MAC-based activity tracking using passive sniffing

Jetse Brouwer
Delft university of technology
Embedded Systems
Mekelweg 4, 2628 CD Delft
Email: j.brouwer-3@student.tudelft.nl

Niels Hokke
Delft university of technology
Embedded Systems
Mekelweg 4, 2628 CD Delft
Email: n.h.hokke@student.tudelft.nl

***Abstract*—Over the course of 5 days more than 1.2 million probe requests from over 91 thousand unique mac address were observed using low cost of the shelf hardware. Using this data, opening times and customer activity in a grocery store could be tracked. Using MAC address filtering the behaviour of individual customers and neighbouring residents is traceable.**

1. Introduction

Smart phones market penetration is ever increasing, with 68.8% the Netherlands is the number eight based on smart phone ownership [1]. While this increased connectivity can be of great convenience to the user, it might also supply other parties with a vast amounts of data. The goal of this paper is to explore the possibilities of tracking crowd activity as well as identifying and studying individuals by sniffing Wi-Fi packets transmitted by the smart phones. By placing a Wi-Fi sniffer at a crowded public place this papers aims to 1) monitor crowd activity over time 2) explore the feasibility of identifying and track behavior of individuals and 3) gather information about MAC randomization as a privacy enhancing feature.

2. Method

2.1. Probe Requests

To discover nearby access point (APs) the client has two options: It can either wait for an access point to announce it self by listening for beacons, or it can actively scan for nearby APs with the use of probe requests. Probe requests are a sub-type of 802.11 Management frames with the goal to discover near by access points. These probes can be either directed at a specific AP or broadcasted to any AP in the vicinity. On reception of such a request the AP sends a probe response with the parameters of the wireless network station such as supported data rates and channel usage. As the client only responds to beacons of networks it knows and wants to associate with, and we are also interested in customers who are not associated with nearby networks we will focus on capturing probe requests. According to the IEEE specification the header must always contain the following fields:

- frame control: frame type identifier
- Address 1: Destination MAC address
- Address 2: Source MAC address
- Address 3: BSSID (only for directed probes)
- Sequence control: Number to identify the probe

2.2. Hardware setup

As the authors had access to a postbox next to the entrance of a grocery store this location was chosen as the monitoring place. Due to the lack of a power outlet near the postbox an alternative power source such as a battery was needed. The chosen hardware-platform was the Raspberry Pi 3 due to its relatively low power consumption, affordability yet providing sufficient computational power.

During 'normal' use of a wireless network interface, the interface operates in 'managed mode' meaning only packets directly addressed to that specific network interface will be handled. To enable capturing of all packages within the network interfaces' reach it has to be put in 'monitor mode', which in our case required patched firmware as the Raspberry Pi 3 interface native did not support monitoring mode.

The Raspberry Pi 3 was equipped with Kali Linux, Debian-based Linux distribution designed for digital forensics and penetration testing. This as it includes patched firmware to enable monitor mode on the Raspberry Pi 3 among other commonly used software for sniffing data.

2.3. Software setup

The Raspberry Pi lacks a battery for the real-time clock, because of this the actual time will be lost at reboot. To synchronize the time an NTP client is started at the startup of the sniffer.

For capturing the packets Tshark (the command line interface implementation of Wireshark) was used. As we are only interested in probe requests we use a capture filter to reduce the amount of data to be stored. This filtering is done with the use of Tshark own capture filter syntax 'subtype probereq'. Tshark is setup to create a new pcap file every 5 minutes to lose at most 5 minutes of data corruption due to unexpected power loss. Besides exporting the PCAP

files, Tshark is configured to also append the arrival time, MAC address and signal strength to a CSV for ease of use.

As the interface can only monitor one channel at a certain time a script is utilized to change the Wi-Fi channel every 0.25 ms. This script was downloaded from [3]

The script run at start up can be seen in 2.3.

Listing 1. mac_tracker.sh

```
#!/bin/bash
echo 'Synchronizing date and time'
ntpdate ntp0.nl.net

echo 'create dir and mount usb-stick'
umount /dev/sda1
mkdir output
mount /dev/sda1 output/

echo 'setting wlan0 to monitoring mode'
airmon-ng check kill
airmon-ng start wlan0

echo 'enable channel hopping'
./chanhop.sh -i wlan0mon &
sleep 1

echo 'start sniffing'
tshark -n -t ad -b duration:300 -T \
fields -e frame.time_epoch -e wlan.sa \
-e wlan_radio.signal_dbm -w data \
'subtype probereq' >> output/output.csv
```

After synchronizing time, the script will try to unmount the USB-drive, create a directory to mount it and then mount the USB-drive.

Using 'Airmon-ng check kill' any background applications that might set the wireless network interface back to 'managed mode' will be killed. Next up wlan0 is set to monitor mode using 'airmon-ng start wlan0'

Then channel hopping is enabled for the created monitoring device. At last Tshark is started with the following arguments:

-n = do not resolve mac address to vendor
-t ad = use unix epoch timestamp output
-b duration:300 create a new pcap file every 5 minutes
-T fields = display the following fields
-e frame.time_epoch = arrival time
-e wlan.sa = sender mac address
-e wlan_radio.signal_strength = signal strength in dBm
-w data = pcap file prefix is 'data'
'subtype probereg' = only capture probe requests
>> output/output.csv = redirect the console output to a csv file for ease of use

3. Results

Over the range of six days the sniffer was placed in the postbox next to the entrance of the grocery store. Every morning at around 7:00 the device was taken in to replace the battery. As this process took on average 5 minutes but was an hour before opening of the store the influence of this reboot is ignored in the analysis. During those days the sniffer crashed once on Thursday, and was only discovered next morning while replacing the battery. In the remaining 5 days the sniffer collected 1.18 million probe requests from 91662 unique mac addresses

3.1. Pre-processing the data:

A single phone can emit up to 2000 probe requests per hour when their screen is active [2] and a person may be in the vicinity of the sniffer for a prolonged period of time, so a single visit might end up as several hundred entries. This together with individuals walking in and out of range of the sniffer requires pre-processing of the information before useful data can be extracted.

To do this a python script was written that removes every entry if that specific MAC address has been seen within the last x minutes. When set to 60 minutes this requires a MAC address to have been away for at least an hour before being admitted as an entry again. This also removes dormant smart phones from neighbouring houses from the data set. Another setting was to remove MAC addresses that have only been spotted once when the main interest was recurrences of addresses.

in appendix A Figure 5 a scatter plot is made with MAC addresses converted to y-coordinates and the time as x-coordinates. The settings used for the pre-processing script were that device probe requests would only be outputted if they were not seen for at least 30 minutes and occurred more than once.

3.2. Customer activity over time

When plotting the unique number of MAC-addresses per time slot it is possible to estimate the total amount of customers entering the store in that time slot. Based on this assumption it is expected to see a raise in the number of unique mac addresses near the opening time (which is 8:00) and a drop after closing time (22:00). A high peak of activity just before dinner is expected as this is the time most people buy groceries. During weekdays it is expected that there will be a peak around lunch, and a major peak before dinner as this is the time most people buy groceries. To get a better impression of the weekdays the measured activity was averaged out, the result is plotted in Figure 1 with a bin size of 30 minutes.

Figure 1 shows a clear increase in activity around 8:30, this is most likely due to customers buying breakfast or lunch before going to office/university. Another peak is observed during lunch time, between 12:30 and 13:30. The highest peak is from 17:30 to 18:30, which corresponds with customers returning from work and doing grocery shopping for dinner. These results confirm our before stated expectations.

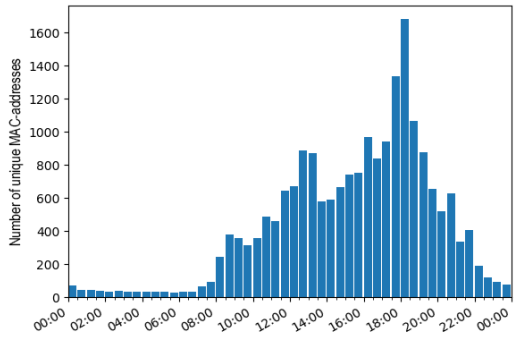

Figure 1. Unique Mac-addresses seen over time on an average weekday (bin size 30 minutes)

During the weekend it is expected that the shopping audiences will be more spread out over the day. This as less people have to work during the weekend and more people do leisure shopping. Because the sniffer is stationed at the entrance, also people passing by (but not entering) are detected.

looking at Figure 2 it can be seen that the customer activity observed around 8:30 is less than half compared to a weekday. However from 12:00 until 19:00 there are at least 1000 unique MAC addresses at any given time. Another interesting observation is that for both work and weekdays there is a small peak right before closing time of people who last minute need to buy something before closing time.

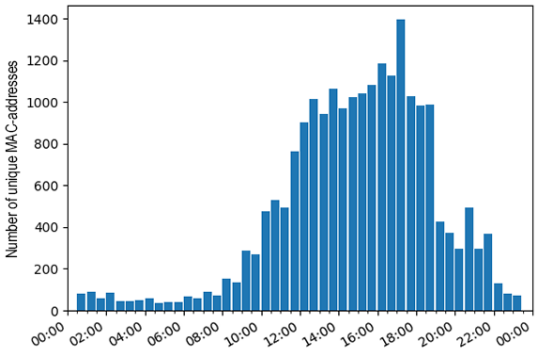

Figure 2. Unique Mac-addresses seen over time on an average weekend day (bin-size 30 min)

3.2.1. Mac randomization as a privacy enhancing feature. As a privacy-feature, smart phones running the latest versio of Android or Ios have the possibility to randomize their mac address when sending probe requests. However this option is not always turned on by default. By observing large amounts of probe requests over the span over several days reoccurring mac addresses can be identified and separated from MAC addresses that are only seen once. From the 91662 at least 5200 MAC addresses were seen with intervals of at least 24 hours in between sightings. This indicates that at least 5200 smart phones do not utilizes MAC spoofing when probing for APs. It is hard to do any estimates on the proportion of spoofing versus non-spoofing since it is very hard to tell how many of the 91622 MAC addresses belong to the same smart phone. Further because the data was only monitored for a week it is possible that from the 91662 addresses a subset does not utilize MAC spoofing but has not visited the store multiple times during that week. Attempts were made to inspect the probe request header to try and link multiple MAC addresses to a single origin. But since the header only contains meta data on supported rates and radio information this was very hard to do. Mainly because these values are the same per model of a phone, and are the same for devices using the same System-on-a-chip IC. Other common options for OS-fingerprinting rely on certain behavior embedded in for example TCP data, but since we are not associated with a Wi-Fi AP we were not able to extract this data from the packets.

The reoccurring MAC addresses can be correlated with costumer loyalty programs like the 'bonus-kaart'. This would give shops the possibility to track costumers based on mac address. With multiple access-points in the store, it would be possible to locate and find the interests of the costumer. Even with randomized mac addresses it would be possible to measure the average time people spend in front of certain shelf's. As this tracking technique is passive it is undetectable by the client. Only a limited amount of phones randomize there MAC address and people are unaware of the setting being enabled or not. The combination of these points makes it almost impossible to find out if one is being tracked or not. For now, turning Wi-Fi off is the only way to prevent this kind of tracking.

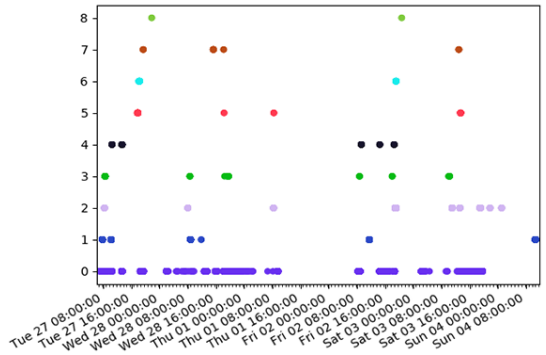

Figure 3. Selected Mac-addresses seen over 5 days (missing data from Thursday)

To test the possibilities of MAC address tracking, the authors own mac-addresses and 6 other interesting mac addresses were filtered out (see figure 3 above). The results correspond with a personal log-book tracking when one of the authors was near the sniffer. The purple color MAC address (at line 0 in the graph) corresponds with the MAC address of the phone of J. Brouwer, who lives within the range of the sniffer. Figure 3 clearly shows when J. Brouwer was home or not. From the original data it was even possible to determine when he was sleeping, as his phone entered a

sleeping mode resulting in less probes per minute when not being used over a long period of time.

The black colored mac address (line 4 in the graph) corresponds with the mac address of the phone of N. Hokke. These data-points show when N. Hokke went near the sniffer, either to shop or pick to up J. Brouwer to go study. Furthermore the person connected to MAC address 5 seems to visit the store every day except Friday, and either in the morning around 8:30 or in the evening around 18:00 on weekdays. On Saturday this person visited the store around 13:30.

3.3. Market share by Vendor

Although not related to activity tracking, the authors thought it would be interesting to also plot the market share per vendor based on the recorded data, as this data is embedded in the MAC address anyway. The market share of the 6 most seen vendors are displayed in Figure 4

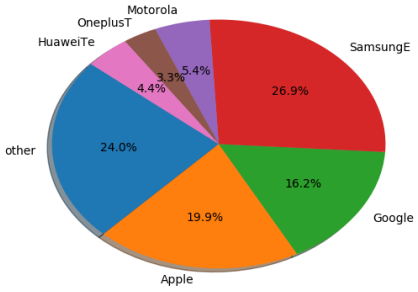

Figure 4. Pi chart of market share by vendor

4. Conclusion and further work

With the use of probe requests it was possible to create an activity graph of a grocery store that matches our expectations and can be used to distinguish working days from weekend days. Doing basic individual analysis it was possible to accurately determine when one of the authors was home, got picked up by the other author and even determine the author's sleeping pattern based on low activity of the phone. Also individual shopping behavior of random customers could be tracked. Future research goals might include spoofing an access point known to a lot of people (e.g. university Wi-Fi or the free Wi-Fi supplied by public transport) to bypass MAC randomization. As a phone will use its real MAC address when trying to associate with known networks. It might also be worthwhile to not only scan for Wi-Fi traffic, but also scan and try to correlate it with Bluetooth traffic.

References

[1] NewsZoo, *Global Mobile Market Report*, 2017

[2] Julien Freudiger, *How Talkative is your Mobile Device? An Experimental Study of Wi-Fi Probe Requests*, WiSec15 June 22-26 2015, New York City, NY, USA

[3] codegist user: hnw *http:codegists.comsnippetshellchanhopsh_hnw_shell*

Appendix A

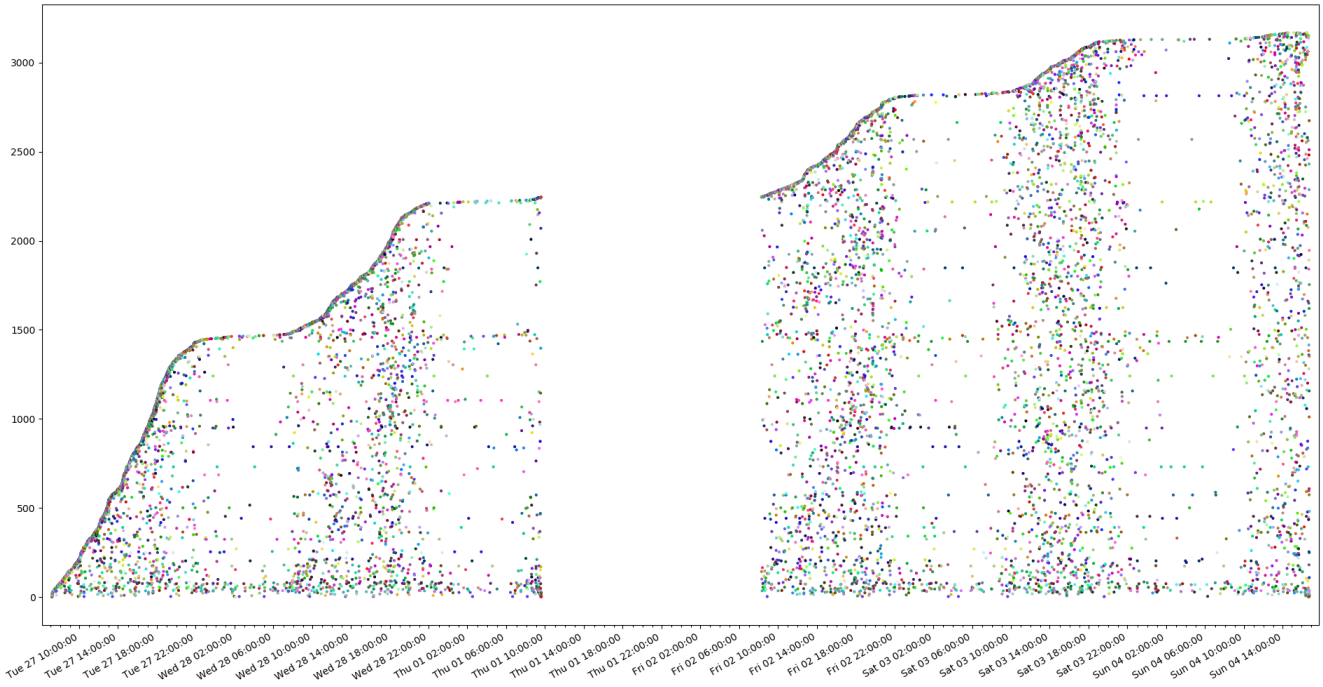

Figure 5. When plotting the last three bytes were used as RGB values for the dot to give the same MAC addresses the same color. the plot was made from left to right, meaning that if the a new mac appeared it would generate a new Y value, if it already existed it would be plotted on a lower. As a result every dot that is not the highest for that given time is a MAC that has been seen before