Open Source Forensics for a Multi-platform Drone System

Thomas Edward Allen Barton and M. A. Hannan Bin Azhar(✉)

Computing, Digital Forensics and Cybersecurity,
Canterbury Christ Church University, Canterbury, UK
{tb1150,hannan.azhar}@canterbury.ac.uk

Abstract. Drones or UAVs (Unmanned Air Vehicles) have a great potential to cause concerns over privacy, trespassing and safety. This is due to the increasing availability of drones and their capabilities of travelling large distances and taking high resolution photographs and videos. From a criminological perspective, drones are an ideal method of smuggling, physically removing the operator from the act. It is for this reason that drones are also being utilised as deadly weapons in conflict areas. The need for forensic research to successfully analyse captured drones is rising. The challenges that drones present include the need to interpret flight data and tackling the multi-platform nature of drone systems. This paper reports the extraction and interpretation of important artefacts found in the recorded flight logs on both the internal memory of the UAV and the controlling application, as well as analysis of media, logs and other important files for identifying artefacts. In addition, some basic scripts will be utilised to demonstrate the potential for developing fully fledged forensics tools applicable to other platforms. Tests of anti-forensics measures will also be reported.

Keywords: Drone forensics · Open source · Mobile forensics · DJI Phantom Android · UAV · Anti-forensics

1 Introduction

Drone crime is a recent phenomenon. In the UK, there was a sharp rise in reported incidents between 2014 and 2015 [1]. The most widespread crime being committed is the transport of contraband, also known as smuggling [1, 2]. The capabilities of drones to carry items [3] and their remote operation makes drones ideal for this type of crime, which has become prolific in the UK and around the world [4]. The cost of even a high-end drone is far outweighed by the inflated value of the cargo [5, 6] meaning drones can be discarded after use. This type of crime has serious impact, and drones used in crime will need to be forensically analysed if caught or shot down. The potential for the misuse of drones to disrupt large scale operations as well as assist in major crime means the identification of suspects is of paramount importance in prevention of further crime. The vulnerability of many sensitive targets to a drone attack should not be ignored, again raising the need for forensic research to successfully analyse captured drones.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018
P. Matoušek and M. Schmiedecker (Eds.): ICDF2C 2017, LNICST 216, pp. 83–96, 2018.
https://doi.org/10.1007/978-3-319-73697-6_6

Open Source techniques provide a number of advantages as they are flexible and meet guidelines on the admissibility of evidence [7]. This paper will cover the use of open source tools and development of some basic scripts to aid forensic analysis of a multi-platform drone system, which include not only the UAV itself but the accompanying mobile platform, application and controlling hardware. The UAV system chosen for analysis was the DJI Phantom 3 Professional (DJI) [8], a quadcopter drone with a variety of features and capabilities. Among commercially available drones, DJI has taken the largest market share of 36% [9] with its Phantom series setting the benchmark for professional drone use. The extensive capabilities of the Phantom include vision, GPS, automatic flight and homing, obstacle avoidance and long range control. These capabilities give the Phantom the potential to be used in various drone related crimes. The remainder of the paper is organised as follows: Sect. 2 describes literature reviews on crimes involving drones and in the area of drone forensic analysis, including techniques used for data extraction and interpretation. Section 3 discusses the methodology used to analyse the UAV and accompanying mobile platform. Section 4 reports the results of analysis, finally Sect. 5 concludes the paper.

2 Literature Review

Although drones are a relatively new technology, some literature exists on both the forensic analysis and their cybersecurity implications. Another technology that goes hand in hand with drones is cameras, implemented either as static recording devices, or for live streaming (sometimes known as vision). This raises a host of privacy concerns for organisations, as well as the public. Many different areas of airspace in the UK are designated no-fly zones [10] because they are considered sensitive areas – these include sites such as airports, military bases and power stations. The ability of drones to capture pictures and videos of operations in these sites presents a significant security threat. As well as the security of infrastructure, individual security may also be compromised. Of the reported incidents mentioned [1], many were simply concerns for public safety. As well as these general incidents, drones are also being used to aid traditional crime, a common example of which is burglary. Using drones, a burglar can survey a potential target site for entrances or exits and security features such as dogs, alarms and cameras in a process known as “casing” or keep an eye out for police [11]. Drones are being utilised as deadly weapons in the countries involved in conflicts [12]. A set of videos released by various forces and militants showed the use of commercially bought and homemade drones as bombers, hitting soft targets such as groups of exposed soldiers and vehicles with customised grenades and High Explosive Dual Purpose (HEDP) rounds [12]. These type of attacks are mostly performed with hovering-type drones, with modifications to add the capacity of dropping bombs [12].

Some important aspects of UAV forensic analysis were highlighted including establishing flight data and establishing ownership [13]. The identification of mobile devices, for comparison, is aided by the presence of artefacts such as account names and details whereas it is possible to operate a drone with little or no identifying artefacts left on it. The digital forensic investigator will also have to interpret recorded flight data. In order to successfully re-create the actions taken by the drone, the understanding of timestamped latitude, longitude and altitude measurements is required, as well as speed, battery level and other data from a host of on-board sensors. A drone system is comprised of a number of different hardware platforms, each containing different artefacts. Some of these component platforms are shown to have physically identifiable artefacts such as serial numbers printed on the casing, which can later be matched up to artefacts recovered using digital forensics [14]. Artefacts related to flight data were successfully recovered from various components of the DJI Phantom 2 Vision+, including the controller, mobile application and the UAV itself [15]. Analysis of recorded media such as photos and videos, stored on the UAV's removable SD card, showed they possessed Exchangeable Image Format (EXIF) metadata that included GPS readings. This can be used in the absence of flight logs, for example if the images were copied to a separate storage media or the UAV was damaged in some way. An analysis of the DJI Phantom 3 Standard version revealed multiple security vulnerabilities [16], as well as establishing how the various components of the DJI Phantom 3 operate with each other. The controller, in this case, is essentially a range extender for sending commands to the UAV via 5 GHz radio signal. The smartphone running the DJI GO application connects to the controller via 2.4 GHz Wi-Fi or by USB connection, which provides access to a network created between the various components. Accessing this network may provide useful in acquiring data, where chip-off analysis is not available [16].

Open source and custom forensics tools provide some significant advantages over commercial toolkits, primarily the ability to be tested by the open source community, meeting what are known as the "daubert" guidelines for the admissibility of evidence provided by expert witnesses [7]. Furthermore, custom tools created by the forensic investigator to perform a specific job are extremely adaptable and, where successful, can be used again in other cases involving similar technology. The rising cost of commercial toolkits can be a barrier to use [17], which makes a stark comparison to the freedom of open source tools. However, commercial status does offer the advantage of support in the form of updates, bug reporting and additional documentation. While previously reported work [13–16] focussed on the extraction of automated flight plans and analysis of media, the investigation presented in this paper will primarily focus on the extraction and interpretation of wider range of important artefacts found both on the internal memory of the professional edition of the Phantom 3 and the controlling application with the use of open source tools. Anti-forensics measures will also be tested.

3 Methodology

The study reported in this paper focusses on the DJI Phantom 3 Professional Edition [8] and the accompanying mobile platform - a Motorola Moto G 3rd Generation, as shown in Tables 1 and 2. The choice of mobile platform in this case reflects the current state of the worldwide smartphone market, which is dominated by Android [18]. Another reason Android was chosen was its huge online developer community, which stems from its open source status. A custom community built version of Android, CyanogenMod [19], was installed on the platform prior to analysis, which included features such as forensically sound rooting without extra modification. The scenario creation was performed before rooting took place. CyanogenMod is based on universal open-source Android software, tested to the same standards as stock operating systems [20]. A secondary platform - a Samsung Galaxy S4 Mini running a stock Android 4.4.4 operating system was tested alongside the main platform to ensure consistency between results, with the same version of the DJI GO application installed. The secondary platform was rooted using a rootkit, Kingo Root [21], which exploits weaknesses in the operating system - a method commonly used on Android systems where native rooting is not supported [22]. Upon examination, there was no noticeable difference in the data structures created by both applications on the internal storage media of the platforms.

Table 1. Drone.

Name	Price	Weight	Camera resolution	Range
DJI Phantom 3 Professional Edition	£699.99	1280 g	4K (12 Megapixels)	5 km

Table 2. Mobile platform.

Name	Model number	Android version	CyanogenMod version	Kernel version	Installed application
Motorola Moto G 3rd Generation	Moto G	5.1.1 (Lollipop)	12.1 (Osprey)	3.10.49-g55f6ac8	DJI GO v3.1.4
Samsung Galaxy S4 Mini	GT-I9195I	4.4.4 (Kitkat)	N/A	3.10.28-5334500	DJI GO v3.1.4

In order to test the devices and generate artefacts, a scenario must be created using the devices. This is a necessary and established part of forensic research [22]. A scenario, in a digital forensics context, is a simulation of a crime using the device to be tested. Because drones, as mentioned earlier, have a great potential to cause concerns over privacy, trespassing and safety, all tests of the devices were to follow legal guidelines on drone safety [10]. The location in which the flights were conducted was suitable for safely testing the capabilities of the drone away from congested areas, and possessed some useful features such as tall building structures and large open space. A chosen standard flight path, consisting of four waypoints within an approximate 150 m radius was established. A number of flights were conducted testing both the manual and automatic function of the drone.

The analysis performed on the UAV and the mobile platform was artefact-driven. Artefacts related to drones were divided into three categories relating to the identification of suspects, interpretation of flight data and the extraction of artefacts from recorded media. The main identification aspect was the method of control of the drone via a smartphone. The DJI Phantom uses a physical controller in conjunction with commands from the smartphone, transmitted to the drone over radio [8]. These methods of control leave footprints on the drone. Identifying artefacts such as MAC (Media Access Control) address, phone model, operating system etc. will be crucial in reducing a suspect pool in investigations.

Flight data was collected during flight via various sensors present in the drone platform including but not limited to GPS, altitude, speed and battery levels. These can reveal details about the flight of the drone that may prove crucial in an investigation, for example the "home" GPS co-ordinate is where the drone took off. Another example is in the event of a drone crash, as battery levels can be correlated with the time that the drone failed.

Media includes any photos or videos taken by the device's camera. The use of drones as bombers mentioned earlier [12] was all recorded via the drone's on-board camera in order to produce videos, and the capture and analysis of such a bombing drone would be able to reveal important intelligence. The DJI phantom is equipped with a high-end camera capable of high resolution photos and videos, making it suitable for this kind of activity.

Because the analysis performed comprised UAV systems, mobile devices, and removable storage, a variety of file systems and interfaces were encountered. Development environments for forensics tools include scripting tools for the Linux operating system such as Bash, Perl and Python, as well as compiled programming languages such as "C". A forensic workstation running Kali, a distribution of Linux, with several forensics and cybersecurity tools was used, as listed in Table 3.

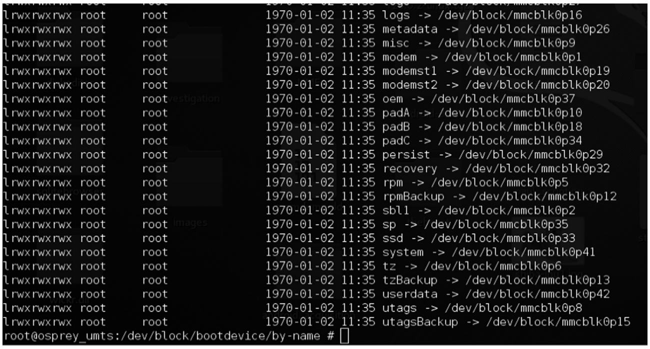
Table 3. Forensic utilities.

Computer used	Operating system	Utilities
Toshiba Satellite L450D	Kali Linux Rolling Update	ls: Listing dd: Data Dump mount: Mount command dmesg: System Logging file: File signature identification script: Terminal recording feature arp: Address Resolution Protocol telnet: Remote Access uname: Version Identification cp: Copy cat: Print file contents bash: Scripting environment

3.1 Mobile Forensics

Mobile forensics was performed to analyse the data of the DJI GO application [23], which was installed via the Android app store. The test mobile platform was a Motorola Moto G 3rd Generation running a customised version of Android, CyanogenMod version 12.1 [19]. This operating system allows for extensive customisation including rooting of the device without needing to subvert operating system security. With the customised operating system, rooting was achieved simply by activating root requests from the developer settings of the phone. Rooting is necessary to acquire portions of the Android internal storage that are protected by the operating system [22], it is the most forensically sound way of acquiring data when chip-off analysis is not available.

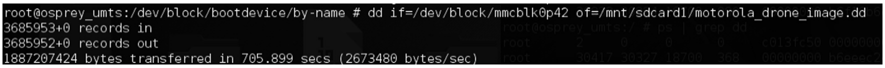
After connecting the test platform to the forensic workstation via USB, access was established through an instance of Android Debug Bridge [24]. Running the command “ls/dev/block/bootdevice/by-name” gave a listing of the mounted partitions on the device, as shown in Fig. 1.



```
lrwxrwxrwx root      root              1970-01-02 11:35 logs -> /dev/block/mmcblk0p16
lrwxrwxrwx root      root              1970-01-02 11:35 metadata -> /dev/block/mmcblk0p26
lrwxrwxrwx root      root              1970-01-02 11:35 misc -> /dev/block/mmcblk0p9
lrwxrwxrwx root      root              1970-01-02 11:35 modem -> /dev/block/mmcblk0p1
lrwxrwxrwx root      root              1970-01-02 11:35 modemst1 -> /dev/block/mmcblk0p19
lrwxrwxrwx root      root              1970-01-02 11:35 modemst2 -> /dev/block/mmcblk0p20
lrwxrwxrwx root      root              1970-01-02 11:35 oem -> /dev/block/mmcblk0p37
lrwxrwxrwx root      root              1970-01-02 11:35 padA -> /dev/block/mmcblk0p10
lrwxrwxrwx root      root              1970-01-02 11:35 padB -> /dev/block/mmcblk0p18
lrwxrwxrwx root      root              1970-01-02 11:35 padC -> /dev/block/mmcblk0p34
lrwxrwxrwx root      root              1970-01-02 11:35 persist -> /dev/block/mmcblk0p29
lrwxrwxrwx root      root              1970-01-02 11:35 recovery -> /dev/block/mmcblk0p32
lrwxrwxrwx root      root              1970-01-02 11:35 rpm -> /dev/block/mmcblk0p5
lrwxrwxrwx root      root              1970-01-02 11:35 rpmBackup -> /dev/block/mmcblk0p12
lrwxrwxrwx root      root              1970-01-02 11:35 sbl1 -> /dev/block/mmcblk0p2
lrwxrwxrwx root      root              1970-01-02 11:35 sp -> /dev/block/mmcblk0p35
lrwxrwxrwx root      root              1970-01-02 11:35 ssd -> /dev/block/mmcblk0p33
lrwxrwxrwx root      root              1970-01-02 11:35 system -> /dev/block/mmcblk0p41
lrwxrwxrwx root      root              1970-01-02 11:35 tz -> /dev/block/mmcblk0p6
lrwxrwxrwx root      root              1970-01-02 11:35 tzBackup -> /dev/block/mmcblk0p13
lrwxrwxrwx root      root              1970-01-02 11:35 userdata -> /dev/block/mmcblk0p42
lrwxrwxrwx root      root              1970-01-02 11:35 utags -> /dev/block/mmcblk0p8
lrwxrwxrwx root      root              1970-01-02 11:35 utagsBackup -> /dev/block/mmcblk0p15
root@osprey_umts:/dev/block/bootdevice/by-name # 
```

Fig. 1. Sample listing of mounted partitions on Android platform.

The mount point for the “userdata” partition, which contains all user-created data including application data, is shown as “/dev/block/mmcblk0p42”. A forensic image of this partition was created using the “dd” command, as shown in Fig. 2. This is a type of physical acquisition, which creates an exact copy of the digital storage media. Before this could take place, a few conditions needed to be met. Firstly the ADB access needed to have root permissions, which was granted by an operating system root request. Secondly, the SD card used to store the image was formatted in the ExFAT (Extended FAT) file system, which has no restrictions on file sizes. Once completed, this created an image on a removable microSD card, which was copied to the forensic workstation for analysis.



```
root@osprey_umts:/dev/block/bootdevice/by-name # dd if=/dev/block/mmcblk0p42 of=/mnt/sdcard1/motorola_drone_image.dd
3685953+0 records in
3685952+0 records out
1887207424 bytes transferred in 705.899 secs (2673480 bytes/sec)
```

Fig. 2. Forensic imaging of “mmcblk0p42” partition using “dd” command.

3.2 UAV

A number of flights were performed with the Phantom, as listed in Table 4. The source of this list is the practical log of flights taken on the day rather than data obtained from analysis of the UAV. Once the flights had been performed, the DJI was taken back to a forensics lab for analysis. The primary method of data storage for the DJI Phantom is the removable micro SD card slot. During the test flight, a 16 GB micro SD card was inserted, which was provided with the UAV itself. To analyse this media, the card was mounted to the forensic workstation and an image was created using the “dd” command. This is a forensically sound method of acquisition as the device does not need to be powered on. An initial check of the image using the Linux "file" command shows the card is formatted in the 32 bit File Allocation Table (FAT32) file system. The SD card's format is commonly found on many mass storage devices and it was analysed using various Linux utilities. The recorded media produced by the phantom stores some useful information, including GPS data, in the EXIF portion of the file. In order to interpret this data, the command line tool "exiftool" [25] was used. Data extracted from the UAV's mass storage devices was correlated with artefacts extracted from the DJI GO mobile application, to highlight links between the controlling application and the UAV.

Table 4. Flight record.

Flight	Start time	Waypoints	End time	Description, notes and recorded media
1	13:57	Travelled a short distance north of the home point before returning	13:18	Test flight for compass calibration
2	14:05	Waypoint 1: 14:06 Waypoint 2: 14:07 Waypoint 3: 14:12 Waypoint 4: 14:14	14:15	Manual flight, GPS assisted, 1 photo and one short video taken at each waypoint
3	14:17	Automatic reconnaissance flight Auto land (return to home) 14:22	14:22	Automatic flight, GPS assisted, using DJI's built-in Point Of Interest (POI) function, which makes the drone rotate around a specified point. Video was recorded the entire flight
4	14:34	(Same waypoints at flight 2, time not recorded due to operator concentrating on flight) Manual landing	14:37	In this flight, foil was attached to the drone covering the GPS module. The drone was operated completely manually independent of GPS. This simulated the intentional obfuscation of GPS signals as mentioned in related work [15, 16]

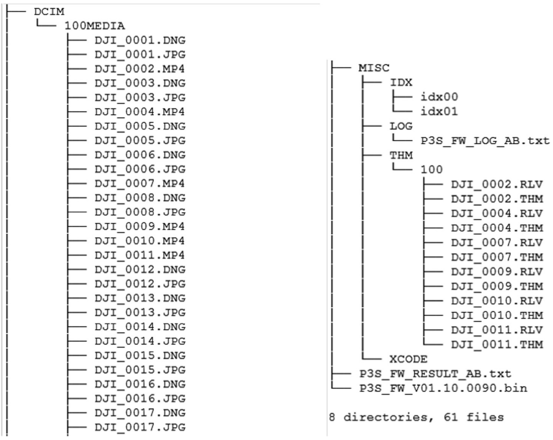
Along with the removable storage, the Phantom also has an internal storage media, a micro SD card, glued on to the centre board of the UAV [14]. To access this storage device, the UAV must be switched on and put into "Flight Data Mode" through the DJI GO application. The UAV was then connected to the forensic workstation via USB and the internal storage was mounted. Analysis of the file system using "fsstat" [26] showed the drive was formatted in FAT32, and a forensic image of the drive was acquired using the "dd" command. Upon examination, the drive contained a number of "FLYXXX.DAT" files - detailed flight logs, created by the Phantom's internal operating system and stored in a proprietary format [14]. These files were logically copied to a removable storage device for further analysis. There are many online services offering interpretation of these files, however uploading evidence to a third party server is not appropriate for a forensic investigation or intelligence purposes, so a tool designed to interpret and visualise these files, "CsvView" [27] was downloaded and installed to a separate machine running Windows, connected to the internet. The tool was established with a Google Maps API key, allowing it to download imagery from the Google Maps database.

4 Results

This section covers the key findings from the analysis described in Sect. 3. The results are broken down into three different areas of interest; the removable SD card used by the UAV, the internal storage of the UAV and the results of the mobile forensic analysis on the DJI GO application.

4.1 SD Card

The DJI Phantom micro SD card image acquired as described in Sect. 3.1 was mounted to the forensic workstation. Output from the "tree" [28] command lists the files and directories of this image. There are two directories, DCIM and MISC, as shown in Fig. 3. The DCIM directory contains a wealth of .JPG, .DNG and .MP4 files, all of which are common media file formats.



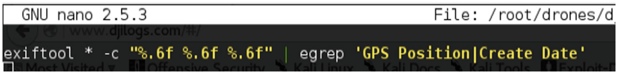
```
├── DCIM
│   └── 100MEDIA
│       ├── DJI_0001.DNG
│       ├── DJI_0001.JPG
│       ├── DJI_0002.MP4
│       ├── DJI_0003.DNG
│       ├── DJI_0003.JPG
│       ├── DJI_0004.MP4
│       ├── DJI_0005.DNG
│       ├── DJI_0005.JPG
│       ├── DJI_0006.DNG
│       ├── DJI_0006.JPG
│       ├── DJI_0007.MP4
│       ├── DJI_0008.DNG
│       ├── DJI_0008.JPG
│       ├── DJI_0009.MP4
│       ├── DJI_0010.MP4
│       ├── DJI_0011.MP4
│       ├── DJI_0012.DNG
│       ├── DJI_0012.JPG
│       ├── DJI_0013.DNG
│       ├── DJI_0013.JPG
│       ├── DJI_0014.DNG
│       ├── DJI_0014.JPG
│       ├── DJI_0015.DNG
│       ├── DJI_0015.JPG
│       ├── DJI_0016.DNG
│       ├── DJI_0016.JPG
│       ├── DJI_0017.DNG
│       ├── DJI_0017.JPG
│       │

├── MISC
│   ├── IDX
│   │   ├── idx00
│   │   └── idx01
│   ├── LOG
│   │   └── P3S_FW_LOG_AB.txt
│   ├── THM
│   │   └── 100
│   │       ├── DJI_0002.RLV
│   │       ├── DJI_0002.THM
│   │       ├── DJI_0004.RLV
│   │       ├── DJI_0004.THM
│   │       ├── DJI_0007.RLV
│   │       ├── DJI_0007.THM
│   │       ├── DJI_0009.RLV
│   │       ├── DJI_0009.THM
│   │       ├── DJI_0010.RLV
│   │       ├── DJI_0010.THM
│   │       ├── DJI_0011.RLV
│   │       └── DJI_0011.THM
│   └── XCODE
├── P3S_FW_RESULT_AB.txt
└── P3S_FW_V01.10.0090.bin

8 directories, 61 files
```

Fig. 3. Sample output of "tree" command.

The file found under the LOG directory was a firmware upgrade log for the UAV. It refers to the file "P3S_FW_v01.10.0090.bin", located on the root of the SD card, meaning that file is the firmware update itself. Other useful information in this log includes a version history of the firmware, up to the current version. The THM directory appears to contain thumbnails generated from each flight. To analyse the EXIF Data of the stored media files, "exiftool" [25] was run against the DCIM/100MEDIA directory. On initial inspection, GPS co-ordinates are stored under a "GPS Position" EXIF tag. To automate the process of extracting the GPS co-ordinates and to create a timestamped GPS flight log, a simple script was created, as shown in Fig. 4. The script executes “exiftool” on all files in the directory, formatting the GPS data to 6 decimal places. The output is then filtered to only contain the GPS Position and Create Date, which denotes when the picture or video was taken.



```
  GNU nano 2.5.3                                        File: /root/drones/d

exiftool * -c "%.6f %.6f %.6f" | egrep 'GPS Position|Create Date'
```

Fig. 4. Script to retrieve GPS data from media EXIF information.

4.2 Internal Storage

The files extracted from the internal storage of the DJI Phantom were analysed using the “CsvView” tool [27]. The DJI Phantom 3 Operating system begins recording flight data from the moment the UAV is switched on. This meant as flights 1–3 listed in Table 4 were performed in the same session of drone activity, the data for those flights were recorded in one file, “FLY012.DAT”. After processing using “CsvView” [27], which converts the file from a “.DAT” to a “.csv” format, the flights were visualised using the “GeoPlayer” function, which utilised the Google Maps API Key mentioned in Sect. 3.2. A copy of this visualisation is shown in Fig. 5, with each flight and waypoints 1–4 and the point of interest (POI) highlighted. Because it is constantly recorded, the GPS data alone is not enough to distinguish between individual flights.

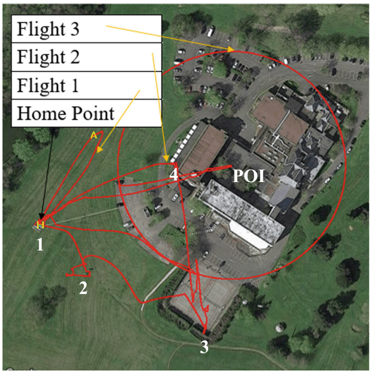
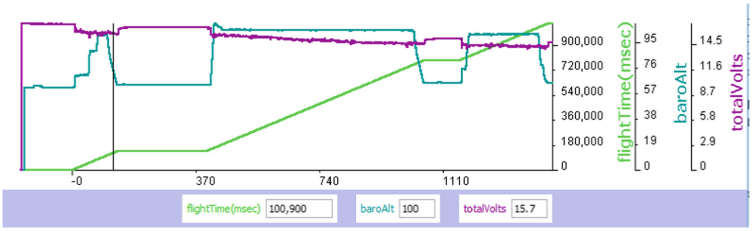


Fig. 5. Annotated visualisation of flights 1–3.

The DJI Phantom flight recorder produces a host of other artefacts. Plotting these artefacts against each other using the “CsvView” [27] tool provides a comprehensive understanding of the actions taken by the drone. Figure 6 shows the flight time (green), which remains constant under periods of non-activity, increasing in a linear function when the drone is in flight, as well as the barometric altitude (blue) and the total voltage level of the battery (purple) of the UAV. When compared with each other, it can be deduced that there was three distinct periods of movement and altitude changes by the drone, were interpreted as flights. The possible artefacts recoverable from these logs are extremely detailed, and are more than necessary to recreate a flight.



flightTime(msec) 100,900 baroAlt 100 totalVolts 15.7

Fig. 6. Flight time, barometric altitude and battery voltage. (Color figure online)

The file “FLY014.DAT” file was identified as being the log for the Flight 4, listed in Table 4. The “GeoPlayer” visualisation for this flight showed that the GPS data recorded was mostly garbage data that had no relation to the actual flight, as shown in Fig. 7.

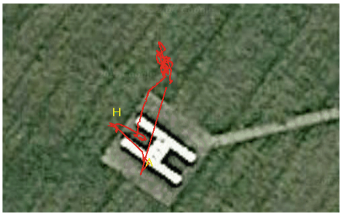
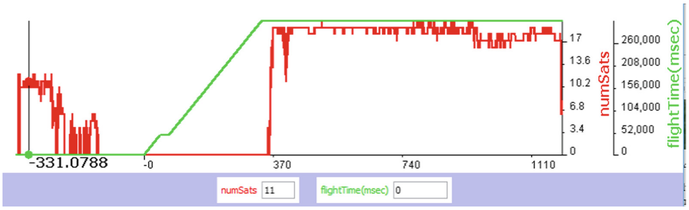


Fig. 7. Garbage GPS data from flight 4.



numSats 11 flightTime(msec) 0

Fig. 8. GPS health plotted against flight time for flight 4.

According to the operator’s previous experience, the recommended amount of GPS signals was about 11, but with the foil obstructing the unit, the Phantom struggled to receive enough GPS data to successfully triangulate a position. To confirm this was the case, the flight time and “numSats” (number of satellites) readings from the flight logs were compared, and showed that during flight, the “numSats” reading was 0, as shown in the time period (X-Axis) of 0 to 370 in Fig. 8. This is interpreted as a lack of available satellites for the UAV to receive data, which was true when the drone was in flight, as described by the flight time. The foil was removed after the flight due to fears of overheating the drone through obstruction of the cooling vents. The data shown in Figs. 7 and 8 confirms findings from related work [15] that the GPS can be obstructed simply by covering the module with aluminium foil. It is quite likely that in a crime scenario, this measure would be taken to prevent later forensic analysis of the flight path, or to evade no fly zones. In this case, investigators must instead rely on other data from the flight log. The DJI Phantom 3 Professional is equipped with accelerometers, which record the acceleration in an axis relative to the UAV in metres/second^2. Accelerometer measurements can be used to reconstruct a flight in 3D space, relative to an arbitrary home point. Inspection of the accelerometer readings showed a period of movement while the UAV was in flight. While it would be possible to perform analysis of this manually, the frequency of measurements taken by the Phantom makes it unreasonable, and it would be better to develop a tool to do this.

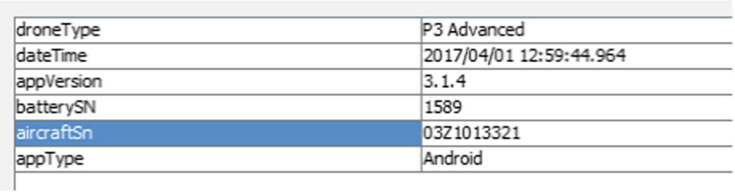
4.3 DJI GO Application

Artefacts from the DJI GO application [23] were located in different locations within the “userdata” partition of the Android test platform, which was acquired using methods described in Sect. 3.1. A list of these directories is shown in Table 5.

Table 5. Useful directories from the DJI GO application.

Path	Type of artefact	Description
/media/0/DJI/dji.pilot/LOG/CACHE	Flight data	Contains a number of logs relating to drone activity
/media/0/DJI/dji.pilot/LOG/CACHE/NFZ	Flight data	This is a log of activity relating to the DJI’s built-in no fly zone function, and contains information such as GPS location
/media/0/DJI/dji.pilot/LOG/ERROR_POP_LOG	Flight data	An error log from the UAV
/media/0/DJI/dji.pilot/DJI_RECORD	Media	A number of video taken during flight named as a date in the format “YYYY_MM_DD_ hh_mm_ss” and stored with the “mp4” file extension. For each video file, there is also a corresponding text file, which contains GPS data, manufacturing information and capture dates
/media/0/DJI/dji.pilot/FlightRecord	Flight data, personally identifying information, serial number	Flight data relating to a number of flights. A string search revealed the presence of the “cccu phantom” string, which was the name assigned to the UAV during setup
/media/0/DJI/dji.pilot/CACHE_IMAGE	Media	Thumbnails of various images and videos taken during flight, seemingly random

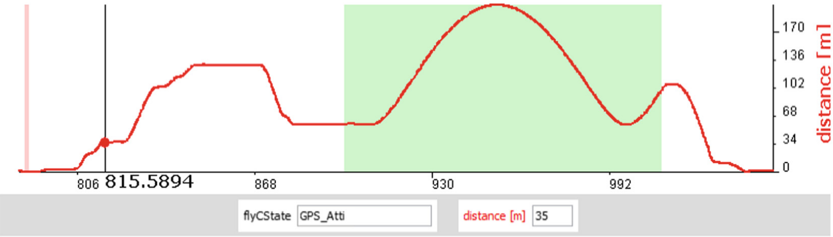
The serial number for the UAV can be extracted from the contents of the DJI GO application and linked to track the specific device used in flight. The data reveals information about the UAV's internal system operations such as updates and errors. A log is also kept of instances when the UAV encountered a no fly zone (NFZ) during flight. Media is present as copies of videos captured during flight are locally stored by the application. Flight data files with the ".txt" extension were extracted from the "FlightRecord" directory. The flight record files extracted from the "FlightRecord" directory were analysed using the "CsvView" [27] tool for comparison to the ".DAT" flight logs extracted from the Phantom's internal storage. Upon inspection, the files were confirmed to be flight data stored in a similar format to the ".DAT" files, but with notable differences. Firstly, the resolution of the recorded data is much lower, with the DJI GO application flight records being between 1 Kb and 1 Mb, whereas the ".DAT" files from the UAV were much larger, often several hundred megabytes. Secondly, files were recorded per flight from take-off to landing rather than per session of activity, meaning it was clearer when distinguishing between flights. The ".txt" files also had noticeably more metadata than the ".DAT" files – including serial numbers of the UAV and the DJI smart battery, application version information and the operating system of the test platform, as shown in Fig. 9.



droneType	P3 Advanced
dateTime	2017/04/01 12:59:44.964
appVersion	3.1.4
batterySN	1589
aircraftSn	03Z1013321
appType	Android

Fig. 9. Metadata from DJI GO application flight log.

As well as the metadata shown in Fig. 9, several other streams of flight data relating to use of the DJI GO application were also available. The "flyCState" attribute described whether the Phantom was in manual or automatic mode. Figure 10 shows the distance of the UAV from the home point plotted against the "flyCState" attribute during the Flight 3.



flyCState GPS_Atti

distance [m] 35

Fig. 10. Flight state plotted against distance from home point for flight 3.

The automatic POI function mentioned in Table 4 generated a clearly visible sine wave (Fig. 10) in the distance measurements during the time when the UAV was in automatic flight mode. This useful artefact identifies when the POI function has been used in a flight. While the GPS data for Flight 4 was also destroyed by the foil covering the GPS receiver, it was also possible to extract the GPS location of the controlling application. This is a crucial finding as it allows for the location of the operator at the time of flight. Anti-forensics measures to counteract this may include GPS spoofing on a software level on the mobile platform, which is possible with free applications available on app markets such as google play.

5 Conclusion

The results from the DJI Phantom 3 Professional show a number of successful methods to retrieve data from the UAV and controlling devices using open source tools. Artefacts present in the flight record data were used to identify key actions taken by the drone using some heuristics and pattern detection. Correlation of these and other artefacts extracted from the mobile platform were enough to establish a connection between the drone and the controlling application. With every drone system, there are many different artefacts spread across a number of devices, file systems, and networks. The forensic analysis of drones requires a correlation of these artefacts to retrieve the actions of the drone. The DJI phantom had an extraordinarily large amount of artefacts associated with it. This was due to having more sensors and a higher resolution of data capture, which stems from its status as a professional device. To recreate the actions of the drone, it was necessary to interpret flight data collected by the UAV. This involved interpreting the movements of the UAV in three dimensional space, as well as data from on-board sensors including accelerometer data and battery levels. A number of useful artefacts were found on the controlling application, and would be enough to identify a suspect.

Further work needs to be done in developing and exploring methods for analysing drone systems in the future, especially integrating the methods discussed in this paper into commercial forensics toolkits. The extraction of data from controlling applications on iOS devices should be explored for comparison to the Android mobile forensics methods demonstrated in this paper. Newer drones, such as the Phantom 4 and the Mavic, will also need to be analysed to explore the differences with previous versions.

References

1. Yeung, P.: Drone reports to UK police soar 352% in a year amid urgent calls for regulation, The Independent (2016). http://www.independent.co.uk/news/uk/home-news/drones-police-crime-reports-uk-england-safety-surveillance-a7155076.html. Accessed 7 Aug 2017
2. BBC news: big rise in drone smuggling incidents (2016). http://www.bbc.co.uk/news/uk-35641453. Accessed 7 Aug 2017
3. UAV Systems international: Tarot T-18 Ready to Fly Drone. https://uavsystemsinternational.com/product/tarot-t-18-ready-fly-drone/3. Accessed 7 Aug 2017

4. Noel, A.: Drone Carrying Three Kilos of Meth Crashes in Tijuana, Vice News (2015). https://news.vice.com/article/drone-carrying-three-kilos-of-meth-crashes-in-tijuana. Accessed 7 Aug 2017
5. Francis, D.: Want to Smuggle Drugs into Prison? Buy a Drone, The Cable - The Foreign Policy Group (2016). http://foreignpolicy.com/2015/08/04/want-to-smuggle-drugs-into-prison-buy-a-drone. Accessed 7 Aug 2017
6. Sullivan, J.P., Bunker, R.J.: Mexican Cartel Strategic Note No. 18: Narcodrones on the Border and Beyond. Small Wars J. (2016). http://smallwarsjournal.com/jrnl/art/mexican-cartel-strategic-note-no-18-narcodrones-on-the-border-and-beyond. Accessed 7 Aug 2017
7. Carrier, B.: Open source digital forensics tools: the legal argument, @stake research report. http://www.digital-evidence.org/papers/opensrc_legal.pdf. Accessed 7 Aug 2017
8. DJI Phantom 3 Professional. https://www.dji.com/phantom-3-pro. Accessed 7 Aug 2017
9. Glaser, A.: DJI is running away with the drone market, recode technology website. https://www.recode.net/2017/4/14/14690576/drone-market-share-growth-charts-dji-forecast. Accessed 7 Aug 2017
10. CAA: Flying Drones. https://www.caa.co.uk/Consumers/Guide-to-aviation/Airspace/Who-manages-UK-airspace-/. Accessed 7 Aug 2017
11. Barrett, D.: Burglars use drone helicopters to target homes, The Telegraph. http://www.telegraph.co.uk/news/uknews/crime/11613568/Burglars-use-drone-helicopters-to-identify-targe-homes.html. Accessed 7 Aug 2017
12. Waters, N.: Death From Above: The Drone Bombs of the Caliphate, Bellingcat open source intelligence. https://www.bellingcat.com/uncategorized/2017/02/10/death-drone-bombs-caliphate. Accessed 7 Aug 2017
13. Horsman, G.: Unmanned aerial vehicles: a preliminary analysis of forensic challenges. Digit. Invest. **16**, 1–11 (2016)
14. Kovar, D.: UAV (aka drone) Forensics, SANS DFIR summit (2015). https://www.sans.org/summit-archives/file/summit-archive-1492184184.pdf. Accessed 7 Aug 2017
15. Maarse, M., Sangers, L., van Ginkel, J., Pouw, M.: Digital forensics on a DJI Phantom 2 Vision+ UAV. MSc System and Network Engineering, University of Amsterdam (2016)
16. Trujano, F., Chan, B., Beams, G., Rivera, R.: Security Analysis of DJI Phantom 3 Standard, Massachusetts Institute of Technology (2016). https://courses.csail.mit.edu/6.857/2016/files/9.pdf. Accessed 7 Aug 2017
17. Huebner, E., Zanero, S.: The case for open source software in digital forensics. In: Huebner, E., Zanero, S. (eds.) Open Source Software for Digital Forensics, pp. 3–7. Springer, Heidelberg (2010). https://doi.org/10.1007/978-1-4419-5803-7_1
18. Woods, V., Meulen, R.V.D.: Gartner Says Worldwide Smartphone Sales Grew 3.9 Percent in First Quarter of 2016. http://www.gartner.com/newsroom/id/3323017. Accessed 7 Aug 2017
19. CyanogenMod android operating system. https://cyngn.com/. Accessed 7 Aug 2017
20. Karlsson, K.J.: Android anti-forensics at the operating system level. M.Sc. thesis, University of Glasgow (2012)
21. Kingo Root Tool. https://www.kingoapp.com. Accessed 7 Aug 2017
22. Barton, T., Azhar, M.H.B.: Forensic analysis of the recovery of Wickr's ephemeral data on Android platforms. In: The First International Conference on Cyber-Technologies and Cyber-Systems, pp. 35–40. IARIA (2016)
23. DJI GO application. http://www.dji.com/goapp. Accessed 7 Aug 2017
24. ADB tool – Android Debug Bridge tool. https://developer.android.com/studio/command-line/adb.html. Accessed 7 Aug 2017
25. Exiftool. http://www.sno.phy.queensu.ca/~phil/exiftool/. Accessed 7 Aug 2017
26. Sleuthkit – fsstat. https://www.sleuthkit.org/. Accessed 7 Aug 2017
27. CsvView tool. https://datfile.net/CsvView/downloads.html. Accessed 7 Aug 2017
28. The "tree" tool. http://www.easydos.com/tree.html. Accessed 7 Aug 2017