

Wi-Fi Internet connectivity and privacy: hiding your tracks on the wireless Internet

Carlos J. Bernardos*, Juan Carlos Zúñiga†, Piers O’Hanlon‡

* Universidad Carlos III de Madrid. E-mail: cjb@it.uc3m.es

† InterDigital. E-mail: JuanCarlos.Zuniga@InterDigital.com

‡ Oxford Internet Institute, University of Oxford. E-mail: piers.ohanlon@oii.ox.ac.uk

Abstract—Internet privacy is a serious concern nowadays. Users’ activity leaves a vast digital footprint, communications are not always properly secured and location can be easily tracked. In this paper we focus on this last point, which is mainly caused by the use of IEEE Layer-2 immutable addresses. Randomization of the addresses used at Layer-2 is a simple, but promising, solution to mitigate the location privacy issues. We experimentally evaluate this approach, by first assessing the existing support of address randomization by the different operating systems, and then conducting several trials during two IETF and one IEEE 802 standards meetings. Based on the obtained results we can conclude that address randomization is a feasible solution to the Layer-2 privacy problem, but there needs to be other mechanisms used at higher layers to make the most benefit from it and minimize the service disruptions it may cause. As a conclusion of the paper and future steps, we discuss the possibility of using a context-based Layer-2 address randomization scheme that can be enabled with privacy features at higher layers.

I. INTRODUCTION

Nowadays, Internet privacy is becoming a huge concern, as more and more devices are getting directly (e.g., via cellular or Wi-Fi) or indirectly (e.g., via a smartphone using Bluetooth) connected to the Internet. This ubiquitous connectivity, together with not very secure protocol stacks and the lack of proper education about privacy make it very easy to track/monitor the location of users and/or eavesdrop their activity. This is due to many factors, such as the vast digital footprint that users leave on the Internet (e.g., sharing information on social networks, cookies used by browsers and servers to provide a better navigation experience, connectivity logs that allow tracking of a user’s Layer-2 (L2) or Layer-3 (L3) address, web trackers, etc.) [1] [2] and/or the weak (or even null in some cases) authentication and encryption mechanisms used to secure communications.

This privacy concern affects all layers of the protocol stack, from the lower ones involved in the actual access to the network (e.g., the Layer-2/Layer-3 addresses can be used to obtain the location of a user) to the applications, especially when browsing or using social networks (e.g., cookies can be used to find out the identity of a user accessing a particular site).

This paper focuses on the privacy threats at the network connectivity level, namely at the Layer-2 and Layer-3 of the protocol stack. We describe the main vulnerabilities that exist today with current operating systems and communication protocols, as well as some of the mechanisms proposed to mitigate the potential attacks that could be used. Then we look

at L2 address randomization as a solution capable of mitigating key privacy threats, which we experimentally assessed on trials conducted during the IETF 91st and 92nd meetings. Finally, we describe the different efforts that are currently going on at the relevant standardization bodies to tackle privacy concerns on the Internet and discuss some possible next steps.

II. BACKGROUND AND PROBLEM STATEMENT

Most mobile devices used today are Wi-Fi enabled (i.e., they are equipped with a IEEE 802.11 wireless interface). Wi-Fi interfaces, as any other kind of IEEE 802-based network interface, have a Layer-2 address (also referred to as *MAC address*), which can be seen by anybody who can receive the signal transmitted by the network interface. The format of these addresses¹ is shown in Figure 1. Addresses can either be universally administered or locally administered. A universally administered address is uniquely assigned to a device by its manufacturer. Most devices are provided with a universally administered address, which is composed of two parts: (i) the Organizationally Unique Identifier (OUI), which are the first three octets (in transmission order) and identify the organization that issued the identifier, and (ii) Network Interface Controller (NIC) Specific, which are the following three octets, assigned by the organization that manufactured the NIC, in such a way that the resulting MAC address is globally unique. Locally administered addresses are set-up by the network administrator, overriding the burned-in address. Universally administered and locally administered addresses are distinguished by setting the second-least-significant bit of the most significant byte of the address (the U/L bit).

Nowadays, it is relatively easy and simple to track a device (and therefore its owner) by observing its Layer-2 and/or Layer-3 address. L2 addresses can be easily observed by a third party, such as the operator of the access infrastructure or even a passive device listening to communications in the same network, especially for the case of over-the-air transmissions such as the ones performed by 802.11 Wi-Fi devices [3]. In an 802.11 network, a station exposes its L2 address in two different situations:

- While actively scanning for available networks, the L2 address is used in the Probe Request frames sent by the station.
- Once associated to a given Access Point (AP), the L2 address is used in frame transmission and reception,

¹We limit the description here to the 48-bit MAC address format, known as EUI-48.

as one of the addresses used in the address fields of an 802.11 frame.

In addition to the L2 problem, traditional L3 address assignment mechanisms such as the IPv6 stateless auto-configuration techniques (SLAAC) [4] generate the Interface Identifier (IID) of the address from its L2 address (via EUI-64), which then becomes visible to all IPv6 communication peers. This potentially allows for global tracking of a device at L3 from any point on the Internet. Besides, the prefix part of the address provides meaningful insights of the physical location of the device in general, which together with the L2 address-based IID, makes it easier to perform global device tracking.

There are some solutions that might mitigate this privacy threat, such as the use of temporary addresses (RFC 4191 [5]), the use of opaque IIDs (RFC 7217 [6] [7]) or even the use of random L2 addresses (as some Operating Systems do when performing active scanning). Next, we briefly describe how these solutions work.

RFC 4191 [5] identifies and describes the privacy issues associated with embedding L2 stable addressing information into the IPv6 addresses (as part of the IID) and describes some mechanisms to mitigate the associated problems. The specification is meant for IPv6 nodes that auto-configure IPv6 addresses based on the L2 address (EUI-64 mechanism). It defines how to create additional addresses (generally known as “temporary addresses”) based on a random interface identifier for the purpose of initiating outgoing sessions. These “random” or temporary addresses are meant to be used for a short period of time (hours to days) and would then be deprecated. Deprecated addresses can continue to be used for already established connections, but are not used to initiate new connections. New temporary addresses are generated periodically to replace temporary addresses that expire. In order to do so, a node produces a sequence of temporary global scope addresses from a sequence of interface identifiers that appear to be random in the sense that it is difficult for an outside observer to predict a future address (or identifier) based on a current one, and it is difficult to determine previous addresses (or identifiers) knowing only the present one. The main problem with the temporary addresses is that they should not be used by applications that listen for incoming connections (as these are supposed to be waiting on permanent/well-known identifiers). Besides, if a node changes network and comes back to a previously visited one, the temporary addresses that the node would use will be different, and this might be an issue in certain networks where addresses are used for operational purposes (e.g., filtering or authentication). RFC 7217, summarized next, partially addresses the problems aforementioned.

RFC 7217 [6] defines a method for generating IPv6 IIDs to be used with IPv6 Stateless Address Autoconfiguration (SLAAC), such that an IPv6 address configured using this method is stable within each subnet, but the corresponding IID changes when the host moves from one network to another. This method is meant to be an alternative to generating Interface Identifiers based on L2 addresses, such that the benefits of stable addresses can be achieved without sacrificing the security and privacy of users. The method defined to generate the IPv6 IID is based on computing a hash function

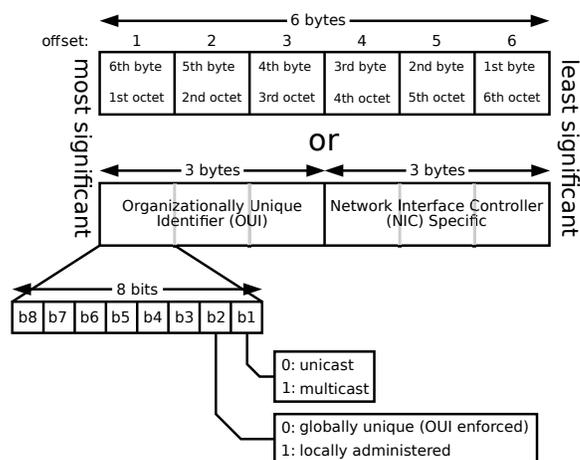


Figure 1. IEEE 802 MAC address format.

which takes as input information that is stable and associated to the interface (e.g., L2 address or local interface identifier), stable information associated to the visited network (e.g., IEEE 802.11 SSID), the IPv6 prefix, and a secret key, plus some other additional information. This basically ensures that a different IID is generated when any of the input fields changes (such as the network or the prefix), but that the IID is the same within each subnet. This solves some of the problems mentioned before.

However, these two solutions do not completely address all the problems and we can cite the following issues that still need to be tackled:

- 1) Existing solutions work in a non-coordinated fashion, which limits the effectiveness of the solution, as privacy has to be tackled at all layers to avoid privacy information leaking.
- 2) Solutions should be flexible, allowing an automatic/pseudo automatic/manual privacy protection activation. In some cases, a permanent address is required, for example due to operational operations (e.g., address-based authentication, access control lists, etc.). Therefore, the solution should allow for enabling/disabling the use of privacy mechanisms depending on the context, such as the location of the device or the characteristics of the network where the device is attaching.

III. LAYER-2 ADDRESS RANDOMIZATION

As described in the previous section, the IEEE 802 addressing includes one bit to specify if the hardware address is locally or globally administered. This allows generating local addresses without the need of any global coordination mechanism to ensure that the generated address is unique. This feature can be used to generate random addresses [8], and therefore makes it more difficult to track a user device from its L2 address. This feature is (partially) being used by some devices, which can be enabled to use random addresses during active Wi-Fi probe scanning².

²More information can be found at <http://www.zdnet.com/article/ios-8-randomizes-mac-addresses/> and <http://blog.airtightnetworks.com/ios8-mac-randomization-analyzed/>

Table I. ADDRESS RANDOMIZATION SUPPORT.

Tool	Platform	Addr. generation mode	Working?
ip	Ubuntu 14.04 - Intel iwlwifi/iwldvm drivers	Manually set	Y
	Fedora 20 - Intel iwlwifi/iwldvm drivers		
ifconfig	Ubuntu 14.04 - Intel iwlwifi/iwldvm drivers	Manually set	Y
	Ubuntu 14.04 - Intel iwlwifi/iwldvm drivers		
macchanger	Ubuntu 14.04 - Intel iwlwifi/iwldvm drivers	Manual & random auto.	Y
	Fedora 20 - Intel iwlwifi/iwldvm drivers		
SpoofMAC	Ubuntu 14.04 - Intel drivers	Manually set & random auto.	Y
	Mac OS X 10.7.5		
	Mac OS X 10.7.5 (MacBook, Wi-Fi:Atheros 5416)		
WiFiSpooF	MAC OS X 10.7.5	Manual & random auto.	Y
	Mac OS X 10.7.5 (MacBook, Wi-Fi:Atheros 5416)		
	v0.9.8.8 on Ubuntu 14.04 - Intel iwlwifi/iwldvm drivers - wpa_supplicant 2.1		
Network Manager	v0.9.10.0 on Debian Jessie - Intel iwlwifi/iwldvm drivers - wpa_supplicant 2.3	Manually set	Y
	v0.9.9.1 on Fedora 20 - Intel iwlwifi/iwldvm drivers - wpa_supplicant 2.0		
	Custom Android 4.2.2 on Nexus 4 and Nexus 7		
Pry-Fi	Custom Android 4.2.2 on Nexus 4 and Nexus 7	Manually set	N
iOS8 built-in randomization	iPhone 4s and iPad Mini v1 iOS 8	Random auto.	N
	iPhone 5s iOS 8.0.2		
PowerShell	Windows 7	Random auto.	Y
MAC Spoofer (changer)	Nexus 4 (Jelly Bean 4.2.2)	Random auto.	Y
	Nexus 5 (Cyanogen12 Android 5.0.2)		Some issues
	Nexus 7 2012 Wi-Fi (Lollipop 5.0.2)		N
	Nexus 7 2012 Wi-Fi (KitKat 4.4.4)		N
	Samsung Galaxy S (Gingerbread 2.3.6)		N

A. Randomizing the MAC address

The idea behind L2 address randomization is simple: to avoid using the burned-in universally administered MAC address on Wi-Fi devices, by configuring an automatically generated random address. This covers not only the actual connection to an Access Point (AP), but also the active scanning that the device performs periodically when not connected to any network. This idea has been explored in the past [8], but in this paper we adopt a very practical approach aimed at (i) giving some guidelines on how to perform L2 address randomization, (ii) assessing if the L2 address randomization can be effectively performed with current mobile devices and operating systems, and (iii) validating the mechanism in real scenarios and evaluating the problems that might appear.

In order to randomize the MAC address used by a Wi-Fi device, it is important to guarantee connectivity to the end-user. Since in this paper we are assessing the feasibility of a simple approach as a first step, we propose randomization of the L2 address in the following two situations (see Fig. 2):

- Every time the device connects to a new wireless network. By “wireless network” we mean an Extended Service Set (ESS), which is a set of connected access points. Each ESS has an ID called Service Set Identifier (SSID). When a device has just booted, come back from sleeping or performed a handover to a different ESS, a new locally administered L2 address is computed and configured on the interface. If the device roams from an AP to a new one of the same ESS, no MAC address change is performed.
- During periodic scanning, when the mobile device is not associated to any wireless network. While the previous situation is the most important one from the point of view of mitigating device tracking, this one provides an additional level of privacy, as it prevents APs and neighborhood nodes to “see” and track a device that is not connected.

We use all the available address space when using locally administered addresses, that is 46 bits (one bit used for U/L

and one used to indicate if the frame is unicast or multi-cast/broadcast). We next report on the experimental evaluation that we did on the feasibility of this randomization practice using existing hardware and major operating systems.

In order to benefit from a randomized Layer-2 address one also needs to take account of certain Layer-3 protocol interactions. The process of network attachment typically involves the acquisition of an IP address using the Dynamic Host Configuration Protocol (DHCP), and in some OSes, such as OSX/iOS and ChromeOS, the use of the Detection of Network Attachment (DNA) [9] protocol. These protocols can leak information about the device that can reduce effectiveness of L2 randomization [10], but this is something that is starting to be addressed within the Dynamic Host Configuration Protocol (DHC) IETF Working group (see Section IV). However in our experiment we have not attempted to tackle these issues as they would require OS level modifications.

B. Experimental evaluation

One of the main contributions of this paper is to describe different ways to perform MAC address randomization on a mobile device equipped with a Wi-Fi interface, as well as to report on our findings based on the major trials conducted at

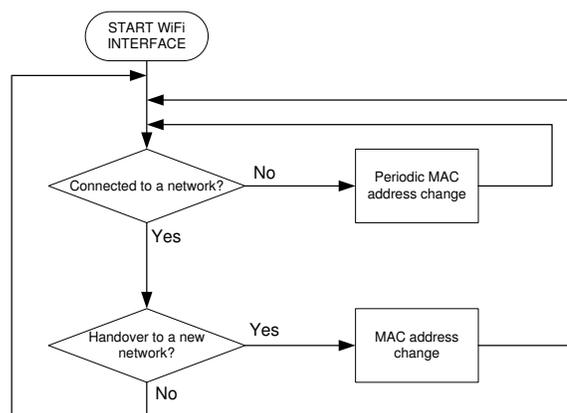


Figure 2. MAC address randomization diagram.

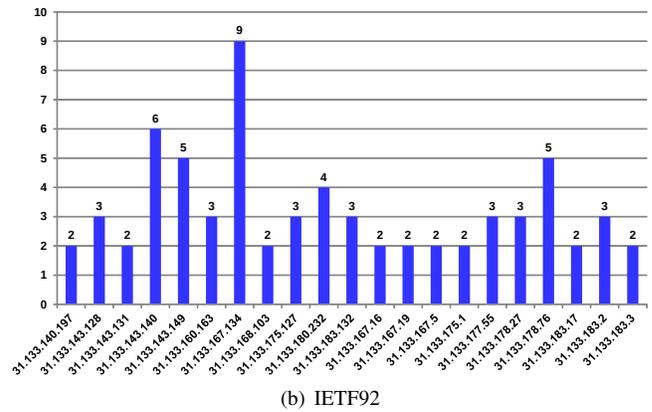
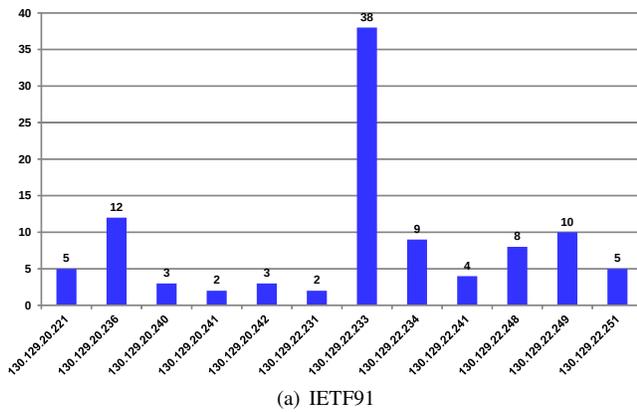


Figure 3. Number of MAC addresses per IP address, for those IPs that were assigned to multiple local MAC addresses.

the IETF and IEEE 802 meetings (between November 2014 and March 2015).

As part of the Internet Privacy efforts in coordination between IETF (IAB/IESG) and IEEE 802, we decided to perform some trials to randomize the MAC address of some user’s Wi-Fi devices. Our first goal was to analyze existing support from the different hardware and operating systems to randomize the MAC address used by the device. A major concern was to minimize complex operations on the terminals to encourage people to participate. A summary of our study is shown in Table I. Note that these results only report on those tools and platforms that we evaluated. Due to the vast variety of hardware and software that exist today, we could not cover all possible options, especially for the case of Android³. Obtained results show that there exist tools that allow L2 address randomization for Linux, Mac OS X and Windows, while iOS and Android devices are poorly supported. More detailed information on this platform feasibility analysis as well as detailed instructions to enable the address randomization, including scripts developed to automatize it, can be found on our wiki page⁴.

The next step towards our goal of assessing if randomizing MAC addresses can be considered as a mechanism to help improving Internet privacy was to trial its use in a real network. We wanted to evaluate the use of L2 address randomization from two different perspectives: (i) the effect on the connectivity experience of the end-user, also checking if applications and operating systems were affected; and (ii) the potential impact on the network infrastructure itself, for example if DHCP pools were exhausted, if an L2 address collision could cause a network disruption, etc. With these requirements in mind, we decided to conduct our experiments on the network deployed for the attendees during the IETF meeting that took place in Honolulu (November 2014), and then repeated the trial at the IETF meeting in Dallas, and the IEEE 802 meeting in Berlin (March 2015). The specifications of the IETF network⁵ include a dual link (for redundancy purposes)

with a minimum of 100 Mbps bidirectional bandwidth for the primary link (average bandwidth utilization is around 35 Mbps and peaked 80 Mbps), no content filtering at all, and both wireless and wired connectivity. This environment provided us with the perfect scenario for our experiment: on the one hand, we have expert users using the scripts we developed for address randomization and reporting their experience using them, and, on the other hand, we have close monitoring from the network side on the impact this experiment had on the network, including low-level logs that allowed us to perform a more detailed post-mortem analysis.

For the first experiment (conducted at the IETF 91, Honolulu, November 9-14, 2014), a specific SSID (`ietf-PrivRandMAC`) was deployed on the wireless IETF Internet infrastructure. It was deployed on all IETF physical APs, as an additional virtual AP. We used WPA PSK security, to avoid non participants accidentally connecting to our trial WLAN. The experimental access network was connected via a different VLAN to the DHCP server (which used an isolated pool of IPv4 addresses) and Internet gateway, providing isolation from the rest of the infrastructure. Participants were asked to notify their participation to a mailing list and use the WLAN address randomization scripts that we developed⁶. We also asked users to set-up the use of the DHCP client identifier (`dhcp-client-identifier`) for debugging purposes⁷, as this ensures that the same IP address is allocated for a client even if it changes its MAC address [11].

We next summarize the main results of the first trial, primarily obtained from analyzing anonymized logs from the DHCP server and `netdisco` management tool. During the week it lasted, 110 local MACs were seen on the trials WLAN network. If we look at the IP address allocation, 29 IP addresses were assigned to local MAC addresses. Out of them: 17 IP addresses were assigned to one local MAC address, e.g., because no DHCP client identifier was used by the client, and 12 IP addresses were assigned to multiple local MAC address

³Due to the large heterogeneity in the Android eco-system and hardware availability, the same tool offers quite different results in different platforms.

⁴https://oruga.it.uc3m.es/802-privacy/index.php/MAC_address_change_tutorial

⁵More details can be found at <https://iaoc.ietf.org/ietf-network-requirements.html>

⁶During this experiment, we supported 3 different OSes: Microsoft Windows (tested on Windows 7), Apple Mac OS X (tested on Version 10.10, alias Yosemite) and GNU Linux (tested on Debian testing/unstable, Ubuntu 13.10, and Fedora 20).

⁷Note that this was done for debugging purposes, as maintaining the same IP address might also introduce privacy concerns.

(Figure 3(a)). While it is hard to estimate the lifetime of a local MAC (i.e., the time it is used in the network) using the logs we had available, we were able to obtain some qualitative results: most of the local MACs never tried to renew the DHCP lease, whereas only a few MACs tried to renew the lease/obtain a new IP. These attempts might have been caused by a change of AP/WLAN network, or a suspend/wake-up, but the OS and user behavior also have an impact. The maximum time seen on the network between the first and last DHCP exchanges for the same IP address was 41 hours 51 min 41 sec, with an average “lifetime” of 4 min 46 sec. In terms of OS participation distribution, 53% were Mac OS X, 40% Linux, and 7% Windows users.

We repeated the experiment at the IETF 92 (Dallas, March 22-27, 2015) and IEEE 802 plenary (Berlin, March 8-14, 2015) meetings. Since no network operation issues were seen during the first trial, in the subsequent trials no isolated network was deployed, and therefore the experiment was conducted on all the attendees wireless networks. The only change that was introduced was using a shorter DHCP lease (e.g., one hour) for those IP addresses assigned to a local MAC. Participants were again asked to notify their participation to a mailing list and use the WLAN address randomization scripts provided, which in this case also included support for Android. In this second experiment, 144 local MACs were seen on the trials WLAN network. If we look at the IP address allocation, 97 IP addresses were assigned to local MAC addresses. Out of them: 76 IP addresses were assigned to one local MAC address, e.g., because no DHCP client identifier was used by the client, and 21 IP addresses were assigned to multiple local MAC address (Figure 3(b)). In terms of OS participation distribution, 50% were Mac OS X, 28.6% Linux, 14.3% Windows and 7.1% Android users.

In terms of impact on the network infrastructure, we detected a behavior on the DHCP server that deserves special attention. We asked users to use a specific `dhcp-client-identifier` so the server delegates the same IP address to a client even if it changes its L2 address. We noticed that if our DHCP server received a request for which it found a matching DHCP lease (i.e., existing `client-id`) within the 25% of the DHCP lease time, the server did not reply. This limits the speed a client can change its L2 address, which besides depends on a configuration parameter on the network side (the DHCP lease time). The implications of this issue requires further analysis.

It is also worth mentioning that the results we have obtained are affected by the software used in the trials. Future MAC address randomization tools should make use of strong address randomization mechanisms to minimize MAC address collisions and potential tracking attacks based on weaknesses of the randomization algorithm used.

C. Context-aware address randomization

L2 address randomization using the local address space is a very powerful privacy tool that will likely become an industry standard to make the tracking of users more difficult. Recently, the IEEE 802 started a Study Group⁸ to generate recommen-

dations and rules for using the local address space. Since L2 address randomization has to be selectively enabled/disabled, a default “always on” or “always off” policy will not be enough. For example, many networks use L2 address access filtering as part of their security policy, or use it to identify allowed users in a public hotspot (i.e., once the user provide the required credentials on a captive portal, L2 address is used to identify and authorize the user). In these kinds of scenarios, L2 address randomization has to be performed more carefully. Likewise, there could be scenarios where the user wittingly wants to be tracked (e.g. with a medical device, or within a constrained and trusted environment), in which case L2 address randomization should be disabled.

We believe the privacy configuration should be influenced by the context of the user. By context, we refer to the information that can be used to take a decision at a given place and time for a given user and service. As examples of context, we have: visible networks (i.e., networks the device could potentially connect to at that time and place), (geo-)location, information provided by the network (i.e., based on 802.11 beacons), etc. For instance, if a user is connected to his/her corporate network, which is trusted, properly secure, and L2 address access filtering is in place, L2 address randomization should be completely disabled. On the other hand, when the user is connected to a public hotspot, the MAC address should be randomized. This is just a simple example, but more complex policies could be expressed depending on the scenario. Besides, the privacy address configuration mechanism should allow the user manually overriding any of the automatic decisions, by selecting which kind of address generation should be used.

IV. STANDARDIZATION EFFORTS

Privacy and security issues have become priority items for the IETF, the Internet Architecture Board (IAB), and the Internet Society. Documents such as RFC 7258 [12] and the recent IAB Statement on Internet Confidentiality⁹ demonstrate the community’s commitment to addressing the issues and concerns raised. The Dynamic Host Configuration (DHC) working group has recently been involved in a number of privacy driven initiatives. It has recently initiated two drafts detailing the privacy implications of DHCP and DHCPv6, and has been working on the DHCP anonymity profile [13] draft that details a specifically limited set of fields and options that should be used if a system wants to maintain anonymity when between successive leases on visited networks. The draft includes such measures as setting the DHCP client identifier field solely based upon the MAC address, instead of the current common practice of using of a fixed trackable identifier, often including the device owner’s name. Furthermore the issue of captive portal access suffers from various privacy and operational issues and potential leakage of personal information, which is being tackled through the definition a new DHCP option [14], and a RA extension, to explicitly inform clients that they are behind a captive portal device, and that they will need to authenticate to get Internet access. The goals are to fix existing Internet technologies and protocols, and to develop more-secure solutions to protect users privacy.

⁸IEEE 802.1 Local Address Study Group: <http://www.ieee802.org/1/pages/lasg.html>

⁹<https://www.iab.org/2014/11/14/iab-statement-on-internet-confidentiality/>

Table II. SUMMARY OF RESULTS DURING IETF 91 AND 92 TRIALS.

Trial	#local MACs	#IPs assigned to local MACs	#IPs assigned to more than local MAC
IETF91	110	29	12
IETF92	144	97	21

Although the IETF is taking major actions on several fronts and via a host of working groups, its privacy and security efforts do not stop there. Coordination and collaboration with other standard organizations on the development of Internet technologies is a necessary next step to providing coherent solutions to today's privacy and security issues. One of the most important standards organizations is the Institute of Electrical and Electronics Engineers (IEEE), which has developed several technologies at the core of Internet connections, including IEEE 802.1 bridges, IEEE 802.3 Ethernet, and 802.11 WLAN (wireless local area network, a.k.a., Wi-Fi). As part of coordinated efforts between these organizations, a joint collaboration between the IETF and the IEEE has been established and an IEEE 802 Privacy Executive Committee (EC) Study Group (SG) was created in July 2014. The group is assessing privacy issues related to IEEE 802 technologies and is planning to develop recommended practices for all IEEE 802 protocols.

One of the privacy issues identified by the group so far relates to the use of media access control (MAC) addresses in over-the-air communications. Protocols such as IEEE 802.11 WLAN openly transmit MAC addresses in several messages. Because MAC addresses, in most cases, are globally unique identifiers that can be associated to personal devices, they can become privacy risks by exposing users to unauthorized tracking.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have highlighted privacy issues related to IEEE 802, and IETF technologies. The coordinated efforts between the two standards organizations led to the creation of the IEEE 802 Privacy EC SG. The group identified privacy issues related to the use of L2 (i.e., MAC) addresses, and also assessed the use of Layer-2 address randomization as a tool to enhance privacy. Three experimental trials were organized during the IETF and IEEE 802 meetings on the meeting networks. We observed that several client drivers support the proposed techniques, no major changes are required on the network configuration, and the probability of address duplication in a network with this characteristics is negligible. The experiments also showed that effective privacy tools should not work in isolation at a single layer, but they should be coordinated with other features. Finally, we point out that further study on the use of short-lived higher layer identifiers (i.e., above L2) and on implications of these changes on current implementations is still required.

ACKNOWLEDGMENT

The authors would like to thank the IETF and IEEE 802 standard network administrators, for all their support in setting up the experiments, especially Chris Elliot, Bill 'wej' Jensen, Jim Martin, Bill Fenner, Warren Kumari, Rick Alfvén and Brandon Height. We would also like to thank Fabio Giust for his help testing the support of MAC address randomization on the different operating systems and developing the automation scripts.

Piers O'Hanlon would like to acknowledge funding from the UK Engineering and Physical Sciences Research Council for the Being There project, grant EP/L00416X/1.

REFERENCES

- [1] H. Metwalley, S. Traverso, M. Mellia, S. Miskovic, and M. Baldi, "The online tracking horde: A view from passive measurements," in *Lecture Notes in Computer Science Traffic Monitoring and Analysis*, vol. 9053. Heidelberg: Springer, 2015, pp. 111–125. [Online]. Available: <http://porto.polito.it/2602582/>
- [2] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, ser. WPES '05. New York, NY, USA: ACM, 2005, pp. 71–80. [Online]. Available: <http://doi.acm.org/10.1145/1102199.1102214>
- [3] P. O'Hanlon, J. Wright, and I. Brown, "Privacy at the link layer," in *"STRINT Workshop: A W3C/IAB workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT)"*, Feb. 2014.
- [4] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Draft Standard), Internet Engineering Task Force, Sep. 2007, updated by RFC 7527. [Online]. Available: <http://www.ietf.org/rfc/rfc4862.txt>
- [5] R. Draves and D. Thaler, "Default Router Preferences and More-Specific Routes," RFC 4191 (Proposed Standard), Internet Engineering Task Force, Nov. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4191.txt>
- [6] 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>
- [7] F. Gont, A. Cooper, D. Thaler, and W. Liu, "Deprecating EUI-64 Based IPv6 Addresses," draft-gont-6man-deprecate-eui64-based-addresses-00, Internet Engineering Task Force, Oct. 2013.
- [8] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis," *Mobile Networks and Applications*, vol. 10, no. 3, pp. 315–325, 2005.
- [9] B. Aboba, J. Carlson, and S. Cheshire, "Detecting Network Attachment in IPv4 (DnAv4)," RFC 4436 (Proposed Standard), Internet Engineering Task Force, Mar. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4436.txt>
- [10] I. Papanagioutou, E. M. Nahum, and V. Pappas, "Configuring DHCP leases in the smartphone era," in *Proceedings of IMC*. New York, USA: ACM, 2012. [Online]. Available: <http://doi.acm.org/10.1145/2398776.2398814>
- [11] T. Lemon and B. Sommerfeld, "Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)," RFC 4361 (Proposed Standard), Internet Engineering Task Force, Feb. 2006, updated by RFC 5494. [Online]. Available: <http://www.ietf.org/rfc/rfc4361.txt>
- [12] S. Farrell and H. Tschofenig, "Pervasive Monitoring Is an Attack," RFC 7258 (Best Current Practice), Internet Engineering Task Force, May 2014. [Online]. Available: <http://www.ietf.org/rfc/rfc7258.txt>
- [13] C. Huitema, T. Mrugalski, and S. Krishnan, "Anonymity profile for DHCP clients," draft-huitema-dhc-anonymity-profile-02, Internet Engineering Task Force, Apr. 2015.
- [14] W. Kumari, O. Gudmundsson, P. Ebersman, and S. Sheng, "Captive-portal identification in DHCP / RA," draft-wkumari-dhc-capport-12, Internet Engineering Task Force, Mar. 2015.