University of Amsterdam

RP 1

WIFI espionage using a UAV

Yadvir Singh

supervised by
Max Hovens & Chris Mavrakis

February 7, 2016

Abstract

Companies buildings can leak different types of signals including wireless network traffic. Although encryption prevents eavesdropping data grams, header information is still readable. This research focuses on what types company information can be deduced from such captured network traffic. This information includes: office occupation, vendor distribution and coarse grained movement tracking. To collect such traffic, UAVs can be used. Although detection systems are available, there deployment does not reach the level of conventional security measures (cameras and fences). This allows an attacker to sniff traffic within company premises.

Acknowledgements

I would like to thank Chris Mavrakis and Max Hovens who helped me in reviewing the report and providing guidance throughout the research.

Contents

1 Introduction

Wireless networks are widely deployed nowadays and can be found at many places including company offices. With the use of wireless networks, leakage of signals can become a risk. Leakage can include WIFI traffic that is not contained within the office building and can be sniffed from the outside. Although traffic might be encrypted, some information can be extracted from packets, including: MAC (Medium Access Control) address of both the recipient and transmitter[24] and the packet size. This traffic can potentially give some insight into the operation of a company and its employees. Additionally, Unmanned aerial vehicle (UAV) might be used advantageously to capture such traffic.

This research is divided into two main parts. Firstly, the structure of 802.11 packets is analysed followed by an on deducible information from captured wireless network traffic. The second part is concerned with the use of UAVs to capture wireless network traffic. This part will examine different types of UAVs, their capabilities and the current state of detection systems. The central research question that will be answered in this research is:

> *Which sensitive company info can be deduced from encrypted wireless network traffic and how effective can the use of a UAV be in capturing traffic that is not attainable using conventional wifi sniffing methods?*

Sensitive information in this context includes: information about the occupation (parts) of the office building, profile of devices (vendor/-type) and rough movement tracking of devices.

The rest of this paper is structured as follows: section 2 will discuss some of the previous work done on analysis of wireless network traffic. Section 3 will talk about the structure of the 802.11 frame and what fields are of interest. This section will also talk about the autonomy of MAC addresses. Section 4 will perform a theoretical analysis on what information can be deduced from captured traffic. Following section 4 is section 5 which will discuss a practical approach to capture and extract header data from wireless network traffic. Section 6 will discuss UAVs and their advantage and disadvantages.
Practical experiments with UAVs are not conducted because of budgetary and time constrains.

2 Related Work

The work done by [4] studies the usage of Google's public WIFI network in Mountain View, California. The study conducted an analysis of the client devices that are connected to determine the vendor and the type of device (phone/laptop/miscellaneous). Categorization is based on first three octets of a MAC address that are vendor specific. By far, most of the devices were manufactured by Apple. The paper also researched the amount of active users throughout the day. The result shows that there are distinct points in the data usage. Peaks are seen during the morning rush (09:00), during lunch time (12:30) and during evening rush (18:00). However, the graph depicting usage in the weekends is much smoother. The types of devices that are active over this 24 hour period also show differences. Modems (Linked to Laptops/Desktops) usage is fairly consistent throughout the day. Smart phone usage however shows peaks at particularly the morning and evening times.

The work of [5] studied the network usage in a corporate environment. The data collection was done by polling all access points every 5 minutes for a period of 28 days. The retrieved data from the APs includes information about the traffic going through the AP and also includes a list containing users connected to that AP. Furthermore, the data collected for each user includes: "detailed information on the amount of data (bytes and packets) transferred, the error rates, the latest signal strength, and the latest signal quality". An analysis of the number of users show the trend of a ”typical” working day. Around 08:00 the number of connected devices strongly rises until a peak is hit at 12:00. There are even devices active outside the working hours, indicating that some employees work longer or leave their devices turned on. The paper also studied the presence of users across 20 working days. Only 12 to 25% of the users were present for 18 or more working days. On the other hand, only 22 to 38 % of the user appear to be present for one or two days which is why they are categorized as visitors. The research also focused on the mobility of user and concluded that a significant amount of users (20 to 45%) move between 2 buildings.

3 Wireless traffic

This section describes the 802.11 frame and fields that are of interest. Additionally, the composition of a MAC address is discussed.

3.1 802.11 Frame

The 802.11 frame shows some similarities with the Ethernet frame although there are some specific fields that are used for wireless links.

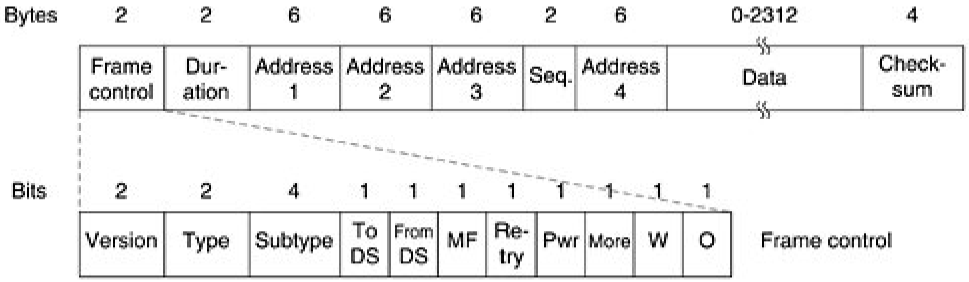

Figure 1: 802.11 frame layout [18].

At the start of the frame we find the Frame Control field which in turn has sub fields. The Frame control field is used to specify the function and type of the frame. Possible frame types include: Data, Management, Control and Reserved. The subtype fields further specifies the function of the frame. The `From DS` and `To DS` specify if a packet is going to a distributed system or coming from a distributed system. These two fields are only used in data type frames.

Following the Duration fields are MAC address fields 1,2 and 3. Opposite to Ethernet frames which only contain a source and a destination MAC address, a 802.11 frame can have up to four MAC addresses. Address 2 is used to store the MAC address of the transmitter. Address 1 is reserved to hold the MAC address of the receiver. Address 3 is usually used to save the final destination. This field can for example, be used by an AP when it does not route the packet at IP level directly. An example would be a STA (Station) that sends a packet to an access point (AP) with address 3 set to the final destination, for example a router[**?**]. In the case of Management frames, address 3 is used to hold the BSSID (Basic service set identifier, also known as the MAC address of the AP) of an AP[16]. The fourth address field is used for routing packets from AP to AP. After the sequence control and address 4 field, the body of the frame follows. The payload of the body typically consists of an IP datagram or an ARP packet. Encryption types such as WPA and WPA2 encrypt only this part of the frame, the header is not encrypted hence readable. The last field is used for error checking.

3.1.1 MAC address autonomy

A common MAC address format that is used, is the EUI-48 scheme. This format consist of 6 bytes with a total of 2^{48} possible addresses. The first three bytes are vendor specific and are also known as the Organizationally Unique Identifier (OUI). The distribution of OUIs is handled by the IEEE. The remaining three bytes can be addressed in any way by the vendor, with the constrain that a MAC address must be unique. The special address `FF:FF:FF:FF:FF:FF` is reserved for broadcasts[12].

4 Extracting Sensitive Data

The following subsections will discuss some of the sensitive information that can be deduced from the 802.11 frame header.

4.1 Unique MAC addresses

As discussed in section 3.1.1 a MAC addresses should uniquely identify a device. By capturing and extracting the MAC addresses in the header, an attacker can get insights into how many devices are currently active inside the sniffing range. When filtering out MAC addresses of the AP and only looking at the source address of packets that are going to the AP, the attacker is left with devices in the range of the AP. These devices can then be related to people, as employees have phones, laptops or other types of devices that personally belong to them. Using this mapping, the attacker can get an insight into the occupation of the office. Additionally, timestamps can be saved during the capture to allow tracking of device presence over a particular time period (e.g working day). The work of [5] analysed this presence for several days. From the results, it concludes that patterns can be seen across the day. Distinct points include when employees start working, lunching and go home. One of the remarks that the study made was that although most employees leave at the end of the day (around 18:00), some device activity is still measured, indicating that there might be employees that work after office hours. This mapping is however not trivial. Employees can for example have multiple devices that relate to the same person but a rough estimate of the occupation can be made.

4.2 Device categorization

MAC addresses have a fixed part that is related to the vendor of a device (see section 3.1.1). Just like IP space, the distribution of these addresses is controlled by an organization, in this case the IEEE. The assignments are made public[13]. and anybody can lookup the vendor that is associated with a particular MAC address. This can reveal if a company is preferring a particular vendor for its devices. Information about vendor use can then be further exploited by searching for vulnerabilities of devices from a particular vendor. Furthermore, vendors can be coupled to device types based on their product portfolio. This gives an insight into what type of devices (laptops, phones) are used in the company. This information can be exploited by using a vulnerability that affects a particular vendor.

4.3 Network analysis

One of the other possibilities is to extract information about network usage from the captured traffic. Besides collecting MAC addresses, the size of each packet can also be retrieved making it possible to monitor data usage per device. Devices that send or receive high volumes of data, might be critical in the operation of companies. Additionally, when packets are routed within the network using a direct link (all address fields in the 802.11 packet are used), an attacker could also monitor communication between devices.

4.4 SSID names

Access points broadcast their SSID names at fixed intervals (if the SSID is not hidden) to announce their presence. Sometimes, SSID names are chosen to reflect the location of the access point such as: Boardroom, Lobby, Conference room etc.... This "*location*" information can be used to monitor the device presence at a particular location. A step further is to categorize devices based on the access point they connect to. An example would be to monitor devices that enter the boardroom frequently at fixed times and dates. This can suggest that these devices belong to high ranked employees.

4.5 Passive movement tracking

Another possibility is passively tracking the movement of devices (and their owners). When employees move inside the building, they may carry their personal devices along with them. Whilst the device is moving, its wireless coverage is moving as well. Along its way and when stationary, the device will try to make a connection to one of the nearby APs. This movement can be tracked by sniffing traffic destined for different APs. An attacker would firstly create a list of MAC addresses that are connected to one AP by monitoring traffic in which the destination MAC address is equal to the target AP. The MAC addresses of APs can be obtained by looking at beacon frames and probe request which are transmitted from APs. Together with the list of MAC addresses connected to an AP, the attacker must also save the position at which he intercepts traffic destined for the target AP. This can for example be achieved using GPS position stamps. The next step is to repeat this process whilst targeting a different AP. This will result in a second list of MAC addresses that are associated with the second AP and the physical position at which traffic of the first and second capture took place. By comparing the two lists of both APs, MAC addresses can be obtained that appear in both lists. MAC addresses that comply with this behaviour appear to have switched from APs. This method can also be expanded by targeting multiple APs.

There are limitations on the effectiveness of this method. Firstly the relative position of a device to the target AP is not known by the attacker. The same holds true for the relative position of the AP to the attacker. The only location information that the attacker has are the GPS coordinates which are saved together at the point of capture. This results in the ability to track a device only at a coarse grained level.

4.6 WIFI shielding

To protect against the leakage of WIFI signals outside buildings, companies can employ WIFI shielding techniques which prevent such leakage. Motivation can include preventing outsiders using free WIFI facilities or blocking hackers from eavesdropping on public networks. Different materials can be used to contain WIFI. Special window films for example let light still pass [17] but also act as WIFI “walls”. Furthermore metal meshes can turn the building into a “Faraday Cage” and there are even special paints that block signals[14]. The use of such techniques make sniffing traffic difficult or even impossible. Companies that employee such shielding can then only be attacked by finding weak spots in the shielding, if any.

5 Experiment

This section will discuss experiments done at an office site. Sections 5.1 and 5.2 will discus how Data is captured and processed. Sections 5.4, 5.5 and 5.6 will discus each experiment individually and the corresponding result.

5.1 Data Collection

Network interface cards (NICs) can operate in different modes. In normal operation, traffic that is not addressed for a particular NIC is discarded by that NIC. Some NICs have the ability to operate in a so called "*monitor mode*" in which all traffic is passed through, even traffic not associated with that NIC. On Linux based operating systems, the `aircmon-ng` utility can be used to enable monitor mode on supported NICs[20]. The `airmon-ng` utility will create a new interface with the prefix `mon`. This interface can then be used to capture all packets on the channel the NIC is listening to.

5.2 Data Processing

To parse and save data, Python is used in combination with the Scapy library[1]. This library allows the creation, modification, injection and sniffing of network packets. The complete data collection and processing is depicted in figure 2.

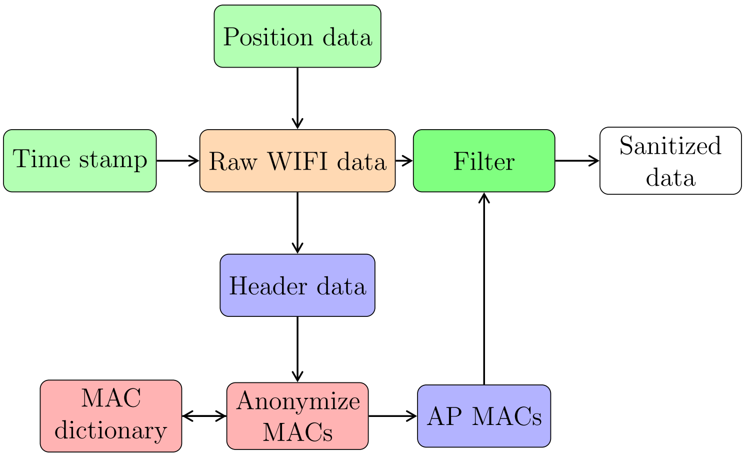

Figure 2: Flowchart that resembles the processing of captured wireless network traffic.

The traffic capture starts by sniffing a particular channel on which a target AP resides. Tools such as `airodump-ng` can reveal access points within range and also state at which channel they are operative. Using the `iwconfig` utility, the channel of the NIC (used to sniff traffic) can be set. Once the configuration is set, the actual collection of traffic is executed. Raw WIFI data is complemented with a time stamp that is saved with each captured packet. Besides time stamps, positional data can also be saved (GPS coordinates) to keep track at which location packets are received. The position data is however not implemented in this set-up because no experiments are done with a UAV.

Whilst collecting WIFI data, the header data is extracted in real time. For this research, the interest lies only on header information and thus all other information is discarded in this step. Information that is extracted from packets includes: MAC addresses in address field 1-4,packet type, packet subtype, packet length and optionally a SSID (if the packet is of type beacon or probe response). From the collected MAC addresses, vendors are extracted using the Manuf library[8].

One of the condition to capture wireless network traffic in the office was the proper anonymization of MAC addresses that appear in the header data. The option of hashing MAC address did not meet the level of anonymity wanted. This was solved by implementing MAC dictionary that contains MAC addresses linked to a unique numeric identifier. After header information is extracted from the packet, the MAC addresses of the address fields are compared to a preloaded MAC dictionary. If the MAC addresses is found in the dictionary, its numeric identifier will be retrieved and used for further processing. If the MAC address is not found in the dictionary, the last numeric identifier in the dictionary is increased by one and awarded to the new MAC address. The new address in combination with its new identifier is then saved in the dictionary for future runs. This dictionary is preloaded every time a new measurement is started. A second condition from the company implied that measurements can only take place on a laptop of the company. The dictionary file that contains the mapping between MAC addresses and numeric identifiers is saved on this laptop and does not leave from this device.

Next, the MAC addresses of access points are extracted. Extraction is based on specific packet types that are only sent by the AP (Beacon frames and probe responses). To target a specific AP, the MAC address of that AP is fed into a filter. This filter examines the raw WIFI data and will filter out any traffic that is not coming from or going to the target AP.

After all processing, the sanitized data contains all traffic to and from the target AP with all MAC addresses anonymized. The sanitized file will contain lines for each captured packet. An example line would be:

```
[1][61][61][3][None][Sitecom Europe BV][Sitecom Europe
BV][noVendor][0][8][2016-02-03 18:02:18.445484][321][
Sitecom_xxxx
```

The following table describes each field:

Field	Description
0	MAC address 1
1	MAC address 2
2	MAC address 3
3	MAC address 4
4	Vendor address 1
5	Vendor address 2
6	Vendor address 3
7	Vendor address 4
8	Packet type
9	Packet subtype
10	Packet subtype
11	Time stamp
12	Packet size
13	SSID (optionally)

5.3 Experiment Set-up

All experiments were conducted on a laptop provided by the company which was running a Kali-Linux instance. Using the `airodump-ng` utility, target access points were selected and the channel of the NIC was manually adjusted (using the `iwconfig` utility) to sniff traffic on the targets APs frequency. To capture all traffic, the `airmon-ng` utility is used to put the NIC in monitor mode. See appendix A for python script that captures and saves traffic.

5.4 Experiment #1

The first experiment examines the effects of different capture time spans on the number of detectable devices. All measurements are made targeting one and the same access point. The duration of the captures is divided into five categories: **1**,**5**,**10**,**20** and **60** seconds. At each test, the number of unique devices that communicate with the target AP are measured.

5.4.1 Results

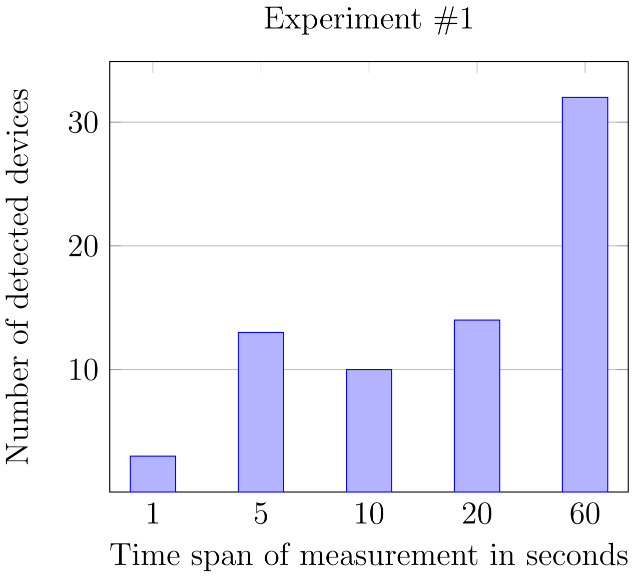

Figure 3: Plot showing the number of devices detected at different time span measurements.

Figure 3 shows the results of experiment 1. At time span of 1 second, only 3 unique devices were detected. Compared to the other results, the few number of detected devices in this time span prevent a decent conclusion to be made with the captured traffic. The results of the 5, 20 second measurements lie closely together with only 1 device detected additionally in the 20s measurement. The odd one is the 10s measurement in which 10 devices are uniquely identified, opposite to the 13 detected in 5s measurement. This can be explained by devices leaving the sniffing area or going to standby mode whilst not sending traffic and/or probe request. At the 60 second measurement, 33 devices are detected.

5.5 Experiment #2

The second experiment focuses on the vendor distribution that can be extracted from MAC addresses. For this experiment, data is collected for 60 seconds targeting one access point. Using the Manuf [8] library, the vendors are extracted. The vendor data that is used by this library is extracted from the Wireshark database. These devices are then categorized based on the product portfolio of the vendor.

5.5.1 Results

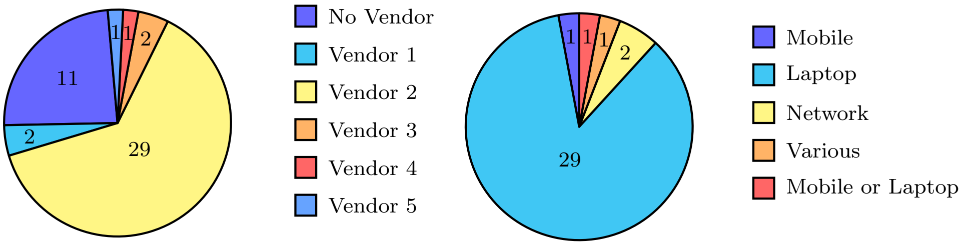

Figure 4: Distribution of devices by vendor(left) and type(right).

Figure 4 shows the top 6 vendor distribution of the 46 unique devices found. The left pie chart shows devices categorized based on their vendor. Due to company policies, the vendors are anonymized. Most noticeable is the presences of vendor 2. It accounts for more than 50 % of all devices. The second largest part of the pie chart is consumed by the devices to which no vendors could be linked. This can however be a shortcoming of the Wireshark database. The right pie charts depicts the distribution of devices based on the product portfolio of their vendor. Devices for which no vendor is found are omitted in this categorization. From this chart, it becomes clear that laptops are the most frequent device in the captured traffic.

5.6 Experiment #3

The third experiment focuses on analysing which devices are connected to which access points and if any devices have shared an access point.

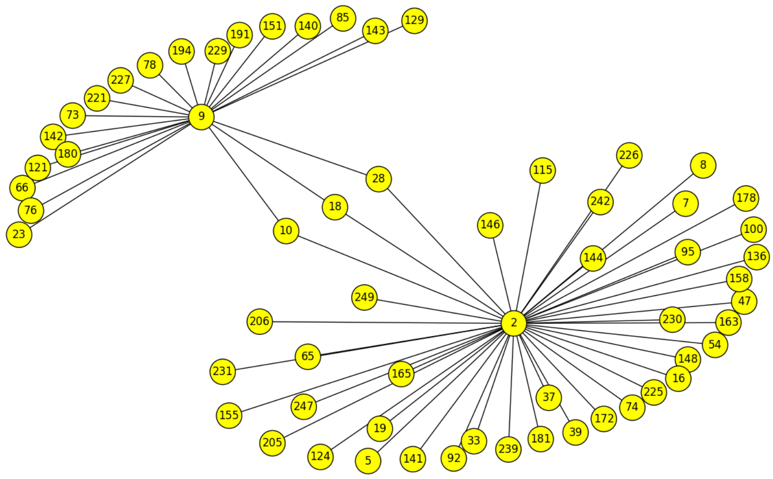

Figure 5: Resulting mapping of experiment #3. The two access points are marked as node 2 and 9.

5.6.1 Results

Figure 5 shows the results of the experiment #3. The two targeted access points are depicted by nodes 2 and 9. All edges to these two nodes are devices that have exchanged traffic with that particular access point. In this figure, three distinct groups can be extracted. One group consists of the devices that solely have communicated with access point 9 and another group that solely talked to access point 2. What is of most interest are the group of devices that appear to have been talking to both access points. These nodes are marked 10,18,28 and appear to have been in the neighbourhood of both access points.

This test however only covers two access points due to the limited time and sniffing permissions. Monitoring more access point should give a more complete overview of device movement between different access points.

6 Usage of UAVs

The definition of a UAV can be formulated as a *"space traversing vehicle that flies a without human crew on board and that an be remotely controlled or can fly autonomously"*[6]. Both the development and use of UAVs has seen a enormous growth in the past three decades. The growth can be related to both commercial and private use, as well as military use. A market survey conducted by the Teal Group forecast that the total turnover in the UAV market will be $ 93 billion in the next ten years[11]. This section will discus different types of UAVs, the trade-off between flight time and payload, and developments on detection systems and rules and regulations.

6.1 UAV types & capabilities

UAVs can be categorized into different classes that relate to the design of the UAV [19]. The four categories include:

Fixed-wing UAV. This type of design refers to a vehicle that has fixed wings, similar to an airplane. The main advantage is that fixed-wing vehicles generally have a long endurance and can fly at high cruising speeds. The main disadvantage is the need of a runway for both take-off and landing.

Rotary-wing UAV These types of UAVs are also called rotor crafts or VTOL (vertical take-off and landing) UAVs. A major advantage of these rotor crafts are their ability to hover and their high manoeuvrability. This makes them good candidates for application like inspection where precession maneuvering is needed. Most commercially available UAVs are based on this design. Rotary wing based vehicles can have 1 (helicopter) and 3 or more rotors. Their main disadvantage is their complexity compared to fixed-wing UAVs.

Blimps This type of UAVs are comparable to airships or balloons. Lift is achieved using a gas that is lighter than air. The main advantage of this type of UAV is its long endurance as little power is need to keep it airborne. Main disadvantages include low fly speeds and large size.

Flapping-wing UAVs Flapping-wing based UAVs are inspired by birds and flying insects. Wings are often small and flexible. One of the advantage is the ability to mask the UAV as a bird such that the

UAV is harder to detect. These type of UAVs are however not that common compared to the fixed and rotary-wing based UAVs.

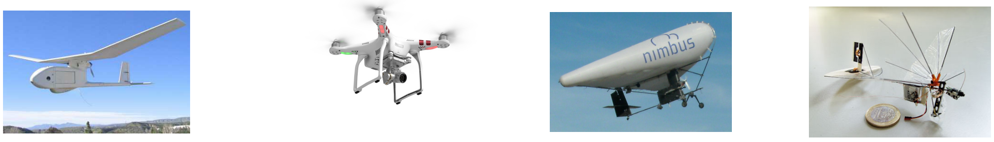

Figure 6: Different types of UAVs. From left to right: Fixed wing[6], Rotary-wing[9], Blimp[2] and Flapping-wing[23].

Out of the four types of UAVs described, the Rotary-wing type is the most favourable solution for collecting wireless network traffic inside or around company premises. One of the main advantages of this type of UAV is its ability to hover (opposite to Fixed-wing types). Steadiness is required to keep in receiving range of the targeted network traffic. Another benefit is the ability to take-off and land vertically, eliminating the need of a runway which might be hard to find in urban environments. Furthermore, rotary-wing UAVs are now widely commercially available compared to blimps and flapping-wing types.

6.2 Flight times & payload

One of the key aspects of UAVs is their endurance level and payload capacity. The development of drones is often geared towards long endurance. This results in a high fuel fraction (weight of the fuel or propellant dived by gross take-off weight) in combination with a low payload fraction. This typically results in UAVs been rated to carry 10 to 20 % of their gross weight[22].

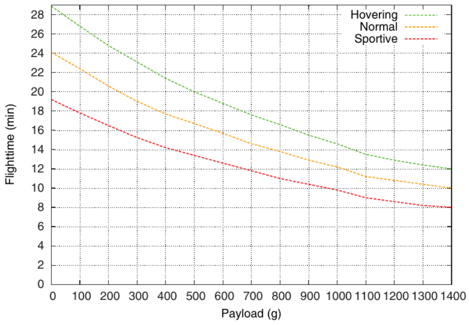

Figure 7: Simulated flight time against payload of the QuadroXL UAV[3].

Figure 7 shows the simulated flight time against payload of the QuadroXL UAV (retail price: €999). The graph shows that with no payload and in normal operation, the UAV has a flight time of 24 minutes, which is comparable with other commercial drones[10]. A small sniffing set-up can consist of: a Raspberry PI (45g), Alfa AWUS036H WIFI adapter (38.5g) and a battery pack (<100g) = 183.5g. This results in a flight time of approximately 21 minutes. With an average speed of 7 m/s and 10% hover time this results in a range of approximately 4 Km (round trip). The ability of rotary-wing UAVs to take-off and land vertically can however be used advantageously. The UAV can land at rooftops or other locations which are physically restricted for an attacker. Whilst landed, the UAV does not need to power its electric propulsion motors and hence use less power resulting in more operational time.

6.3 Detection

One of the advantages that UAVs carry is their ability to pass through conventional (ground based) security measures[21]. and the lack of widely deployed detection equipment. However some recent incidents got the attention of national security agencies and law enforcement resulting in the urge to develop these detection systems. These systems for UAVs fall into two categories: active and passive detection. Passive detection includes: visual, acoustic, thermal/infrared, and UAV communications/control frequencies monitoring [21]. Active detection on the other hand relies on using radar. Although conventional radar is able to detect large object in the sky (such as air planes), small sized UAVs generally do not have enough surface material to reflect radar signals. Thales Netherlands is one of the companies that brought a special radar system to market that is designed to detect UAVs[7]. It uses conventional radar technology with an algorithm that distinguishes UAVs from other objects such as birds.

Following detection is the interception of UAVs. This field is focussed on altering and/or interfering with the control and behaviour of UAVs as swell as forcefully taking down UAVs[21]. One of the methods used to accomplish this is interruption. Interruption includes: operation interruption, jamming and spoofing. Jamming is concerned with interrupting the control signal from the operator to the UAV, but also interruption of the sensors on-board the UAV. GPS for example, is used by UAVs to autonomously fly a predefined course. Jamming such signals can alter the course or cause malfunctioning. Spoofing is used to send control commands to UAV whilst pretending they come from a legitimate source. An example is SkyJack[15], an open-source project that demonstrates the ability to takeover control of the Parrot AR drone by exploiting unencrypted communication between the operator and the UAV. Interception is also possible by using forceful methods such as ballistic missiles to incapacitate a UAV, although the risk of collateral damage must be accounted for.

Although specialistic equipment such as the radar system developed by Thales is available, it is not as widely deployed as conventional security measures such as cameras and fences. This still allows attackers to enter company premises with a UAV and sniff network traffic without the same risk as physically entering the premises.

6.4 Rules and regulations

The popularity of UAVs for recreational use and commercial use has lead to new legislation, both in The Netherlands and outside. Dutch legislation makes a difference in recreational and commercial usage. Important rules include: the operator must always have a direct visual line of sight with its UAV, UAV may not achieve altitudes in excess of 120 meters, UAVs are allowed to have a weight of 25 kg max (4 kg in the future) and UAVs are not allowed to operate at night.

7 Conclusion

Overall different types of information are deducible from corporate wireless encrypted network traffic. Firstly, the MAC address allows to uniquely identify devices. These devices can than be mapped to employees to get an overview of the occupation level at different locations. Combined with time stamps, activity across the day can be measured.

Secondly, a distribution of vendor usage can be made by looking at the OUI of collected MAC addresses. This can reveal if a company is preferably using devices from a particular vendor which might be attackable by exploiting specific vulnerability. Furthermore, vendors can be categorized on the device types(laptops, phones etc...) they manufacture, resulting in obtaining the device type distribution of a company.

Thirdly, MAC addresses and packet length can reveal information about the network usage per device. When packets are routed within the company's network (all four address fields are used), communication between devices can be monitored.

Also, the SSID names broadcast by access points can reveal the location of the AP. Naming schemes such as Boardroom_XX or Lobby_Company_Name can be used to identify devices that are within range of these APs. By recording devices connected to these specific APs, further classification of devices is possible. An example would be devices that are frequently connected to the Boardroom AP. If a same group of devices is connected to the AP at regular time and date intervals, then these devices can be related to specific group of (high ranked) employees.

Lastly, the movement of employees can be tracked at a coarse grained level. By monitoring devices connected to different (physically separated) APs, an attacker can get an overview of devices and their connections to different APs. This can give an insight into the movement of owner (employee) of that device. There are however limitations to the accuracy achieved as no information is available on the relative position of the device to the AP and the relative position of the AP to the attacker.

However, one of the limiting factors can be WIFI shielding that is incorporated in the company building. Special types of glass, film and even paints are used to prevent leakage of WIFI signals outside building walls. Attacks can then only be conducted on weakness in the shielding, if any.

Additionally, UAVs can be advantageously used to capture corporate wireless network traffic. Rotary-wing based models allow to hover in mid air and offer a good flight-time with modest sniffing gear fitted. Although the urge to develop a detection and interception systems has led to several practical systems, they are not as widely deployed as conventional security measures (such as cameras and fences). These conventional security measures are of little to no resistence for UAVs.

8 Future work

Due to budegtary and time constrains, no practical test with UAVs were done in this research. Further research can be done on implementing and testing the use of UAVs for the collecting wireless network traffic.

One of the domains that can be further investigated on the practical implementation on UAVs is the use of different antenna types. The most common antenna is the omnidirectional antenna. This type of antenna provides 360 degree coverage. This might not be useful when trying to receive traffic from a practical direction. Directional antennas (such as flat, grid and yagi types) can help as they have a more focused signal. These signals are often an oval shaped pattern in which the beam is only a few degrees wide. Further research can also be done to limit the reception range when on desires to capture traffic nearby.

Also, experiment #1 (subsection 5.4) researched the affect of time on the amount of device detected. Further research can be done to find the optimal parameters (time-span of measurement, capturing location, equipment effects).

Furthermore, the use of WIFI shielding can be inspected to find how well WIFI signals are contained by the different types of WIFI shielding methods. Research can also focus on possible weaknesses in the deployment of such shielding.

References

[1] Scapy. http://www.secdev.org/projects/scapy/.

[2] 4dp8i. Air vehicle in metaplane configuration. https://commons.wikimedia.org/wiki/File:MetaplaneInFlight1.PNG.

[3] Benjamin Adler, Junhao Xiao, and Jianwei Zhang. Autonomous exploration of urban environments using unmanned aerial vehicles. *Journal of Field Robotics*, 31(6):912–939, 2014.

[4] Mikhail Afanasyev, Tsuwei Chen, Geoffrey M Voelker, and Alex C Snoeren. Analysis of a mixed-use urban wifi network: when metropolitan becomes neapolitan. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 85–98. ACM, 2008.

[5] Magdalena Balazinska and Paul Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 303–316. ACM, 2003.

[6] Guowei Cai, Jorge Dias, and Lakmal Seneviratne. A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends. *Unmanned Systems*, 2(02):175–199, 2014.

[7] Andrew Chuter. Mini drones spark heightened interest in countering threat. http://www.defensenews.com/story/defense/air-space/strike/2015/06/20/small-drones-raise-interest--combating-threat/28977373/.

[8] Michael Huang (coolbho3k). Parser library for wireshark's oui database. https://github.com/coolbho3k/manuf.py.

[9] DJI. phantom-3-standard. http://www.dji.com/product/phantom-3-standard.

[10] My First Drone. Best drones for sale and why. http://myfirstdrone.com/tutorials/buying-guides/best-drones-for-sale/.

[11] Teal Group. Press release: Uav production will total $93 billion. http://www.tealgroup.com/index.php/teal-group-news-media/item/press-release-uav-production-will-total-93-billion.

[12] IEEE. Guidelines for 48-bit global identifier (eui-48). https://standards.ieee.org/develop/regauth/tut/eui48.pdf.

[13] IEEE. Registration authority. https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries.

[14] Tom Jowitt. New paint promises high-speed wi-fi shielding. http://www.infoworld.com/article/2674148/networking/new-paint-promises-high-speed-wi-fi-shielding.html.

[15] Samy Kamkar. Skyjack. http://samy.pl/skyjack/.

[16] Sumanth Kavuri. Understanding the address fields in 802.11 frames. http://80211notes.blogspot.nl/2013/09/understanding-address-fields-in-80211.html.

[17] Lessemf.com. Scotchtint(tm). http://www.lessemf.com/plastic.html.

[18] NCSU. http://www4.ncsu.edu/~aliu3/802.bmp.

[19] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer Science & Business Media, 2010.

[20] Chris Sanders. *Practical packet analysis: Using wireshark to solve real-world network problems*. No Starch Press, 2011.

[21] Ryan J Wallace and Jon M Loffi. Examining unmanned aerial system threats & defenses: A conceptual analysis. *International Journal of Aviation, Aeronautics, and Aerospace*, 2(4):1, 2015.

[22] Dyke Weatherington and U Deputy. Unmanned aircraft systems roadmap, 2005-2030. *Deputy, UAV Planning Task Force, OUSD (AT&L)*, 2005.

[23] Team WTC. Dossier delfly. http://www.tudelft.nl/actueel/dossiers/archief/delfly/.

[24] Zytrax. Tech stuff - 802.11 mac (media access control). http://www.zytrax.com/tech/wireless/802_mac.htm.

A Python script

See repository at `https://github.com/cheatas/RP-1` for all data processing scripts.

```
"""

Author: Yadvir Singh

Description:

Program that captures network traffic and
   extracts header data.
This data is then saved to a file in conjunction
    with a time stamp.

"""
import sys
from scapy.all import Dot11, Dot11Elt, sniff
import datetime
import manuf


#A counter used to track the number of captured
   packets.
counter = 0
macCount = "0"
venCount = "0"
macDict = {}



def getVendor(mac, parser):
        try:
                vendor = parser.get_comment(mac)
        except:
                vendor = "noVendor"
        return vendor

def readDict(fileHandle, dictionary):
        count = 0

        lines = fileHandle.readlines()
        if len(lines) == 0:
                print "MAC file empty\n"
                return "0"
        else:
                lastLine = lines[-1].split("][")
                count = lastLine[1].rstrip()
                print "Read last line from MAC
                   file\n"

        for line in lines:
                fields = line.split("][")
                dictionary[fields[0]] = fields
                   [1].rstrip()
        return count


#Function that parses the input parameters to
   acquire the output file.
def init():

        if(len(sys.argv) == 2):
                print "No argument supplied!
                   Suplly file name."
                sys.exit()

        inputFile = sys.argv[1]
        macDict = sys.argv[2]

        try:
                inputFileHandle = open(inputFile
                   , 'w')
                inputMacHandle = open(macDict, '
                   r+')
        except:
                print "invalid file !"
                sys.exit()

        return inputFileHandle, inputMacHandle

#Function to load a dictionary from a file.
def checkDict(mac, dictionary, count, fileHandle
   ):
        mac = str(mac)

        if mac in dictionary:
                return dictionary[mac], False
        else:
                count = str(int(count) + 1)
                dictionary[mac] = count
                fileHandle.write(str(mac) + "]["
                    + count + "\n")
                return count, True



#Main function that captures and parses network
   traffic.
def parse(x, filename, macDict, macFile, parser)
   :
        global counter
        global macCount


        packetLen = len(x)

        #Remove unwanted layers.
        if x.haslayer(Dot11Elt):
                x[Dot11Elt].remove_payload()
        else:
                x[Dot11].remove_payload()


        mac1 = x.addr1
        mac2 = x.addr2
        mac3 = x.addr3
        mac4 = x.addr4

        ven1 = getVendor(mac1, parser)
        ven2 = getVendor(mac2, parser)
        ven3 = getVendor(mac3, parser)
        ven4 = getVendor(mac4, parser)
```

```
mac1,new = checkDict(mac1, macDict,
   macCount, macFile)
if(new):
        macCount = mac1

mac2,new = checkDict(mac2, macDict,
   macCount, macFile)
if(new):
        macCount = mac2

mac3,new = checkDict(mac3, macDict,
   macCount, macFile)
if(new):
        macCount = mac3

mac4,new = checkDict(mac4, macDict,
   macCount, macFile)
if(new):
        macCount = mac4

#Write header data to the output file.
filename.write("
   [{0}][{1}][{2}][{3}][{4}][{5}][{6}]
[{7}][{8}][{9}][{10}][{11}]".format(mac1
   ,mac2,mac3,mac4,ven1,ven2,ven3,ven4,x
   .type,x.subtype,datetime.datetime.now
   (),packetLen))


#The SSID is given in either the Beacon
   frame or a Probe response.
if x.type == 0 and (x.subtype == 8 or x.
   subtype == 5):
        if(x.info == ""):
                filename.write("[ Hidden
                    ]")
        else:
                filename.write("[ " +
                   str(x.info))
        print x.info,

filename.write("\n")
```

```
        #We print the same data that is written
           to the file to give some
        #visual feedback during the program
           execution.
        print("{0},{1},{2}".format(mac1,mac2,
           mac3))


        counter = counter + 1
        print "--------------------" + str(
           counter)



parser = manuf.MacParser()
target, macFile = init()
macCount = readDict(macFile, macDict)

#store=0 is needed to prevent the storage of
   packets in memory
caputre = sniff(iface="mon0",prn = lambda x:
   parse(x, target, macDict, macFile, parser),
   store=0)
```