Attack On Drones

Yan Songyang
School of Software Engneering
Xi'an Jiaotong University
lapalacayim@gmail.com

Li Siyu
College of Cybersecurity
Sichuan University
sy_lee_real@icloud.com

Wan Ziyi
College of Cybersecurity
Sichuan University
scuderrickwan@gmail.com

Ye Junyan
School of Information and Software Engineering
University of Electronic Science and Technology
2239308357@qq.com

Li Hanxing
School of Cyber Science and Engineering
Wuhan University
t0918555@u.nus.edu

ABSTRACT

In this article we investigate the security problems of the Parrot AR.Drone 2.0 and Hubsan H107 quadcopters. We will set focus mainly on obvious security vulnerabilities in the Wi-Fi and 2.4GHz radio connection. We will show how to eavesdrop video streams of the AR.Drones and take over it's control using Man-In-The-Middle attack. We will illustrate the approach of radio replay attack on the Hubsan H107. Besides the realization of attacks, we put forward some suggestions on how the drones could be secured from unauthorized access.

Categories and Subject Descriptors

H.4 [**Input/Output and Data Communications**]: Data Communication Devices

Keywords

Drone, Remote Control, Man-In-The-Middle, Replay Attack

1. INTRODUCTION

The drone, also known as the unmanned aerial vehicle(UAV), is the Aircraft without any human pilot controlling inside. Drones are widely used in many fields, including military, industry, agriculture, photography and so on. We can simply divide them into civilian and military.

The civilian drones, which we are focusing on, are applied to countless aspects of our lives. Farmers use drones to spray pesticides and monitor crop's growth. E-commerce companies such as Amazon use drones to deliver express. Photographers use drones to create from a special perspective. And also people, just like us, using drones just for fun. All in all, the use of drones has penetrated into many aspect of our lives.

Quadcopter, one of the most common types of civilian drones, using Wi-Fi connection or 2.4GHz radio as its way of communication, generally. Some professional drone companies such as DJI may have their own private communication protocols and they are beyond our research.

Within Wi-Fi connection, a drone is WLAN AP(Access Point) itself. If the user wants to control the drone, he needs to connect his smartphone(or other smart devices) with control application to the WLAN created by the drone. And there may be steps to authenticate during the connection establishment process. Then user's smartphone could issue instructions to the drone and receive real-time image transmission from drone, after getting the Wi-Fi connection to the drone.

And the other way of communication is using 2.4GHz radio. This is a much simpler communication mechanism, which means it has a lower cost and usually used on low-end drones. 2.4GHz radio technique is widely used in lots of remote controls, and remote control used in toys including drones is the one of the application scenarios. The remote control sending instructions via an agreed frequency in 2.4GHz ISM, and the drone receive these instructions. The massage transmit via 2.4GHz radio broadcasts to anyone that can receive it, which means there isn't any connection between remote control and the drone. Not to mention identity verification. So this is a primitive way of control, which is easy to attack.

2. ATTACK ON THE AR.DRONE 2.0

2.1 Technical Specification

The AR.Drone 2.0, equipped with various sensors, uses a Linux operating system based on the kernel version 2.6.27.44. All control commands, telemetry data and the video streams are handled via (unencrypted) 2.4GHz WLAN communication with the controlling device. Users can use iOS and Android devices to control the drone via the official application AR.FreeFlight. Figure 1 shows the controller interface running on iPad.

Figure 1: Parrot AR.FreeFlight control interface

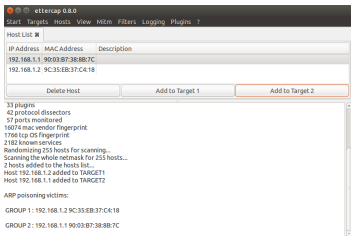

Figure 2: Ettercap Configuration

2.2 Video Signal Interception

According to AR.Drone Developer Guide[6], AR.Drone 2.0 video stream is transmitted on TCP socket 5555. AR.Drone 2.0 will start sending frame immediately when a client connects to the socket.

For drones like JJRC-H37 produced by JIANJIAN TECHNOLOGY, they broadcast video signal to every client. Hence every device that connected to the drone can watch the video stream simultaneously. While AR.Drone 2.0 only sends video stream to the controller. To eavesdrop the video stream sent by AR.Drone 2.0, attacker can do as follows:

1. Connection to the drone,
2. Determination of the IP address of the control device,
3. Using arp spoofing to intercept data,
4. Decode video stream.

In this project, we use Ettercap[2] to conduct Man-In-The-Middle attack using arp poisoning. The drone's ip address is 192.168.1.1 and the controller's ip address is 192.168.1.2. The configuration of Ettercap is shown as figure 2.

AR.Drone SDK 2[3] provides us a tool called ardrone_navigation to decode video stream. It receives data stream from drone. The video module inside will decode the stream and show it in a seperated window. Figure 3 shows the user interface of ardrone_navigation.

2.3 Highjack Attack

2.3.1 AT Commands

The controller uses port 5556 to send commands in a UDP packet to port 5556 of the drone[6]. These commands are

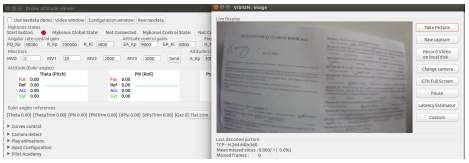

Figure 3: GUI of ardrone_navigation

Table 1: REF commands

Command	Function
ATREF=<sequence>,290718208	Take off
ATREF=<sequence>,290717696	Land
ATREF=<sequence>,290717952	Emergency Stop

called AT commands. Because of the instability of UDP connection, the communication protocol allocates ascending sequence number to different commands. This prevents older commands with lower sequence numbers incoming later (due to transmission errors) from executing[5]. This protocol provides an attack method. Attacker can conduct a man-in-the-middle attack with a sequence number which is always higher than the one being sent from the legitimate user.

An AT command begins with the fixed string "AT*", followed by either REF, PCMD or CONFIG. REF commands are single commands such as land or takeoff. PCMD commands are used for flight control. CONFIG commands are used for sending new configuration. To take over a drone, using REF commands is enough. The format command of REF command is

$$AT * REF =< sequence >, < UI > . \quad (1)$$

Different REF commands are listed in table 1.

2.3.2 Attack Process

The attack consists of the following phases:

1. Connection to the drone,
2. Determination of the IP address of the control device,
3. Sending fake land packets with high sequence number.

After booting up, the drone will set up a Wi-Fi hotspot named "ardrone2_" followed by a random number with 6 digits. The connection to the drone is possible because the network is not protected by any encryption or other access restriction techniques.

Once the connection has been built, attacker can scan the subnetwork to determine the IP address of the control device.

To prevent multiple phones trying to control the drone, the drone only accepts packets from the source IP of the controller. But since it's UDP, it's simple to spoof the IP address of the controller. Attacker can send a land command to force the drone to come to the ground[1].

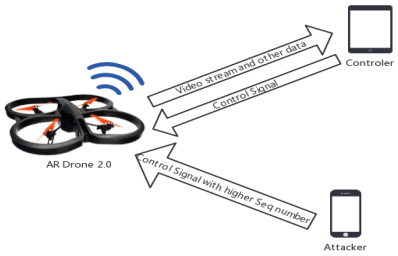

Figure 4: Hijack Attack Scenario

2.3.3 *Using Android Device to Conduct Attack*

After the previous analysis, we have mastered the WIFI-controlled drone attack method and the whole process. The drone uses the application(FreeFlight 2.0) to control takeoff, landing, flight direction and capture image acquisition. We decided to create another Android application to attack this drone. Then this application can be controlled by a third party and take control of the drone, force the drone to land, and modify the flight direction.This attack scenario is shown in Figure 4.

The attack consists of the following phases:

1. The third-party user uses the Android packet capture application to obtain the target drone IP address,
2. After the third-party user enters the obtained IP address in this application, he can click the button to send a command to control the drone.

Jpcap provides a class JpcapSender for sending packets, which can be used to send IPPackets and its subclasses, including IPPacket, ICMPPacket, TCPPacket, UDPPacket[4]. After defining a corresponding package, we can use the SendPacket function to send packets. Meanwhile, Jpcap can be used to construct IP data packets to implement sending fake IP.

The creation of this new Android app is primarily convenient for any third-party user who can hijack drones directly at the application layer.

3. ATTACK ON HUBSAN H107

3.1 Communication Principle

Communicating via 2.4GHz radio is widely used in many scenes, and controlling is one of them. The remote control sending instructions via an agreed frequency in 2.4GHz ISM, and the drone receive these instructions. In many types of drones, such as the Hanson H107 Drone we used in our experiment, the message transmit via 2.4GHz radio broadcasts to anyone that can receive it, which means there isn't any connection between remote control and the drone. Not to mention identity verification. Because of the broadcast mechanism and connectionless transmission without identifications, this is a primitive way of control, which is easy to attack. The attack model is shown in Figure 5.

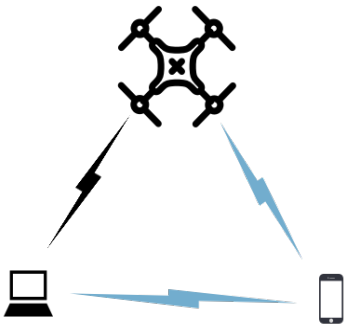

Figure 5: Radio Attack Model

3.2 Attack

Since this drone use broadcast to pass signals. There's no connection mechanism between controller and drone and it does not use a mechanism to verify the controller, the attack on this drone is relatively simple. We can use the LimeSDR device to attack the drone. Firstly, we can set up the attack environment on the PC with LimeSDR, gqrx and GNURadio, and then monitor the radio signal in GNURadio. The running drone can simultaneously monitor the radio signal at 2.4GHz and receive it. To a specific broadcast band signal, the corresponding command can be obtained by decoding.

3.3 Result

Since the drone does not have a mechanism to establish a connection. We can't get his full control, all we can do is interfere with the controller's control, making it insensitive or ineffective. The easiest way is to replay the attack and resend the received signal after a period of time. At this point, the signal from LimeSDR will cause the drone to get out of control or even fall.

4. DEFENSE

The availability of inexpensive drones requires an analysis of potential security and safety threats for such devices. Potential consequences that might be caused by the usage of such drones could be damaging.

To secure the AR.Drone, an encrypted Wi-Fi connection should be adopted. If the attacker fails to join the network hosted by the drone, attacks mentioned above are impossible. Furthermore, as provided in [5], developers can cross-compile patch programs to secure the Wi-Fi connection in a more reliable way.

To defense against replay attack on Hubsan H107, the developers should configure the protocol stack so that it will check the serial number of the data it receives [7].

5. CONCLUSIONS

We have attacked AR.Drone 2.0 and Hubsan H107, using Wi-Fi drone and 2.4GHz radio respectively.

For the AR.Drone, we can intercept its video signal and seize the control of the drone completely, by sending command messages with a larger sequence number.

As for the drone controlled by 2.4GHz radio, we have found its communication frequency. We can interfere with the normal flight of the drone by sending command signal which we have recorded. Because of the radio broadcast mechanism, we can't block the original radio signal completely, so we just interfere it.

To prevent the two drones from being attacked, we offered some approaches to securing Wi-Fi and radio connection.

6. ACKNOWLEDGMENTS

We would like to offer our sincere appreciation and thanks to Professor Hugh Anderson for his guidance, enthusiasm and encouragement during the project. He is always generous in providing all kinds of equipment we might need and we are equally grateful to the School of Computing for this. And we also thank our teaching assistant Tan Wang Leng for his insight and expertise. Without them, we wouldn't have gone thus far.

7. REFERENCES

[1] drone-hacking. `https://github.com/markszabo/drone-hacking`. Accessed: 2016-05-11.

[2] Ettercap home page. `https://www.ettercap-project.org/`. Accessed: 2019-07-23.

[3] Parrot for developers. `https://developer.parrot.com/`. Accessed: 2019-07-23.

[4] C. Peng. Jpcap-based tcp/ip packet capture and transmission. *Journal of Changji University*, 2008.

[5] J. Pleban, R. Band, and R. Creutzburg. Hacking and securing the ar.drone 2.0 quadcopter - investigations for improving the security of a toy. 01 2014.

[6] P. E. Stephane Piskorski, Nicolas Brulez and F. D'Haeyer. *AR.Drone Developer Guide*. 5 2012.

[7] Q. Yang and L. Huang. *Inside radio: an attack and defense guide*. Springer, 2018.