Predator-drone, one drone to rule them all

Florent Fayollas and Antoine Vacher,
Masters students at TLS-SEC

***Abstract*—Drones are now present almost everywhere. Reserved before to military, this technology is now accessible to everyone. Their feild of use is wide: surveillance, agriculture, media coverage, etc. However, some people are using them in malicious way, to spy, to bomb, etc. In this work, we developed a tool whose goal is to take control of multiples civil drones.**

***Index Terms*—Security, UAV, Drone, Hijack, Parrot AR.Drone 2.0, Syma X5C-1, WiFi, RF 2.4 GHz, Deauth attack**

I. Introduction

A. Overall introduction

Unmanned aerial vehicles (UAV), commonly known as drones, are aircrafts without human pilot aboard. An UAV is a component of an unmanned aircraft system (UAS), which include an UAV, a ground-based controller and a communication system between the two. However, many UAVs can take certain decisions autonomously during their flight.

These systems were originally developed by military and used in too dangerous missions for humans. In the past few years, their use became generalized to many sectors such as academic, commercial, and even recreational. As concrete examples, drones are now used for surveillance, agriculture and aerial photography.

Their field of use is continuously growing. As an example, Amazon is working on drones to be used for their deliveries: the future Amazon Prime Air service. Amazon managed to perform its first fully autonomous delivery on December 7, 2016 [1].

B. Motivation of this work

The use cases stated before are mostly advantageous for our society. Yet, malicious usages also exists. A first concrete example is the recreational drone use by Daesh to bomb on Syrian frontline [2]. Another one, more recent, is the flight over London Gatwick airport by a non-identified drone, which caused a blockage of the whole airport [3].

Thereby, it is necessary to protect from these threats. Military has developed solutions, such as control link jamming. The DroneGun, constructed by DroneShield [4], is a great example. Nevertheless, such solutions do not exist for civilians.

We could think on making military solutions accessible publicly, but these solutions often take down the drone, without considering the final state of the drone. This means that these solutions can cause a crash. This is not acceptable for civilians. As an example, we could think of a drone flighting over a crowd, that a crash would harm.

C. Our goals

The objective of this work is to develop a tool able to take control of multiple commercial drones, without implying their crash. This tool will be embedded on a predator drone that will cover a restricted flying zone.

In ou study, we focused on taking control of two commercial drones: the Parrot AR.Drone 2.0 and the Syma X5C-1. We also addressed the protection means that could be deployed to shield these drones from our attacks. Then, we discussed about the embedding question on a predator drone.

II. Hijacking a Parrot AR.Drone

The Parrot AR.Drone 2.0 was released on January 2012. It is a quadcopter drone developed by the French company Parrot SA. The user can control it thanks to a WiFi control link, with his smartphone and a dedicated app. In this WiFi control link, the drone is the access point and the user is a client.

In December 2013, Samy Kamkar published an attack called SkyJack [5] on this drone. This attack can be use to take drone control with the following steps:

1) Search for opened WiFi networks and search for their clients thanks to `airodump-ng`
2) Filter Parrot WiFi networks
3) Disconnect found client[1] with `aireplay-ng`
4) Connect to the opened WiFi network
5) Run a controlling program to take drone control

This attack needs two WiFi adapters. The first one is used to scan for WiFi networks and perform the de-authentication attack. The second one is used to connect to the hacked network.

We enhanced this attack, keeping in mind the embedding goal, using the Python Scapy library [6]. In addition, we used the PyRIC library [7] to manage WiFi adapter.

A. Opened WiFi discovering

To list available opened WiFi networks, we can perform an active scan, sending probe request packets and listening for probe response packets. However, doing this, we become easily detectable. Thus, we decided to perform a passive scan, only listening for beacon frames. These packets contains lots of data about the access point that sent it: channel used, SSID[2] and BSSID.[3]

We implemented this solution using Scapy's `sniff` function. This allowed us to remove the `airodump-ng` dependency. In addition, we are now stealthier than this program.

B. Disconnecting client

To disconnect real pilot of the drone, we used `aireplay-ng`, as Samy Kamkar had done. We first tried to use Scapy to send de-authentication packets, but they were not sent correctly.

[1] The client will be the real pilot of the drone.
[2] Name of the WiFi network
[3] MAC address of the access point

C. Controlling software after attack

To control the drone after hijacking, we used `ardrone-webflight`. This program is a web server that serves drone's video and allow us to control a Parrot AR.Drone through a web browser and a keyboard (or a gamepad).

D. Attack prevention

A first solution to prevent our attack is to use Parrot's security solution. This register MAC address of the real pilot in the drone. Then, any other devices from the registered one will be rejected when connecting to the WiFi network. However, MAC address can be spoofed to bypass this security solution.

Another option is to configure the drone access points to not send beacon frames. Doing this, the access point will be hidden and our passive scan will not work. Nonetheless, we could perform an active scan and the hidden access point will answer.

A last solution is to use an encrypted WiFi network, such as WEP or WPA2. This solution was adopted by Parrot to secure their new drones. The de-authentication attack will work, but not the connection to the WiFi: we would need the secret key.

III. Hijacking a Syma X5C-1

The Syma X5C-1 is a small drone, controlled thanks to a RF 2.4 GHz remote controller. Its protocol was Reverse-engineered in June 2016 [8].

A. Development of a hacking tool

We developed a Python script to hijack the drone. This tool uses a nRF24L01+ module to listen for Syma neighbouring drones and to take their control.

The listening consists in searching for any valid packet, according to Syma protocol, on the four radio channels used by a Syma drone.

To take control, we transmit a packet build from the position of a gamepad's joysticks and buttons. This transmission is performed faster than the one of the real remote controller. Thus, as the drone tries to perform all received orders, we can control it.

B. Attack characterization

We built a log tool that listen for Syma packets on a specific radio channel. We used it to build the figure 1. We clearly can see when the attack is performed because of the increase of received packets.

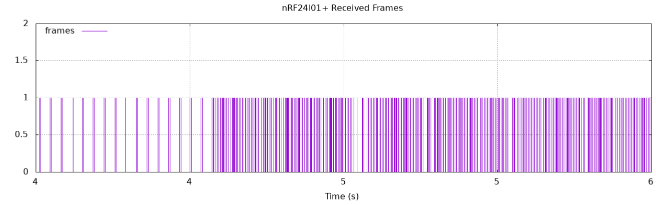

Fig. 1. Received frame before and during attack

Our attack is hard to understand from the real pilot point of view. Indeed, he will lose control of the drone in just a second.

C. Attack prevention

There is no simple and efficient way to protect from this attack. Indeed, we could imagine to scramble all the drone's radio channel, but the control link would be lost for everyone.

We could yet think of scrambling all drone's radio channel just when the lawful remote controller is transmitting. This solution could work, but would requires an extremely accurate scrambling.

A last solution could be to increase the transmission rate of the lawful remote controller when such an attack is detected. Nevertheless, we can't predict the drone's comportement in such a case because of packets collisions. The control link will most probably be lost for everyone.

IV. Embedding tool on a predator drone

We then wanted to embed our tool on a predator drone. To be drone-independent, we decided to install our tool on a Raspberry Pi Zero W [9]. This board runs Debian Stretch and has an SPI connectivity that can be used with the nRF24L01+ module. In addition, this board is really lightweight and can be powered by an external battery.

We used a Bluetooth PAN[4] to establish a SSH connection to the Raspberry Pi. Doing this, we released the integrated WiFi adapter that we used as a second WiFi adapter for the Parrot attack. The first adapter (in monitor mode) was an external USB WiFi dongle.

We first thought about running the Parrot controlling software (`ardrone-webflight`) on the Raspberry Pi. However, it was using too much resources. We decided to deport it on the attacker's computer, thanks to the magic of packet routing on the embedded board.

V. Conclusion and future applications

References

[1] https://www.amazon.com/Amazon-Prime-Air/
[2] https://ctc.usma.edu/islamic-state-drones-supply-scale-future-threats/
[3] https://www.bbc.com/news/uk-england-sussex-46623754
[4] https://www.droneshield.com/
[5] https://samy.pl/skyjack/
[6] https://scapy.net/
[7] https://github.com/wraith-wireless/PyRIC
[8] https://blog.ptsecurity.com/2016/06/phd-vi-how-they-stole-our-drone.html
[9] https://www.raspberrypi.org/products/raspberry-pi-zero-w/

[4] Personal Area Network