POSTER: Security Analysis of Personal Unmanned Aerial Vehicles

Peng Chen(✉) and Hao Chen

ShanghaiTech University, Shanghai, China
{chenpeng,chenhao}@shanghaitech.edu.cn

Abstract. Personal unmanned aerial vehicles (UAVs) have become popular in recent years. While their extreme mobility enables exciting new applications, they also raise security concerns. However, currently we understand little about UAV's vulnerabilities, feasible attacks, and defense options. Toward securing UAVs, we analyzed two of the most popular personal UAVs. We discovered a series of vulnerabilities, including insecure communication channels and misuse of cryptography. By exploiting these vulnerabilities, an attacker can eavesdrop on the data acquired or transmitted by the aircraft, impersonate the aircraft to send bogus data to the user's mobile device, hijack the camera on the aircraft or the aircraft itself, and prevent the aircraft from communicating with the user's mobile device.

Keywords: Unmanned aerial vehicles · Security vulnerabilities · DJI

1 Introduction

Personal UAVs have become popular in recent years. As they can fly to areas where human access is infeasible, dangerous, expensive, or inconvenient, they have many applications in photography, delivery, and wildlife protection. Unfortunately, security has not been a priority for personal UAVs manufacturers. In the current fast growing market for personal UAVs, manufacturers care more about functions, cost, and applications. In the research community, there is no comprehensive study on UAV's vulnerabilities, feasible attacks, and defense options. To make progress toward securing UAVs, we conducted an empirical security analysis of the DJI series of quadcopters, which are among the most popular personal UAVs.

A UAV consists of an aircraft, a hardware remote controller, a commodity mobile device (e.g., an Android-powered device or iPhone) running a mobile app for the UAV. In this paper, we examined two representative DJI UAVs: *Phantom 2 Vision+* [3], *Phantom 3 Professional* [4].

2 Phantom 2

Phantom 2 Vision+ is a quadcopter consisting of an aircraft, remote controller, and range extender. The aircraft has a WiFi module, video module, receiver,

R. Deng et al. (Eds.): SecureComm 2016, LNICST 198, pp. 743–746, 2017.
DOI: 10.1007/978-3-319-59608-2_41

NAZA V2 controller, gimbal, and camera. The aircraft communicates independently with the remote controller and the mobile app DJI Vision, where the remote controller controls the flight and the mobile app controls the camera.

2.1 Vulnerabilities

Phantom 2 Vision+ fails to provide secure networks, and its servers fail to authenticate clients. Since neither of the WiFi APs in the UAV is encrypted, any one can connect to them. Neither the users manual nor DJI Vision allows the user to enable encryption on these networks. Any DJI Vision app can connect to the UAV without authentication, after which the app can control the camera, ground station etc. However, the TCP server created by *ser2net* [7] in the WiFi module accepts at most one connection at any time. The UAV designer might have intended to use this mechanism to protect the UAV owner, because her app will likely connect to the UAV first after she powers on the UAV.

2.2 Hijack Aircraft Communication

The goal of this attack is to hijack the communication between the aircraft and victim DJI Vision app. Since *ser2net* allows only one TCP connection at any time, we must close the existing TCP connection between the WiFi module and victim app before we could connect to the WiFi module. We achieved this by the TCP reset attack [5,8]. The requiresite parameters for the attack (IPs, ports, and sequence numbers) can be sniffed in the packets transmitted between the app and the aircraft. After the TCP reset attack closed the existing connection between the WiFi module and the victim app, we connected to the WiFi module. This connection allowed us to prevent victim app from acquiring live video, to exfiltrate photos and videos from aircraft, to control aircraft's camera and hijack aircraft (Sect. 2.4).

2.3 Attack on Video Module

By reverse engineering the video module, we found that it uses a modified version of UDP-based data transfer (UDT) [6] for communicating with the app and H.264 for video codec. Similar to the TCP reset attack described in Sect. 2.2, we can sniffed the IPs, ports and sequence numbers of the UDT packets transmitted from the aircraft to the app. Then we created our UDT packets using these parameters and sent them to the victim app from our malicious device. Our attack caused the following damages: we sent packets containing our bogus video and verified that the app indeed was playing our video; after receiving some crafted UDT packets for a while, DJI Vision crashed; we sent UDT packets to DJI Vision as fast as we could and observed the communication between aircraft and app was disabled.

2.4 Ground Station

Ground station allows the user to create flight tasks via DJI Vision. After we reverse engineered the protocol for controlling ground station. we were able to hijack the aircraft by sending upload job, upload point, and joystick commands. We were also able to exfiltrate existing job and waypoints on the aircraft. This would be useful for forensic analysis or reconnaissance.

3 Phantom 3 Professional

In Phantom 3, the mobile device connects via a USB cable to the remote controller, which connects to the aircraft via a DJI-proprietary wireless system. The DJI Pilot app on the mobile device connects to the remote controller via Android debug bridge (ADB) [1]. DJI Pilot creates a TCP server at the host 0.0.0.0 and the port 22345. Once the remote controller powers on, it runs a TCP client to connect to the server via ADB port forwarding.

3.1 Insecure Server

A vulnerability in the TCP server created by DJI Pilot is that it listens to 0.0.0.0, which the official Android security tips advise against [2], since this allows any host that can address the mobile device (e.g., an attacker on the same wireless LAN) to connect to this server. To fix this vulnerability, the server should listen to `localhost` (or 127.0.0.1), because ADB forwards the remote controller's connection request from `localhost`. This way, the server will reject any connection request that originates outside the mobile device.

3.2 Hijack Aircraft

The TCP server created by Pilot does not authenticate clients. However, it accepts at most one connection at any time, so no other program can connect to this server if the remote controller has already connected to it. The Pilot developers might have intended to use this mechanism to protect the Pilot and aircraft, but we found three attacks to circumvent it.

Win Race Against Remote Controller. As soon as Pilot starts, the remote controller requests connection. However, we found that if our malware app also requested connection repeatedly when Pilot started, our malware almost always won the race against the remote controller, after which the remote controller could never connect. Now that our malware impersonated the remote controller (and transitively, the aircraft), it was able to send bogus data, such as fake photos or videos, to Pilot, and Pilot could no longer communicate with the aircraft.

Kill and Impersonate Pilot. If the remote controller is already connected to Pilot's TCP server when our malware starts, the malware must close the existing connection. One way to achieve this is to push Pilot into the background and then invoke the *killBackgroundProcesses* method in the *ActivityManager* class[1]. The malware can send an attractive bogus notification to trick the user to click it. The click brings the malware to the foreground and pushes Pilot to the background. After killing Pilot, the malware creates a TCP server to impersonate Pilot to communicate with the remote controller.

4 Conclusion

We studied the risks of UAVs and conducted an empirical analysis of three popular DJI UAVs. We discovered a series of vulnerabilities, including insecure communication channels and misuse of cryptography. We have demonstrated that, by exploiting these vulnerabilities, an attacker can eavesdrop on the data acquired or transmitted by the aircraft, impersonate the aircraft to send bogus data to the user's mobile device, hijack the camera on the aircraft or the aircraft itself, and prevent the aircraft from communicating with the user's mobile device.

5 Other Works and Responsible Disclosure

We also analysed DJI Matrice 100, DJI's mobile and onboard SDKs, founding that the steps of UAV activation and developer authorization are insecure.

We notified DJI of all the discovered vulnerabilities and verified attacks between June and August of 2015. DJI was confident that they would be able to fix the vulnerabilities by the time of this conference.

References

1. Android Debug Bridge. http://developer.android.com/tools/help/adb.html
2. Android Security Tips. http://developer.android.com/training/articles/security-tips.html
3. DJI Phantom 2 Vision+. http://www.dji.com/product/phantom-2-vision-plus
4. DJI Phantom 3 Professional. http://www.dji.com/product/phantom-3
5. Floyd, S.: Inappropriate TCP Resets Considered Harmful. RFC, United States (2002). 3360
6. Gu, Y., Grossman, R.L.: UDT: UDP-based data transfer for high-speed wide area networks. Comput. Netw. **51**(7), 1777–1799 (2007)
7. ser2net. http://ser2net.sourceforge.net/
8. Watson, P.: Slipping in the Window: TCP Reset attacks (2004)

[1] Our malware needs the `KILL_BACKGROUND_PROCESSES` permission to invoke this operation.