2023 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-9336-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SP46215.2023.10179376

IPvSeeYou: Exploiting Leaked Identifiers in IPv6 for Street-Level Geolocation

Erik C. Rye
University of Maryland

Robert Beverly
CMAND

***Abstract*—We present *IPvSeeYou*, a privacy attack that permits a remote and unprivileged adversary to physically geolocate many residential IPv6 hosts and networks with street-level precision. The crux of our method involves: 1) remotely discovering wide area (WAN) hardware MAC addresses from home routers; 2) correlating these MAC addresses with their WiFi BSSID counterparts of known location; and 3) extending coverage by associating devices connected to a common penultimate provider router.**

We first obtain a large corpus of MACs embedded in IPv6 addresses via high-speed network probing. These MAC addresses are effectively leaked up the protocol stack and largely represent WAN interfaces of residential routers, many of which are all-in-one devices that also provide WiFi. We develop a technique to statistically infer the mapping between a router's WAN and WiFi MAC addresses across manufacturers and devices, and mount a large-scale data fusion attack that correlates WAN MACs with WiFi BSSIDs available in wardriving (geolocation) databases. Using these correlations, we geolocate the IPv6 prefixes of >12M routers in the wild across 146 countries and territories. Selected validation confirms a median geolocation error of 39 meters. We then exploit technology and deployment constraints to extend the attack to a larger set of IPv6 residential routers by clustering and associating devices with a common penultimate provider router. While we responsibly disclosed our results to several manufacturers and providers, the ossified ecosystem of deployed residential cable and DSL routers suggests that our attack will remain a privacy threat into the foreseeable future.

1. Introduction

Media Access Control (MAC) addresses are designed to be globally unique layer-2 network interface hardware identifiers. Most modern network interfaces, including Ethernet, WiFi, and Bluetooth, utilize 48-bit IEEE MAC addresses [1]. For several well-known reasons – notably manufacturer fingerprinting [2] and the ability to track devices by an identifier that remains static across network changes [3], [4], [5] – MAC addresses are considered sensitive.

MAC addresses are typically confined to layer-2, and thus cannot readily be discovered by a remote attacker who is not attached to the same subnet. A historical exception is the use of MAC addresses to automatically select the host bits of an IPv6 client, a process known as SLAAC EUI-64 addressing [6]. Due to the aforementioned vulnerabilities, modern operating systems instead typically generate IPv6 addresses with random host bits [7], [8], [9].

Prior work in high-speed active IPv6 network topology techniques [10] has helped overcome the challenge of finding active hosts and networks amid the vast IPv6 address space [11]. Recent research in IPv6 periphery discovery [12] produced a large corpus of Customer Premises Equipment (CPE) devices, i.e. residential home cable and DSL modems providing IPv6 service. Surprisingly, more than 60M of these CPE deployed in the Internet use legacy EUI-64 addresses, likely because they run older operating systems and legacy configurations inherent in embedded devices.

Beyond this relatively minor privacy weakness, our key insight is that many of these CPE devices are System-on-a-Chip (SoC) designs, e.g. [13], with multiple network interfaces where *each interface is assigned a MAC address predictably from a small range.* For example, an all-in-one device with a Wide Area Network (WAN), Local Area Network (LAN), and WiFi interface where the WAN address is `AA:BB:CC:DD:EE:01`, the LAN address is `AA:BB:CC:DD:EE:02`, and the WiFi BSSID (Basic Service Set Identifier; the Access Point (AP) wireless MAC address) is `AA:BB:CC:DD:EE:03`. While the number of MAC addresses allocated and offsets can differ widely across devices, manufacturers, and implementations, we develop an inference technique that permits us to predict the most likely BSSID given the WAN MAC address.

We can then search for the BSSID in available public wardriving [14], [15] databases, e.g. WiGLE [16], Apple Location Services [17], and others [18], [19]. The ability to bind a CPE IPv6 address to its corresponding WiFi BSSID leads to our core contribution: *street-level geolocation* of the IPv6 network prefixes assigned to these CPE (cf. Figure 2).

Our geolocation inferences are not limited to devices and implementations using legacy EUI-64 addressing. Where EUI-64 and non-EUI-64 devices are both deployed in a provider, we cluster those devices connected to the same upstream provider router to establish a feasible location for non-EUI-64 devices. Thus, a single EUI-64 device connected to a provider router may potentially compromise the geolocation privacy of *other* customers that are also connected to that provider router. Because CPE software is rarely upgraded, and the devices themselves are infrequently

DOI 10.1109/SP46215.2023.00024

replaced, we expect our findings to remain a threat into the foreseeable future. Our privacy attack, *IPvSeeYou*, makes the following primary contributions:

- An algorithm to infer, per-CPE manufacturer and device, the offset between its WAN and WiFi interface MAC addresses (§3).
- Validation of IPvSeeYou on a subset of geolocation inferences with a median error of 39 meters, suggesting our technique is accurate and precise (§6).
- Street-level geolocation of 12M IPv6 CPE– and the IPv6 customer prefixes they connect – across 146 countries[1] using our offset algorithm and performing data fusion with wardriving databases (§7).
- Extension of the technique to additional IPv6 CPE by clustering geolocated devices and associating them with non-EUI-64 devices (§8).
- Disclosure to several equipment manufacturers and service providers, and steps toward remediation (§9).

While our attack affects a large subset of deployed IPv6 routers, primarily residential devices, several conditions must be satisfied in order to successfully geolocate a router with IPvSeeYou: a router must 1) be responsive to active probes; 2) use EUI-64 IPv6 addresses (§3.2); 3) have a predictable offset between the WAN MAC address and its BSSID (§3.4); and 4) have a BSSID in a geolocation database we query (§3.3). We discuss these limitations further in §4 and ethical considerations of our work in §5

2. Background and Related Work

2.1. IPv6 Addressing

Devices commonly auto-generate their interface IPv6 addresses through Stateless Address Autoconfiguration (SLAAC) [21], [6], [22], [8], [9] rather than via assigning addresses statically or DHCPv6. Early IPv6 standards encouraged the use of EUI-64 IPv6 addresses [23], [24], wherein the lower 64-bits of the 128-bit address – the Interface Identifier (IID) – embed the interface's MAC address. The embedding (a modified EUI-64) first sets the Universal/Local bit, then inserts the bytes `0xFFFE` between the third and fourth bytes of the MAC. Figure 1 displays an example EUI-64 IPv6 address.

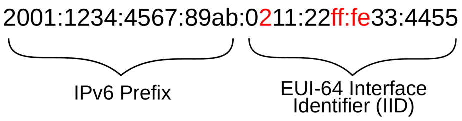

2001:1234:4567:89ab:0211:22ff:fe33:4455
IPv6 Prefix
EUI-64 Interface Identifier (IID)

Figure 1: An EUI-64 IPv6 address constructed by embedding the MAC `00:11:22:33:44:55` in the IID.

Modern devices, particularly end systems, no longer employ EUI-64 SLAAC addressing for several reasons. First, a static, unique IID allows an adversary to track devices over time and address space changes. Second, MAC addresses are globally unique, with contiguous blocks of 2^{24} bits (known as Organizationally Unique Identifiers (OUIs)) assigned to manufacturers. Not only does embedding the MAC address in the IID expose the device manufacturer, work has shown that it is possible to infer the specific device model [2]. Instead, modern operating systems typically form IPv6 addresses using randomly generated IIDs [7], [8], [9].

Despite known security and privacy issues inherent in EUI-64 addressing, and the introduction of SLAAC Privacy Extensions (PE) [25] over 20 years ago, previous studies [10], [11], [12] discovered millions of CPE devices using EUI-64 SLAAC. Our work focuses on these devices, which primarily include residential home cable and DSL routers.

2.2. IP Geolocation

IP addresses are logical network identifiers; while addresses and hostnames may identify a network or operator and hint at the location, the associated device may physically be anywhere. Further, the device may not wish to reveal its location, or may be unable to geolocate itself. As a result, a rich body of work has developed IP geolocation techniques that allow a third-party to map arbitrary IP addresses to physical locations. Multiple IP geolocation services, e.g. [26], [27], [28], exist to support applications such as advertising, content and language customization, content geo-fencing, law and policy enforcement, anti-fraud, authentication, and forensics [29], [30], [31].

Well-known methods for third-party IP geolocation include: 1) registry databases, e.g. whois and the DNS [32]; 2) constraint-based techniques that leverage speed-of-light propagation delay to triangulate an address [33], [34]; 3) network topology [29], [35]; and 4) privileged feeds [36].

While these geolocation services impinge on the privacy of the devices and users, they generally provide course-grained location, e.g. city. Several studies have found significant inaccuracies in techniques and databases as compared to ground truth. For instance, Poese et al. found 50-90% of ground truth locations present in commercial databases had greater than 50 kilometers of error [37]; more recently Komosnỳ et al. studied eight commercial geolocation databases and found mean errors ranging from 50-657 kilometers [38].

In contrast to these prior techniques, our approach seeks to geolocate IPv6 addresses at a street-level granularity. While Wang et al. similarly sought street-level geolocation, albeit for IPv4, their technique requires geolocation targets to reside in a high-density location and nearby permissive passive landmarks [31]. Finding acceptable landmarks, even in dense locations, is a significant hurdle in the modern age of shared hosting services. In this paper, we show that not only can IPvSeeYou provide accurate street-level geolocation, it is effective on a large number of IPv6 prefixes.

2.3. Related Work

Wright and Cache observed that the WiFi and Bluetooth MAC addresses of mobile devices are often sequential, allowing passive adversaries to correlate these identifiers [39].

[1] We count ISO-3166-1 two-letter country codes throughout and use the term "countries," although some are dependent territories [20].

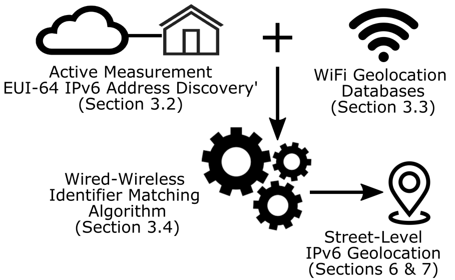


Figure 2: IPvSeeYou: to geolocate IPv6 CPE, we fuse MAC addresses from EUI-64 IPv6 addresses (§3.2) with WiFi BSSIDs from geolocation databases (§3.3). Our matching algorithm (§3.4.3) produces an inferred per-OUI offset.

Our work also leverages the idea of predictable MAC assignment across different link-layer technologies, but does not require physical proximity to the target and focuses on CPE rather than mobile devices.

The extraordinarily large address space of IPv6 removes the need for Network Address Translation (NAT). Whereas NAT is ubiquitous in residential IPv4 networks, IPv6 restores an end-to-end connectivity model whereby the CPE device is a routed hop. This requires a novel approach to CPE discovery in IPv6. Our "edgy" algorithm is specifically aimed at discovering the IPv6 network periphery, i.e. the CPE that connect customer edge networks to the IPv6 Internet [12]. With edgy, we previously discovered 5M unique MAC addresses in 16M EUI-64 IPv6 addresses, but did not attempt to correlate these MAC addresses with wireless identifiers or geolocate them, as we do in this work.

Recent work has sought to understand IPv6 addressing. Fiebig et al. [40] and Borgolte et al. [41] used DNS response semantics to discover active addresses within reverse zones. Murdock et al. [42] generated target addresses and test for liveness using active measurements. Li and Freeman [43] examine user-level IPv6 behavior and address dynamics from the vantage of a large online social network, and how to best implement effective IPv6 filtering.

Prior work has exploited IPv6 to mount tracking campaigns. Berger et al. reversed the keyed hash function used to generate IPv6 flow labels, thereby permitting device tracking [44] even when the device uses random addresses; this vulnerability has since been mitigated by common operating systems. More recently, Rye et al. leveraged CPE IPv6 EUI-64 addresses to track connected devices across prefix and IID changes [45]. Our work similarly exploits IPv6 EUI-64 addresses to attack user privacy, but via precision geolocation for a subset of users and devices. While limited prior work has examined IPv6 geolocation [46], to the best of our knowledge IPvSeeYou is one of the first techniques to exploit specific properties of IPv6 for geolocation.

3. Methodology

Our starting point exploits CPE implementations with two key facets: Internet-facing WAN interfaces that use EUI-64 addresses, and predictable MAC address assignment across the device's wired and wireless interfaces. EUI-64 addressing allows an adversary to remotely obtain MAC addresses from vulnerable devices by eliciting responses from network probes (e.g. traceroute, Yarrp [47], zmap6 [48], [49]). Predictable MAC address assignment allows an adversary to map wired MAC addresses obtained from active network probing to wireless BSSIDs from WiFi geolocation databases. While this section focuses on exploiting CPE that use EUI-64 addresses, we extend the technique's potential coverage to CPE that do not use EUI-64 SLAAC in §8.

Figure 2 outlines our methodology; in §3.2 we discuss our EUI-64 IPv6 corpus, §3.3 describes our BSSID geolocation data, and §3.4 gives our algorithm for linking EUI-64 IPv6-derived MAC addresses with BSSIDs. First however, we provide an example of IPvSeeYou to build intuition.

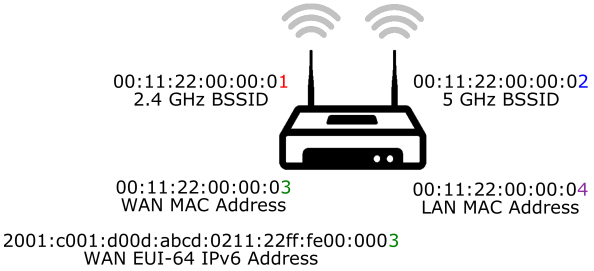


Figure 3: A vulnerable CPE router. The four interfaces (2.4 and 5 GHz 802.11 BSSIDs, LAN and WAN) are assigned sequential MAC addresses. The EUI-64 SLAAC WAN IPv6 address is discovered via active network scans, while the BSSIDs are found in WiFi geolocation databases.

3.1. Example

Figure 3 depicts a router that is vulnerable to the IPvSeeYou geolocation technique. In this simple example, each router interface is addressed sequentially from the same `00:11:22` OUI; furthermore, it generates its WAN IPv6 address using the modified EUI-64 derived from the WAN interface MAC address. This allows its WAN MAC address to be discovered by active network measurements that elicit a response from the CPE. Its WiFi BSSIDs are contained in crowdsourced databases, such as WiGLE [16], that give a precise geolocation for the BSSID. The offset distance between the WAN MAC address and BSSIDs (in Figure 3 the offsets are -1 and -2) typically remains fixed throughout an OUI, or at a minimum, for device models within an OUI. This allows an attacker who has discovered CPE MAC addresses from active network scans to *predict* the device's BSSID(s) and look them up in WiFi geolocation data, or *fuse* previously-obtained data sources together.

3.2. IPv6 EUI-64 Corpus

Our previous work on IPv6 "periphery discovery" [12] employs an iterative, targeted scanning algorithm to find CPE routers using Yarrp [47]. We refine this original technique to also use non-TTL limited ICMP6 echo request probes and performed additional Internet-wide scanning campaigns from July 2020 through July 2021.

The resulting IPv6 periphery discovery corpus includes a large number of EUI-64 IPv6 responses. EUI-64 IPv6 addresses are readily identifiable and easily reversible – the MAC addresses are decoded from EUI-64 response addresses by removing the fourth and fifth bytes (`0xFFFE`) of the IID, then inverting the U/L bit. The corpus contains approximately 347M EUI-64 IPv6 addresses with nearly 61M unique MAC addresses. Note that MAC addresses can appear in multiple EUI-64 IPv6 addresses due to ephemeral prefix leases provided through temporary-mode DHCP [45] and devices moving to new networks. Smaller numbers of repeated MAC addresses occur due to address reuse.

Of the 61M total MAC addresses derived from the EUI-64 IPv6 dataset, approximately 0.2% (126,730) were observed in EUI-64 addresses in multiple Autonomous Systems (ASes). While this dispersion may be attributable to customers and devices changing service providers, we exclude them from analysis to eliminate potential occurrences of non-unique MACs. The Appendix provides a detailed analysis of the OUI and AS distribution of the WAN corpus.

3.3. WiFi Geolocation Data

While our methodology is agnostic to the source of geolocation data, this study uses five sources, including three open-source databases [18], [19], [50], the WiGLE API [16], and Apple's WiFi geolocation service [17].

The three databases contain 20M (Alexander Mylnikov) [18], 15M (OpenBMap) [50], and 29M (OpenWifi.su) [19] BSSIDs respectively, along with associated geolocations. Because these databases rely on crowdsourcing, they are biased toward the locations of contributors.

In addition to the three databases, we issued wildcard queries for the OUIs in our corpus via the WiGLE API. As the standard API rate-limits were prohibitive, we coordinated with the WiGLE administrators to increase our daily query limit to obtain 1,367,700 geolocated BSSIDs. As with the other databases, WiGLE's coverage is dependent on the location of the crowdsourcing contributors.

Finally, we also obtain BSSID geolocation data using Apple's WiFi geolocation service [17]. Apple provides this API for its products to geolocate themselves as part of its Location Services suite of tools; the API accepts an 802.11 BSSID as a search parameter. If Apple has geolocation data for the BSSID, it returns these data, optionally with additional location information for APs in close proximity. The purpose of returning the additional geolocation information is presumably to short-circuit API requests from the same client as it encounters these additional nearby AP BSSIDs.

We used the Apple geolocation API as an oracle to validate the existence of BSSIDs we suspect are related to our EUI-64 MAC addresses. We queried the location service for BSSIDs at offsets increasing from 0 from our wired MACs from §3.2. When we guessed a valid BSSID, the geolocation service returns not only the coordinates of the guessed BSSID, but additionally up to 400 nearby BSSIDs and their geolocations [51]. We stopped querying for BSSIDs within an OUI if no offset value between successive WAN MAC addresses produced a valid BSSID. This results in our largest WiFi geolocation dataset, with 444,860,460 unique BSSIDs.

In total, our geolocation data contains 450,018,123 distinct BSSIDs in 238 countries and territories. We use the IEEE OUI database to map OUIs to manufacturers [52]. Table 3 in the Appendix summarizes macro-level characteristics of the geolocation data. Given the potentially sensitive nature of the data we collect and aggregate, we sought and followed guidance from our IRB; see §5 for details.

3.4. Inferring WAN-to-BSSID MAC offsets

Key to our method is correlating addresses between network interfaces on a CPE device. Given a WAN IPv6 address with an embedded MAC address, we wish to determine the MAC address of a corresponding WiFi interface on the CPE. In the trivial case, the WiFi BSSID MAC is exactly one greater than the WAN MAC address. However, the assignment of MAC addresses is vendor and device dependent. For example, Figure 3 shows a CPE with four interfaces: LAN, WAN, and two different WiFi radio frequencies. In this example, the two BSSID MAC address values are one and two less than the WAN MAC address. In practice, more complex allocations exist and there is a wide variety of deployed implementations.

3.4.1. Challenges. To enable our data fusion, we require a mapping of the offsets between interface MAC addresses on a per-OUI basis. Unfortunately, vendors do not publish their MAC address assignment policy, and even a single vendor frequently uses different strategies for different devices.

Thus, given the huge variety of vendors and deployed CPE devices in the Internet, we develop an algorithm to statistically infer the MAC address offsets. We utilize the large number of WAN and WiFi MAC addresses in our corpora (summarized in the Appendix) to capture this diversity and build a database of offsets for different devices.

A naïve approach to building the mapping is simply associating a WAN MAC with the numerically closest BSSID. Such an approach can fail simply due to missing data points; for instance, if only the device's WAN address is present in our data, or only the BSSID is present. In these cases a false association can be made where the two linked addresses belong to different devices. These errors can be mitigated in part by inferring the number of addresses allocated per device and preventing associations between two addresses that differ by more than the size of the allocation.

More subtle errors can, however, occur. Figure 4 illustrates how a simplistic algorithm fails for a particular CPE we purchased and for which we have ground truth (a CPE from the vendor AVM, which is prevalent in our data). This device uses a block of seven contiguous MAC addresses for

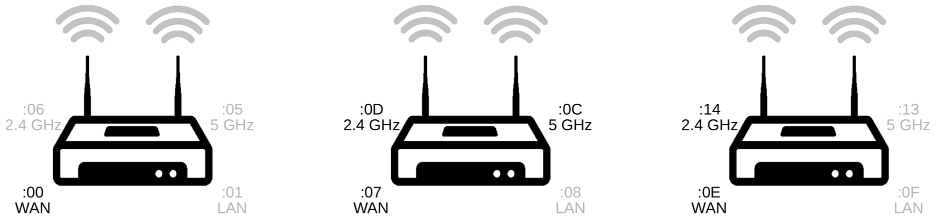


Figure 4: Example where three WAN MACs and three WiFi BSSIDs are observed (first five bytes omitted; observed MACs shown in bold). We first infer the number of MAC addresses allocated per device based on all data observed within the OUI, then compute the most likely WAN-to-BSSID *offset*. Missing data and multiple in-block BSSIDs (e.g. guest WiFi and different frequencies) complicate the inference. Each device is allocated seven MAC addresses, and the offset is +6. Note that a naïve closest matching (e.g. between `0x0D` and `0x0E`) is incorrect.

its various interfaces[2]. The lowest MAC address is given to the WAN address while the highest is given to the 2.4 GHz WiFi interface. Thus, the true offset is +6. Because of this allocation, the nearest match association can result in the WAN address of one device being associated with the BSSID of a different device; for example MACs ending in `0x0D` and `0x0E` in Figure 4. Also in the data are MACs corresponding to different WiFi radio frequencies. For instance, this model of device also has a 5 GHz WiFi interface, but at an offset of +5.

Further, our data may include a single MAC address for a device, i.e. either just the WAN or just the BSSID. Missing data is common, and can occur simply because wardrivers never encounter a device, a network blocks our probes, or there is other filtering. For example, Figure 4 shows one device where our data includes only the WAN MAC (at position `0x00`). In these cases, the nearest matching BSSID may be a multiple of the true offset, for instance +12. Missing data, multiple in-block BSSIDs, and very sparse or dense OUIs therefore complicate the inferences.

3.4.2. Algorithm. Our algorithm infers the most likely offset between the WAN MAC address and BSSID for a given OUI. First, we determine the OUI's mostly likely allocation size division, i.e. how many MAC addresses are allocated per device. We sort all of the BSSIDs in the OUI to build a distribution of intra-MAC distances. Thus, for n input BSSIDs, we compute $n - 1$ distances between these points. We find the most frequent distance and then determine how many of the samples in the distribution correspond to a multiple of this distance by computing the greatest common divisor (gcd). If the fraction of distances that are multiples of this distance are high, then we correspondingly have high confidence that the inferred allocation size is correct.

Given the inferred allocation size, the algorithm next iterates through each EUI-64 MAC address in ascending sorted order for every OUI with at least 100 WAN MAC and 100 BSSID instances. Because the matching WiFi MAC address may be at either a positive or negative offset, the algorithm finds both the closest corresponding BSSID less than, and greater than, the EUI-64-derived MAC, subject to the constraints that these must be within a window determined by the inferred allocation size in the previous phase. Finally, the algorithm infers the offset for this device to be the most common offset among all the matches.

During execution of this algorithm, both correct and false associations will be made, for instance the false association to the -1 offset BSSID versus the +6 offset BSSID in the example of Figure 4. However, the intuition is that, in aggregate, it will be more common for a single device to be present in the data with *both* its addresses than for two different devices with adjacent addresses. While exceptions can exist, especially for OUI with a large number of devices present in our data, in practice, statistically choosing the offset produces the correct inference for our ground truth devices. We again compute the fraction of devices for the OUI that conform to our inferred offset such that we can have an associated confidence measure.

3.4.3. Correlating IPv6 EUI-64 MACs and BSSIDs. Given a MAC address embedded in an EUI-64 IPv6 address, the final step in our technique is to utilize our offset database to look up the offset to the BSSID given the OUI in question. We then lookup the corresponding inferred BSSID in the wardriving databases to make our final geolocation inference. Note that if the OUI is not contained in our offset database, we cannot make a geolocation determination.

3.4.4. MAC-to-BSSID offset confidence. As Figure 4 illustrates, offset inference is complicated by manufacturer differences and available observations. Assuming a relatively dense number of observations of WAN addresses and BSSIDs, our algorithm intuitively accumulates the most "matches" at the correct offset and lesser counts at regular intervals (+/- the number of MAC addresses allocated to individual devices). In addition to 18 ground-truth devices we purchased (§6.3), we employ statistical measures to infer and validate offsets for each OUI. For instance, Figure 5 displays a PMF of the offset value for an Arris OUI. The peak offset is -2, while smaller impulses occur at intervals 16 addresses away. In this particular case, Arris allocates a span of 16 MACs to each device. Thus, the other points are a *harmonic* of the -2 offset (where the distance is to a

[2] While a block size of seven is immediately conspicuous for being odd, prime, and not on a nybble boundary, both our ground-truth and inference algorithm reveal that this is the true manufacturer allocation policy.

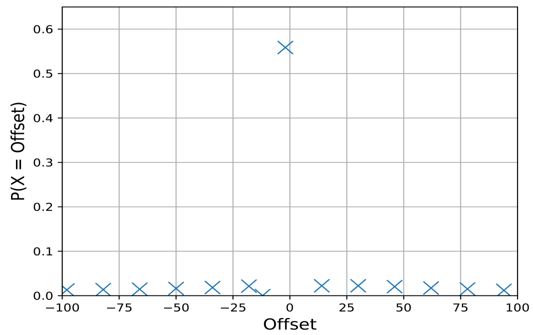


Figure 5: Probability mass of offsets between observed Arris addresses in the `00:1D:D1` OUI. The peak offset is -2 with other points at multiples of 16 away, i.e. harmonics of the -2 offset. 99.9% of the probability mass supports the inferred -2 offset, with 15,165 total offset inferences.

different device). Indeed, 99.9% of the probability mass in this plot supports the -2 inference.

4. Limitations

While our methodology allows us to make geolocation inferences for a large number of IPv6 networks – we use IPvSeeYou to geolocate more than 12M routers in the wild – it is limited to specific CPE behaviors and deployments.

First, IPvSeeYou relies on CPE that use EUI-64 WAN IPv6 addresses and are responsive to our active probing. Second, IPvSeeYou is only effective against all-in-one CPE that include a built-in WiFi base station. In contrast, some home networks contain a standalone cable modem with an Internet-facing EUI-64 IPv6 address that is then connected by Ethernet to an AP. In this case, it is impossible for us to predict the BSSID from the WAN MAC address, as they are two entirely distinct devices.

While many all-in-one CPE use a single integrated SoC, our associations are limited to devices where all interfaces are allocated MACs from the same OUI. If the device's BSSID resides in a different OUI, our methodology will not find any potential offsets, and thus will be ignored.

Further complicating the offset inference are instances where addresses from a single OUI are divided among multiple device models, a phenomenon observed by Martin [2]. If multiple devices with more than one distinct WAN MAC address to BSSID offset value exist within a single OUI, our algorithm does not capture this nuance, instead choosing the predominant offset value as learned from the data. However, our confidence measure identifies these heterogeneous OUIs as problematic. While we ignore OUI with low-confidence offsets, a more sophisticated algorithm could additionally infer these granular OUI allocations in the future.

Finally, the underlying BSSID geolocation data we use to geolocate a IPv6 address may itself be incorrect or outdated. Additionally, devices geolocated in the past may have since moved, introducing inaccuracy in our geolocations.

While these constraints limit the attack's scope, we note that: 1) all-in-one CPE with predictable offsets are common; 2) the validation we perform, while limited, confirms both the technique's viability and accuracy; and 3) we extend the technique's coverage in §8 to associate CPE which do not meet the above constraints with *other* CPE that do.

5. Ethical Considerations

Fundamental to our work are MAC addresses, which uniquely identify network interfaces, and, hence, devices. While not a user identifier per se, MAC addresses can be leveraged to track users or combined with other meta-data – as we explicitly show. As such, we submitted our research plan and protocols to our institution's IRB, who cleared the study. Our IRB noted that we have no way to associate any of our data with individuals, and that there was the potential for overall societal benefits from the research by improving privacy for millions of residential Internet users.

To minimize risk, we treat MAC addresses and any correlated geolocations as private data. We only publish, share, or release aggregated analyses on the data, and ensure that the raw data at rest remains encrypted.

While the results of our research could be misused, we aim to ultimately improve privacy protections by highlighting this vulnerability. In addition, we have responsibly disclosed the privacy weaknesses of exposing MAC addresses in IPv6 to network equipment vendors and a large residential service provider. At least one vendor is in the process of issuing a patch to update their equipment's behavior, and the residential service provider is currently deploying measures to mitigate our attack. In light of these factors, we believe, as does our IRB, that the beneficence of our work significantly outweighs any potential harm or risk it may present.

6. Validation

We employed a multi-pronged strategy for validation, including crowd-sourcing measurements, purchasing selected CPE devices, and collaborating with a large residential ISP.

6.1. Crowd-sourced Measurements

Our first validation experiment enlisted the help of volunteers. We designed a custom web page that first tests for IPv6 connectivity and, if the client has operational IPv6, logs the client's address and requests the client's location via the HTML5 geolocation API. If the user consents, we obtain tuples of IPv6 address and precise geolocation. From the client address, we obtained their CPE router's IPv6 address via active probing (edgy) and employed the IPvSeeYou inference procedure. We publicized our measurement site via the SIGCOMM Slack channel.

Of the 50 participants with residential IPv6 service from 31 ASes in Europe, North America, and Asia that participated in our user study, the majority (84%) did not have CPE using EUI-64 IPv6 addresses. Of the eight with EUI-64 IPv6 addresses, IPvSeeYou successfully geolocated five.

The true geolocation for four of these five CPE agreed with IPvSeeYou's inference within 50 meters. By contrast, MaxMind's GeoLite2 geolocation database geolocated these addresses to between 500 meters and 421 kilometers from the true location, with a mean error of 106 kilometers and median error of 1.34 kilometers. The IPvSeeYou geolocation of the fifth device was 0.68 kilometers from the HTML5 geolocation, while the MaxMind location was over 300 kilometers from both the HTML5 and IPvSeeYou geolocations.

For the three devices for which we did not obtain an IPvSeeYou geolocation, one was due to an inability to determine the MAC address offset. This occurred because we lacked a sufficient number of observations of WAN MACs and BSSIDs to make an offset inference. IPvSeeYou was able to determine the offset for the remaining two CPE, but the inferred BSSIDs were not found in our wardriving data. This validation experiment demonstrates that IPvSeeYou can provide highly precise geolocations for some devices, and highlights some real-world challenges, such as CPE without EUI-64 addresses and address offset inference failures. Due to the small sample size of the crowd-sourced measurements, we ultimately collaborated with a large North American ISP for more representative validation, as detailed next.

6.2. Provider Validation

We coordinated with a large United States-based residential and business ISP that offers IPv6 to obtain validation. From our complete dataset, we randomly sampled 1,350 responsive CPE addresses from the provider's address space. We used a bulk reverse geocoding service [53] to map our inferred geo-coordinates to a US ZIP code and supplied the <IPv6 CPE address, customer subnet, ZIP code> tuples to the provider. Of the full set, the ISP was able to provide validation for 486 CPE (36%) due to missing records and address churn. Of this subset, 80% of our ZIP code geolocation inferences agreed with the provider's ground-truth ZIP codes.

Due to stringent customer privacy regulations, the provider could unfortunately not provide further detail about individual errors or error bounds. We find this 80% agreement encouraging given: 1) ZIP code geo-granularity is widely accepted as useful for marketing and demographic correlations (there are approximately 44,000 ZIP codes in the United States); 2) natural address churn and reuse during the delay between providing our inferences and receiving validation; and 3) the potential that we inferred a ZIP code adjacent to the correct ZIP.

6.3. Ground-truth Hardware

Finally, as a third source of complementary validation, we purchased 18 used hardware devices from OUIs prevalent in our corpus. Not only did these ground-truth CPE inform our offset algorithm, they exposed real-world pathologies and limitations of IPvSeeYou.

Some of the OUIs of MAC addresses derived from our active scan dataset have no corresponding BSSIDs in our wireless data; this means that entire OUIs are "matchless." This behavior is primarily due to manufacturers allocating MAC addresses to the wired and wireless interfaces of a CPE from different OUIs. For example, we procured several Technicolor [54] CPE routers distributed by Comcast [55], for its Xfinity Internet service. Five devices addressed their wired and wireless interfaces from different OUIs (e.g. `FC:91:14` and `78:F2:9E`, respectively.) Despite a heterogeneous mix of OUIs on a single CPE, some patterns still exist. For instance, as the lower 24 bits of the wired interface MACs increase, so too do the lower 24 bits of the BSSIDs in the `78:F2:9E` OUI, albeit non-uniformly.

Among the 18 ground-truth CPE we procured, our offset algorithm correctly returned no inference for all ten devices with different OUIs for WAN MAC and BSSID allocations. Among the eight for which our algorithm did return an offset, five were correctly predicted and three were incorrectly predicted. Further investigation revealed that the three incorrectly-predicted offset exemplars have no close MAC observations in our active scan data, potentially indicating the purchased devices do not use EUI-64 IPv6 addresses and thus do not appear in our corpus.

7. Results

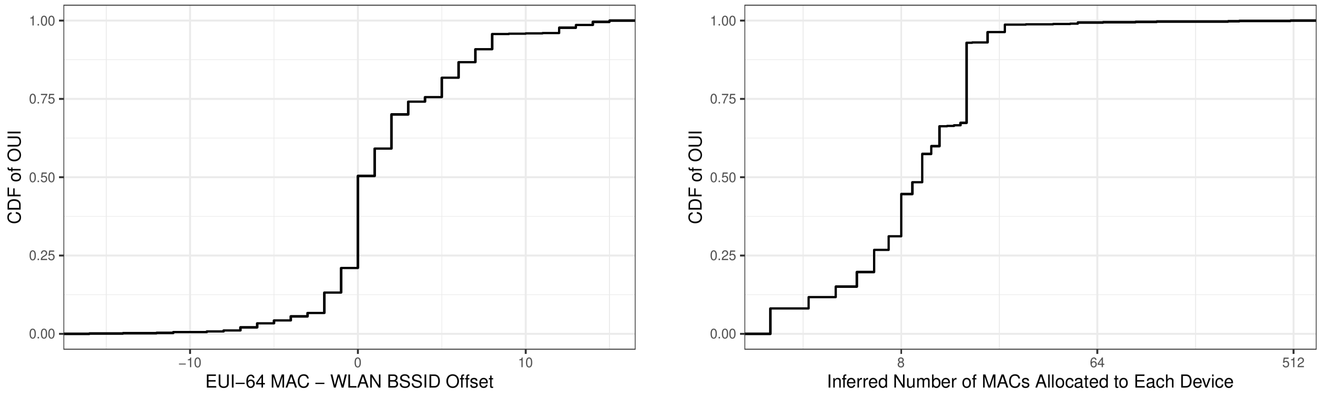
We next present results of IPvSeeYou on our data, first by characterizing the inferred CPE WAN-to-BSSID offsets and then in terms of the geolocations it produces.

7.1. Offsets

Using our methodology (§3.4.2) we compute the inferred WAN-to-BSSID offset value for each OUI with more than 100 data points in the EUI-64 WAN data, i.e. only those OUI where we have enough data to make a meaningful inference. We then filter OUI by the fraction of data points that are consistent with the inferred offset or a harmonic of the inferred offset. We apply a liberal filter that excludes OUI where fewer than 5% of the data points are offset consistent.

We discuss some scenarios that may cause an OUI with many EUI-64-derived MAC addresses to have few or no matches in §4. Our rationale for only filtering the lowest-confidence offset OUI is to accommodate instances where the OUI contains multiple device models that implement different offsets, as well as to handle instances where there are a large number of observations in one of the two media (wired MACs or wireless BSSIDs) but few in the other.

After filtering, 1,008 unique OUIs remain, or 3% of the 32,345 registered OUIs (see Appendix for additional details.) These 1,008 OUIs cover 31,720,611 distinct EUI-64-derived (WAN) MAC addresses in the corpus, approximately 52% of the total discovered from probing (§3.2). Of these ∼31M EUI-64 MAC addresses, 12,125,839 have a predicted BSSID found in our geolocation database (a "match") at



(a) CDF of 1,008 analyzed OUIs' inferred MAC-to-BSSID offsets. All inferred offsets fall within the range of -16 to 15; 0 (the WAN MAC and BSSID are the same) is the most common offset.

(b) The inferred number of MAC addresses allocated to each device as a CDF of the analyzed OUIs. Note that the vast majority of OUIs (∼90%) have inferred allocation sizes of 16 or fewer.

Figure 6: Inferred wired MAC to wireless BSSID offset and MAC address allocation size CDFs.

the inferred offset value derived from our matching algorithm. This represents ∼38% of the EUI-64-derived MAC addresses from the 1,008 filtered OUI and ∼20% of the entire EUI-64-derived MAC address corpus.

Figure 6a depicts the cumulative fraction of inferred offset value across OUIs we analyze. The inferred offsets range between -16 and 15, with a statistical mode of zero. The range of inferred offset values is unsurprising; device manufacturers assigning MAC addresses sequentially to interfaces will naturally produce small offset distances between any two, and the range suggests that typically single-device MAC addresses do not stray more than a nybble away from each other. More surprising is the slightly more than one-quarter of the OUIs that produced an inferred offset of zero between the wired MAC and wireless BSSID.

At least two potential scenarios likely explain the root cause of the zero mode in the distribution. First, a device may lack a link-layer identifier suitable to create an EUI-64 IPv6 address for a particular interface, e.g. the cellular interface of a hotspot device. In this case, the MAC address of a different interface is used (the BSSID) to create the EUI-64 IPv6 address; since the network prefixes should differ on each interface, no address collision will occur. A second cause for an inferred WAN-to-BSSID offset value of zero is MAC address reuse between the wired and wireless interfaces. Because MAC addresses allocated from a vendor's OUI are assumed unique [1], this scenario suggests misuse of IEEE-assigned MAC address space.

7.2. Geolocation Inferences

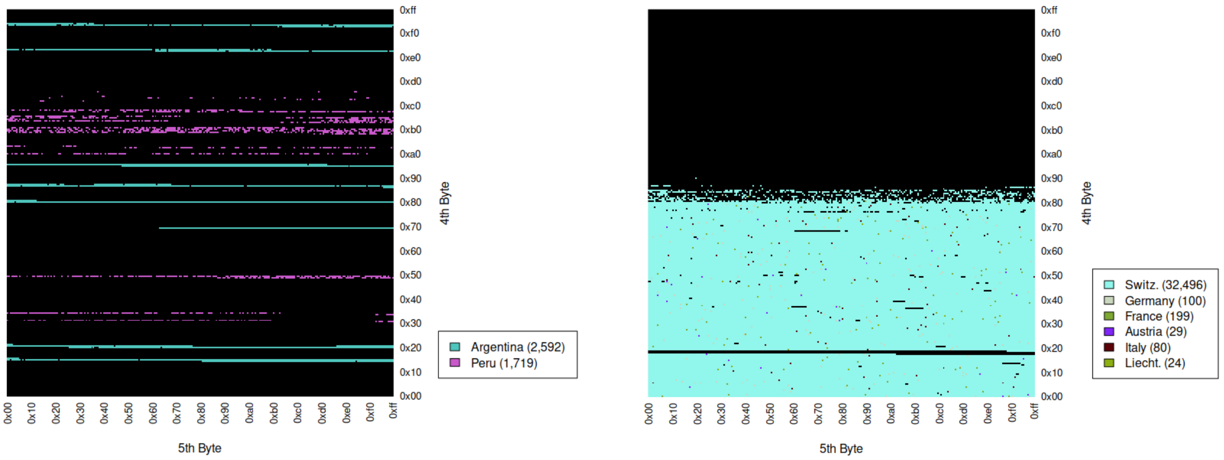
Table 1 summarizes our results from matching EUI-64 IPv6 address-derived MAC addresses with WiFi BSSIDs. Our algorithm pairs at least one IPv6 address from 1,114 unique ASNs with a geolocated BSSID, representing approximately 5% coverage of the ∼27k IPv6 ASNs announced in the global BGP routing table. Of the 347M unique EUI-64 IPv6 addresses in our corpus, 118,429,034 (∼34%) contain an embedded WAN MAC address that pairs with a geolocated BSSID. Further, due to provider prefix cycling [45] and address churn, multiple EUI-64 IPv6 addresses with the same encoded WAN MAC map to the same BSSID. Only 12M unique MAC addresses are encoded in the 118M EUI-64 IPv6 addresses that have a WAN MAC-BSSID correlation.

Germany is the most frequently-geolocated country, with more than a quarter of the total address matches. This is primarily due to a large number of WiFi routers made by AVM GmbH under the brand name "Fritz!Box". We note that, for this reason, we purchased AVM CPE and explicitly validated IPvSeeYou on these devices – and thus have high-confidence in the geolocation inferences for this large subset.

IPvSeeYou enables insight into the geographic distribution of devices within an OUI. Figure 7 displays the breakdown of country-level geolocations we obtained for EUI-64 IPv6 addresses within two different OUIs. Figure 7a, representing a Mitrastar OUI, shows that there are distinct bands of MAC addresses allocated to devices operated in different, non-adjacent South American countries. Figure 7b, on the other hand, shows an Askey Corporation OUI in which the vast majority of MAC addresses allocated to their devices are geolocated to a single European country.

TABLE 1: IPvSeeYou geolocation results; summary of matches between EUI-64-derived WAN MAC addresses and BSSIDs from WiFi geolocation databases.

Geolocations	**Country**	**Geolocations**	**OUI**
3.5M (29.2%)	DE	603k (5.0%)	A0:65:18 (VNPT Tech.)
1.5M (12.2%)	US	374k (3.1%)	10:86:8C (Arris)
1.3M (10.6%)	VN	254k (2.1%)	3C:7A:8A (Arris)
1.2M (9.6%)	FR	249k (2.1%)	A4:F4:C2 (VNPT Tech.)
1.0M (8.2%)	BR	247k (2.0%)	E0:28:6D (AVM GmbH)
3.7M (30.3%)	142 Other	10.4M (86%)	1,003 Other
12.1M (100%)		12.1M (100%)	



(a) A Mitrastar OUI (`CC:D4:A1`) displays bands of MAC address space that geolocate to different, nonadjacent countries.

(b) An Askey Corp. OUI (`1C:24:CD`) whose MAC addresses are geolocated primarily to a single nation.

Figure 7: Inferred country-level geolocation distribution for MAC addresses in two OUIs. Points represent 4^{th} (y-axis) and 5^{th} (x-axis) bytes of MAC addresses colored according to geolocated country.

7.3. EUI-64 IPv6 Geolocation Comparison

To further explore IPvSeeYou, we compared our geolocation capability with current widely-used IP geolocation databases. IP geolocation databases are known to contain inaccuracies e.g. [37], [38], and are generally used for applications that only require country or city-level accuracy. However, these databases provide comparative insight into IPvSeeYou's goal of providing street-level geolocation.

We first used the popular MaxMind GeoLite2 geolocation database [27] contemporaneous with the active scan data (April 2021) to obtain coordinates for the EUI-64 IPv6 addresses in our corpus (§3.2). For each MAC address in our WAN MAC address corpus, we retrieved the EUI-64 IPv6 address it was embedded in. In cases where a MAC address appeared in more than one EUI-64 IPv6 address due to periodic prefix cycling by ISPs, we randomly selected one of the IPv6 addresses for comparison. Then, for each of the BSSIDs matched to the WAN MAC addresses present in our EUI-64 corpus, we compared the BSSID geolocation to the MaxMind IP geolocation. Like the EUI-64 IPv6 addresses, BSSIDs appeared in our geolocation data multiple times with different coordinates as well. Again, we chose one of the geolocations randomly to use as the canonical location.

Figure 8 is a CDF of the geodesic distance difference between IPvSeeYou inferences and MaxMind geolocations. Because we do not know whether MaxMind or our inference is correct, this metric is simply the distance between the two points. However, we note that when the wardriving database is accurate and up-to-date, we expect our inference to better represent the true location. While approximately 2,800 (0.02%) WAN MAC-BSSID pairs have MaxMind and wardriving geolocations within 100m of each other, about 75% of all MAC-BSSID pairs have IP and BSSID geolocations more than 8 kilometers apart. The median difference between MaxMind and our wardriving database locations is 26 kilometers, indicating that the locations routinely differ on a city- or regional-level. In the extreme case, geolocations provided by MaxMind and our data differ by thousands of kilometers. There are several potential reasons for drastic geolocation differences. First, a router may move between EUI-64 address discovery and BSSID geolocation. Secondly, an incorrect BSSID inference from a WAN MAC address may erroneously match a device in a different geographic region than the correct inference would have. Finally, the MaxMind geolocation data may also be incorrect or stale.

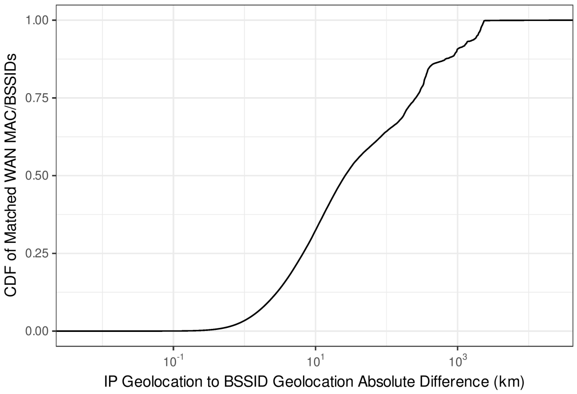


Figure 8: CDF of geolocated IPv6 addresses displaying the distance difference between MaxMind and IPvSeeYou.

We further analyzed the 20% of geolocations with the largest difference between MaxMind and IPvSeeYou locations (2,423,991 pairs with geolocation differences ≥ 335km). Of these 2.4M WAN MACs encoded in IPv6 addresses, five of the ten most common OUIs belong to the Arris Corporation, accounting for 45% (1,093,184 of 2,423,991). This company produces one of the consumer-grade routers issued by a large US ISP, leading to a high degree of OUI and ISP homogeneity among the most ex-

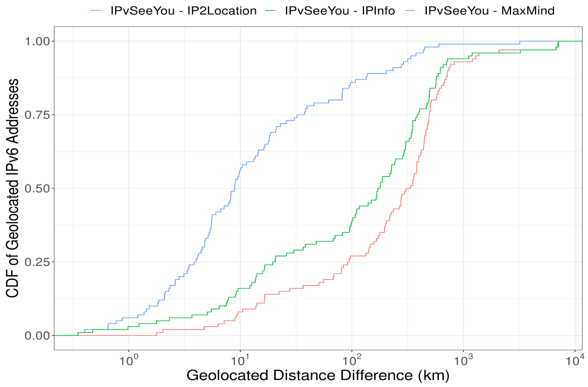


Figure 9: CDF of geolocation distance differences between IPvSeeYou and popular IP geolocation databases.

treme geolocation differences. All but ten of these devices are geolocated to the same point in a lake in Kansas, USA by MaxMind's GeoLite2 (presumably representing a default location), while IPvSeeYou produces 1,093,076 unique geolocations throughout the provider's coverage area. Another three of the ten most common OUIs belong to Mitrastar, accounting for ∼9% of the top 20% of geolocation discrepancies. Of these 213,465 geolocated devices, 96% (205,594) are geolocated to a point in Guanabara Bay, Brazil by MaxMind, while IPvSeeYou reports 213,439 distinct locations.

The number of unique geolocation data points in the wardriving and MaxMind datasets suggests that the wardriving data is closer to ground truth. Of the 347M unique IPv6 EUI-64 addresses in the IPv6 corpus, MaxMind returns only 22,676 distinct geolocations, indicating that MaxMind places millions of IPv6 addresses at the same positions. In contrast, the wardriving data we obtained comprises 433M distinct BSSID geolocations of 450M total BSSIDs. This means that far fewer BSSIDs are geolocated to the same point in our wardriving data; those that do geolocate to the same point are frequently sequential, indicating that they are likely the BSSIDs for two different WiFi frequency bands. Focusing specifically on our 12.1M MAC to BSSID matches, MaxMind returns 10,133 distinct IPv6 geolocations, while IPvSeeYou returns 12.1M.

Finally, we compared IPvSeeYou's geolocation across other popular IP geolocation services, including MaxMind, IP2Location [28], and IPinfo.io[56] by sampling a random 100 geolocated addresses from our corpus and comparing the resulting geolocation from each service. Although none of these sources of data are ground truth, Figure 9 shows that IPvSeeYou is most consistent with MaxMind, with a median geolocation distance difference of about 10 kilometers. IPInfo and IP2Location have much larger geolocation differences, with median differences of approximately 250 and 500 kilometers from IPvSeeYou, respectively.

8. Infrastructure and Non-EUI-64 Geolocation

Thus far, our IPv6 geolocation capability applies only to CPE devices that use EUI-64 addressing and the prefixes associated with those devices. In this section, we extend our technique to permit more general geolocation of CPE devices via association with geolocatable CPE. While extending our coverage comes at the cost of reduced accuracy, it allows for geolocation of devices with unknown offsets, missing BSSID in the available wardriving databases, and CPE that do not use EUI-64 addressing.

In typical deployments, multiple CPE devices connect to, and are aggregated by, an upstream provider router. Further, the network link between a CPE and its upstream router is generally relatively short due to protocol specifications and physical constraints. For example, the DOCSIS standard for cable modems defines a maximum distance between the CMTS and modem of 100 miles (160 kilometers), but with a "typical maximum separation of 10-15 miles." [57]

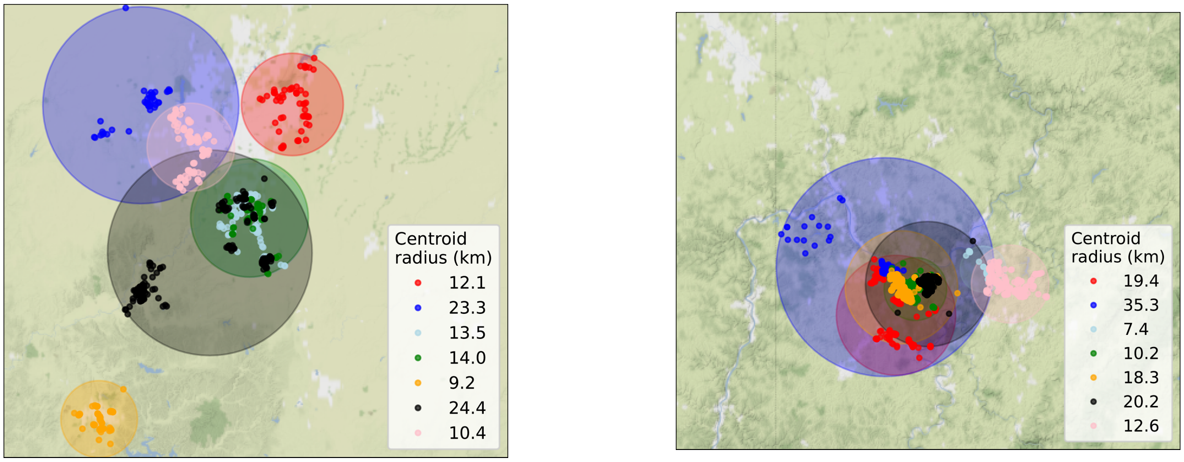
We therefore leverage IPvSeeYou-geolocated EUI-64 CPE to locate: 1) upstream provider last mile infrastructure; 2) EUI-64 CPE that we cannot geolocate using our methodology in §3.4.3; and 3) non-EUI-64 CPE. Our basic intuition is straight-forward: known locations of CPE devices can be used to infer the location of unknown CPE if they connect to the same provider router (and, hence, are likely in close physical proximity). Further, when our assumptions about the distance between a CPE and the router to which it connects are incorrect, e.g. in a virtualized network topology, this error will be reflected by a wide dispersion of geolocated devices and thus evident and detectable.

8.1. Infrastructure Case Study

To explore the feasibility of this intuition, we probed the path toward each prefix behind all EUI-64 routers in our corpus that are connected to a large United States residential ISP. For this probing, we used Yarrp from a well-provisioned vantage point; Yarrp is a high-speed randomized IP topology prober that reveals the sequence of router interfaces along the data plane path in the same fashion as traceroute [47], [10]. We then grouped successfully geolocated EUI-64 IPv6 CPE by their penultimate hop, i.e. the CPE's upstream provider router.

Figure 10 depicts the geolocation of EUI-64 CPE mapped by IPvSeeYou in two metropolitan areas of the United States. Each CPE is a dot on the map, while each color corresponds to a common penultimate hop (provider router) revealed via Yarrp. For each infrastructure router, we additionally compute the centroid of the set of geolocated CPE it serves. Then, we plot a lighter circle of the ISP router's color around the centroid with minimum radius r such that all geolocated CPE fall within r kilometers. This illuminates an inferred coverage area for each provider router.

As an illustrative example showing both the power of the technique as well as the complexity of real-world deployments, Figure 10a maps seven distinct clusters in Indianapolis, IN, USA corresponding to seven different provider routers. Fifty-seven CPE geolocate in the red cluster and 173 CPE geolocate to the black cluster. The red CPE form a fairly dense grouping, with all CPE within 12.1 kilometers



(a) Indianapolis, IN, USA CPE

(b) Pittsburgh, PA, USA CPE

Figure 10: CPE geolocations for two large US metropolitan areas; colors represent a common penultimate (provider) router. Non-EUI-64 CPE that connect to the same router are inferred to be within the same cluster as their EUI-64 counterparts.

of the geolocations' centroid. The black CPE, however, are distributed between two distinct geographic clusters approximately 10 kilometers apart. Further, the northeastern grouping in the black cluster substantially overlaps with both the green and light blue clusters. For candidate CPE that either do not use EUI-64 addressing or that we fail to geolocate directly using IPvSeeYou, infrastructure router clustering provides an indirect, coarse geolocation mechanism.

Figure 10b shows a more complicated, yet equally compelling, example from the Pittsburgh, PA, USA area. Here, there are again seven different colors representing seven unique provider routers for the geolocated CPE. In this example, significant overlap exists in the provider router service areas. All seven penultimate hops have at least one CPE device located inside of the dark blue coverage area, and the red, orange, green, and black coverage areas substantially intersect. This result is expected, as multiple ISP infrastructure routers may be necessary to support deployments in dense metropolitan regions.

Due to our ability to deduce the service coverage range of a provider's last-mile infrastructure, even a single device using EUI-64 addressing can potentially compromise the geolocation privacy of *all* of the devices that connect to the same infrastructure. A CPE using random IPv6 addresses is therefore not sufficiently protected – simply living near an IPv6 EUI-64 CPE device can be a privacy vulnerability. This further implies that, even if EUI-64 becomes less widely deployed, legacy equipment that is infrequency updated or refreshed implementing EUI-64 will continue to enable geolocation.

8.2. Accuracy

8.2.1. Volunteers. As in 6.1, we solicited volunteers to assist with validating our methodology for geolocating non-EUI-64 CPE. The participants disclosed their LAN IPv6 subnet, which we then used to obtain their CPE WAN IPv6 address via Yarrp and to discover other CPE devices assigned adjacent subnet allocations. The nearby CPE devices with EUI-64 addresses were then used to infer the BSSID assigned to the same device using our offset inferences computed in §3.4, which in turn were used for precise geolocation using IPvSeeYou.

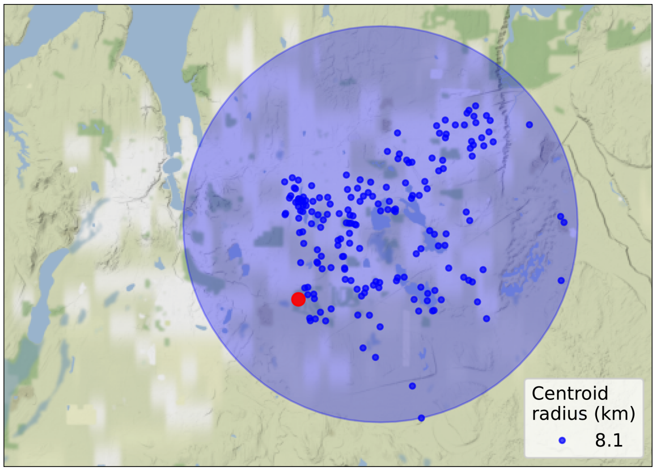


Figure 11: Olympia, WA, USA geolocated CPE (blue) used to infer the location of a known ground-truth device (red) with a non-EUI-64 IPv6 address.

Figure 11 presents geolocated EUI-64 CPE in blue located nearby one volunteer's non-EUI-64 CPE device (represented as the red point) for which we have known ground-truth location (in Olympia, WA, USA). The ground-truth device is located 4.75km from the IPvSeeYou inferred centroid of the geolocated EUI-64 CPE, demonstrating in this example that our non-EUI-64 geolocation methodology produces correct and highly accurate results.

Five additional volunteers from §6.1 whose CPE could not be geolocated directly using IPvSeeYou were geolocated using IPvSeeYou on EUI-64 CPE in adjacent subnet allo-

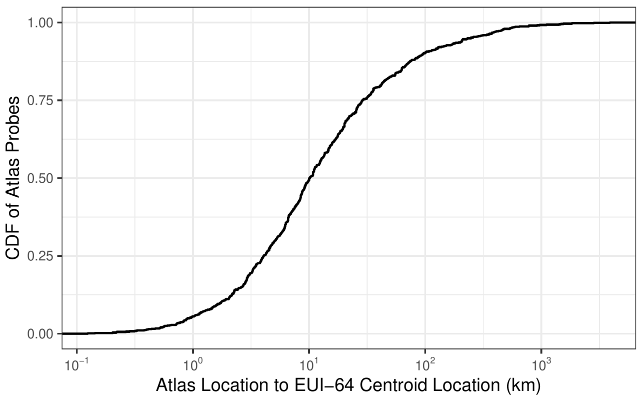


Figure 12: CDF of RIPE Atlas probes displaying the distance between the reported probe location and IPvSeeYou-derived location.

cations. These non-EUI-64 devices' true locations all fall within between 550 meters to 9 kilometers of the centroids of the EUI-64 CPE allocated prefixes adjacent to our non-EUI-64 ground truth.

8.2.2. RIPE Atlas. For additional evaluation of IPvSeeYou's ability to geolocate non-EUI-64 CPE, we utilize RIPE Atlas [58] "probes." Probes are lightweight measurement nodes installed in homes and networks. Currently, Atlas has approximately 25,000 probes distributed throughout the world [58]. Probe owners self-report their device's physical coordinates into the RIPE Atlas database when registering. When Atlas data is queried by non-owners, RIPE inserts an error of up to one kilometer to preserve the owner's privacy [59]. While some probe owners may intentionally input incorrect geolocation coordinates, we assume that most users disclose a reasonably accurate device location to RIPE and examine the data in aggregate.

We consider only RIPE probes with IPv6 connectivity, and those in residential networks. We first eliminate RIPE probes that indicate they are in a data center or use a known tunnel broker IPv6 prefix, which reduces the total number of probes we examine to approximately 3,500. Then, we initiate Yarrp traces to address space adjacent to the Atlas probes to elicit responses from nearby CPE routers. For 893 probes, we obtain at least one EUI-64 address in the same /48 prefix as the probe device we are attempting to geolocate. In this experiment, all CPE in the same /48 have the same penultimate infrastructure hop (provider router). For a given probe, we use IPvSeeYou to geolocate the EUI-64 CPE attached to the same provider router (in the same /48) as the probe. Then, we find the centroid of the associated EUI-64 CPE geolocations and use it as the inferred geolocation of the probe.

Figure 12 displays the error between the IPvSeeYou inferred Atlas probe locations and the reported location of the probe. The latter includes both the RIPE-injected error and any error introduced by the device owner when self-reporting their probe location. Nonetheless, the median distance is approximately ten kilometers, indicating that our methodology consistently geolocates the large and widely distributed set of RIPE probes with high accuracy. In some instances, our methodology detects probes that have likely changed locations without an accompanying RIPE update. For instance, one probe was purportedly in Los Angeles, CA, USA, but all other EUI-64 CPE in the same /48 as the probe geolocate to a Seattle, WA, USA suburb.

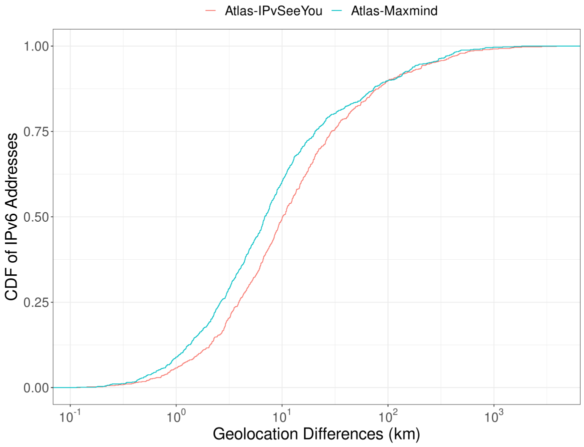


Figure 13: A CDF of RIPE Atlas nodes (non-EUI-64) depicting the distance difference between the reported location and MaxMind and IPvSeeYou-centroid geolocations.

Finally, we compare our inferred centroid locations to MaxMind's IP geolocations. Figure 13 displays the difference between the RIPE-reported location (including the error) and the MaxMind and IPvSeeYou-centroid geolocations. Note that none of these geolocations (MaxMind, IPvSeeYou-centroid, or RIPE) represents ground truth, and, in the case of the IPvSeeYou-centroid, the geolocation is an aggregate of other nearby CPE geolocations. However, there is general agreement between RIPE Atlas' error-injected reported location, MaxMind, and IPvSeeYou. We claim only feasibility, rather than improved accuracy over commercial databases, for non-EUI-64/infrastructure geolocation.

8.3. Coverage Gain from EUI-64 Clustering

Using our technique for associating clusters of geolocated EUI-64 CPE with non-geolocatable EUI-64 and non-EUI-64 CPE, we last evaluate the coverage gain. The coverage gain is simply the net increase in additional CPE that can be geolocated by leveraging the locations of other CPE connected to the same provider router. As an example, we perform Yarrp traces to random IIDs in each /64 within a single /48 of the large American ISP previously considered. In this exemplar /48, we discover 3,825 distinct CPE addresses, including 1,776 (46%) EUI-64 and 2,049 IPv6 addresses with random IIDs. Because /60 subnets are allocated to end-users in this /48, we would expect to see at most 4,096 unique IPv6 CPE addresses; thus, the /48 is nearly completely allocated and discoverable.

Employing IPvSeeYou from §3, we geolocate 180 of the 1,776 EUI-64 CPE in the service provider's /48 to the Olympia, Washington, USA metropolitan region, displayed in Figure 11. Assuming that the distribution of non-geolocatable EUI-64 and non-EUI-64 CPE does not vary significantly from the 180 geolocated CPE, we should expect to find these "hidden" devices in approximately the same location as the 180 geolocated CPE. Thus, our methodology enables the geolocation of all 3,825 CPE as opposed to the 180 for which we have precise geolocations, representing a CPE coverage gain factor of approximately 20.

9. Remediation

Despite privacy-preserving mechanisms for dynamically generating IPv6 addresses existing for two decades [25], tens of millions of CPE devices continue to use EUI-64 IPv6 addresses. EUI-64 SLAAC addressing, when combined with predictable MAC address assignments to device interfaces, enables an adversary to conduct the type of data fusion attack we outline in §3 and demonstrated the feasibility of in §7. The most straightforward solution to our attack is for more CPE vendors to employ a random addressing mechanism [7], [8], [9] to generate its WAN IPv6 address.

Toward the goal of encouraging vendors to adopt countermeasures to our attack, we responsibly disclosed our findings to the manufacturers of devices tied to over 7M EUI-64-derived MAC addresses in our corpus. Based on our findings, one manufacturer plans to transition to using modern randomized IIDs when the device generates WAN IPv6 addresses. A second vendor disputed our findings, specifically that their devices were exposing MAC addresses via EUI-64, despite clear evidence to the contrary.

Several additional mitigations for our attack also exist. CPE manufacturers can protect their devices from our data fusion attack by randomizing their MAC address allocation patterns. This breaks the linkage we infer between the EUI-64-derived WAN MAC address and BSSID. This mitigation requires the vendor to ensure that no fixed patterns exist in MAC address allocation, and maintain a strict account of which randomly-chosen addresses have already been allocated to avoid duplicate MAC assignment. Furthermore, continued use of EUI-64 SLAAC addressing still permits targeted attacks and device tracking. Therefore, we view this as a suboptimal solution.

A third protection mechanism against our attack is the use of randomized MAC addresses for either or both of the WAN MAC and 802.11 BSSID. Adopting MAC address randomization on either the WAN or 802.11 interface, wherein a new, random MAC address is generated at each power cycle or when a new EUI-64 IPv6 address is generated, would also prevent linking the two identifiers, and thus the geolocation fusion attack. However, the technical difficulty of implementing MAC address randomization in CPE devices is likely to be as or more difficult than simply enabling random addressing. Further, a CPE device with BSSIDs that change over time might be erroneously construed as an attacker conducting an "Evil-Twin" attack [60], [61], [3].

A final protection mechanism is to disable ICMPv6 responses on the CPE. This prevents an attacker from obtaining the CPE WAN address through active scans. However, in disabling ICMPv6, the operator also loses the ability to use ICMPv6 responses to troubleshoot networking issues. In many low-cost CPE, no mechanism to disable ICMPv6 is even exposed.

We strongly suggest the use of random IPv6 addresses due to the limitations of other potential mitigations. However, as noted in §8, unless *all* CPE devices employ these mitigations, even a single EUI-64 device can aid an attacker in geolocating non-EUI-64 CPE devices.

10. Summary and Conclusions

In this work, we demonstrated a location privacy vulnerability that exists in millions of deployed IPv6 devices. Despite best current practices discouraging the use of EUI-64 IPv6 addresses and their disuse in most modern endpoint operating systems, many CPE devices continue to generate IPv6 addresses by embedding the interface's MAC address in the lower 64 bits of the IPv6 address. Further, many CPE manufacturers assign MAC addresses predictably to the interfaces on their devices. Due to these two design choices, we were able to fuse two large datasets, one consisting of EUI-64-derived MAC addresses, the other of WiFi BSSID geolocations, to correlate these identifiers and discover the physical location of millions logical IPv6 addresses.

Toward this end, we developed a novel algorithm to determine the number of MAC addresses assigned to individual CPE devices and infer the offset between the WAN interface MAC address and WiFi BSSID. We found that over 12M EUI-64 IPv6 addresses in 146 countries could be matched to WiFi BSSIDs. Not only does this privacy vulnerability impact device owners whose products implement this legacy technology, but nearby devices connected to the same provider router can also be geolocated due simply to their proximity to EUI-64 CPE. The insecurity of even a few legacy devices jeopardizes the privacy of all of their neighbors, no matter how privacy-conscious.

Due to the consequences of our location privacy attack, we contacted several CPE vendors as well as a large ISP. Our results lead to the deprecation of EUI-64 addressing by a one manufacturer, and mitigation of the vulnerability within the network of the large residential service provider.

However, residential routers are rarely updated and infrequently replaced. Thus, the ossified ecosystem of deployed residential cable and DSL routers implies that IPvSeeYou will remain a privacy threat into the foreseeable future.

Acknowledgments

We thank kc claffy, Oliver Gasser, Will van Gulik, Thomas Krenc, Josh Kroll, Jeremy Martin, Justin Rohrer, and the anonymous reviewers for invaluable feedback, bobzilla from WiGLE, and Kevin Borgolte for shepherding. Authors Rye and Beverly are founders of sixint.io, which specializes in IPv6 intelligence and security solutions.

References

[1] "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture," *IEEE Std 802-2014 (Revision to IEEE Std 802-2001)*, pp. 1–74, 2014.

[2] J. Martin, E. C. Rye, and R. Beverly, "Decomposition of MAC Address Structure for Granular Device Inference ," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Dec. 2016.

[3] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, "Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, 2016, pp. 413–424.

[4] E. Fenske, D. Brown, J. Martin, T. Mayberry, P. Ryan, and E. Rye, "Three Years Later: A Study of MAC Address Randomization In Mobile Devices And When It Succeeds," *Proceedings on Privacy Enhancing Technologies*, vol. 3, pp. 164–181, 2021.

[5] M. Cunche, "I Know Your MAC Address: Targeted Tracking of Individual using Wi-Fi," *Journal of Computer Virology and Hacking Techniques*, vol. 10, no. 4, pp. 219–227, 2014.

[6] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 2462 (Standards Track), Internet Engineering Task Force, Dec. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2462.txt

[7] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 4941 (Standards Track), Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4941.txt

[8] F. Gont, "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)," RFC 7217 (Proposed Standard), Internet Engineering Task Force, Apr. 2014. [Online]. Available: http://www.ietf.org/rfc/rfc7217.txt

[9] F. Gont, S. Krishnan, T. Narten, and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6," RFC 8981 (Standards Track), Internet Engineering Task Force, Feb. 2021. [Online]. Available: http://www.ietf.org/rfc/rfc8981.txt

[10] R. Beverly, R. Durairajan, D. Plonka, and J. P. Rohrer, "In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery," in *Proceedings of ACM Internet Measurement Conference (IMC)*, Nov. 2018.

[11] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczyński, S. D. Strowes, L. Hendriks, and G. Carle, "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists," in *Proceedings of ACM Internet Measurement Conference (IMC)*, 2018.

[12] E. C. Rye and R. Beverly, "Discovering the IPv6 Network Periphery," in *International Conference on Passive and Active Network Measurement*, 2020, pp. 3–18.

[13] Broadcom, "Broadcom BCM3390 DOCSIS3.1 modem/gateway SoC," https://www.broadcom.com/products/broadband/cable/modems/bcm3390.

[14] H. Berghel, "Wireless Infidelity I: War Driving," *Commun. ACM*, vol. 47, no. 9, p. 21–26, Sep. 2004. [Online]. Available: https://doi.org/10.1145/1015864.1015879

[15] C. Hurley, *WarDriving: Drive, Detect, Defend: A Guide to Wireless Security*. Elsevier, 2004.

[16] "WiGLE – All the Networks. Found by Everyone." 2021, https://wigle.net.

[17] Apple, "Location Services and Privacy," 2021, https://support.apple.com/en-us/HT207056.

[18] A. Mylnikov, "Geo-Location API Download Section," 2021, https://www.mylnikov.org/download.

[19] openwifi.su, "OpenWifi.su Dataset," 2021, http://openwifi.su/db/.

[20] The International Organization for Standardization, "ISO 3166 Country Codes," 2022, https://www.iso.org/iso-3166-country-codes.html.

[21] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 1971 (Standards Track), Internet Engineering Task Force, Aug. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1971.txt

[22] S. Thomson, T. Narten, and T. Jimei, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Proposed Standard), Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4862.txt

[23] R. Hinden and S. Deering, "IPv6 Addressing Architecture," RFC 3513 (Standards Track), Internet Engineering Task Force, Apr. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3513.txt

[24] ——, "IPv6 Addressing Architecture," RFC 4291 (Standards Track), Internet Engineering Task Force, Feb. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4291.txt

[25] T. Narten and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 3041 (Standards Track), Internet Engineering Task Force, Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3041.txt

[26] Akamai, "Edgescape," 2020, https://developer.akamai.com/edgescape.

[27] MaxMind Inc, "MaxMind GeoLite Databases," 2021, https://dev.maxmind.com/geoip/geoip2/geolite2/.

[28] Hexasoft, "IP2Location," 2022, https://www.ip2location.com.

[29] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards IP Geolocation using Delay and Topology Measurements," in *Proc. ACM SIGCOMM*, 2006, pp. 71–84.

[30] B. Huffaker, M. Fomenkov, and k. Claffy, "Geocompare: A Comparison of Public and Commercial Geolocation Databases," 2011, pp. 1–12.

[31] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang, "Towards Street-Level Client-Independent IP Geolocation," in *USENIX NSDI*, vol. 11, 2011, pp. 27–27.

[32] V. N. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," in *Proc. ACM SIGCOMM*, 2001, p. 173–185. [Online]. Available: https://doi.org/10.1145/383059.383073

[33] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-Based Geolocation of Internet Hosts," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1219–1232, 2006.

[34] Z. Hu, J. Heidemann, and Y. Pradkin, "Towards Geolocation of Millions of IP Addresses," in *Proc. ACM IMC*, 2012, p. 123–130. [Online]. Available: https://doi.org/10.1145/2398776.2398790

[35] B. Eriksson, P. Barford, B. Maggs, and R. Nowak, "Posit: A Lightweight Approach for IP Geolocation," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 2, p. 2–11, Oct. 2012. [Online]. Available: https://doi.org/10.1145/2381056.2381058

[36] E. Kline, K. Duleba, Z. Szamonek, S. Moser, and W. Kumari, "A Format for Self-Published IP Geolocation Feeds," RFC 8805 (Informational), RFC Editor, Fremont, CA, USA, Aug. 2020. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8805.txt

[37] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?" *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 53–56, 2011.

[38] D. Komosnỳ, M. Vozňák, and S. U. Rehman, "Location Accuracy of Commercial IP Address Geolocation Databases," 2017.

[39] J. Wright and J. Cache, *Hacking Exposed Wireless: Wireless Security Secrets & Solutions*. McGraw-Hill Education Group, 2015.

[40] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, and G. Vigna, "Something From Nothing (There): Collecting Global IPv6 Datasets From DNS," in *Proceedings of the 18th Passive and Active Measurement Conference*, Mar. 2017.

[41] K. Borgolte, S. Hao, T. Fiebig, and G. Vigna, "Enumerating Active IPv6 Hosts for Large-scale Security Scans via DNSSEC-signed Reverse Zones," in *Proceedings of 39th IEEE Symposium on Security and Privacy*, May 2018.

[42] A. Murdock, F. Li, P. Bramsen, Z. Durumeric, and V. Paxson, "Target Generation for Internet-wide IPv6 Scanning," in *Proceedings of ACM Internet Measurement Conference (IMC)*, 2017.

[43] F. Li and D. Freeman, "Towards A User-Level Understanding of IPv6 Behavior," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 428–442.

[44] J. Berger, A. Klein, and B. Pinkas, "Flaw Label: Exploiting IPv6 Flow Label," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1259–1276.

[45] E. C. Rye, R. Beverly, and kc claffy, "Follow the Scent: Defeating IPv6 Prefix Rotation Privacy," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, Nov. 2021.

[46] T. V. H. Tran, "Ipv6 geolocation using latency constraints," Master's thesis, 2014.

[47] R. Beverly, "Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery," in *Proceedings of ACM Internet Measurement Conference (IMC)*, Nov. 2016.

[48] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-Wide Scanning and its Security Applications," in *22nd USENIX Security Symposium (USENIX Security 13)*, 2013, pp. 605–620.

[49] tumi8, "ZMapv6: Internet Scanner with IPv6 capabilities," 2021, https://github.com/tumi8/zmap.

[50] radiocells.org, "OpenBMap Dataset," 2021, https://radiocells.org/.

[51] H. Seiwert, "Github Repository for iSniff GPS," 2021, https://github.com/hubert3/iSniff-GPS.

[52] IEEE, "OUI Database," 2021, http://standards-oui.ieee.org/oui/oui.txt.

[53] "Latitude and longitude finder," 2022, https://www.latlong.net/.

[54] "Technicolor Home Page," 2021, https://www.technicolor.com/.

[55] "Xfinity," 2021, https://my.xfinity.com/.

[56] "IPInfo.io," 2022, https://www.ipinfo.io.

[57] Cable Television Laboratories, "Data-Over-Cable Service Interface Specifications, Operations Support System Interface Specification," https://www.cablelabs.com/wp-content/uploads/2015/08/CM-SP-OSSIv3.0-I05-071206.pdf.

[58] RIPE, "RIPE Atlas," 2021, https://atlas.ripe.net/.

[59] ——, "Are the locations of probes made public?" 2021, https://atlas.ripe.net/about/faq/#are-the-locations-of-probes-made-public.

[60] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel, "Hacker's Toolbox: Detecting Software-Based 802.11 Evil Twin Access Points," in *2015 IEEE consumer communications and networking conference (CCNC)*, 2015.

[61] K. Bauer, H. Gonzales, and D. McCoy, "Mitigating Evil Twin Attacks in 802.11," in *2008 IEEE International Performance, Computing and Communications Conference*, 2008, pp. 513–516.

[62] R. Padmanabhan, J. P. Rula, P. Richter, S. D. Strowes, and A. Dainotti, "DynamIPs: Analyzing Address Assignment Practices in IPv4 and IPv6," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, 2020, pp. 55–70.

[63] Team Cymru, "IP to ASN mapping," 2021, https://www.team-cymru.org/IP-ASN-mapping.html.

[64] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A Study of MAC Address Randomization in Mobile Devices and When it Fails," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 365–383, 2017.

Appendix: WAN and WiFi Corpora

Because the same MAC address can appear in many EUI-64 IPv6 addresses, we characterize our data using the MAC address as the primary unit of measure. For example, Versatel 1&1's (AS8881) prefix delegation policy causes many CPE devices to generate a new EUI-64 addresses every 24 hours [45]; in other providers, the delegated prefix and CPE EUI-64 address may remain stable for several months [62]. In our data, the maximum number of IPv6 addresses to a single WAN MAC in our corpus is 5,652, with a mean of 9.8 and median of 1 IPv6 addresses per WAN MAC. Note that these statistics depend substantially on the networks probed and the duration of the probing.

The 60,571,842 WAN MAC addresses in our corpus are embedded in EUI-64 IPv6 addresses from ASes corresponding to 200 different countries and territories as determined by Team Cymru's IP-to-ASN lookup service [63]. China contributes the largest fraction at 35%; the top ten countries each add over 1 million MAC addresses to the total. Although China leads all countries in MAC address count, Comcast, an American Internet Service Provider (ISP), is the top AS with over 10 million distinct MAC addresses. Comcast dominates the US contribution with 91% of the US EUI-64-derived MAC addresses; Guangdong Mobile, the leading Chinese AS, contributes only 37% of the total Chinese MAC addresses by contrast.

The OUI and manufacturer data we collect indicate that we discover 463,188 distinct OUIs embedded in EUI-64 IPv6 addresses. However, only 32,345 OUIs are listed in a recent IEEE OUI database [52]. This discrepancy has several potential root causes.

One basis for this variation is due to the randomness involved in generating temporary [7], [9] or stable [8] random addresses. Because EUI-64 IPv6 addresses are identified through a static `0xFFFE` in the fourth and fifth byte positions of the IID, a random process would be expected to produce *false-EUI-64* IIDs with probability $p = \frac{1}{65,536}$ (or $p = \frac{1}{131,072}$ if we require the U/L bit to be set). This type of falsely-identified MAC address is highly unlikely to result in a MAC-BSSID offset inference and IP-BSSID geolocation because there are unlikely to be any BSSIDs in the false MAC address' OUI.

A second underlying cause for the inflated OUI count is due to *real* EUI-64 IPv6 addresses being formed from a MAC address whose OUI is not registered in the IEEE OUI database. We observe ample evidence of unregistered OUIs in EUI-64 IPv6 addresses in our corpus. Four of the top five OUIs we observe resolve to the China Mobile IOT Company; the fifth (`F0:3C:91`) is not listed among IEEE-registered OUIs. However, all instances of MAC addresses using this OUI originate in EUI-64 IPv6 addresses from an American cloud hosting provider's networks (Linode). Table 2 summarizes our WAN MAC address data derived from EUI-64 IPv6 addresses.

Table 3 displays our BSSID geolocation dataset discussed in §3.3. The most commonly-geolocated country for our BSSIDs is the US, followed by Germany, Brazil, France, and Japan. Of note, China, which is the most-common country from our WAN MAC dataset, ranks 32 in most-common BSSID geolocations.

As with our EUI-64-derived MAC address dataset, we observe a significantly larger number of OUIs in our BSSID data than exist in the IEEE OUI database (850,083 vs 32,345). Again, several root causes for this discrepancy are possible.

First, many APs will invert the U/L bit of their BSSID to form virtual WLANs using a single NIC. We see evidence of this occurring; for instance, in our data we observe the TP-Link OUI `C0:4A:00` occur nearly 379k times in our BSSID corpus. This OUI with the U/L bit inverted (`C2:4A:00`) also appears in our data in 3,108 BSSIDs.

Other potential causes include users spoofing AP BSSIDs from unassigned OUI space, or wireless client addresses being erroneously uploaded to geolocation databases as AP BSSIDs. Because probe requests are typically sent from ephemeral, random source MAC addresses in modern mobile operating systems [4], [3], [64], probe requests entering the geolocation corpus could potentially add a large number OUIs to the BSSID corpus.

Nonetheless, the majority of our BSSID data come from IEEE-assigned OUIs. Over 75% (333,996,812 of 442,543,751) of unique BSSIDs come from allocated OUI space, while ~89% (392,926,586 of 442,543,751) of OUIs are from allocated OUI space or allocated OUIs with the U/L bit inverted.

TABLE 2: Summary of top countries, ASes, and OUIs of MACs embedded in EUI-64 IPv6 addresses. MAC addresses found in more than one AS are not included to account for potential MAC address reuse.

Country	**Count**	**AS**	**Count**	**OUI / Manufacturer**	**Count**
CN	21,425,581 (35.4%)	Comcast (AS7922)	10,188,218 (16.8%)	14:AD:CA / China Mobile IOT	904,783 (1.5%)
US	11,196,587 (18.5%)	Guangdong Mobile (AS9808)	8,004,879 (13.2%)	F0:3C:91 / Unknown	885,386 (1.5%)
DE	9,265,924 (15.3%)	Deutsche Telekom (AS3320)	6,353,101 (10.5%)	B0:30:55 / China Mobile IOT	875,657 (1.4%)
BR	3,404,573 (5.6%)	France Telecom (AS3215)	2,746,829 (4.5%)	FC:8E:5B / China Mobile IOT	839,804 (1.4%)
FR	2,753,927 (4.5%)	China Unicom (AS4837)	2,399,925 (4.0%)	FC:F2:9F / China Mobile IOT	738,947 (1.2%)
195 Other	12,525,250 (20.7%)	12,651 Other	30,878,890 (51.0%)	463,183 Other	56,327,265 (93.0%)

TABLE 3: Summary of BSSID geolocation dataset by geolocated country, BSSID manufacturer, and data source. The total number of unique BSSIDs is less than the sum of the individual data sources due to some BSSIDs existing in multiple datasets. A small number of BSSIDs (particularly small constants such as `00:00:00:00:00:01`) geolocate to multiple countries.

Country	**Count**	**OUI / Manufacturer**	**Count**	**Source**	**Count**
US	119,591,390 (26.6%)	A0:65:18 / VNPT Technology	2,206,621 (0.5%)	Apple API	444,860,460
DE	78,034,169 (17.3%)	98:9B:CB / AVM GmbH	1,352,222 (0.3%)	OpenWifi.su	29,340,881
BR	37,245,817 (8.3%)	3C:A6:2F / AVM GmbH	1,320,865 (0.3%)	Mylnikov	20,226,869
FR	32,464,391 (7.2%)	7C:FF:4D / AVM GmbH	1,311,865 (0.3%)	OpenBMap	15,384,623
JP	28,170,359 (6.3%)	38:10:D5 / AVM GmbH	1,282,799 (0.3%)	WiGLE	1,367,700
233 Other	154,588,509 (34.3%)	850,083 Other	442,543,751 (98.3%)	Total	450,018,123