devices).
- Identifying significant exposure of IPv6 networks, absolutely and in comparison to IPv4 (i.e., 25% of IPv6 end-hosts lack filtering, compared to 5.6% of IPv4 hosts).
- An in-depth analysis of IPv6 open networks, including network structures, the discovery of around 9K end-host networks in IPv6 from over 400 ASes.
- Identifying specific routers that contribute to exposed IPv6 networks, and verifying their default configuration does not provide a network-level firewall.
- Discuss recommendations for securing IPv6 networks.

2 Related Work

IPv6 Security. The past several years have seen an increasing number of large-scale measurements of IPv6 security. A number of these studies have focused on the task of discovering active IPv6 hosts, such as through public data sources (e.g., the IPv6 Hitlist uses DNS records [16] and Borgolte et al. enumerated addresses through DNSSEC [5]) or target generation algorithms [37, 47, 69], and evaluating the security characteristics of identified hosts. While these previous works have identified vulnerable or misconfigured IPv6 network services, the population of hosts characterized skews heavily towards routers and servers due to the nature of the datasets used. Such hosts, by their nature, are typically configured to be Internet accessible (although certain services may be unintentionally exposed). In comparison, our work focuses on the security evaluation of client end-hosts and their networks, which traditionally in IPv4 are not meant to be publicly reachable.

The most similar work to our study is Czyz et al. [9], who identified dual-stack servers and routers (i.e., available simultaneously on both IPv4 and IPv6) and evaluated differences in the available network services. As separate end-host firewalls are employed for IPv4 and IPv6, this measurement in essence assessed the differences in firewall policies between the two IP protocols on the same device. Through their measurement, they identified that a dual-stack host often had more open services on IPv6 than on IPv4. Given that IPv6-only services are likely rare, this finding indicated that the dual-stack hosts had more permissive firewall policies for IPv6 specifically, and in many cases, likely did not enable the IPv6 firewall altogether. While our study bears similarities in its focus on comparing open services between IPv4 and IPv6, we seek to understand and compare IPv4 and IPv6 networks broadly, beyond

only considering dual-stack hosts. As a consequence, our study encapsulates hosts and networks that may be available only on one protocol (rather than both). Furthermore, the method designed by this prior work scoped the study specifically to servers and routers, whereas our study caters towards client end-hosts and networks.

An additional relevant work is Olson et al. [52], who examined the firewall deployments of 10 home routers and found that two of them did not have a firewall enabled by default for IPv6. This finding motivates our larger-scale investigation into the exposure of IPv6 end-host networks, to determine the extent to which such router configurations drive IPv6 security concerns in practice.

NTP for IPv6 Address Collection. Other works similarly use the NTP Pool to collect IPv6 client addresses. Rye et al. [61] use this data collection method, but only use ICMP to identify host responsiveness, and do so to evaluate IPv6 hitlists rather than IPv6 network openness. ICMP responsiveness does not effectively measure firewall policies as ICMP is blocked differently, particularly in IPv6, where ICMPv6 is used for Path MTU Discovery. Our major findings are distinct from those in prior work.

Additionally, a recent work from Klopsch et al. [34] also collected and scanned client addresses from the NTP Pool. They set up 11 NTP servers and added them to the NTP Pool. They scanned five protocols (i.e., HTTP, SSH, AMQP, MQTT, and CoAP) and compared results with the IPv6 Hitlist. They highlighted that NTP Pool clients have a higher fraction of hosts with an outdated SSH version and open AMQP and MQTT services, compared to the IPv6 Hitlist. However, this recent work focuses on understanding the differences between addresses extracted from the NTP Pool and those in the IPv6 Hitlists, as well as identifying scans using the NTP Pool. In our work, however, we focus on the filtering policy of end-host IPv6 networks (instead of end-host devices) and compare it with that of IPv4 networks (as opposed to the IPv6 Hitlist). Our methods are different and novel compared to the prior work (e.g., we consider the TCP port state to infer the firewall policy). We found new client-specific services widespread across IPv6 networks, which are rarely seen in IPv4, with severe security implications, and have not been discovered in prior work.

3 Host Data Collection

In this study, we seek to evaluate the IPv6 end-host network exposure. As IPv6 does not broadly deploy NATs due to its abundant address space, we hypothesize IPv6 networks are more Internet-exposed due to a lack of firewalls. At a high level, our method entails identifying such hosts and assessing their network services to determine exposure. Our measurement pipeline, illustrated in Figure 1, consists of several parts. Here, we describe the first: the host data collection using NTP pools. Whereas prior work on IPv6 address discovery focused on uncovering primarily servers [5, 9, 16, 47, 69], we focus on obtaining IPv6 end-host addresses. As Rye and Levin [61] demonstrate the use of the NTP Pool to harvest IPv6 end-host addresses, we deployed NTP servers in the NTP Pool to collect end-host devices beyond servers (e.g., personal computers, smartphones, and IoT devices, shown in prior work [46, 61, 62]). We discuss ethics of host collection in Section 8. We elaborate on our deployment of NTP servers in the NTP Pool in Section 3.1, followed by NTP client characterization in Section 3.2.

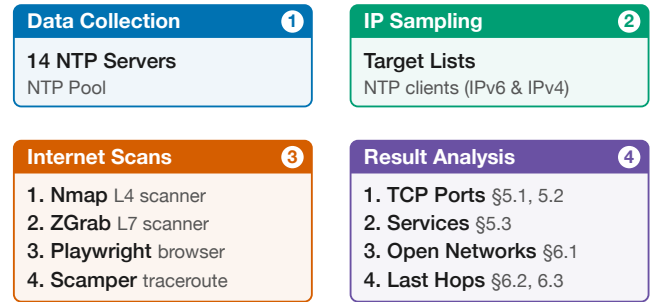


Figure 1: Overview of our measurement pipeline. We collect IP addresses from the NTP servers, sample them in both IP versions, perform various measurements towards them. Then, we analyze various attributes from our measurements to characterize the hosts and label the networks.

3.1 NTP Deployment

We collected IP addresses through the NTP Pool, a volunteer-driven platform that provides time synchronization service to clients, especially Linux-based devices (e.g., Debian, CentOS, Ubuntu, Android), as shown in prior work [46]. Here, we describe our NTP server deployment in terms of their location, uptime, traffic received, and, finally, the addresses we collected. We cover the ethical considerations of using the NTP Pool in Section 8.

Server Demographics. We deployed 14 NTP servers in the NTP Pool: 13 on Azure VMs and 1 in an academic network. We followed the NTP Pool Project’s best practices for joining [57], including syncing with multiple NTP servers and disabling NTP management queries to avoid contributing to amplification denial-of-service attacks. We deployed at least one server per continent, aligning with populous regions, Azure availability, and IPv6 adoption. Our servers cover 12 countries on 6 continents: three in the US (North America) and one each in Brazil (South America); the UK, France, and Germany (Europe); South Africa (Africa); Qatar, UAE, India, Japan, and Hong Kong (Asia); and Australia (Oceania). Each supported both IPv6 and IPv4. We did not use these servers for scanning.

Server Uptime. The NTP Pool measures NTP servers’ round-trip time to ensure server availability. If an NTP server is consistently unavailable, the NTP Pool platform will email the administrator and exclude it from serving in the NTP Pool. During our data collection period, our servers maintained availability, and we did not receive emails from the NTP Pool about excluding our servers.

Server Traffic. The traffic assigned by the NTP Pool to each NTP server varies by server location, and between IPv6 and IPv4. We used a parameter, *Net Speed* (available on the NTP Pool management portal), to control the level of traffic expected to be handled by each server. With the default configuration of the NTP node, the received number of IP addresses in IPv4 was significantly higher than that in IPv6 due to the difference in client populations. We therefore tuned the parameter such that we collected similar numbers of addresses across both IP versions, e.g., if an IPv4 server receives more IP addresses, we set a lower amount of *Net Speed*.

Collected Addresses. Figure 2 shows the number of collected NTP IP addresses throughout the data collection period. Despite

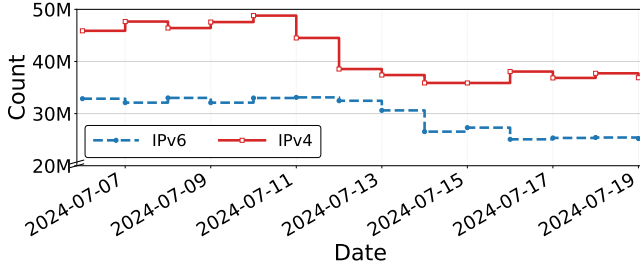


Figure 2: Number of collected IP addresses per day for all NTP servers. (Remarks: Y-axis is truncated.)

tuning the *Net Speed* parameter, certain servers in the NTP Pool collected significantly more NTP clients in one address space than the other, causing the difference in collected IP addresses in total between IPv6 and IPv4. For example, the vantage point in Hong Kong collected around 18X more median daily IP addresses in IPv4 than IPv6. To avoid the bias created by specific regions that collect excessive IP addresses in one of the address spaces, we sampled the same amount of IP addresses from each vantage point. We elaborate on our sampling approach in Section 4.

3.2 Host Characterization

To investigate the characteristics of the collected NTP clients, we analyze data for each day over our measurement period (i.e., two weeks from July 6th, 2024) to identify their locations and AS distributions and present median daily figures in this section.

We use RouteViews [13] to determine the announced BGP prefixes of the IP addresses collected on the same day. For the AS information, we use MaxMind’s GeoLite2 ASN database [42]. The median number of IPv6 addresses collected per day was 31M, spanning a median of 42.8K BGP prefix announcements and 8.8K ASes. In IPv4, we collected a daily median of 38M addresses spanning 359K BGP prefixes and 46K ASes. We detail the statistics of the collected IP addresses in Appendix A (Table 3). The diversity of collected data shows the representation of our findings to the broader Internet. We note the limitation that we do not capture clients that do not use the NTP Pool (e.g., Windows uses its own NTP servers [26]). However, as we lack such devices in both IPv6 and IPv4, evaluated hosts between two protocols are still comparable.

Location distributions. We geolocate with MaxMind [42] and find country-level locations for 99.7% of collected IPv6 addresses and 99.9% of IPv4. Both IPv6 and IPv4 addresses are geographically diverse, spanning over 175 regions (IPv6) and 240 regions (IPv4). In IPv6, the top regions are the United Arab Emirates (19.6%), Brazil (15.8%), the United States (15.0%), Germany (10.3%), and India (7.1%). In IPv4, the top regions are China (28.4%), the United States (13.0%), Japan (9.2%), India (9.0%), and Brazil (8.0%).

The number of collected IP addresses correlates to the number of NTP servers available in the regional pool zone and the adoption of IPv6 in the region. For example, we received the most traffic from UAE because our server was the only server in the region for IPv6 such that it received most of the NTP traffic from the UAE.

AS distribution. We collected IP addresses from 8,868 distinct ASes in IPv6 and 46,347 in IPv4. We use the ASdb dataset to categorize their types [72]. All the top ASes in IPv6 and IPv4 are labeled as either Internet Service Providers (ISPs) or Phone Providers, in contrast to categories like hosting or cloud providers. In IPv6, the top ASes are major ISPs or telephone operators in the region (e.g., Comcast, AT&T, and T-Mobile in the US; Deutsche Telekom and Vodafone in Germany; Orange in France). In IPv4, the top ASes include Chinanet and China Mobile, the major ISPs and telecoms in the region; as in IPv6, Comcast is among the top. Overall, across all collected IP addresses, over 82% (IPv6) and 73.4% (IPv4) come from ASes that ASdb classifies as ISPs, and 3.6% (IPv6) and 13% (IPv4) from Phone Provider ASes.

By number of BGP prefixes, the top ASes in both IPv6 and IPv4 are mostly telecommunication companies (e.g., in IPv6, Bharti Airtel (AS 45609) with 2,235 prefixes and China Mobile Communications (AS 9808) with 1,848 prefixes). These networks fit our goal of measuring end-host networks.

4 Host Evaluation Method

In this section, we describe other parts of our measurement pipeline (Figure 1), which evaluate host and network exposures. Section 4.1 describes how we sample IP addresses from the collected ones for network scans. Section 4.2 elaborates on four tools we use to gather host/network-related information (e.g., TCP port states, application-layer responses, and the last hop to the hosts). In Section 4.3, we cover our methods for service identification and heuristics for firewall detection, open network evaluation, and router analysis based on scan results. Finally, we address limitations in Section 4.4.

4.1 IP Sampling

Every 15 minutes (which we refer to as a scan window), we sample *new* IP addresses collected from our NTP servers for network scans. We consider an IP address *new* if it has not been scanned in the past 24 hours. We additionally balance our sampling across our servers and IP versions to reduce bias toward certain regions (e.g., we observed more IP addresses from the UAE in IPv6 than IPv4).

Sampling Timeframe. We sample IP addresses that have not been scanned within the last 24 hours to avoid re-scanning the same device frequently. In contrast, we scan the same IP address again if we observe it more than 24 hours since the last scan. We choose a 24-hour window because it is the default address lifetime of the privacy extension in IPv6 [18]. We acknowledge that IP address lifetimes can vary based on the use of privacy extension and the assignment practice of network operators [53]. The consideration of potential impacts on our analysis is covered in Section 4.4.

Balancing Sampling Across Servers and IP Versions. As discussed in Section 3.1, the same server can have a significant difference in collected IP addresses between two IP protocols. To maintain uniformity of the addresses to measure, we randomly sample 8,000 IPs from each server in each scan window. We balance the number of sampled IP addresses per server to ensure we compare IPv4, IPv6, and each server equally. Because we typically receive less IPv6 traffic in each server, this is the bounding factor in our sampling.

4.2 Network Scanning

We deploy the remaining parts of our scanning pipeline (Figure 1) in a US academic network, outside of the institution’s firewall (thus avoiding interference). We did not use our NTP servers for scanning purpose.

We perform various network measurements on the sampled NTP clients in IPv6 and IPv4 concurrently. We use existing tools (i.e., Nmap [40], ZGrab [71], Playwright [56], and scamper [39]) to gather information about the collected hosts to investigate services potentially exposed on end-hosts. We also aim to understand the cause of potential network exposure by identifying the gateway device of end-hosts.

Measuring Ports with TCP SYN Scans. To discover potential services, we perform TCP scans of common ports of client services (e.g., Android applications [70], IoT devices [32], database systems [30], and Apple iTunes Syncs [55]). We use Nmap’s TCP SYN scan [40] (illustrated as the L4 scanner in Figure 1). The port state is determined by sending a TCP SYN packet to a particular destination port, then labeling the port as: 1) *open* if the host replies with SYN-ACK, 2) *closed* if the host replies with an RST, and 3) *filtered* if the host is unresponsive or replies with an ICMP error message, aligning with Nmap’s terminologies [40].

Although it would be more informative to scan every port on every host, we are realistically constrained by both network capacity and time. To survey more ports efficiently, we perform *generic scans* on all sampled addresses with fewer ports, and apply *extensive scans* on a small, balanced subset of these addresses, sampled by subnet, with more ports. The *extensive scans* cover the top 1K TCP ports [45] and 11 other ports (i.e., TCP ports 162, 623, 1883, 5672, 6379, 8883, 9333, 27017, 30303, 42135, 59777) that are related to security-sensitive or client-centric services (e.g., databases, remote access controls, cryptocurrency, and Android applications [70]). The selected ports cover both popular services used by servers (e.g., SSH, HTTP) and clients (e.g., ADB, iPhone) to establish an understanding of potential open services and device types of the scanned hosts. We added these ports to *extensive scans* to explore if these ports are commonly open in IPv6, as these services could be protected by a firewall or a NAT in IPv4 due to the service nature. We apply *extensive scans* to 10% of subnets randomly sampled from the target list. In each subnet, we randomly sample one IP address. This scan approach affords broader port coverage for a smaller set of addresses across diverse networks, while avoiding excessive scan traffic in aggregate. During initial exploration, we additionally found several ports widely open in the extensive scans (i.e., TCP/5555, TCP/7000, TCP/9080, and TCP/62078), that are known to commonly have client-specific services. Our *generic scans*, which evaluate a larger population of hosts, cover these four ports and five others related to remote access and database services (i.e., TCP/161, TCP/162, TCP/623, TCP/3283, TCP/6379), and the top 100 TCP ports [45], applied to all remaining sampled addresses.

Application-layer data. To gather information on the service running on the open port, we establish application-layer connections over TLS and HTTP. We additionally capture screenshots of HTTP pages to better identify services.

We use the HTTP probe from ZGrab to collect HTTP responses by initiating GET requests [10]. We attempt HTTP on ports that are

assigned in IANA [3] with HTTP (i.e., TCP/80, TCP/8000, TCP/8008, TCP/8009, TCP/8080, TCP/8081, TCP/8088, TCP/8888). Additionally, because HTTP is one of the most popular unexpected services across all possible ports [30], we also attempt HTTP with the top open ports that do not have a default or broadly assigned service (e.g., TCP/7000, TCP/9080, and TCP/49152).

Because ZGrab does not support JavaScript execution [10], we use the Playwright [56] crawler to crawl HTTP-responsive services, allowing for JS evaluation and page redirection. We log each unique HTTP response body by its SHA256 hash and only crawl with Playwright if an HTTP service’s response hash has not been previously observed. We find that the rendered HTTP page often includes useful information to identify the type of services or devices (e.g., router models are often rendered on the webpage).

Next, to collect TLS certificates, we use ZGrab’s TLS probe module [71], which performs TLS handshakes. We attempt TLS on ports commonly used for HTTPS: TCP/443, and TCP/8443. We exclusively do TLS measurements towards these ports because of their abundant popularity; however, acknowledge that other TLS protocols could be analyzed in future work.

Traceroute Experiments. When we analyze our scan results (i.e., TCP open ports, application-layer responses), we find five client-specific services (i.e., Android ADB (TCP/5555), Chromecast (TCP/8443), AirPlay (TCP/7000), iOS iTunes Sync (TCP/62078), and Netflix (TCP/9080)) that are popular in IPv6 and are designed to run in the local network. We describe how we identify these services in Section 4.3. As these services are not designed to open on the Internet, they can serve as a strong sign of an open network. To explore these open networks, we seek to identify the last hop device (router) of these client-specific services, which safeguards end-host network security [52], and evaluate their security postures (e.g., filtering policy) and model/vendor fingerprinting (Section 4.3).

After each application-layer scan, we parse the responses and identify the hosts with any client-specific services. Similar to how we sample IP addresses to scan, we record networks with any client-specific services within 24 hours. If we find other scanned hosts in the same subnet, we perform traceroute experiments on those hosts to enlarge the coverage of hosts so that we can better construct the network structure, especially when we find multiple end hosts sharing the same last hop, indicating a shared gateway. We analyze the distribution of the number of collected IP addresses per various sizes of subnet (i.e., /64, /56, /48, /40) to determine the subnet level to look at, as shown in Figure 8 in Appendix A. We observe that /56 has a significant amount (27%) of subnets with multiple IP addresses; meanwhile, /48 has a shifted distribution. Given that /48 is not typically assigned to end-users [73], we select /56 as a subnet level for the traceroute measurement. We discuss limitations in Section 4.4.

We use Scamper [39], an open-source tool that supports traceroute measurements in parallel. We use the default UDP probes, which target a range of unassigned UDP ports (i.e., UDP/33435–33465). For other parameters, we set to probe at 1000 packets per second. We set 8 as the gap limit (i.e., the number of consecutive unresponsive hops before terminating probes to the target), 30 as the hop limit (i.e., the maximum number of hops to probe), and 3 probes per hop. Note that we use 30 hops as the limit as it covers

most of the routers (e.g., only 0.004% IPv6 routes reached the 30th hop). CAIDA's Ark platform also exemplifies that most paths do not exceed 30 hops [14].

From the traceroute results, we identify a last hop device of the target if both the target and the device at TTL-1 are responsive to the UDP probes. A limitation of this method is that it requires the evaluated hosts to have a public IP address; therefore, we are not able to measure IPv4 hosts behind NATs. However, we find it useful in IPv6 when NAT is not necessary.

4.3 Result Analysis

In this section, we discuss how we utilize our measurement results to explore the exposure of the end-host networks in IPv6, including identifying running services, detecting firewalls, applying heuristics to label networks, and analyzing the network infrastructure to understand what may cause openness.

In this work, we focus on analyzing a broader view of network openness instead of comparing openness between IPv6 and IPv4 of dual-stack hosts that are identified via DNS records in prior work [9]. Because end-hosts typically do not have public DNS records, they cannot be analyzed using this method (i.e., there is no record to match the IPv4 and IPv6 addresses). Thus, we compare aggregate network characteristics for IPv4 and IPv6.

Identifying services. When we analyze the potential services running on the top open ports in IPv6 and IPv4, we consider a combination of information (e.g., the assigned service by Internet Assigned Numbers Authority (IANA), official documentation, academic literature, experimentation with devices we obtain, and the collected HTTP and TLS responses).

We find that TCP/5555 is used by the Google ADB service and Google Chromecast uses TCP/8008, TCP/8009, and TCP/8443 from its official documents [21, 22]. Prior work additionally found that Netflix uses TCP/9080 on mobile applications, and iOS devices use TCP/62078 for iTunes data transfer [55, 70].

For the top ports without known services, we analyze the HTTP responses collected from our network scans (Section 4.2). We find the appearance of AirTune in the HTTP header of responses from TCP/7000, and we find Open AirPlay, a reverse-engineering version of AirPlay, documents the use of TCP/7000 [33].

To validate that a service is running on a port, we obtain a corresponding device and perform network scans on it (e.g., by scanning an Android phone running a Netflix application). We further record the application-layer responses of the services and fingerprint them in our measurement pipeline. Eventually, we identified **five client-specific services (i.e., ADB/5555, Chromecast/8443, iTunes Sync/62078, Netflix/9080, AirPlay/7000)** from the top open ports. We provide the details of each service in Section 5.3.

Understanding default firewall behavior. Identifying if a firewall is in place in a network is a key element to determine if a network is open. We experiment with the following OSes to determine their default behaviors: Android 14 and 10, iOS 18, iPadOS 18, macOS 14.4.1, Linux (Ubuntu 20.04), and Windows 10. After receiving a TCP SYN packet on a non-listening port, we note whether the system resets a connection (closed port) or drops the packet (filtered port). The results show that all of these OSes reset a connection with a TCP RST packet for a non-listening port, except Windows,

which has a default firewall that ignores such a connection. We discuss limitations in Section 4.4.

This behavior differs from common firewall software, such as iptables and ip6tables, that will drop the packet and do not reset the connections. Therefore, we can infer that a host with all closed ports lacks a firewall, and a host with all filtered ports would have a firewall when it does not have any open service. However, in practice, a firewall can be configured to reset certain ports to avoid disruption of services (e.g., it is common to reset traffic to port TCP/113 [67].) Furthermore, we may encounter packet loss and rate-limiting, i.e., we do not find a host with all the ports closed. Thus, we need a threshold to determine if a host has a firewall by looking at the distribution of the number of filtered ports per host.

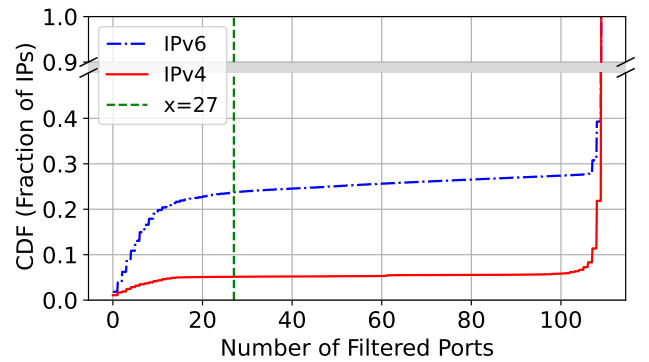


Figure 3: CDF of scanned IPs by number of filtered ports (number of scanned ports=109). The figures are based on the median fraction of IP addresses for each number of filtered ports per day, over our two-week measurement period. (Remark: the gray area marks the omitted y-axis range 0.5–0.9.)

Detecting firewalls. In our generic scan results, we observe that most of the hosts (i.e., 70% (IPv6) and 95% (IPv4)) have at least 100 filtered ports out of 109 scanned ports, as shown in Figure 3. These hosts are mostly filtered and leave few ports open or closed. We additionally observe that the CDF plateaus after the fraction of IP addresses crosses 0.25 (i.e., 27 filtered ports, the vertical line in Figure 3), which includes an insignificant amount of hosts that we are uncertain whether it is firewalled (e.g., filtering 50 ports on purpose) or open (e.g., we do not receive all the responses due to rate-limiting or packet loss). To be conservative, we treat them as filtered hosts, and therefore, we label a host as having a firewall (FW) if it has more than 25% of scanned ports being filtered; otherwise, it is labeled as open (OP).

Labeling open networks. Applying our firewall detection heuristic, we first label if a host is OP or FW. Then, we analyze two groups of networks (i.e., all networks and networks with client-specific services). For each type of network, we group the results by subnets of varying sizes and announced BGP prefixes. Then, we analyze the distribution of OP and FW hosts to label if a network is open. We present two approaches to label open networks, on opposite extremes: 1) a network with all OP IPs (the most restrictive), and 2) a network with at least one OP IP (the least restrictive). We exclusively

analyze networks with multiple IPs to avoid the potential bias created by a network with only one scanned IP. For the networks with client-specific services, since those services are not supposed to be exposed, these networks should have a higher degree of openness. Thus, we analyze the openness of these networks and compare them with the general scanned networks.

Router analysis. As the router is the guardian of the end-host network security, we aim to understand the routers from multiple perspectives, e.g., the filtering policy, the port openness, and the vendor/model. To measure the routers, we perform various measurements towards the last-hop devices of NTP clients (from our traceroute experiments in Section 4.2): TCP SYN scans with the top 1,000 ports with Nmap, followed by the application-layer connections to the same ports using HTTP. We use the port results to compare the relationship of the router filtering policy with the end-hosts in the same network. It offers insights into whether a filtering router affects the openness of the end-host, e.g., a router can have a firewall for itself, but not for the network, which causes the network being open. Afterward, we analyze the open ports to infer more information about the gateway devices (e.g., TCP/80 is the popular open port in IPv6, which is widely used by home routers for hosting a login page [69]). We seek to identify the vendor/model of the routers in the open networks by studying the HTTP payload, TLS certificates, and the OUI, derived from the EUI-64 IPv6 addresses. In addition, we use a fuzzy hashing algorithm, ssdeep, to cluster HTTP pages and manually label each cluster, which has been used in prior work when clustering Web interfaces of exposed ICS systems [63]. We compare the ssdeep hashes of a day of results with the rest of the days and found that 98.8% (median) of IPs per day have matched ssdeep hashes with those on July 8th, 2024. As the router types are similar across days, we present results of a single day in Section 6.2, Section 6.3, and Section 6.4.

4.4 Limitations

Our methods for host evaluations (i.e., IP address lifetime, IP grouping, and firewall detections) are empirical and will not be able to capture all possible device and network behaviors. However, our methods are evidenced to cover most cases in our study and are equipped with multiple approaches to mitigate potential limitations.

IP address lifetime. Using a 24-hour window for our IP sampling has two possible implications: (1) re-scanning the same device if its IP address churns slower than 24 hours; (2) over-counting the same device if it churns within 24 hours. In the first case, it does not affect our results as we analyze in 24-hour windows, so we do not count the same host multiple times. In the latter scenario, we avoid over-estimation by analyzing results with various subnet levels. For example, we analyze popular open ports that appear across BGP prefixes instead of counting the full IP addresses in Section 5. As address assignment does not often switch to a different BGP prefix, especially in IPv6 [53], even if IP addresses churn within 24 hours, as they share the same BGP prefix, we will not overcount the same open host multiple times. Moreover, Padmanabhan et al. [53] showed that churning within 24 hours is limited in both IPv6 and IPv4. If an open host appears to churn its IP address within 24 hours and switch to a different BGP Prefix, we will count them as a separate case. However, it is not common in practice. Therefore,

using 24 hours as a timeframe for IP sampling and result analysis is a reasonable threshold.

Subnet size. Similarly, each network has its practice of prefix assignment size [28]. For our traceroute experiment, we try to find hosts in the same network as those identified with client-specific services, which suggests an open network. However, it is challenging to have a one-size-fits-all solution that informs if multiple hosts are from the same end-host network based on their IP addresses. We empirically choose /56 for IPv6 as they have numerous IPs per subnet based on our data while not too broad as a subnet size for end-host networks. The potential implication is that we could have missed the end hosts in a network that has a size larger than /56.

When analyzing network openness, we group our results with multiple subnet sizes and BGP prefixes and identify consistent patterns across all of them. We offer a diverse picture of openness between IPv6 and IPv4 that avoids assuming a specific pair of subnet sizes when comparing two protocols.

Firewall detections. Our firewall detection heuristics rely on the default behavior of how the OS reacts to a TCP SYN packet to a non-listening port. However, one can configure a firewall to have a different behavior than the default (i.e., ignoring a connection instead of resetting it). In this case, we may miscount a firewall as open. To avoid over-estimation of network openness, one approach to evaluate networks is to only count a network as open if all the IPs do not have a firewall. We also assess networks with multiple IPs to reduce the chance of over-claiming openness. Despite the different ways we analyze the data, we still observe consistent results. Therefore, even if there are inaccuracies for some hosts/networks, the overall pattern remains the same.

Meanwhile, for Windows devices, although they do not reset a TCP connection by default, our NTP Pool-based clients are less Windows-based at the same time since they use a dedicated time server provided by Microsoft instead of using NTP Pool servers [26]. Our scan results resemble this assumption when we found fewer IPv4 clients open on Windows-related ports, compared to Censys data, which covers the entire IPv4 Internet (Section 5.1). We believe our analysis is unlikely to be affected by Windows devices.

5 Open Host Analysis

During the measurement period, we sampled and scanned a median of 10.1M IPv4 addresses and 9.2M IPv6 addresses a day. For IPv4, this spanned a daily median of around 278K BGP prefixes, 34K ASes, and, 240 countries/regions. For IPv6, this spanned a daily median of about 38K BGP prefixes, 7.5K ASes, and 212 countries/regions.

In this section, we present the result of our measured clients in terms of open ports and identified services. The port openness offers insight into a device's potential services and characterizes the type of device we are measuring [30]. In Section 5.1, we first compare our IPv4 scan results with Censys's dataset [6], where they continually scan the IPv4 address space on over 3,500 ports, to understand the representativeness of our NTP clients in terms of difference and similarity. In Section 5.2, we compare the port openness between IPv6 and IPv4. We find that the top open ports of IPv6 clients are distinct from IPv4. IPv6 clients are more likely to have client-centric services running on open ports than IPv4 that are identified in Section 5.3. The disparity of port openness

and available services between IPv6 and IPv4 signals the network exposure of IPv6, as we are able to discover client-specific services that are meant to use in a local network.

5.1 Comparing IPv4 Scan Results with Censys

To understand the representativeness of our method, we compare our IPv4 results to Censys [10]. As Censys regularly scans the entire IPv4 address space, their dataset is representative of all hosts on public IPv4 addresses. Given that it is not feasible to scan the entire IPv6 space, and that both IPv6 and IPv4 clients are from the NTP Pool, this allows us to infer how representative our IPv6 clients are.

Here, we compare the top open ports of the Censys dataset with our scan results from one day. We find that our results are similar overall, but acknowledge that we likely miss devices that do not use the NTP Pool (e.g., Windows devices and devices in regions where we do not have an NTP server).

Top open ports of NTP clients and Censys. We find that our IPv4 client top open ports are similar to those in Censys. Eight of the top 10 open ports are shared by both datasets (i.e., TCP port 443, 80, 22, 5060, 8443, 2000, 8080, 21). TCP/1723 (PPTP) and TCP/8000 (HTTP) are not in the top 10 in Censys, but still ranked in the top 20 (19th and 20th, respectively). These top ports are commonly used by customer-premises equipments (CPEs), e.g., routers, or for port-forwarding [54]. As Linux-based devices (e.g., routers and VPN servers) widely use the NTP Pool, this could result in a larger appearance of TCP/1723 and TCP/8000 in our results.

On the other hand, TCP/25 (SMTP) and TCP/3389 (MS WBT Server) are in the top 10 ports in Censys but not in the NTP Pool. However, these ports still rank high in the results. For example, TCP/25 ranks 13th, and TCP/3389 ranks 24th. This shows the similarities in the common ports seen across two different datasets.

Ports more prevalent in Censys than NTP. We compare the top ports on IPv4 hosts between our dataset and Censys and evaluate the differences proportionally by calculating the difference between the percentage of hosts with a port open on Censys and in our data. We identify three services that are represented less in our dataset compared to Censys. (1) **RPC (TCP/135), RDS (TCP/3389)**: These ports are used by Windows servers according to Microsoft documentation [43]. Windows devices, by default, use their own dedicated NTP servers managed by Microsoft [26]. Although users can configure a third-party NTP server, the default configuration might cause fewer devices to use the NTP Pool, and thus we capture fewer; (2) **HTTP (TCP/8008)**: 69% (1.93M/2.8M) of Censys's results with TCP/8008 open are from an AS in Korea, i.e., AS4766 (Korea Telecom). In contrast, we only scanned 4,798 IPs from AS4766 on the same day. We attribute this to our lack of servers in Korea, allowing us to collect few hosts; (3) **MySQL (TCP/3306)**: 30% of Censys results with this port open are from a single AS in Cyprus (1.59M/5.8M), AS47583 (AS-HOSTINGER). From the NTP pool, we only scanned 427 hosts in this AS. Again, we attribute this to our lack of servers in Cyprus.

To summarize, the top open ports are similar to those of our IPv4 scans and Censys. Although we find the limitations in our data due to vendors that use a dedicated NTP server or the locations of our servers, we determine our dataset is suitable for evaluating the port openness in IPv6.

5.2 Comparing Port Openness in IPv6 and IPv4

Since certain devices use a specific set of ports [31], we group the measured hosts by the open port combinations. Comparing the difference between the top port combinations between IPv6 and IPv4 can show the difference in potential types of devices that we are able to reach. We additionally group hosts by their announced BGP prefix to understand how our results change across networks. Thus, we also circumvent bias created by devices that are only popular in certain regions.

The results from both extensive and generic scans show significant differences in port openness between IPv6 and IPv4. Figures 4a and 4b show the top 10 open port combinations in IPv6 and IPv4 of scanned BGP prefixes across two weeks of extensive scans. The IPv6 top ports include TCP/5555, TCP/49152 and TCP/62078, TCP/9080, and TCP/7000, which are rarely seen in IPv4 (< 0.1% of scanned BGP prefixes). These ports are known to be used by services that run in local networks (e.g., TCP/5555 (ADB) [21], TCP/62078 (lockdown used by iOS devices) [55], TCP/9080 (Netflix [70])). In contrast, the top ports in IPv4 are TCP/443, TCP/80, TCP/2000, and TCP/22, which are commonly used by web servers/CPE devices [44].

In generic scans, where we scan 10 times more IP addresses, the top port combinations and the disparity between IPv6 and IPv4 retain. The top open port combinations are very similar between extensive and generic scans, as discussed in Appendix B. For example, the top port in IPv6, TCP/5555, is found in more than 10% of the scanned IPv6 BGP prefixes compared to nearly none in IPv4. In addition, a specific open port profile (TCP/8008, TCP/8009, TCP/8443) appears as one of the top port combinations in IPv6; meanwhile, rarely discovered in IPv4. The other top ports in IPv6 are similar to those found in extensive scans. The difference in port openness between IPv6 and IPv4 indicates that we reach different types of devices across the scanned networks, incentivizing us to study further the services running on the top IPv6 open ports.

5.3 Identified Services

As illustrated in Section 4, we use ZGrab [10] to establish application-layer connections to the top open ports in Section 5.2. From the responses of the top 10 IPv6 ports, we were able to identify five client-specific services (i.e., Android Debug Bridge, Chromecast, Netflix, iOS iTunes Sync, and AirPlay). These five services were chosen as strong indicators of end-host network exposure as they 1) should be local/LAN services rather than public services, and 2) were not common in IPv4 (< 0.1% of scanned BGP prefixes).

Android Debug Bridge (ADB), TCP/5555. TCP/5555 is the most popular open port in our scans in IPv6. It is commonly used by ADB for devices running on Android 10 or below [21]. Exposing ADB services to the public Internet can cause severe security issues because it allows one to perform remote code execution on exposed devices. For example, malicious actors had exploited the open ADB services in 2018 to spread botnet malware [38, 48]. To confirm that we find exposed ADB services instead of other services (e.g., Soft-ether VPN [49]) that use the same port, we developed a probe according to the protocol specification [17, 65] and incorporated it into our scanning pipeline. To filter the ADB responses, we match if the response payload has an ADB message command (i.e., SYNC, CNXN, AUTH, OPEN, OKAY, CLSE, WRTE). Table 1 shows the breakdown

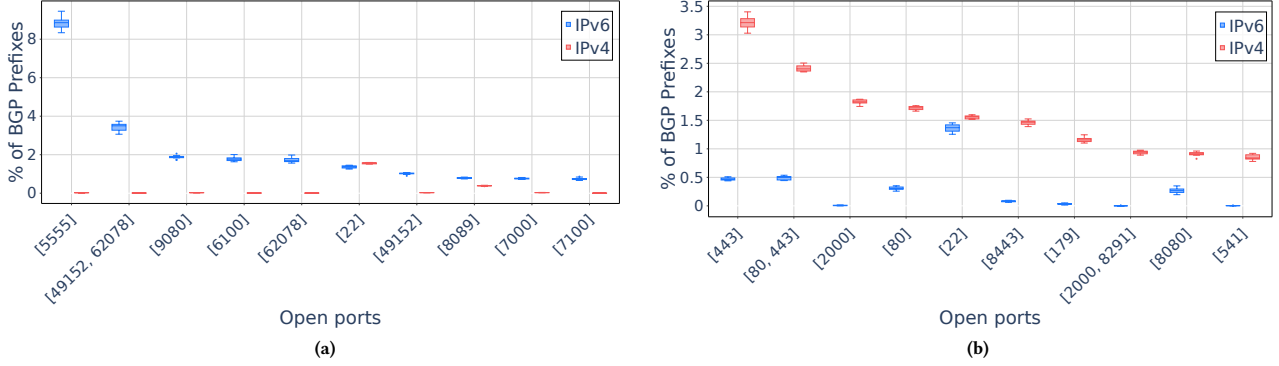


Figure 4: Top 10 open port combinations in IPv6 (a) & IPv4 (b) from extensive scans. In each case, we also plot the results in the other IP version for comparison. We highlight that it is more common to have only one port open, and the distribution of open ports is different in each case. (Remarks: Y-axes are truncated, and IPv6 results are on the left side for each port combination.)

of the daily responses of the ADB probes on TCP/5555 over two weeks of results. The results show that most of the hosts opening on TCP/5555 in IPv6 are indeed ADB devices (31,567 hosts or 88.8% of the probed hosts), as opposed to only 0.4% (43) of probed hosts with an open port on TCP/5555 in IPv4. In IPv6, we also found that many ADB devices (14,904/41.8%) are open to connections without requiring user authentication. Furthermore, all of those ADB devices offer a feature of `shell_v2` to the ADB clients, which grants one shell access to the device remotely. This would cause severe security issues as one can simply take over the device, run remote code execution, and potentially infect it with malware, as in previous incidents [38, 48].

	IPv6	IPv4
Number of probes	35,670.5	11,360.5
ADB responses (%)	31,567.5 (88.8%)	43.0 (0.4%)
CNXN (%)	14,904.5 (41.8%)	6.5 (0.1%)
CNXN, <code>shell_v2</code> (%)	14,904.5 (41.8%)	6.5 (0.1%)
AUTH (%)	16,600.0 (46.9%)	34.0 (0.3%)
Non-ADB responses (%)	4,078.5 (11.2%)	11,319.0 (99.6%)

Table 1: Breakdown of daily median ADB responses over our two-week measurement period.

Chromecast, TCP/8443. Chromecast is a device that enables other devices (e.g., smartphones and laptops) in the same network to control and stream content on the smart TV. Google TV has a built-in Chromecast service [20]. Chromecast is known to use TCP/8008, TCP/8009, and TCP/8443, according to the official documentation [22, 23]; meanwhile, it suggests not forwarding them to the Internet. Exposing it to the public Internet can cause a denial of service (e.g., an unauthorized user can turn off a TV or cast a specific video content on the TV, which happened in the past as known as CastHack [68]). We identify a Chromecast service by checking if the issuer’s distinguished name of the collected server certificate from TLS/8443 matches the Chromecast one.

Netflix, TCP/9080. Wu et al. [70] showed the use of TCP/9080 in the Android Netflix application. We confirmed this by scanning our devices (Android 15) when it was running Netflix, and observed traffic with that port when casting video. Furthermore, we discovered that TCP/9080 accepted a payload sent to an endpoint of `(/ddr)` with specific JSON data. While the usage of the API is not documented, we found that one can cause Netflix to crash by sending a malformed payload remotely. Leaving the Netflix application exposed publicly can be susceptible to a denial-of-service attack. The Netflix security team has acknowledged our findings and planned for remediation. To fingerprint this service safely, we chose not to elicit responses using an undocumented API. Instead, we sent an HTTP GET request to TCP/9080 and found that it replied with an HTTP status of 204 No Content. Thus, we incorporated this into our pipeline to filter out hosts with a different HTTP response.

iOS devices, TCP/62078. TCP/62078 is used by `lockdownd` in iOS devices, and this feature has been used for OS fingerprinting [55]. The port is used to transfer files between a macOS and an iOS device over a Wi-Fi connection. Leaving TCP/62078 on the Internet might have potential security implications as `lockdownd` controls access to the device. One can send an Apple’s proprietary payload to TCP/62078 to elicit a device’s information (e.g., OS version). We scanned an iPhone and an iPad and confirmed that they open on TCP/49152 and TCP/62078. IANA does not assign these two ports, so it is less likely that a non-iOS device has opened the same two ports. We regard a host as an iOS device if it only opens TCP/49152 and TCP/62078. Similar to fingerprinting Netflix, as `lockdownd` is not officially documented, to avoid potential disruption of services, we refrained from establishing application-layer connections to TCP/62078. We note that while Apple devices use their own time servers for NTP [2, 46], we identify that their NTP domain (`time.apple.com`) can sometimes resolve to NTP Pool servers¹, particularly in non-US ISPs, resulting in some Apple devices connecting with our NTP Pool servers.

¹For example, when we queried for `time.apple.com` using an open DNS resolver in FPT Telecom in Vietnam, one of the IP addresses returned had reverse DNS records pointing to NTP Pool domain names (i.e., `2.asia.pool.ntp.org` and `hk.pool.ntp.org`).

AirPlay, TCP/7000. When we analyzed the HTTP response from TCP/7000, we found that the HTTP header server field was related to AirTunes. It was related to AirPlay for streaming audio/video to TVs or home audio systems [33]. One might be able to exploit AirPlay vulnerabilities [1] at scale. To identify such a service, we checked the appearance of AirTunes in the HTTP header field of a response from an open port TCP/7000.

IPv6				
Service	IP	BGP Prefix	AS	Region
ADB	31567.5 (0.3%)	4632.5 (12.3%)	2378.0 (31.8%)	75.0 (35.3%)
Airplay	1075.5 (<0.1%)	124.0 (0.3%)	69.0 (0.9%)	30.0 (14.3%)
Chromecast	7675.0 (0.1%)	883.0 (2.4%)	206.0 (2.7%)	53.5 (24.9%)
iTunes	33910.0 (0.4%)	2557.0 (6.8%)	722.5 (9.7%)	91.0 (42.9%)
Netflix	4803.0 (0.1%)	1463.5 (3.9%)	478.0 (6.4%)	71.0 (33.5%)
Total	78468.0 (0.9%)	6705.0 (17.8%)	2704.5 (36.1%)	107.5 (50.6%)

IPv4				
Service	IP	BGP Prefix	AS	Region
ADB	43.0 (<0.1%)	38.0 (<0.1%)	27.5 (0.1%)	12.0 (5.0%)
Airplay	20.0 (<0.1%)	20.0 (<0.1%)	15.0 (<0.1%)	7.0 (2.9%)
Chromecast	213.0 (<0.1%)	182.0 (0.1%)	75.5 (0.2%)	21.0 (8.8%)
iTunes	241.5 (<0.1%)	154.0 (0.1%)	76.0 (0.2%)	22.5 (9.4%)
Netflix	10.5 (<0.1%)	9.0 (<0.1%)	7.0 (<0.1%)	6.0 (2.5%)
Total	522.0 (<0.1%)	391.0 (0.1%)	145.0 (0.4%)	34.5 (14.3%)

Table 2: Distribution of client-specific services by IP, BGP prefix, AS, and region. Numbers are the median of daily results over two weeks. Percentages are the fraction of scanned populations with a specific service.

Widespread Client-Specific Services in IPv6. We found that the client-specific services are widespread in IPv6 (Table 2). 78K IP addresses with those services span over 6.7K BGP prefixes, 2.7K ASes, and 107 regions per day. In contrast, they are much fewer in IPv4 (i.e., 522 IPs from 391 BGP prefixes, 145 ASes, and 34 regions). The top client-specific services are iTunes (33.9K/41.8%), ADB (31.6K/40.2%), and Chromecast (7.68K/9.78%) in IPv6. The same type of services are much less popular in IPv4 in comparison, where we found 241 iTunes (46.2%), 213 Chromecast (40.8%), and 43 ADB (8.2%).

Takeaways: IPv6's top open ports are distinct from those of IPv4, where we identified five client-specific services (e.g., ADB) not meant to be exposed to the Internet that are widespread in IPv6 (17x more BGP prefixes and 2.7x more ASes than IPv4).

6 Open Network Analysis

In this section, we combine findings on the host level (i.e., firewall behavior and identified client-specific services) to broadly analyze network openness on multiple subnet sizes and using BGP prefixes (Section 6.1). In Figure 5, we show the distribution of IP addresses across different subnet sizes and BGP prefixes, illustrating how different subnet sizes exhibit varying populations (hence why we evaluate multiple subnet sizes). Then, we investigate IPv6 networks that are classified as open (i.e., with client-specific services and

lacking in filtering). Here, instead of grouping addresses by various subnet sizes and BGP prefixes, we group addresses by their last hop router identified with traceroute. We perform three analyses of these open IPv6 networks: the address relationship between the end hosts and their last hop router (Section 6.2), comparing the filtering policy of last hop router and end hosts (Section 6.3), and fingerprinting common last hop router models (Section 6.4).

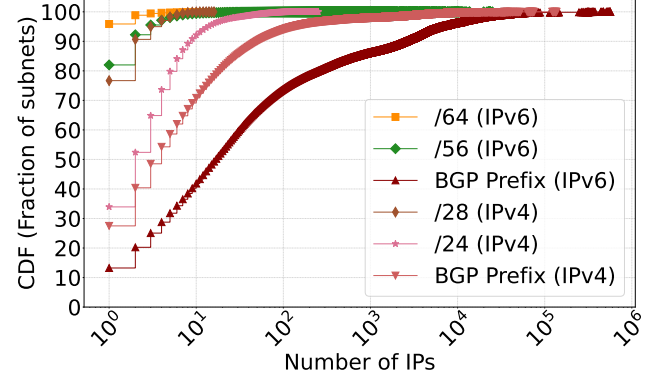


Figure 5: CDF of subnets by number of scanned IP addresses

6.1 IPv6 and IPv4 Network Openness

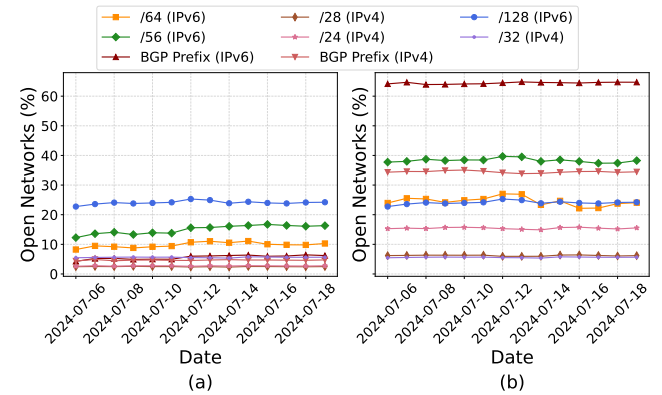


Figure 6: Network openness of scanned networks evaluated with two open network labeling methods, (a): networks with all open IP addresses, (b): networks with at least one open IP address

Figure 6 shows the percentage of open networks in IPv6 and IPv4 over two weeks of scan results, under both definitions of open networks (for networks with multiple IP addresses). At a high level, for both openness definitions, IPv6 is consistently more open than IPv4 across various subnet sizes. When conservatively considering networks with all open IP addresses (Figure 6a), the median percentage of open networks in IPv6 is around 10% of /64s and 15% of /56s, which is 4-8 times higher than for all IPv4 subnet sizes (i.e., /24, /28). When using the less restrictive open network definition of a single open address (Figure 6b), we see the

same pattern. About 25% of /64s and 38% of /56s are open in IPv6, compared to 15% of /24s and 7% of /28s in IPv4. Beyond subnets, we also include full addresses (i.e., /128 for IPv6, /32 for IPv4) in Figure 6 as a reference (particularly as NAT is widely deployed in IPv4 so /32 addresses often represent networks). Comparing the openness of end-sites, IPv6 hosts are still more open than IPv4. About 25% of IPv6 hosts are open, compared to around 5% in IPv4.

In addition, we analyze results using the BGP prefix. Under the conservative open network definition, the fraction of open networks is more similar between IPv6 and IPv4 when using BGP prefixes, although IPv6 is consistently still more open. (We also note that, as shown in Figure 5, BGP prefixes tend to have more addresses in IPv6 than IPv4, so the conservative definition is stricter for IPv6 BGP prefixes.) Under the more liberal definition (at least one open IP), we can differentiate that IPv6 BGP prefixes are nearly 2 times more open than IPv4, as seen in Figure 6b. Around 65% of IPv6 BGP prefixes have at least one open IP address in IPv6 compared to 35% for IPv4 BGP prefixes. While it is unclear how to align subnet sizes across IPv4 and IPv6, we observe broadly and consistently that IPv6 exhibits a higher degree of network openness.

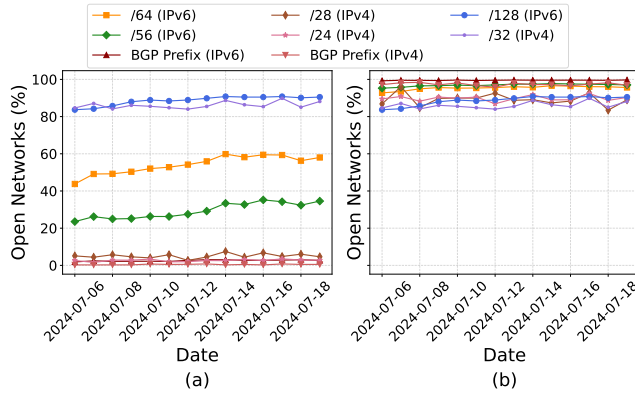


Figure 7: Network openness of scanned networks with client-specific services evaluated with two open network labeling methods, (a): networks with all open IP addresses, (b): networks with at least one open IP address

Significance of client-specific services. We apply the same analysis to the networks with client-specific services, as shown in Figures 7a and 7b. Compared to the general scanned networks, we find that networks with client-specific services are much more open with both the most and the least restrictive open network definitions. For example, around 58% of /64s with client-specific services (and multiple IP addresses) have all open IP addresses, as shown in Figure 7a, around 6 times higher than the general scanned networks. When we consider networks with at least one open IP address (Figure 7b), both IPv6 and IPv4 networks are nearly all open. Under both openness definitions though, IPv6 networks again remain consistently more open than IPv4 networks. The results exemplify that a network with client-specific services is likely to be open; thus, we further analyze this type of network in the following sections.

6.2 IPv6 Open Network Infrastructures

To further investigate networks with client-specific services, we try to locate the routers in these networks by using traceroute to the end-hosts identified with a client-specific service. As mentioned in Section 4.3, we analyzed the results of a day (July 8th, 2024) and found it similar to other days over the measurement period. Thus, we present the results of a single day in the following sections (i.e., Sections 6.2, 6.3, and 6.4).

In our traceroute experiments, 46.3% (35K) of probes were successful (i.e., both the end-host and its last hop are responsive) in IPv6. We were able to map 35K end-hosts to 31.7K routers. We group results by their last hop and apply the same heuristics used in Section 6.1 to label open networks (note many networks based on these last hop routers have only one address). We find 28.4K open networks (89.6%), a high proportion similar to results in Section 6.1.

When we compare the addresses between the end-hosts and their last hops, the most common shared prefix is /64, accounting for 31.5% (8,962) of networks from 405 ASes. The top AS with this type of network (/64s) was AS12322 (Free SAS), a major ISP in France, which contributes 40% of these networks, followed by one of the largest ISPs in Japan, Softbank Corp. (AS12322), at 28.4%. The second and the third most common shared prefixes between the router and the end host are /47 and /32, which account for 7.76% and 7.58% of the networks respectively. The top ASes are Etisalat (AS8966) in UAE (/47), and V tal (AS7738) in Brazil (/32), which are popular telecoms in the region.

6.3 Security Policy of Routers and End-Hosts

Similar to the scans to NTP clients, we run port scans with Nmap's top 1,000 TCP ports to the routers in these networks in IPv6. Applying the firewall detection heuristics (Section 4.3), we suspect over one-third of routers (10.3K) lack a firewall.

When we compare the filtering policy between routers and end-hosts, for all networks (including those with only one address behind a router), over 93.3% of end-hosts and their last hops are both non-filtered (i.e., no firewall). This shows how the lack of a firewall in routers correlates with or impacts the security policy of the end hosts. On the other hand, considering routers with a firewall, we find only around 11.6% of hosts behind them are filtered. This is a contrary case where a router implements filtering rules only for itself, but not for the network. Both situations demonstrate how routers are not firewalling their internal networks.

Next, we analyze networks with multiple IPs to exclude skew from those with a single IP address, and the results are similar. Of approximately 1K networks with multiple IPs, around 35.9% of routers are firewalled. Most of the hosts (86.6%) behind those routers likewise are classified as lacking a firewall. Only 4.27% of hosts with a firewalled router are likewise appearing as firewalled.

In IPv6, we find that overall 23.8% of the scanned routers have at least one open port. The top open port combinations are TCP/80 (4.8%), TCP/53, 80, 443 (4.5%), and TCP/21 (1.78%), which are commonly used by home routers.

6.4 Identifying Router Models

We analyze 28K routers in IPv6 open networks. To identify a router's vendor and model, we consider HTTP Header/Body payloads and

Subject/Issuer Distinguished Names of TLS certificates, collected from the application-layer connections. We also look up the OUI of EUI-64 addresses, which indicates the device manufacturer.

Top identified vendors and models. Of 28K routers, we identify 97 vendors, which covers around 19.9% (5,651) of routers. Furthermore, we identified 63 models that include 9.3% (2,647) of scanned routers. The top vendors are D-LINK (1,343), Intelbras (995), Huawei (831), Nokia (662), ZTE (249), and FiberHome (211). On the other hand, the top router models are D-LINK DIR-853/ET (1,284), Nokia GPON Home Gateway (450), Huawei HG8145V5 (139), VNPT iGate GW040-H (123), Greatek GWR-1200AC (99), and FPT ISP CPE (94). We notice that some of these router models are ISP-specific (e.g., D-LINK DIR-853/ET is a specific model for Etisalat, an ISP in UAE, and VNPT iGate GW040-H is used by VNPT, a telecom in Vietnam).

Patterns of router models across ASes. The routers with an identified vendor are distributed over 1,131 ASes. We observe that some router vendors and models appear consistently within an AS, indicating that they may be CPEs offered by the ISP. Among 552 ASes with multiple routers, 59% of ASes have more than 90% of routers sharing the same vendor, and 80.8% of ASes have half of the routers with the same vendor. Similarly, for identified router models across 322 ASes, 78.6% of ASes with multiple routers (110) have more than 90% of routers sharing the same model. Furthermore, 91.4% of them have over half of the routers with the same model. For example, in AS8966 (Emirates), 98.7% (1,284) of routers are D-LINK DIR-853/ET; in AS45899 (VNPT Corp), 86.5% (122) are VNPT iGate GW040-H, in AS17072 (Totalplay), 84% (79) are Huawei HG8145V5.

Security concern of ISP CPEs. Interestingly, we also find numerous router models that are only found in a single AS. These 32 router models cover 1,477 devices. Among them, 18 models are found with multiple devices. These routers are likely the ISP CPEs, due to its unique appearance in a single AS, and we suspect these devices potentially lack a firewall for IPv6 by default, given that we find the prevalence of the same model of routers within the same AS. For example, D-LINK DIR-853/ET is a specific model made for Etisalat, according to D-Link documentation [7], and it is found only in AS8966 (Etisalat); VNPT iGate routers are found only in AS45899 (VNPT Corp).

While acquiring each of these router models for testing is challenging, we instead review the device user manuals and identify the default IPv6 firewall settings. For example, we found multiple Galaxy 5G Mobile Wi-Fi SCR01 routers, and all from AS2516 (KDDI). This particular device is a product of the ISP operated by KDDI. The router user guide [8] shows that the default firewall configuration does not include IPv6. The lack of an IPv6 firewall by default could cause widespread exposure to networks that use this device model.

Takeaways:

- IPv6 networks are substantially more open than IPv4 ones, regardless of considered network granularities.
- Network openness appears independent of the network router's own filtering or firewall policy.
- Common routers (e.g., ISP-specific CPEs) are observed in open IPv6 networks, suggesting that poor default router configurations lead to many open IPv6 networks.

7 Concluding Discussion

In this work, we explored the exposure of IPv6 end-hosts, identifying significant security concerns arising from the widespread lack of firewalling or filtering in IPv6 networks. Such a phenomenon likely stems from the adoption and deployment of networking hardware that, while having implicit security behaviors (i.e., NAT) in IPv4, lacks explicit security protections in IPv6. These exposures put IPv6 networks and devices at risk. Here, we reflect on the implications of our findings and synthesize recommendations for the security community to improve Internet security moving forward.

Differences Between IPv4 and IPv6. While the differences between IPv4 and IPv6 do not contribute directly to observed security differences, the ecosystem, and the maturity of devices and configurations around these networks yield a vastly different security posture. We note a parallel exists between the state of the IPv4 Internet in the worm era before the large-scale deployment of NAT and firewalls, and the nascent state of large-scale IPv6 deployments. If lessons from securing the IPv4 Internet are not transferred (and adapted as necessary) to IPv6, history may repeat itself with widespread Internet security incidents.

Network Firewalls/Filtering. Our work reveals a significant lack of firewalls and filtering at the network level in end-host—especially residential—networks. This points to an urgent need to bring consumer expectations of accessibility that arose out of implicit security technologies into line with the actual explicit security posture of these devices and networks. Network firewalls and filtering are particularly salient as they encapsulate the entire device population on the network, shielding even end-hosts lacking security controls.

End-Host Firewalls. The service exposure identified in this work also points to a lack of end-host firewalling in the IPv6 context, which is particularly concerning for oft-vulnerable hosts such as IoT devices. Furthermore, distinct classes of devices that are rarely publicly exposed in IPv4, such as smartphones, may now be widely reachable in IPv6. Previously, this lack of end-host firewalling was largely obscured by the inaccessibility of such devices. Our work points to an urgent need for routers and networks to adhere to basic security best practices, such as enabling end-host firewalls, even on devices not expected to be on the public Internet.

Visibility of IP Addresses. The vast size of the IPv6 address space affords some limited protection to exposed services. However, recent advances in IPv6 Internet Scanning, as well as the exposure of client IPs by virtue of basic Internet activity (e.g., the use of NTP [61]), reduce the protections of address space size, requiring urgent changes in how networks are secured. We note that the size of the IPv6 address space affords creative use of addressing to mitigate these risks; services not intended for public access could bind to different privacy-preserving addresses that are used for client activity, adding an additional layer of obfuscation. Such changes would help mitigate the exposure risk of IPv6 networks, but regardless, urgent configuration, and firewall changes are needed to ensure the security of the IPv6 Internet.

8 Acknowledgements

We thank the anonymous reviewers for their constructive feedback. This work was supported by the National Science Foundation (NSF) CNS Award 2247514 and 2319315, NSF CAREER Award 2239183, and Graduate Research Fellowship (GRFP) under Grant No. DGE-2545526. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Ethical Considerations

As our study entails large-scale Internet measurements, we consider our experiments in the context of the ethical principles from the Menlo report [4]: Beneficence, Respect for Persons, Justice, and Respect for Law and Public Interest.

Beneficence. Experiments with inherent risk must reduce the risk such that the benefits outweigh the potential negatives. We believe that our findings and disclosures will significantly improve users' IPv6 security and mitigate harms to end-users stemming from a misconfigured IPv6 firewall. We disclosed issues to three stakeholders: vulnerable service vendors, network operators, and router vendors. We received some positive responses, such as Netflix acknowledging our report and indicating that they plan to fix it. Our disclosures to network operations and router vendors aimed at broadly increasing the adoption of firewalls on IPv6 (by default), protecting vulnerable end-user services.

Meanwhile, we minimized potential harm to individuals by rate-limiting scans and randomizing scan order to avoid overloading specific network blocks, consistent with best practices [11]. Additionally, we did not perform vulnerability scanning, such as attempting authentication or executing/testing exploits. Instead, we only performed TCP-SYN scans and application-layer scans such as HTTP GET requests or handshakes. Lastly, our NTP servers were configured following the NTP Pool's recommendations [57] and provided high-availability NTP service to ensure we did not provide degraded NTP service that causes harm to users.

Respect for Persons. It is impossible to obtain informed consent from the individual users that receive Internet scans. Rather, we focused on identifying ourselves and providing means to terminate scanning. Keeping with the norms and practices of prior studies, we communicated the research nature of our work in a couple forms: 1) Our scanning machines were configured with reverse DNS and WHOIS records that indicate who we are and our research intent, and hosted webpages that convey who we are, what we do, and how to opt-out of scans (we received 13 requests that we promptly complied), consistent with best practices [11]; 2) Our servers were grouped into the same organization inside the NTP Pool system, which provided a contact email, organization, and communicated research intent via naming.

Additionally, we sought to follow similar procedures as prior work that collected and studied client IP addresses, including from the NTP Pool [50, 61] and BitTorrent [60], albeit with the addition of active scanning post-collection. We complied with the NTP Pool Terms-of-Service [59], and ensured our servers performed the intended NTP operations correctly. We believe our NTP servers positively contributed to the NTP Pool capacity without impacting other NTP servers or operators.

Justice. Our large-scale study was global and did not filter by individual characteristics (besides balancing across geographic regions), and the results benefit the Internet as a whole. Furthermore, the IP addresses studied, who bear the risks from our scans, are among those that specifically benefit from our findings and disclosures.

Respect for Law and Public Interest. We are transparent in methods and results, and we are accountable for actions, especially through publishing our work. We do not expose any specific IP addresses. We have conducted responsible disclosure to both the vendors of the affected routers with the configuration issues we identified and the networks we found to be exposed. Our work reveals the widespread exposure of IPv6 networks, which raises significant security concerns whose resolution is a public good. These issues will likely require community-wide efforts to resolve, which this research enables.

We briefly note that our institution's IRB does not consider this human subject research, as IP addresses and Internet hosts are not considered human subjects.

Considerations for Research using the NTP Pool. After our measurements, we became aware of prior NTP Pool-based scanning involving Shodan. Shodan had scanned the NTP Pool clients in 2016 [19], and they were removed by the NTP Pool maintainer. However, we note that our use of the NTP Pool, methods, data usage, and intent differ significantly. Shodan scanned in order to display full addresses and scan results publicly on their website, with a goal of monetizing scan data. They also did not have an opt-out process. In contrast, our measurement methods were carefully designed around ethical considerations (e.g., rate-limiting the scan rate, offering opt-outs, limiting application-layer scans for gathering banner information only, and signaling our research intent and identity through DNS records), and we only analyzed aggregate statistics for the purposes of improving users' IPv6 security and disclosing our findings to the stakeholders. We have not, and will not, make scan addresses or specific scan results public. Broadly, we believe our study differs significantly from the Shodan incident, and that its benefits outweigh the potential risks.

Due to the public and distributed nature of the NTP Pool, where anyone could join, and that our method abides by existing best practices [11, 57, 59], we did not initially reach out to the lead NTP Pool maintainer prior to performing our experiments. However, through community discussions after the research, we learned of potential NTP Pool maintainer concerns with active scanning of NTP clients. We therefore contacted the lead NTP Pool maintainer. We disclosed our research by sharing our paper draft and describing our work, its method, and its research goals. We sought the maintainer's feedback on our work, suggestions for any actions to take, and perspectives on how the research community should engage with the NTP Pool in the future. The lead NTP Pool maintainer responded acknowledging our disclosure email, and ultimately, no concerns have been raised to date.

In lieu of best practices directly from the maintainers, we point to recent communication from other researchers who have previously engaged with the NTP Pool as initial guidance [36]. This communication suggests that future research carefully consider how it uses the NTP Pool to collect client data, especially with

active measurements, and engage the NTP Pool maintainer as a stakeholder prior to executing a study.

Open Science

We provide open access to our study's source codes at <https://github.com/IPv6-Security/v6-exposure>, including the scanning pipeline described in Section 4, and result analyses in Section 5, and Section 6.

References

- [1] Apple. 2025. About the security content of visionOS 2.3 - Apple Support. <https://support.apple.com/en-us/122073>.
- [2] Apple. 2026. Use Apple products on enterprise networks - Apple Support. <https://support.apple.com/en-us/101555>.
- [3] Internet Assigned Numbers Authority. 2025. Service Name and Transport Protocol Port Number Registry. <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>.
- [4] Michael Bailey, David Dittrich, Erin Kenneally, and Doug Maughan. 2012. The Menlo Report. *IEEE Security & Privacy*.
- [5] 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 *IEEE Symposium on Security and Privacy*.
- [6] Censys. 2025. Research Access to Censys Data. <https://docs.censys.com/docs/research-access-to-censys-data>.
- [7] D-Link Corporation. 2025. D-Link Wi-Fi ET on the App Store. <https://apps.apple.com/ae/app/d-link-wi-fi-et/id1459210558>.
- [8] KDDI Corporation. 2021. Galaxy 5G Mobile Wi-Fi User Guide. https://www.au.com/content/dam/au-com/support/service/mobile/guide/manual/scr01/pdf/scr01_torisetu_shousai.pdf.
- [9] 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 *Network and Distributed System Security Symposium*.
- [10] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *22nd ACM Conference on Computer and Communications Security*.
- [11] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security Symposium*.
- [12] Facebook. 2024. Facebook IPv6 | Total Adoption. https://www.facebook.com/ipv6/?tab=ipv6_total_adoption.
- [13] Center for Applied Internet Data Analysis. 2024. Routeviews Prefix to AS mappings Dataset (pfx2as) for IPv4. <https://www.caida.org/catalog/datasets/routeviews-prefix2as/>.
- [14] Center for Applied Internet Data Analysis. 2025. Archipelago (Ark) Measurement Infrastructure. <https://www.caida.org/projects/ark/>.
- [15] G. Huston. 2015. IPv4 Address Exhaustion in APNIC. *APNIC Blog*. <https://blog.apnic.net/2015/08/07/ipv4-address-exhaustion-in-apnic/>.
- [16] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. In *International Workshop on Traffic Monitoring and Analysis*.
- [17] Benoit Goby. 2012. adb/protocol.txt - platform/system/core - Git at Google. <https://android.googlesource.com/platform/system/core/+f4ed516643ee8ed3a59ad1a8048f7ce5f47f93fb/adb/protocol.txt>.
- [18] Fernando Gont, Suresh Krishnan, Thomas Narten, and Richard P. Draves. 2021. Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6. RFC 8981. doi:10.17487/RFC8981
- [19] Dan Goodin. 2016. Using IPv6 with Linux? You've Likely Been Visited by Shodan and Other Scanners. *Ars Technica*. <https://arstechnica.com/information-technology/2016/02/using-ipv6-with-linux-youve-likely-been-visited-by-shodan-and-other-scanners/>
- [20] Google. 2024. About Chromecast with Google TV - Google Fiber Help. <https://support.google.com/fiber/answer/12244429?hl=en>.
- [21] Google. 2024. Android Debug Bridge (ADB) | Android Studio | Android Developers. <https://developer.android.com/tools/adb>.
- [22] Google. 2024. Google Nest & Chromecast security tips - Chromecast Help. <https://support.google.com/chromecast/answer/9216542?hl=en>.
- [23] Google. 2024. Network requirements for cast moderator - Chrome Enterprise and Education Help. <https://support.google.com/chrome/a/answer/12256492?hl=en>.
- [24] Google. 2026. IPv6 - Google. <https://www.google.com/intl/en/ipv6/statistics.html>.
- [25] Sarthak Grover, Mi Seon Park, Srikanth Sundaresan, Sam Burnett, Hyojoon Kim, Bharath Ravi, and Nick Feamster. 2013. Peeking behind the NAT: an empirical study of home networks. In *ACM Internet Measurement Conference*.
- [26] Robin Harwood. 2023. Windows Time service tools and settings | Microsoft Learn. <https://learn.microsoft.com/en-us/windows-server/networking/windows-time-service/windows-time-service-tools-and-settings?tabs=config>.
- [27] Fahad Hilal, Patrick Sattler, Kevin Vermeulen, and Oliver Gasser. 2024. A First Look At IPv6 Hypergiant Infrastructure. In *International Conference on Emerging Networking EXperiments and Technologies*.
- [28] Amanda Hsu, Frank Li, and Paul Pearce. 2023. Fiat Lux: Illuminating IPv6 Apportionment with Different Datasets. In *ACM Measurement and Analysis of Computing Systems*.
- [29] Geoff Huston. 2008. The changing foundation of the Internet: Confronting IPv4 address exhaustion. *The Internet Protocol Journal*.
- [30] Liz Izhikevich, Renata Teixeira, and Zakir Durumeric. 2021. LZr: Identifying Unexpected Internet Services. In *USENIX Security Symposium*.
- [31] Liz Izhikevich, Renata Teixeira, and Zakir Durumeric. 2022. Predicting IPv4 Services across All Ports. In *ACM Special Interest Group on Data Communication*.
- [32] Peter Jose, Said Jawad Saidi, and Oliver Gasser. 2023. Analyzing IoT Hosts in the IPv6 Internet. *arXiv preprint arXiv:2307.09918*.
- [33] Ben Klein. 2024. Service Discovery - Unofficial AirPlay Specification. https://openairplay.github.io/airplay-spec/service_discovery.html.
- [34] Michael Klopsch, Constantin Sander, Klaus Wehrle, and Markus Dahlmans. 2025. Time To Scan: Digging into NTP-based IPv6 Scanning. In *ACM Internet Measurement Conference*.
- [35] World IPv6 Launch. 2024. IPv6 Adoption, World IPv6 Launch. <https://www.worldipv6launch.org/measurements/>.
- [36] Dave Levin. 2026. IPv6 Address Learning via NTP Pool Servers: A Cautionary Tale. https://www.linkedin.com/posts/dave-levin-658b2564_the-ipv6-observatory-activity-7446545607957082112-k6V0.
- [37] Xiang Li, Baojun Liu, Xiaofeng Zheng, Haixin Duan, Qi Li, and Youjun Huang. 2021. Fast IPv6 network periphery discovery and security implications. In *IEEE/IFIP International Conference on Dependable Systems and Networks*.
- [38] Hubert Lin, Lorin Wu, Vit Sembera, Chunbo Song, and Tim Yeh. 2018. Open ADB Ports Used to Spread Possible Satori Variant | Trend Micro (US). https://www.trendmicro.com/en_us/research/18/g/open-adb-ports-being-exploited-to-spread-possible-satori-variant-in-android-devices.html.
- [39] Matthew Luckie. 2010. Scamper: a scalable and extensible packet prober for active measurement of the Internet. In *ACM Internet Measurement Conference*.
- [40] Gordon Lyon. 2024. TCP SYN (Stealth) Scan (-sS) | Nmap Network Scanning. <https://nmap.org/book/synscan.html>.
- [41] Gregor Maier, Fabian Schneider, and Anja Feldmann. 2011. NAT usage in residential broadband networks. In *Passive and Active Measurement Conference*.
- [42] MaxMind. 2024. GeoIP2 Databases | MaxMind. <https://www.maxmind.com/en/geoip2-databases>.
- [43] Microsoft. 2025. Service Overview and Network Port Requirements - Windows Server | Microsoft Learn. <https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/service-overview-and-network-port-requirements>.
- [44] MikroTik. 2022. Manual:IP/Services - MikroTik Wiki. <https://wiki.mikrotik.com/wiki/Manual:IP/Services>.
- [45] Daniel Miller. 2024. nmap/nmap-services at master · nmap/nmap. <https://github.com/nmap/nmap/blob/master/nmap-services>.
- [46] Giovane C. M. Moura, Marco Davids, Caspar Schutjijser, Cristian Hesselman, John Heidemann, and Georgios Smaragdakis. 2024. Deep Dive into NTP Pool's Popularity and Mapping. In *ACM Measurement and Analysis of Computing Systems*.
- [47] Austin Murdoch, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. 2017. Target Generation for Internet-wide IPv6 Scanning. In *ACM Internet Measurement Conference*.
- [48] 360 Netlab. 2018. ADB Miner: More Information. <https://blog.netlab.360.com/adb-miner-more-information-en>.
- [49] Daiyuu Nobori. 2024. Specification - SoftEther VPN Project. <https://www.softether.org/3-spec>.
- [50] Sadia Nourin, Erik Rye, Kevin Bock, Nguyen Phong Hoang, and Dave Levin. 2025. Is Nobody There? Good! Globally Measuring Connection Tampering without Responsive Endhosts. In *IEEE Symposium on Security and Privacy*.
- [51] Erik Nygren. 2022. Akamai Blog | 10 Years Since World IPv6 Launch. <https://www.akamai.com/blog/trends/10-years-since-world-ipv6-launch>.
- [52] Karl Olson, Jack Wampler, and Eric Keller. 2023. Doomed to Repeat with IPv6? Characterization of NAT-centric Security in SOHO Routers. *Comput. Surveys*.
- [53] Ramakrishna Padmanabhan, John P. Rula, Philipp Richter, Stephen D. Strowes, and Alberto Dainotti. 2020. DynamIPs: Analyzing Address Assignment Practices in IPv4 and IPv6. In *International Conference on Emerging Networking EXperiments and Technologies*.
- [54] Thomas Papastergiou, Roberto Perdisci, and Manos Antonakakis. 2022. Returning to Port: Efficient Detection of Home Router Devices. In *IEEE Conference on Communications and Network Security*.
- [55] Vasile Claudiu Perta, Marco Valerio Barbera, and Alessandro Mei. 2014. Exploiting Delay Patterns for User IPs Identification in Cellular Networks. In *Privacy Enhancing Technologies*.

- [56] Playwright. 2024. Fast and reliable end-to-end testing for modern web apps | Playwright. <https://playwright.dev/>.
- [57] NTP Pool Project. 2024. Configuration recommendations for servers joining the pool. <https://www.ntppool.org/join/configuration.html>.
- [58] NTP Pool Project. 2024. [pool.ntp.org](https://www.ntppool.org/): The Internet Cluster of NTP Servers. <https://www.ntppool.org/>.
- [59] NTP Pool Project. 2024. Terms of Service. <https://www.ntppool.org/en/tos.html>.
- [60] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. 2016. A multi-perspective analysis of carrier-grade NAT deployment. In *ACM Internet Measurement Conference*.
- [61] Erik Rye and Dave Levin. 2023. IPv6 Hitlists at Scale: Be Careful What You Wish For. In *ACM Special Interest Group on Data Communication*.
- [62] Teemu Ryttilahti, Dennis Tatang, Janosch Köpper, and Thorsten Holz. 2018. Masters of Time: An Overview of the NTP Ecosystem. In *IEEE European Symposium on Security and Privacy*.
- [63] Takayuki Sasaki, Akira Fujita, Carlos H. Gañán, Michel van Eeten, Katsunari Yoshioka, and Tsutomu Matsumoto. 2022. Exposed Infrastructures: Discovery, Attacks and Remediation of Insecure ICS Remote Management Devices. In *IEEE Symposium on Security and Privacy*.
- [64] Internet Society. 2024. IPv6 Adoption, Internet Society Pulse. <https://pulse.internetsociety.org/en/technologies/>.
- [65] Callum Styan. 2024. [cstyan/adbDocumentation](https://github.com/cstyan/adbDocumentation): Better documentation of the ADB protocol, specifically for USB uses. <https://github.com/cstyan/adbDocumentation>.
- [66] G. Van de Velde, T. Hain, R. Droms, B. Carpenter, and E. Klein. 2007. *Local Network Protection for IPv6*. Technical Report.
- [67] Nicholas Weaver, Robin Sommer, and Vern Paxson. 2009. Detecting Forged TCP Reset Packets. In *Network and Distributed System Security Symposium*.
- [68] Zack Whittaker. 2019. Hackers hijack thousands of Chromecasts to warn of latest security bug. <https://techcrunch.com/2019/01/02/chromecast-bug-hackers-havoc>.
- [69] 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*.
- [70] Daoyuan Wu, Debin Gao, Rocky K. C. Chang, En He, Eric K. T. Cheng, and Robert H. Deng. 2019. Understanding Open Ports in Android Applications: Discovery, Diagnosis, and Security Assessment. In *Network and Distributed System Security Symposium*.
- [71] ZGrab. 2025. [zmap/zgrab2](https://github.com/zmap/zgrab2). <https://github.com/zmap/zgrab2>.
- [72] Maya Ziv, Liz Izhikevich, Kimberly Ruth, Katherine Izhikevich, and Zakir Durumeric. 2021. ASdb: A System for Classifying Owners of Autonomous Systems. In *ACM Internet Measurement Conference*.
- [73] Jan Zorž, Sander Steffann, Primož Dražumerić, Mark Townsley, Andrew Alston, Gert Doering, Jordi Palet, Jen Linkova, Luis Balbinot, Kevin Meynell, and Lee Howard. 2017. Best Current Operational Practice for Operators: IPv6 Prefix Assignment for End-Users—Persistent vs. Non-Persistent, and What Size to Choose. <https://www.ripe.net/publications/docs/ripe-690/>.

A Host Statistics and Demographics

Here, we elaborate on the statistics and demographics of the IPv6 addresses we collected via our NTP servers.

Table 3 summarizes the statistics (i.e., minimum, median, maximum, and standard deviation) of daily collected IPs in IPv6 and IPv4 over two-weeks of our measurement period. In general, we collected more IPv4 addresses than IPv6. The median number of daily collected IPv6 addresses is around 31.3M (IPv4: 38.3M). The standard deviation of the number of BGP prefixes and ASes is around 1% of the median in IPv6, meaning the number of BGP prefixes and ASes is similar across days; we observed similar behavior in IPv4.

Figure 8 shows the distribution of collected IPs per subnet in IPv6 and IPv4. This distribution helps us choose a prefix size that captures multiple IPs per subnet without being too broad for an end-user network.

B Top Open Ports in Generic Scans

The top open combinations in IPv6 appear in both generic and extensive scans are TCP/5555, TCP/49152 and TCP/62078, TCP/9080, and

IPv6				
Stat	# of IPs	# of /64s	# of BGP prefixes	# of ASes
min	25,069,143.0	22,452,940.0	42,084.0	8,642.0
median	31,359,011.0	24,822,131.5	42,841.0	8,868.5
max	33,114,585.0	26,446,968.0	43,279.0	8,930.0
std	3,487,207.4	1,654,404.5	366.1	88.4

IPv4				
Stat	# of IPs	# of /24s	# of BGP prefixes	# of ASes
min	35,875,057.0	2,898,741.0	347,753.0	44,206.0
median	38,307,267.5	2,979,420.0	359,611.0	46,347.5
max	48,797,264.0	30,083,99.0	365,259.0	46,860.0
std	5,094,424.6	40,857.3	6,001.8	1,049.5

Table 3: Statistics of collected data in IPv6 and IPv4 per day, over our two weeks measurement period.

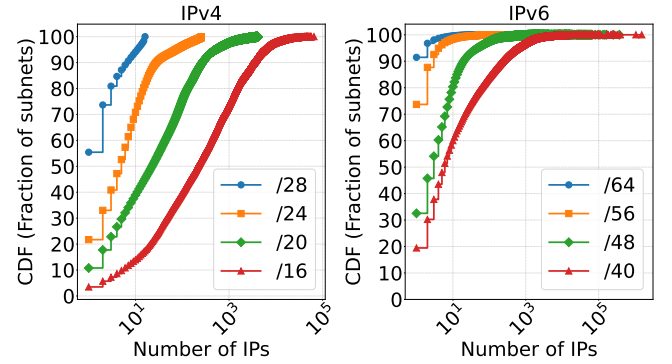


Figure 8: CDF of subnets by number of IP addresses)

TCP/7000, TCP/8008, TCP/8009, and TCP/8443 (used by Chromecast) appear in the top 10 ports in the generic scans but not the extensive ones (ranked 13th). These ports are used by client-specific services that suggest network openness. Similarly, in IPv4, the top ports in generic are TCP/443, TCP/80 and TCP/443, TCP/8443 that are widely used by routers.