

Marauder's Map: Sniffing MAC addresses in the MIT wireless network

Ali-Amir Aldan, Omer Cerrahoglu, Erjona Topalli, Xavier Soriano

{aliamir, omerc, erjonat, xsoriano}@mit.edu

Abstract

There are intrinsic security problems in wireless networks. Malevolent third-parties can use them to attack users and breach privacy protocols. Man in the middle attacks are possible and relatively cheap to implement as our paper will demonstrate. Public information can be used to infer private data about individuals.

We present in this paper the Marauders Map, a system designed to track and visualize movement of individuals on the MIT campus. To meet this end we employ packet sniffing. Marauders Map sniffs portable devices, such as smartphones and laptops, that have embedded wireless cards in them. These Wi-Fi enabled devices periodically broadcast in plaintext their unique MAC address along with other potentially sensitive information. This makes users in a wireless network vulnerable, especially in a place like MIT where students have their smartphones on most of the time and use MIT's wireless networks frequently. Since most areas in MIT are publically accessible and compromisable, we argue that by using several methods of data tracking and extraction we can discover a mapping between an individual and their MAC address, as well as generate a location map of users. We demonstrate this through a small demo. Ability to build such a system creates serious implications about peoples privacy and security.

1. Introduction

WiFi technology is ubiquitous today, having been deployed extensively in public and private areas. Most smart devices use Wi-Fi technology as the medium of communication and possess Wi-Fi network interfaces. This allows cheaper and faster access to the Internet compared to other technologies like GSM. At MIT particularly, with a campus-wide network providing

free Internet, students are constantly connected to Wi-Fi networks. Since most of the student-busy areas of MIT are publicly accessible, Wi-Fi frames of a large body of students are open to potential attacks. Of our concern is a particular vulnerability of IEEE 802.11 standards - despite the data in frames being encrypted in secure networks, headers of the frames are sent as plaintext. Those headers expose MAC addresses of the devices in the communication. As a result, an adversary might leverage MAC address information in combination with signal strength to track the owner of the device.

This paper considers an attack that tracks users and exploits the publicly available MAC addresses. The attack is enhanced using a simple real-time visualization tool inspired by Harry Potters Marauders map ([1]), the demo of which is presented in Section 5. This attack is based on previous work. [2] describes a system that accurately estimates position of devices by using signal strength, which could be used to estimate trajectories of the devices, as was shown in [4]. Additionally, *TODO: citation here* described the vulnerability of unencrypted frame headers and described possible ways of connecting MAC addresses to individuals. We employ these techniques to describe a system tailored for campuses of universities.

The paper is organized as follows. Section 2 presents problem statement and the motivation behind our attack. In Section 3 we give a brief technical introduction to the Wi-Fi technology and the existing approaches to the attack. In Section 4 we talk about our attack model. A detailed description of Marauders Map and a demo implementation are given in Section 5. Finally, in Section 6 we discuss possible prevention mechanisms.

2. Problem Statement and motivation

It is a well known fact that one may listen to broadcasts of packets from devices to wireless access points. Those broadcasts happen at link-level, meaning that they contain the plaintext MAC address of the communicating devices. Our team decided to explore whether one can learn MAC addresses of people on campus, track their whereabouts using those packets, and what are the implications of such a security issue.

With unlimited access to the network information, MIT system administrators could easily set up such a tracking mechanism. However, we hope that they would never do such a thing. The problem that we want to explore is that of an adversary, potentially a stranger to MIT, who might clandestinely place small low cost devices around campus in order to achieve such a

goal.

Our motivation behind this attack is to prove that such a low cost mechanism is possible, and to raise awareness about this security design flaw. Whereas this problem has been well-known and brought up several times in the interweb for years, there are many people who do not suspect this privacy issue.

3. Background

3.1. The ever-growing presence of Wi-Fi

Wi-Fi-like technology has been around for several decades, but it was not until the end of the 90s that it took the name of Wi-Fi and became commercially available [10]. Ever since then, the world has experienced a real wireless revolution with faster and safer protocols being developed at faster rates[11] [12]. With more than a hundred million [13] Wi-Fi hotspots in the world and a rapid growth of IoT devices, the world is more connected every single day. However, its omnipresence has also made it easier and cheaper to exploit the flawed design decisions that were made in the early years of this technology. Many of these flaws are well-known and have been repeatedly mentioned in articles and blog posts floating on the Internet. One question that keeps coming back is how much does a MAC address of a device tell about its user. There are some that argue that MAC addresses represent the Identity of a computer system, and it certainly acts as an important parameter to how secure the wireless network communication is. The more paranoid describe darker scenarios: from how Wi-Fi can be used by big companies to track you in a city and learn about your shopping patterns to how an adversary could sniff the packets transmitted by an individuals device, and use the contents of the packets to learn the individuals identity, whereabouts, and potentially critical personal information transmitted via these insecure networks.

3.2. Wi-Fi technology

In this section we will talk about the technology behind Wi-Fi. Wi-Fi is a technology that allows electronic devices to connect to wireless LAN networks. A typical Wi-Fi network is composed of access points and stations. Stations are devices equipped with a wireless interface card and provide a network physical layer on top of the radio link of another station. APs are

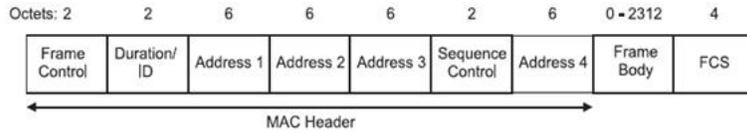


Figure 1: Structure of the MAC Header.

typically connected to the LAN through a wire. Stations communicate with each other using radio frequencies (RF) between 2.4 GHz and 2.5 GHz.

Wi-Fi technology is based on 802.11 protocols. Wireless networks broadcast their packets using RF or optical wavelengths, which make it easy for anyone to listen in, to manufacture their own spoof packets and perform various attacks of different nature in the network. Users of these devices are in constant risk from malicious third-parties.

The packets transmitted through the Wi-Fi network are called frames. Access Points (or APs) provide 802.11 frame distributions to stations connected to them. Frames can be classified into three classes: management frames, control frames and data frames. Management frames establish and maintain communication, control frames help in data delivery and data frames contain source and destination MAC address, SSID and TCP/IP datagram. The figure below presents the structure of a 802.11 frame: 20 bytes long header and a payload followed by a 4 bytes checksum. Address 1 and Address 2 are the source and destination addresses respectively.

MAC Address Both wireless and wired network interface cards have them. They stand for Media Access Control addresses. A MAC address is a unique 48-bit number, assigned to devices at the time of manufacture. It is often represented as a string of six octets separated by colon, e.g., 87:12:4E:22:G3:E8. However, though the MAC address is supposed to be unique, not all manufacturers make sure this is the case, and since it is written in software, it can easily be rewritten and there are multiple softwares that can do that. ¹

The MAC addresses can be used as a means of identification of a Wi-Fi-enabled device in the network, and frames sent by a device with a specific MAC address could be used to determine the location of the device when received signal strength at multiple locations is taken into account.

¹The APs also have a 0 to 32 byte long Service Set Identifier (SSID) that is also commonly called the network name. The SSID is used to segment the airwaves for usage. We wont use SSIDs in our attack.

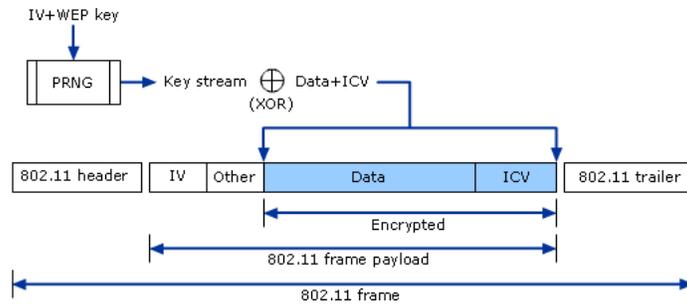


Figure 2: structure of a 802.11 frame: a 30 bytes long header and a payload followed by a 4 bytes checksum. Within the header the Address fields contains MAC addresses: Address 1 designates the source and Address 2 the destination. Note that while the payload could be encrypted, the header, which contains MAC addresses, is sent in the clear as plaintext

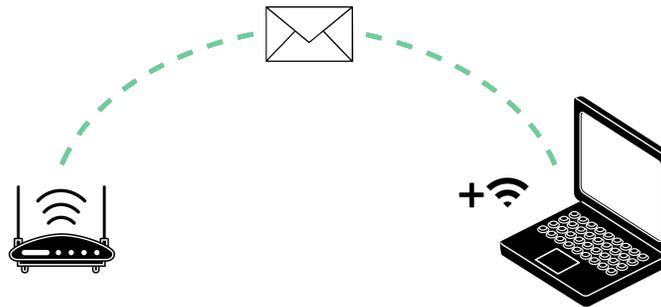


Figure 3: RadioTap header is placed by the receiver and contains signal strength (RSS)

Figure above presents the structure of a 802.11 frame: a 30 bytes long header and a payload followed by a 4 bytes checksum. Within the header the Address fields contains MAC addresses: Address 1 designates the source and Address 2 the destination. Note that while the payload could be encrypted, the header, which contains MAC addresses, is sent in the clear as plaintext.

RadioTap Additionally, frame headers sometimes contain signal strength (RSS) information. These headers implement RadioTap protocol. RadioTap is an interesting protocol because it is placed in the frame by a receiver capable of sensing signal strength, rather than by the sender. Marauders Map as well as other wireless indoor positioning systems, described next, rely on this header to locate their targets.

3.3. Wireless Indoor Positioning

3.4. Wireless tracking of MAC addresses

4. Our Attack Model

4.1. Batman vs. The Joker

In order to describe our attack model we need to travel to the alternate Earth 857 Universe. In this world we will meet Bruce, a young MIT student who has not take 6.857 does not know much about security. He is very fond of technology so he owns several wireless and IoT devices and in particular he always carries his cellphone with him. He also is Batman.

We meet the Joker, Batmans adversary and archenemy. He also goes to MIT, but he is evil. He took 6.857 and learned about thinking adversarially. Surprisingly for an MIT student, the Joker has a lot of free time though a limited amount of cash. He has decided to use his sparse resources to learn Batmans secret identity.

The Joker uses his money to buy a number of Raspberry Pis and Wireless Network cards capable of going on Monitor mode to sniff for packets. The Joker wants to meet with Batman, hoping that he is carrying his cellphone with him, and distract him for long enough to sniff his MAC address out using one of the Raspberry Pis. For this purpose and following the long-held tradition of MIT, the Joker performed a hack (MIT hack, not a computer hack), but he didnt follow the rules because he is a villain after all. He destroyed MIT property by bending and breaking the Bexley Memorial Park Tower. He sat in the site and laughed hysterically until Batman showed up. He then engaged Batman on an extensive conversation about anarchy and how he got the scars in his face, while secretly he ran his sniffing code in a Pi hidden in the debris of the tower.

When he was sure he had captured Batmans MAC address the Joker was ready to leave, so he redirects Batmans attention to New House, where he has again created chaos by exploding all the hot water pipes causing flood, and trapping the students between hot water and the exits. Batman goes to assist the trapped New House students and let the Joker escape via the MIT tunnels this time. Then, the Joker learned Batmans MAC address and could start stage two of his plan. He programmed all of his Pis as sniffers to target specifically Batmans MAC address, and he planted his sniffers in several strategic locations around the MIT campus: many in the Infinite Corridor, and large lecture halls such as 10-250, 26-100, and 32-123. Over the lapse of a couple of weeks, he retrieved a map of locations and times

where the MAC address had shown up and based on this information he inferred Batmans class schedule. He showed up early for several of Batmans classes, carrying sniffer with him, and waited until Batmans MAC address appeared. The Joker now knows that Batman is Bruce.

4.2. A more realistic scenario

What we described in the earlier section is an example of targeted MAC address attack. A lot of work has been done regarding this problem in [4]. Our approach goes a little further. We are not interested in particular targets, rather our attack aims at knowing several MAC addresses, acknowledge our victim's presence in an specific area, and even map out their movement. Depending on the number of Raspberry Pis in place, one can get a high level idea of the movement of any MAC address (such as where they take a turn on the infinite, or if they were in one of the large classrooms), or specific details of the movement of a MAC address at any point of time. If one wants to track only a specific target (or a small number of them), one could first plant the Raspberry Pis in the strategical locations presented earlier, and then on the movements we got we could place them in more specific locations to get more accurate details.

A real adversary could be interested on stalking a member of the community of MIT or another prestigious university. This is a more realistic scenario given that the children of many important political leaders, renowned scientists and business figures attend to such universities. The adversary would be more silent than the Joker in the process of extracting the MAC address. Of course he would not destroy MIT buildings, but instead he could possibly seat close to the victim in a number of different places such Stata Cafe, 5th floor Student Center, and Barker Library and then narrow down to only the repeated MAC addresses in all locations. Eventually, he will get exactly the MAC he is looking for. The intentions of the adversary may be genuinely evil and pragmatic, and by creating such a tracking mechanism and given that he can extract the MAC address of individual users, he could learn personal information about his victims: not only their class schedules, but also their routines, where they usually eat, and even where they live (if they live on campus). In this way, a resourceful adversary would get access to a tracking system not unlike Harry Potters infamous Marauders map and, in this way, jeopardize the safety and integrity of members of our community.

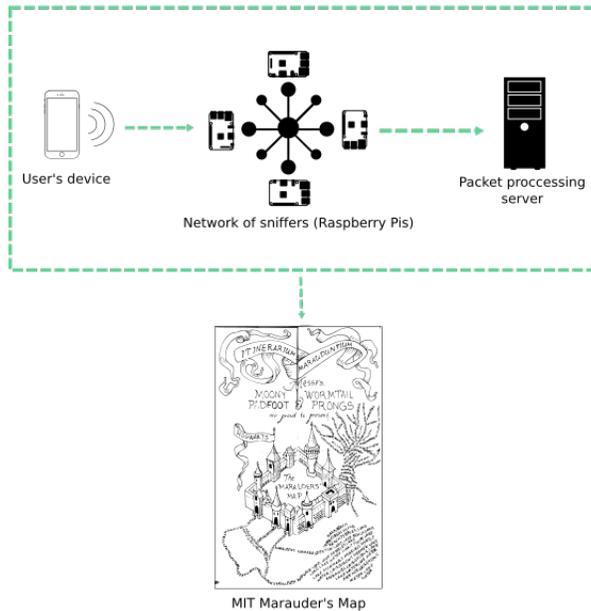


Figure 4: Marauders Map

5. Marauders Map

5.1. Overview

Marauders Map consists of two modules, the sniffers and the server. Sniffers listen on all packet headers and forward signal strengths, MAC addresses and timestamps to the server via Access Points in their range. Note that the range of a sniffer is at least 20 m, as per our own experimental evidence, and when a packet is received, a RadioTap header is automatically added to it. The header contains the signal strength and frequency of the channel where the packet was sent to. One can use this metrics to estimate the distance from the packet to the source (i.e. the distance between the sniffer and the smartphone/laptop) using the free-space path loss formula: $20 * \log(dist) + 20 * \log(freq) + K = signalStrength$, as we did in our baseline implementation. We note, however, that this estimation could be enhanced further as described in [2]. A final touch to the map, is an individual to MAC address mapping algorithm, which we describe in subsection 5.2.

5.2. Sniffers

For each sniffer, the server stores a mapping of MAC addresses to the latest observed signal strength and timestamp. It then estimates locations of the devices for each MAC address by performing a triangulation on calculated distances from sniffers. This triangulation takes into account the campus map and the location of APs. Our demo, described in section 5.3, leverages this calculation to estimate location of a device in a corridor using two sniffers. Using this triangulation, the server estimates current locations of the devices and plots them in a User Interface (UI) for the owner of the map to see.

Apart from the mapping to latest messages from sniffers, the server stores all of the messages in a table with MAC addresses as rows. This is done to estimate the resulting trajectory from the accumulated data and analyse MAC addresses with the purpose of identifying potential mappings between individuals and movement patterns. We describe three ways to identify individuals.

One method is to stalk a target individual and collect a trace of packets captured by the sniffing device. This is a method that is described in [4]. The second method makes use of extracting data from the server and extrapolating useful identifying information. Using this method the attacker can pull up all of the classrooms visited by each MAC address. Since most locations and room numbers for classes are available online, an attacker can associate each MAC address with potential list of classes. It then can use data of occurrences near dorm entrances. Extrapolating information this way can help create a schedule and dorm assignment of the individual carrying the device with a given MAC address. To identify a MAC address an adversary can use the schedule of the individual. We assume a wide variety in class schedules of students at MIT. With that in mind, second approach should narrow the choice of MAC addresses to a few. In order to arrive at a single decision, adversary may approach the carriers of the devices from afar by observing the map, to see which of the MAC addresses correspond to the target.

Finally, an attacker perform a reverse mapping from a MAC address to an individual. An adversary can place a single well-hidden camera in the infinite and store sets of faces every 10 minutes in sets that will be linked to the set of the MAC addresses being sniffed by the one Raspberry Pi sniffer located in the infinite. Later, the attacker can pull up the mapping of sets and process the data. A clustering algorithm, such as tSNE for example, may be used to identify a single individual that appears in most of those

occurrences. Due to the Infinite being a central place at MIT, a wide range of MAC addresses may be mapped this way.

5.3. Baseline implementation

We have implemented a simple demo, that mimics the final design. In our implementation, we used two Macbooks as sniffers, due to lack of WiFi cards able to operate in monitor mode for our Raspberry Pis, and one of the Macbooks, additionally, played the role of the server. For sniffing, we used a C++ library libtins, which enables listening on packets from channels. With libtins, we looked for packets with RadioTap (for signal strength) and Dot11 (for MAC addresses) headers. Communication between the server and the sniffers was implemented using a messaging broker called RabbitMQ. Further details may be found in the code.

5.4. Further work and Issues

5.4.1. Demo Issues

In our demo, we used a vanilla implementation of distance estimation together with a simple circle intersection logic. Such an approach resulted in a few problems: (1) Distance estimations were noisy, (2) Distance estimations change based on obstacles between the sniffer and the device. In addition to that, sniffers should periodically switch between channels in order to not miss devices that are communicating on other channels. This introduces a reduced frequency in acquisition of data. Despite these factors, as was shown in previous works, it is possible to employ a more accurate positioning system.

5.5. Hardware Issues

The sniffer code was developed on a Mac OS environment for ease debugging. This brought some expected complications when compiling it in the Raspberry Pis. Additional effort is necessary when performing the linking of shared libraries, and manual manipulation of the Wi-Fi card functionality (enabling, disabling, mode changing) was also needed. Also, with an inferior processing power, a lower frequency in acquisition of data is to be expected. This results in a even less smooth path tracing.

5.6. Extensions

A potential extension is the use of more specialized, but less customizable wireless tracking devices, which are already used as the hardware for commercial IPS solutions. One option is Open-Mesh which is described as

a facilitator for ultra low-cost, zero config, plug and-play wireless mesh networks *cite facilitator* and are used as APs in malls, hotels, apartments, etc.

We have also begun to explore the possibility of a three node sniffing system. One of the main considerations for such system is the likely possibility that the three circles may not intersect. To address this problem, we have drawn upon the Euclidean Geometry concept of the line called radical axis. A point belongs to the radical axis of a pair of circles if and only if the difference of the squares of their respective point-to-center distances and radii are the same. Not only it is a known fact that the three radical axes of three circles are concurrent, but it also provides a measure of the distance error with a similar morphology to a mean square error calculation.

5.7. Results

In our experiments, we identified that the range in which our sniffers were able to hear the devices was at least 20 meters. The rate at which packets from a given device are received is approximately 5-10 packets per second. The error margin in distance estimation with our simple approach was on average, rounded up, 3 meters. However, the distance varied a lot depending on the location of our measurements (indoors of a dorm, tunnels, classroom). The demo of our Marauders Map basic execution and results can be found here: <http://mit.edu/xsoriano/www/demo1.mp4>.

6. Prevention

6.1. MAC Randomization

Apple has implemented an interesting feature in their iOS 8 software: whenever the iPhone tries to connect to a Wi-Fi router (i.e. when it is sending probe messages), it randomizes the MAC address it sends [16][17]. However, for this feature to work the iPhone needs to be in sleep-mode and it must happen before a Wi-Fi connection is established. While there are many other conditions that need to hold for MAC randomization to happen (as explained in [16]), even if the function were to work as expected, it would not prevent the attack we presented.

An iPhone does not randomize the MAC address after it is connected to a Wi-Fi router: it actually uses its real MAC address. As a victim is moving, it will only send the randomized MAC while trying to connect and we would not be able to detect the device in this initial situation. However, once the

device connects to the Wi-Fi router, it will start using its real MAC address, and our method will be able to track the device. This means that while on locations such as the Infinite Corridor, we might not have many packets from a target MAC (though we will presumably have some, which will enable us to make guesses about the actual trajectory). However, the sniffers near classrooms, where the device is going to stay for a more prolonged period of time, will be able to receive almost all packets from the target MAC.

One should still note that this protection mechanism is nevertheless very effective against two subtly different attacks, which is apparently implemented by some department stores to infer shopping habits). In the first attack, the store plants sniffers in many locations throughout the store, and receive the packets each phone is sending. Note that most, if not all, of these packets will be probe packets, so by randomizing the MAC address the store wont be able to infer whose device sends the packets. The second attack prevents a device from broadcasting its previously known SSIDs along with its MAC address, since that could serve as a potential fingerprint for attackers to use. Hence, since, as explained in **citation**, the iOS 8 MAC randomization only works under very specific and stringent conditions, the feature is less effective in our setup.

6.2. MAC Encryption

In terms of security, the best solution is to always encrypt the MAC address. Each communicating party will have a public and a secret key (for example, we could use identity based encryption). When a device sends a message to a router, it will encrypt its message (and, most importantly in our case, its MAC address) using the public key of the router. This obviously implies that the device knows the public key of the router, so it must have somehow learned it. This can happen when the device learns the MAC address of the device, which happens through the address resolution protocol ([?]). The device will also send its public key, so the router can use it when sending messages to the device.

However, the device cant send the public key in the clear, as then the adversary can use the public key to infer information about the device, although as it is only sent once, it is still better than what is currently implemented. The device will encrypt its public address using the public key of the router. The router, which is the only one holding its secret key, will be able to decrypt the message from the device, and knows where to send its response. When the router sends a message to the device, it will use the public key

of the device to encrypt the MAC address of the device. The device will be able to decrypt the MAC address and know the message was meant for it. There are, however, some costs of implementing the solution presented in the previous paragraph. Implementing it would mean changing the link-layer level of networking, which already is a daunting task. Furthermore, there is the cost of encrypting and decrypting; also, whereas before a party could check the messages meant for it by ensuring the destination MAC was its own, with this protocol the device has to decrypt the MAC address first and then check whether the two MAC addresses match.

7. Conclusion

Attacks abusing 802.11 protocols are widely known and as we have shown they pose serious threats to privacy of users. There are existing prevention mechanism, however they are not perfect and do not satisfy all security requirements for a campus environment. We have described a potential solution, which has the drawback of needing to change the software on devices. With this in mind, we hope that our work will help as more researchers and students look into solving this problem.

8. References

- [1] Marauder's Map, http://harrypotter.wikia.com/wiki/Marauder's_Map
- [2] P. Bahl , V. N. Padmanabhan, *RADAR: an in-building RF-based user location and tracking system*, INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Volume:2)
- [3] A.B.M. Musa, Jakob Eriksson, *Tracking Unmodified Smartphones Using Wi-Fi Monitors*, SenSys12, November 69, 2012, Toronto, ON, Canada
- [4] Mathieu Cunche, *I know your MAC Address: Targeted tracking of individual using Wi-Fi*, International Symposium on Research in Grey-Hat Hacking - GreHack, Nov 2013, Grenoble, France. 2013. jhal-00858324j
- [5] http://elinux.org/RPi_USB_Wi-Fi_Adapters

- [6] https://en.wikipedia.org/wiki/Comparison_of_open-source_wireless_drivers
- [7] <https://www.open-mesh.com/skin/frontend/default/open-mesh/images/om2p.pdf>
- [8] https://en.wikipedia.org/wiki/Radical_axis
- [9] <http://mit.edu/xsoriano/www/demo1.mp4>
- [10] Wi-Fi name patent <http://tsdr.uspto.gov/documentviewer?caseId=sn75799629&docId=IPC20070420145537#docIndex=19&page=1>
- [11] History of Wi-Fi <https://getvoip.com/history-of-wifi/>
- [12] A brief history of Wi-Fi <http://www.economist.com/node/2724397>
- [13] Wi-Fi Growth Map <https://www.ipass.com/wifi-growth-map/>
- [14] So how much does a MAC address tell about you? <http://helvick.blogspot.com/2010/06/so-how-much-does-mac-address-tell-you.html>
- [15] <http://www.howtogeek.com/169540/what-exactly-is-a-mac-address-used-for/>
- [16] <http://blog.mojonetworks.com/ios8-mac-randomization-analyzed/>
- [17] <http://mpact.zebra.com/documents/iOS8-White-Paper.pdf>