Automated Detection of IPv6 Privacy Leakage in Home Networks

Ali Zohaib
University of Massachusetts Amherst

Amir Houmansadr
University of Massachusetts Amherst

Abstract

A promising feature of IPv6 is allowing devices to change their IP addresses periodically, thereby enhancing privacy against surveillance and censorship. However, legacy deployments of IPv6 are known to leak device identities, as the IP addresses associated with each device are a function of the device's MAC address. To address this privacy leakage, the community has developed privacy extensions to the IPv6 addressing mechanism.

Unfortunately, despite the many efforts towards privacy-preserving addressing standards, the use of (the leaky) legacy addressing is prevalent across the IPv6 address space, especially among residential routers and Internet of Things (IoT) devices. This specifically exposes home broadband users to a variety of tracking and surveillance risks. Recent research shows that even a single leaky device can compromise the whole home network it resides in, i.e., allowing an adversary to track all users across that network, correlate users' activities over time, or extract users' precise geolocation.

We observe that because of the large number of devices with different configurations, users are largely unaware of what devices on their home network might be using the leaky legacy IPv6 addressing. In addition, users trust their ISPs for adopting privacy best practices for IPv6 but lack visibility into their policies. For instance, a user may not know if their ISP rotates their network prefix. In this paper, we develop and present a tool that allows users with minimal technical expertise to scan their local home networks to identify the IPv6-leaking devices and observe their ISP's prefix rotation policy.

1 Introduction

IPv6 came into existence nearly two decades ago. Its deployment, however, has seen a large uptick in recent years, primarily because of the increased demand for Internet-connected devices in mobile and residential broadband networks. As of September 2022, Google reports almost 40% of its traffic is IPv6 [4] and APNIC data shows 32.9% of the global Internet users are capable of using IPv6 [1]. This growth in IPv6 usage requires a deeper focus on its security and privacy aspects. Of particular concern is the address generation mechanism that embeds hardware identifiers in user addresses, also known as SLAAC EUI-64 addressing [22]. As network interfaces have globally unique and static hardware identifiers, i.e., Media Access Control (MAC) addresses, IPv6 addresses that carry MAC addresses expose sensitive information to the upper layers of the network stack. This creates an array of privacy threats that could enable an adversary to track users across networks, perform address-based activity correlation, carry out device fingerprinting or extract a user's precise geolocation [5]. Given the aforementioned privacy concerns, most newer devices and operating systems use modern privacy standards such as privacy extensions to generate random addresses.

However, despite the many efforts made to improve privacy in IPv6 addressing, recent works show that long-standing concerns with legacy addressing across the IPv6 address space still plague the IPv6 ecosystem and can be effectively exploited to compromise user privacy [15, 16, 18, 19]. In [16], researchers demonstrate that if a home network router, commonly referred to as customer premises equipment (CPE), is using a legacy addressing standard that relies on EUI-64 (Extended Unique Identifier), it can be used as a tracker for other devices on the home network that uses IPv6 using active measurements. In [19], Saidi et al. broaden this finding and report that even if the CPE and the ISP employ best privacy practices such as using privacy extensions and prefix rotation, the presence of a single device on a home network that uses a EUI-64 address can in-turn act as an identifier for all IPv6-enabled devices on the network thereby affecting the whole end-user network's privacy. They identify that IoT devices are a major source of this privacy leakage in consumer home networks. Further, Rye et al. [15] present a sophisticated technique that can allow an attacker to extract a user's location that is accurate up to the street level, based on their IPv6 address.

In this context, we observe the source of the majority of

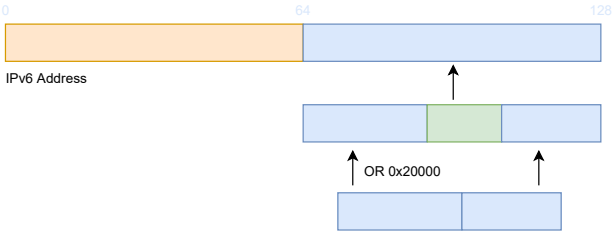

Figure 1: An EUI-64 IPv6 address constructed by embedding the 48-bit Media Access Control (MAC) address in the interface identifier portion.

IPv6 addressing problems lie in the different configurations of IPv6 standards in different devices ranging from CPEs to smart devices. Residential home-network users are largely impacted by privacy-leaking IPv6-enabled devices but there is a lack of visibility and awareness about the risks among end-users. Users purchase and may own devices without the knowledge of IPv6 support in them. Additionally, users trust their ISPs for employing privacy best practices (e.g prefix rotation) but cannot easily know their policies. Our motivation behind this work is to empower and educate end-users about their home IPv6 network.

We would like to provide users, who have minimal technical expertise to gain insight into the potential privacy leakage that is caused by legacy addressing configuration on devices on their home networks. To this end, we create a tool that allows users to locally scan their home networks to enumerate the IPv6-enabled devices on their network and identify devices that use legacy addresses. Additionally, our tool enables users to observe the IPv6 prefixes assigned by their ISP to their home networks over a period of time to identify their prefix rotation policy if there exists one. We also discuss the possible solutions and future directions that could enable an overall privacy-preserving IPv6 ecosystem.

Availability: Our implementation of the tool is freely available at https://github.com/SPIN-UMass/v6localscan. The tool is developed in Python and can be currently run on Linux/macOS-based systems. Pre-compiled binaries and application files for respective operating systems will be made available soon.

2 Background

In this section, we briefly introduce IPv6 concepts and discuss the security and privacy problems that are relevant to this work.

Addressing Schemes: IPv6 extends the address space to 128 bits per address from the 32 bits per address of IPv4. Of the 128 bits, the first 64 bits are associated with the routing prefix and the last 64 bits are dedicated to LAN-specific information which may be assigned in different ways. Addressing schemes include Stateless Address Auto Configuration (SLAAC) [8,9,20–22], Dynamic Host Configuration Protocol Version 6 (DHCPv6) [13] and manual assignments. While in DHCPv6, the router assigns the full 128-bit addresses to hosts, with SLAAC, the router sends the network portion of an address that may be up to 64 bits to the clients. Clients can then, themselves, choose the host part that constitutes the least significant 64 bits of the address. This host part is also called the interface identifier or IID. Historically, the host part was generated using a deterministic function of the interface's IEEE hardware Media Access Control (MAC) address known as the Extended Unique Identifier - 64 Bit (EUI-64). Figure 1 shows how an EUI-64 based address is generated. At first, in the last 64 bits of an address, the Universal/Local bit is set, then the bytes `0xFFFE` are inserted between the third and fourth bytes of the MAC address. Since EUI-64 addresses are derived from MAC addresses, they are globally unique but expose to layer-3 the host's layer-2 information such as the manufacturer, model or operating system. Additionally, they can be used to track devices across networks and correlate activity over time [6].

Privacy Extensions As a solution to the problem of fixed addresses, *privacy extensions* [9] are used in most modern devices wherein the client chooses a random lower 64-bit interface identifier that changes frequently.

Prefix Rotation: Since ISPs assign unique prefixes to clients, IP-based tracking is possible when prefixes are static. As a solution to this problem, some providers provide 'temporary mode' DHCPv6 where prefixes assigned to customers change periodically.

3 Related Work

The prevalence of EUI-64 addresses in the IPv6 address space is documented in multiple works [15–17]. A recent study by Zirngibl et al. [23] highlighted the presence of 282 million EUI-64 based addresses derived from 22.7 million MAC addresses in the active *IPv6 Hitlist* [7] service. Similarly, Rye et al. [16] reported a large number of CPEs that used EUI-64 IIDs and showed how users can be tracked despite the prefix rotation by the ISPs. They also present a privacy attack in [15] on residential routers based on EUI-64 addresses that allows an attacker to observe the *precise* location of a user by inferring the BSSID of the user's WiFi via their router's Wide Area Network (WAN) interface MAC address and using the BSSID to geo-locate user through wardriving (geolocation) databases. In [19], researchers passively collected traffic data from a large European ISP to analyze the possibility of tracking customer networks based on a single EUI-64 address. They found that 19% of all customers of the ISP had at-least

one device running a legacy addressing scheme and show that even a single device using a EUI-64 address can defeat the purpose of ISP-deployed prefix rotation and privacy extensions adopted by other device vendors. They highlight that IoT devices contribute the most to the privacy-leakage.

Another body of work that our work is related to is home network scanners. IoT-Inspector [11] is one such tool that targets users who wish to understand how the smart devices at their home, communicate with the Internet. Nevertheless, it only focuses on IPv4-running devices. IPv6 local scanning is available in tools such as NMap [12] and SI6 Network's IPv6 Toolkit [3]. However, we observe that these tools require a certain level of technical expertise to operate or do not provide a graphical user interface.

4 Methodology and Implementation

In this section, we describe the requirements for our tool, our methodology and enlist the implementation details.

4.1 Requirements

Our main requirement in creating the tool is to make it easy to install and run. Another requirement is to have a friendly User Interface (UI) that allows a user to easily scan and see the devices. A web interface would incentivize users to run the tool and visualize their IPv6-enabled home network.

4.2 Implementation

Compared to the task of globally scanning all IPv6 addresses, where the size of the address space becomes a challenge, local scanning is much simpler. RFC 7707 [10] notes the scanning of a local network can be done with Internet Control Message Protocol version 6 (ICMPv6) echo requests. Unlike IPv4, IPv6 does not support broadcast messages. Hence, to enumerate IPv6-enabled devices, ICMPv6 echo requests are sent to the multicast address group. There is a shortcoming to this approach though; because the implementations vary on different devices based on different standards, not all devices respond to the echo requests. For instance, as noted in [10], Windows systems (Vista, 7, etc.) do not respond to such requests. Therefore, similar to other local scanning tools, we use other types of ICMP probes to elicit responses from devices.

Our tool operates as follows:

- Upon running the tool, a packet sniffer listens for IPv6 packets on the network. Any multi-cast packets received such as Neighbour Solicitation or Router Advertisement messages are captured to initialize the enumeration of IPv6-enabled devices. At the same time, the tool opens up a browser window showing the list of discovered devices. Figure 2 shows a screenshot of the web interface listing all devices that are recognized by the tool.

- ICMPv6 echo requests are sent to the multicast address `ff02::1` for each of the source addresses configured on the machine based on the announced prefixes. As noted in [10], because the source addresses are different for each of the echo requests, devices on the network also respond with different source addresses, allowing the enumeration of most of the addresses in use on the local network.

- To elicit responses from devices that do not respond to ICMPv6 multicast echo requests, we send ICMPv6 packets with an unrecognized option of type `10xxxxxx`. This results in devices responding with ICMPv6 Parameter Problem error messages.

- We then send multiple mDNS requests to the multicast address `ff02::fb` to collect information about other devices and active services on the network.

Observing Prefix Assignments from the ISP: To observe the prefixes assigned by the ISP to the user, we collect prefix information from Router Advertisement messages. The delegated prefix is part of the message that allows a client to self-assign a full address via SLAAC. The tool saves this information to the user directory with the respective timestamp. Since prefix rotation may occur after different time lengths from ISPs (depending on their policy), running the tool over a period of time would reflect the changing prefix in the user interface, if the end user's ISP uses a prefix rotation. Figure 2 shows the graph that can be viewed to observe the number of the unique prefix assigned to the client over time.

Manufacturer Labelling: After enumerating the IPv6-enabled devices, we filter the devices that use EUI-64 based addresses. By reversing the process that is shown in Figure 1, the MAC address can be extracted from a EUI-64 address i.e removing the `ff:fe` bytes from the IID and checking for the U/L but. We then collect the first three bytes of the MAC address that constitute the Organization Unique Identifier (OUI) of the manufacturer. For the mapping, we use the IEEE OUI database [14] that contains details about the name and address of the manufacturer that registered the OUI. Since we are only focusing on devices with EUI-64 addresses, in the majority of the cases, the tool is able to resolve the organizational details. We acknowledge that determining the manufacturer of a device's network interface may not always provide a complete identification of the device. To improve our identification methods, we plan to incorporate techniques that may require collecting network traffic from devices in the future.

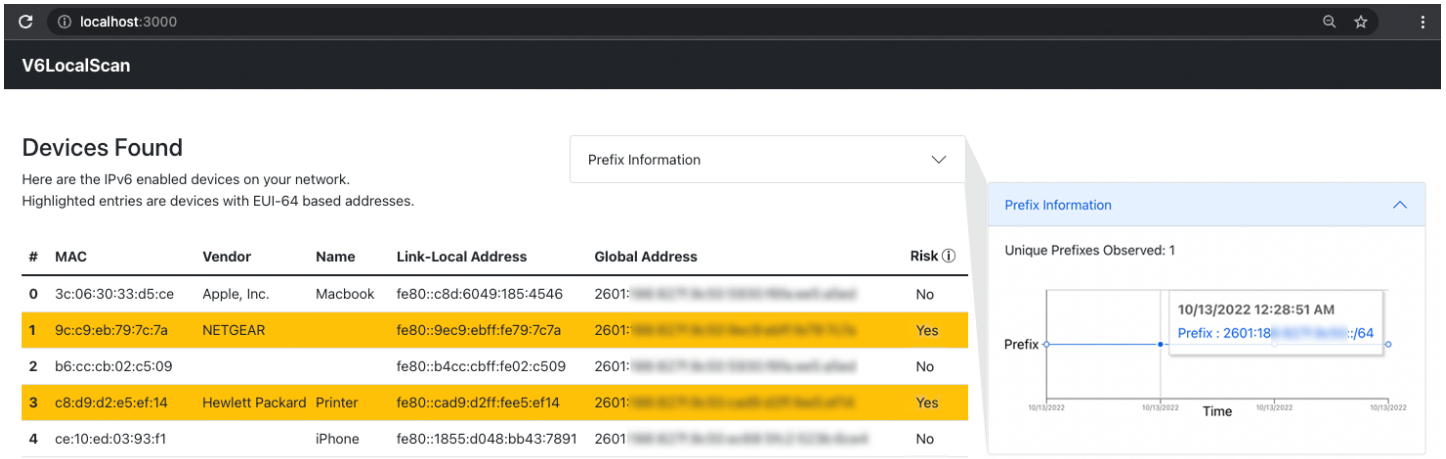

Figure 2: A screenshot of our tool's web interface that shows a list of IPv6-enabled devices on the network. Devices with MAC addresses embedded in their IPv6 addresses are highlighted. Observed prefixes announced by the router are shown at different timestamps in the right dropdown box. In this example, there is no prefix rotation employed by the ISP.

5 Discussions

The identification of devices running legacy IPv6 addressing standards in home networks is the first step toward the development of a privacy-preserving IPv6 ecosystem. The immediate question that follows is what can a user do about such devices? The most straightforward solution to avoid privacy leakage via EUI-64 MAC addresses is to employ random address mechanisms on all IPv6-enabled devices be it the router, gateway, mobile, or any IoT device. Unfortunately, although privacy-preserving techniques for IPv6 addressing have been around for as long as IPv6 itself, millions of devices still continue to use EUI-64 addresses. A change away from this requires action from both hardware and software vendors but we see mixed reactions from both. For instance, in [15], Rye et al. give an account of a large CPE manufacturer that failed to acknowledge that their devices were leaking MAC addresses via EUI-64 despite being presented with clear evidence. On the other hand, other large manufacturers like Apple are in full support of IPv6 privacy and employ the best practices [2]. We believe this problem is not just of systems but also of policy, regulation, and culture. This would require efforts from all stakeholders including device manufacturers, OS developers, ISPs, and users. We propose that device manufacturers and operating systems developers should self-regulate their products to use privacy extensions by default. ISPs should develop a policy to use prefix rotation for all IPv6 assignments. And customers should check and ask for details about devices/Internet services they purchase so manufacturers and ISPs are pressurized to undertake the best IPv6 privacy practices.

6 Future Work

While the current version of the tool that we have built works locally, in the next iteration of this work and similar to [11], we plan to crowd-source labeled data in a privacy-preserving manner via our tool to generate a dataset of IoT/mobile/CPE devices and manufacturers that employ legacy IPv6 standards on their devices. We aim to improve device identification by using techniques such as network traffic analysis and allowing users to add self-identified labels. This will enable us to gather more accurate data on device names and manufacturers while ensuring that personally identifiable information is not collected. Given enough users are able to run our tool on their home networks, it would enable a larger characterization of manufacturers (more specifically devices) that is not possible via active probing. Our hope is that insights from our tool will encourage more users to scan their home networks and generally increase awareness about IPv6.

Conclusion

In this paper, we address the issue of privacy leakage in IPv6-enabled home networks from an end-user's point of view. We first enumerate the potential issues that have been shown to impact IPv6 users. We then present a tool that provides users an insight to the user about the IPv6 deployment on their network to view 1) prefixes assigned to them by their ISP and whether they are rotated 2) the IPv6-enabled devices that use legacy configurations of the standard. We hope our work would encourage more users to understand the issues pertaining to IPv6 privacy so they can drive the efforts to develop a more privacy-preserving IPv6 ecosystem.

Acknowledgement

This work was supported by the NSF grant 1953786.

References

[1] IPv6 Measurement Maps. https://stats.labs.apnic.net/ipv6. (Accessed on 10/07/2022).

[2] IPv6 security - Apple Support. https://support.apple.com/guide/security/ipv6-security-seccb625dcd9/web. (Accessed on 10/14/2022).

[3] IPv6 Toolkit – SI6 Networks. https://www.si6networks.com/research/tools/ipv6toolkit/. (Accessed on 10/11/2022).

[4] IPv6 – Google. https://www.google.com/intl/en/ipv6/statistics.html. (Accessed on 10/07/2022).

[5] A. Cooper, F. Gont, and D. Thaler. Security and Privacy Considerations for IPv6 Address Generation Mechanisms. RFC 7721, RFC Editor, March 2016.

[6] Tianyu Cui, Gaopeng Gou, Gang Xiong, Zhen Li, Mingxin Cui, and Chang Liu. SiamHAN: IPv6 Address Correlation Attacks on TLS Encrypted Traffic via Siamese Heterogeneous Graph Attention Network. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 4329–4346. USENIX Association, August 2021.

[7] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D Strowes, Luuk Hendriks, and Georg Carle. Clusters in the expanse: understanding and unbiasing IPv6 hitlists. In *Proceedings of the Internet Measurement Conference 2018*, pages 364–378, 2018.

[8] F. Gont. A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC). RFC 7217, RFC Editor, April 2014.

[9] F. Gont, S. Krishnan, T. Narten, and R. Draves. Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6. RFC 8981, RFC Editor, February 2021.

[10] Fernando Gont and Tim Chown. Network Reconnaissance in IPv6 Networks. RFC 7707, March 2016.

[11] Danny Yuxing Huang, Noah Apthorpe, Frank Li, Gunes Acar, and Nick Feamster. Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–21, 2020.

[12] Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure, Sunnyvale, CA, USA, 2009.

[13] T. Mrugalski, M. Siodelski, B. Volz, A. Yourtchenko, M. Richardson, S. Jiang, T. Lemon, and T. Winters. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 8415, RFC Editor, November 2018.

[14] Institute of Electrical and Electronics Engineers (IEEE). Organizationally Unique Identifier (OUI) MAC Address Registry. https://standards-oui.ieee.org/oui/oui.txt. (Accessed on 10/10/2022).

[15] Erik Rye and Robert Beverly. IPvSeeYou: Exploiting Leaked Identifiers in IPv6 for Street-Level Geolocation. *arXiv preprint arXiv:2208.06767*, 2022.

[16] Erik Rye, Robert Beverly, and Kimberly C Claffy. Follow the scent: defeating IPv6 prefix rotation privacy. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 739–752, 2021.

[17] Erik C Rye and Robert Beverly. Discovering the ipv6 network periphery. In *International Conference on Passive and Active Network Measurement*, pages 3–18. Springer, 2020.

[18] Erik C. Rye, Jeremy Martin, and Robert Beverly. EUI-64 Considered Harmful, 2019.

[19] Said Jawad Saidi, Oliver Gasser, and Georgios Smaragdakis. One Bad Apple Can Spoil Your IPv6 Privacy. *SIGCOMM Comput. Commun. Rev.*, 52(2):10–19, jun 2022.

[20] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 1971, RFC Editor, August 1996.

[21] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 2462, RFC Editor, December 1998.

[22] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862, RFC Editor, September 2007. http://www.rfc-editor.org/rfc/rfc4862.txt.

[23] Johannes Zirngibl, Lion Steger, Patrick Sattler, Oliver Gasser, and Georg Carle. Rusty Clusters? Dusting an IPv6 Research Foundation. In *Proceedings of the 2022 Internet Measurement Conference*. ACM, October 2022.