



CREaTE

Canterbury Research and Theses Environment

Canterbury Christ Church University's repository of research outputs

<http://create.canterbury.ac.uk>

Please cite this publication as follows:

Azhar, M. H. B., Barton, T. and Islam, T. (2018) Drone forensic analysis using open source tools. *Journal of Digital Forensics, Security and Law*, 13 (1). pp. 7-30. ISSN 1558-7223.

Link to official URL (if available):

<https://doi.org/10.15394/jdfsl.2018.1513>

This version is made available in accordance with publishers' policies. All material made available by CReaTE is protected by intellectual property law, including copyright law. Any use made of the contents should comply with the relevant law.

Contact: create.library@canterbury.ac.uk



See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/324994744>

Drone Forensic Analysis Using Open Source Tools in The Journal of Digital Forensics, Security and Law; Available at: <https://commons.erau.edu/jdfsl/vol13/iss1/6/>

Article · May 2018

CITATIONS

0

3 authors:



M A Hannan Bin Azhar
Canterbury Christ Church University

23 PUBLICATIONS 56 CITATIONS

[SEE PROFILE](#)



Thomas Barton
Canterbury Christ Church University

7 PUBLICATIONS 3 CITATIONS

[SEE PROFILE](#)



Tasmina Islam
University of Kent

3 PUBLICATIONS 2 CITATIONS

[SEE PROFILE](#)

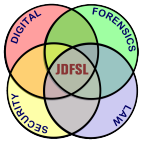
Some of the authors of this publication are also working on these related projects:



Drone Forensics [View project](#)



Security and reliability of biometrics [View project](#)



3-31-2018

Drone Forensic Analysis Using Open Source Tools

M A Hannan Bin Azhar

Canterbury Christ Church University, hannan.azhar@canterbury.ac.uk


Thomas Edward Allen Barton

Canterbury Christ Church University, forensicstom@gmail.com

Tasmina Islam

International Association of Engineers, tasmina.ukc@gmail.com

Follow this and additional works at: <https://commons.erau.edu/jdfsl>

 Part of the [Computer Law Commons](#), and the [Information Security Commons](#)

Recommended Citation

Azhar, M A Hannan Bin; Barton, Thomas Edward Allen; and Islam, Tasmina (2018) "Drone Forensic Analysis Using Open Source Tools," *Journal of Digital Forensics, Security and Law*: Vol. 13 : No. 1 , Article 6.

Available at: <https://commons.erau.edu/jdfsl/vol13/iss1/6>

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

EMBRY-RIDDLE
Aeronautical University[®]

SCHOLARLY COMMONS

(c)ADFSL



Drone Forensic Analysis Using Open Source Tools

Cover Page Footnote

shorter version of this paper was presented in 9th EAI International Conference on Digital Forensics & Cyber Crime, Prague, Springer-Verlag, October 2017

DRONE FORENSIC ANALYSIS USING OPEN SOURCE TOOLS

M A Hannan Bin Azhar¹, Thomas Edward Allen Barton¹ and Tasmina Islam²

¹Computing, Digital Forensics and Cybersecurity

Canterbury Christ Church University, Canterbury, United Kingdom

Email: hannan.azhar@canterbury.ac.uk, forensicstom@gmail.com,

²Member of International Association of Engineers

<http://www.iaeng.org>

Email: tasmina.ukc@gmail.com

ABSTRACT

Carrying capabilities of drones and their easy accessibility to the public have led to an increase in crimes committed using drones in recent years. For this reason, the need for forensic analysis of drones captured from the crime scenes and the devices used for these drones is also paramount. This paper presents the extraction and identification of important artefacts from the recorded flight data as well as the associated mobile devices using open source tools and some basic scripts developed to aid the analysis of two popular drone systems- the DJI Phantom 3 Professional and Parrot AR. Drone 2.0. Although different drones vary in their operations, this paper extends the extraction and analysis of the data from the drones and associated devices using some generic methods which are forensically sound adhering to the guidelines of the Association of Chief Police Officers (ACPO).

Keywords: Digital forensics, Drone forensics, Open source tools, DJI Phantom, AR Drone 2.0.

1. INTRODUCTION

Drones, also known as unmanned aerial vehicles (UAV), are being increasingly popular amongst public due to their accessibility and affordability. This popularity is not only helping in rapid growth of global commercial market of UAVs (Majendie & Chia, 2018; Moskwa, 2016) but also inevitably increasing drone crimes (Yeung, 2016). The carrying capabilities of drones over long distances (UAV, 2018) and their remote operation make drones ideal for transport of contraband, also known as smuggling (BBC, 2016; Yeung, 2016), the most widely committed drone crime. This type of drone crime has become prolific in the UK and around the world (Dinan, 2017;

Mikelionis, 2018), such as dropping weapons, phones, drugs into prisons or delivering drugs or sometimes arms in and out of a country bypassing borders. As well as smuggling, the camera mounted onto a drone, either as a static recording or a live streaming device, raises significant data privacy concerns for organisations and public. Also, the ability of drones capturing pictures or videos of operations in designated no-fly-zone areas of airspace (CAA, 2015), such as, airports, military base and power stations, presents a significant security threat. Drone-mounted cameras are also being used for traditional crime such as burglary (Barrett, 2015;

Siddique, 2017). Furthermore, drones are being utilised as deadly weapons in countries involved in conflict. Mostly hovering-type drones such as the DJI Phantom or similar, are used in these type of attacks (Hambling, 2017; Waters, 2017).

Due to the rise in criminal activities, the need for forensic analysis of the captured drones has augmented immensely. After capturing the drone, a forensic analysis can provide a lot of information about the potential suspect of a crime based on the data gathered from on-board sensors and other electronics that assist with flight and navigation, as well as the camera and digital storage. This can also help in preventing further crime. Interpreting the flight data and tackling the multi-platform nature of drones are the major challenges in forensic analysis of drones. This paper presents 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 with the use of open source tools as they are flexible and meet guidelines on the admissibility of evidence (Carrier, 2002). Additionally, some basic scripts will be used to aid the forensic analysis of two commercially popular drone systems, demonstrating the potential for developing more robust forensic tools applicable to other platforms. The chosen drones for analysis are -the DJI Phantom 3 Professional (DJI, 2018) and Parrot AR. Drone 2.0 Power Edition (Parrot, 2017), for ease of subsequent reference the drones are designated as DJI and A.R respectively throughout the paper. The DJI is a quadcopter drone with a variety of features and capabilities and dominating with 70% and Parrot A.R is in second with 7% of market share among commercially available drones (Valentak, 2017). These drones are different in their

operation and their capabilities and some generic methods reported in (Barton & Azhar, 2018) will be applied in this paper for the analysis and comparison of results between the two drone models.

The remainder of the paper is organised as follows: Section 2 reviews the existing research in relation to forensic analysis of drones, Section 3 discusses the methodology used to analyse the drones and accompanying mobile platform. In Section 4 results of the analysis are reported and finally Section 5 concludes the paper.

2. LITERATURE REVIEW

A forensic analysis of UAV system using Parrot Bebop UAV is reported in (Horsman, 2016). The author highlighted some key areas of analysis including acquisition of data, establishing flight data and the ownership, these pose a variety of challenges to the digital forensic investigators. Firstly, the presence of identifying artefacts such as name and address is not essential for drone operation, it is possible to operate a drone with little or no identifying artefacts left on it. Which is in complete contrast to forensic investigation involving mobile devices where an abundance of personal information is available in the devices. Secondly, to re-create the actions taken by the drone, interpretation of the recorded flight data is essential, which is not a likely skillset of forensic investigator. At a minimum, the understanding of timestamped latitude, longitude and altitude measurements is required, as well as speed, battery level and other data from a host of possible on-board sensors.

Modern drone systems are comprised of a number of hardware platforms, which makes drones a valuable source of forensic artefacts, creating the need for forensic research into the area. Some of these component platforms contain physically identifiable artefacts such as

serial numbers printed on the casing, which can later be matched up to artefacts recovered using digital forensics (Kovar, Dominguez, & Murphy, 2015). Another work (Maarse, Sangers, Ginkel, & Pouw, 2016) presented a forensic analysis of DJI Phantom 2 Vision+, where flight data related artefacts were successfully recovered from various components of the UAV, including the controller, mobile application and the UAV itself. The recorded media of the Phantom 2 Vision+ was found to possess Exchangeable Image Format (EXIF) metadata including GPS (Global Positioning System) information. 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, co-ordinates extracted from EXIF data can be used to recreate a flight.

An analysis of the DJI Phantom 3 Standard edition (Trujano, Chan, Beams, & Rivera, 2016) revealed that an IPv4 network is created between the components of the UAV system including the drone, controller, on-board camera and mobile devices. The controller relays commands to the drone via radio signal. The smartphone running the DJI GO application connects to the controller via Wi-Fi or by USB connection, which provides access to the network. De-compilation of the DJI GO application revealed the Service Set Identifier (SSID) and password required to gain access to this network. The authors also discussed (Trujano et al., 2016) several security issues related to DJI Phantom 3 Standard edition.

A detailed security analysis of the DJI Phantom 3 Advanced edition is presented in (Luo, 2016). The author identified various security issues and their countermeasures by several analyses including firmware analysis, GPS analysis, radio signal analysis and Software Development Kit (SDK) authentication.

The work reported in (Jain, Rogers, & Matson, 2017) analysed the basic structure of five commercially available drones and proposed some steps, that would aid the digital investigation of drones. Some of the steps include, risk assessment, collection of data from the crime scene, identification of drone category, weight, fingerprint available on the drone, data signal in Wi-Fi or Bluetooth, memory card, and lastly, documentation of every steps.

Another article (Pleban, Band, & Creutzburg, 2014) analysed the security threats on A.R Drone 2.0, for example, attack through the Telnet or FTP server or through Wi-Fi by de-authenticating the real user and proposed encryption to secure the drone. A.R drones use an embedded Linux operating system that governs the flight, camera and network interfaces. The drone provides an unsecured (by default) wireless access point. Once connected, root access to the operating system is granted via an anonymous telnet port. Root access presents a number of options for acquisition, including imaging internal storage partitions and logical-level copying (Horsman, 2016).

In a recent work (Barton & Azhar, 2018) utilised open source tools for forensic analysis of a multi-platform UAV system. Unlike commercial toolkits, open source and custom forensics tools have 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 (Carrier, 2002). The freedom of using open source tools is another advantage over the costly commercial (Zanero & Huebner, 2010). Furthermore, successful custom tools created for one specific case, can be adapted in other cases involving similar technology. However, the support available in the form of updates, bug reporting and additional

documentation is an advantage of commercial toolkits.

Most of the studies reported above focused on the extraction of automated flight plans and analysis of media utilising methods applicable to specific models of drones. The investigation presented in this paper will focus on the extraction and interpretation of wider range of important artefacts found both on the internal memory of the drones and the controlling application with the use of open source tools and some generic methods that can be applied to both the DJI Phantom 3 Professional and the A.R. Drone 2.0 Power edition, as well as testing anti-forensics measures.

3. METHODOLOGY

The study reported in this paper focusses on two drones and the accompanying mobile platform - a Motorola Moto G 3rd Generation, as shown in Tables 1 and 2. Android is chosen as the mobile platform because of its dominance in mobile market (Gartner, 2018)

and its huge online developer community, which comes from its open source status. Prior to analysis, a custom community built based on universal open-source Android software, CyanogenMod (CyanogenMod, 2017), was installed on the platform. CyanogenMod includes forensically sound rooting feature which does not require further modification and is tested to the same standards as stock operating systems (Karlsson & Glisson, 2014). The scenario creation was performed before rooting took place. Alongside the Motorola Moto G 3rd Generation, a Samsung Galaxy S4 Mini running a stock Android 4.4.4 operating system as a second platform, rooted using Kingo Root (KingoApp, 2017), was tested to ensure consistency between results, with the same version of the DJI GO application installed. No noticeable difference was found in the data structures created by both applications on the internal storage media of the platforms.

Table 1.
Drones

Name	Specifications			
	<i>Price</i>	<i>Weight</i>	<i>Camera Resolution</i>	<i>Range</i>
DJI Phantom 3 Professional	£699.99	1280g	4K (12 Megapixels)	5Km
A.R Drone 2.0	£299.99	380g / 420g	720p (0.9 Megapixels)	50m

Table 2.
Mobile Platforms

Name	Model Number	Android Version	CyanogenMod Version	Kernel Version	Installed Application
Motorola Moto G 3 rd Generation	Moto G3	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

A scenario was created using the devices by simulating the use of the drones in a crime, to generate the required data for acquisition and analysis as this is a necessary and established part of forensic research (Azhar & Barton, 2016). Because of potential privacy and safety concerns of using drones, as mentioned earlier, selection of the location and tests of the devices were conducted following legal guidelines on drone safety (CAA, 2015). A suitable remote area with tall building structures and open space was chosen to test the capabilities of the drones. Four waypoints over about a 150m radius were established to test both the manual and automatic function of the drones.

An artefact-driven analysis was performed on the UAVs and the mobile platforms and were divided into three categories. The first of these is identification of suspects. In this case, a suspect is most likely to be the user of the drone, and therefore the main area of interest in identification is the method of control, especially via smartphone. Each drone included in this project uses a slightly different method of control, all with smartphones. The DJI Phantom, for example, uses a physical controller in conjunction with commands from the smartphone, transmitted to the drone (DJI, 2018) whereas the A.R Drone 2.0 uses direct connection from the smartphone to the drone via Wi-Fi and Bluetooth respectively. Each of these methods will leave a different footprint on the drone and identifying artefacts such as MAC (Media Access Control) address

and phone model, operating system etc. will be crucial in reducing a suspect pool in investigations.

Another category of artefacts related to drones is the interpretation of the flight data. These were collected via various sensors present on the UAV systems. Some key data of interest were GPS readings, battery levels, altitude, acceleration, speed. Analyses of these data can reveal the actions of the drone during flight. For example, GPS co-ordinate can reveal from where the drone took off, or in the event of a crash, battery levels can reveal the time when the drone failed as it can be correlated with time. These data can also be used to re-construct the flight, which is especially important when the drone has been used in smuggling or other flight-related crime.

The category of artefacts related to drones is the extraction of artefacts from recorded media which includes any photos or videos taken by the device's camera. All of the drones are fitted with at least one camera which is controllable through the smartphone application or controller. The use of drones as bombers by ISIS was all recorded via the drone's on-board camera in order to produce videos (Waters, 2017), and the capture and analysis of such a bombing drone would be able to reveal actual and potential targets and measures such as evacuation can take place. The DJI is equipped with a high-end camera capable of high resolution photos and videos,

while the A.R is equipped with two fixed lower resolution cameras.

Because of the acquisition and the analysis of the artefacts performed on multi-platforms, including the 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 were performed to analyse the data of the DJI GO (DJI, 2018) and A.R Freeflight (Parrot, 2017) applications, which were installed via the Android app store. As mentioned in Section 3, a Motorola Moto G 3rd Generation running a customised version of Android, CyanogenMod (CM) version 12.1 (CyanogenMod, 2017), was used as the test mobile platform. This operating system provide rooting, which is necessary to access portions of internal storage that are protected by the operating system’s security (Azhar & Barton, 2016). With this customised operating system the root access was granted natively

without needing to install third-party rooting software, which is a forensically sound option when methods such as chip-off analysis are not available. Once the test platform was connected to the forensic workstation via USB, root terminal access was granted using Android Debug Bridge (ADB) (Android, 2017). The “userdata” partition was identified by running the command “ls /dev/block/bootdevice/by-name” as shown in Figure 1. A forensic image of this partition was created using the “dd” command, as shown in Figure 2. This created an image on a removable microSD card attached to the test platform, which was copied to the forensic workstation for analysis.

```
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_ums:/dev/block/bootdevice/by-name #
```

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

```
root@osprey_ums:/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)
```

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

However, upon attempting to mount the image, the format was not recognised. Identifying this partition in the output of the “mount” command on the test platform revealed that the “userdata” partition is formatted in the Flash Friendly Filesystem, or “f2fs,” which is designed specifically for flash storage devices (Lee et al., 2015). After checking compatible filesystems on the forensic

workstation using the “cat /proc/filesystems” command, it appeared that the “f2fs” file system was not supported by Kali. To overcome this, the “f2fs-tools” (f2fs-tools) package for debian was installed. However, on attempting to mount the image again, the mount command returned errors, shown in the output of the “dmesg” command in Figure 3.

```
root@lab:/media/root/SAMSUNG/University/Dissertation/Analysis/images# dmesg | tail
[ 960.180197] device-mapper: ioctl: 4.33.0-ioctl (2015-8-18) initialised: dm-devel@redhat.com
[ 960.214816] loop: module loaded
[ 1075.651068] F2FS-fs (dm-0): Magic Mismatch, valid(0xf2f52010) - read(0x0)
[ 1075.651075] F2FS-fs (dm-0): Can't find valid F2FS filesystem in 1th superblock
[ 1075.651112] F2FS-fs (dm-0): Magic Mismatch, valid(0xf2f52010) - read(0x0)
[ 1075.651115] F2FS-fs (dm-0): Can't find valid F2FS filesystem in 2th superblock
[ 1075.651119] F2FS-fs (dm-0): Magic Mismatch, valid(0xf2f52010) - read(0x0)
[ 1075.651121] F2FS-fs (dm-0): Can't find valid F2FS filesystem in 1th superblock
[ 1075.651123] F2FS-fs (dm-0): Magic Mismatch, valid(0xf2f52010) - read(0x0)
[ 1075.651125] F2FS-fs (dm-0): Can't find valid F2FS filesystem in 2th superblock
```

Figure 3. Error messages from attempting to mount “userdata” forensic image

At a glance, it seemed that the file system was corrupted, making the image unreadable. To fix this, the image would have to be modified. A copy of the image was made in order to maintain forensic soundness if modifications were later questioned in a court of law. Then, the tool “fsck.f2fs” from “f2fs-tools,” a version of the “fsck” tool designed to

work with “f2fs” file systems, was used on the image. The “fsck” tool automatically scans for file system errors and corrects them. A portion of the output is seen in Figure 4. After this was completed, the image was successfully mounted on the forensic workstation and was ready for analysis, as seen in Figure 5.

```

root@lab:/media/root/SAMSUNG/University/Dissertation/Analysis/Images# fsck.f2fs motorola_image_final.dd
Info: Segments per section = 1
Info: Sections per zone = 1
Info: sector size = 512
Info: total sectors = 9535232 (4655 MB)
Info: MKFS version
"Linux version 3.10.49-gf9e7acc (hudsoncm@ilclbld109) (gcc version 4.8 (GCC) ) #1 SMP PREEMPT Mon Jan 4 07:50:33 CST 2016"
Info: FSCK version
from "Linux version 3.10.49-g55f6ac8 (build02@cyanogenmod) (gcc version 4.8 (GCC) ) #1 SMP PREEMPT Mon Nov 16 19:09:19 PST 2015"
to "Linux version 4.3.0-kali1-amd64 (debian-kernel@lists.debian.org) (gcc version 5.3.1 20160101 (Debian 5.3.1-5) ) #1 SMP Debian 4.3.3-5kali4 (2016-01-13)"
Info: superblock features = 1 : encrypt
Info: superblock encrypt level = 0, salt = 00000000000000000000000000000000
Info: total FS sectors = 9535232 (4655 MB)
Info: CKPT version = 4541f
Info: checkpoint state = 6 : compacted_summary orphan_inodes sudden-power-off
[ASSERT] (sanity_check_nid: 388) --> nid[0x7026] nat_entry->ino[0x7026] footer.ino[0xbe0e]
[FIX] (fsck_chk_orphan_node:1515) --> [0x7026] remove from orphan list
    
```

Figure 4. Running “fsck.f2fs” tool on forensic image.

```

root@lab:/mnt/analysis# ls
adb          app-lib      camera       data          fota          lost+found   power_log    security      tombstones
anr          app-private  camera_dump  dontpanic     hardware_revisions  media        power_supply_logger  shared        tpapi
app          audio        connectivity dpm           hostapd       mediadrms   resource-cache  ss-ram-dumps  user
app-asec    backup       dalvik-cache drm            local         misc         rfs          time          wapi_certificate
    
```

Figure 5. Successfully mounted forensic image.

3.2 Drones

As mentioned in Section 3, flight data such as GPS readings, altitude, speed, acceleration and battery levels were collected via various sensors present on the both the UAV systems.

1) DJI Phantom 3 Professional: An operational diagram of DJI with potential

artefacts is shown in Figure 6. Following the methodology described in Section 3, a number of flights were conducted with the DJI Phantom, as listed in Table 4. This list is the practical log of flights taken on the day rather than data obtained from analysis of the UAV.

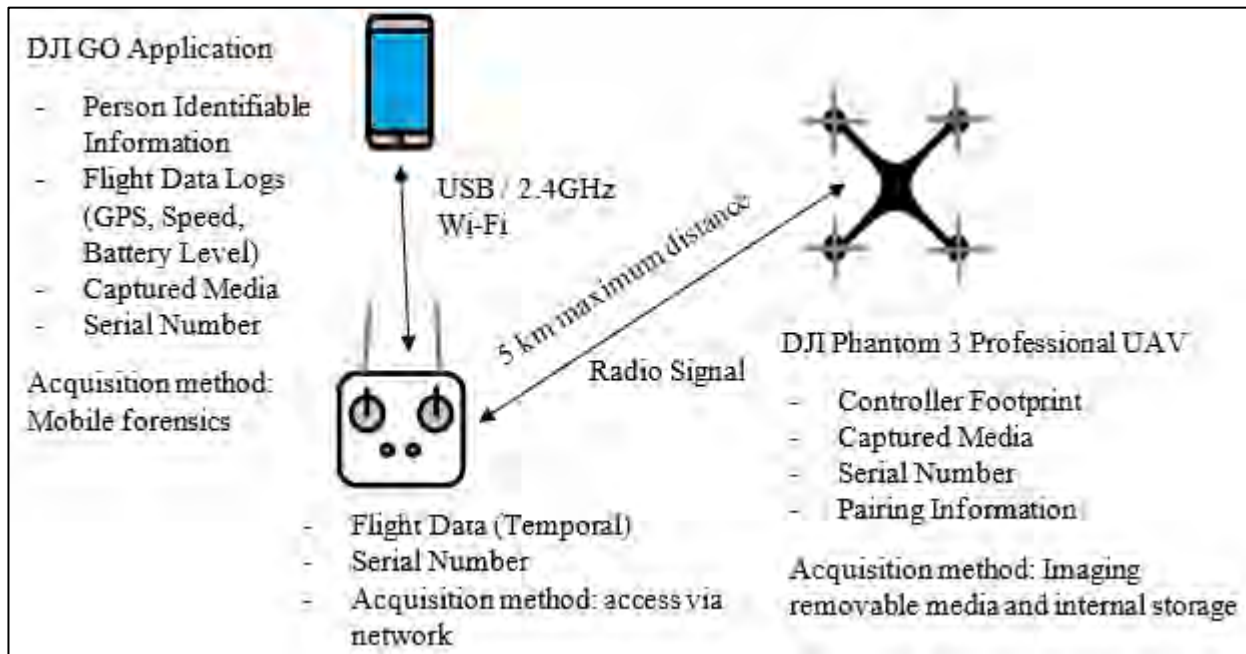


Figure 6. DJI Phantom 3 Professional operation and potential artefacts

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 16GB 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 DJI 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” (exiftool) was used. Data extracted from the UAV’s mass storage devices were then correlated with artefacts extracted from the DJI GO mobile application, to highlight links between the controlling application and the UAV. The controller in this case does not seem to have any digital storage capacity of interest and was excluded from the scope of this investigation.

Another area of interest is the UAV’s internal mounted storage. This is a micro SD card permanently attached to the main board of the UAV. 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 drive, named “DJI FLY LOG,” was mounted. Analysis of the file system using “fsstat” 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, which were detailed flight logs, created by the DJI’s internal operating system and stored in a proprietary format (Kovar et al., 2015). These files were 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” 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.

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].

2) A.R Drone 2.0: An operational diagram of the A.R Drone 2.0 with potential artefacts is shown in Figure 7. A single flight, with the aim of collecting photos and media from the UAV and generating data on the A.R free flight application, was performed according to the procedure listed in section 3. Further flight was decided against due to safety concerns over wind levels.

The structure of the A.R Drone 2.0's system results in three areas of interest for forensic artefacts, including the artefacts generated by the A.R Freeflight application and stored on the UAV's internal and removable storage. The application data was acquired using the methods described in section 3.1, and the removable storage media, a 512Mb flash drive formatted in the FAT32 file system, was connected to the forensic workstation and a forensic image created using the "dd" command for later analysis.

The A.R Drone 2.0 does not have any hardware ports allowing access to the internal

storage, meaning the only method of access was through the UAV's Wi-Fi network. This method has been used by both digital forensics and cyber security researchers to acquire data and investigate the drone (Horsman, 2016). When switched on, the A.R becomes a Wi-Fi hotspot with the SSID "ardrone2," without any form of authentication. This has become a hot topic for security researchers and has even allowed for the development of automated drone hacking tools which target the weak security of drones such as the A.R (Pleban et al., 2014). Connecting to this network and interfacing with the UAV will invariably change digital data on the device. Therefore, all actions taken should be in accordance with the Association of Chief Police Officers (ACPO) good practice guidelines for handling digital evidence (ACPO, 2012) principles 2; the person acquiring the data must be competent and give evidence for their actions, and 3; that an audit trail of processes should be created and preserved. The lack of other

routes into the A.R.'s internal memory is enough to fulfil principle 2 if actions taken are later questioned, and for principle 3, a

complete log will be kept using the Linux "script" tool, which records all terminal commands in a text file.

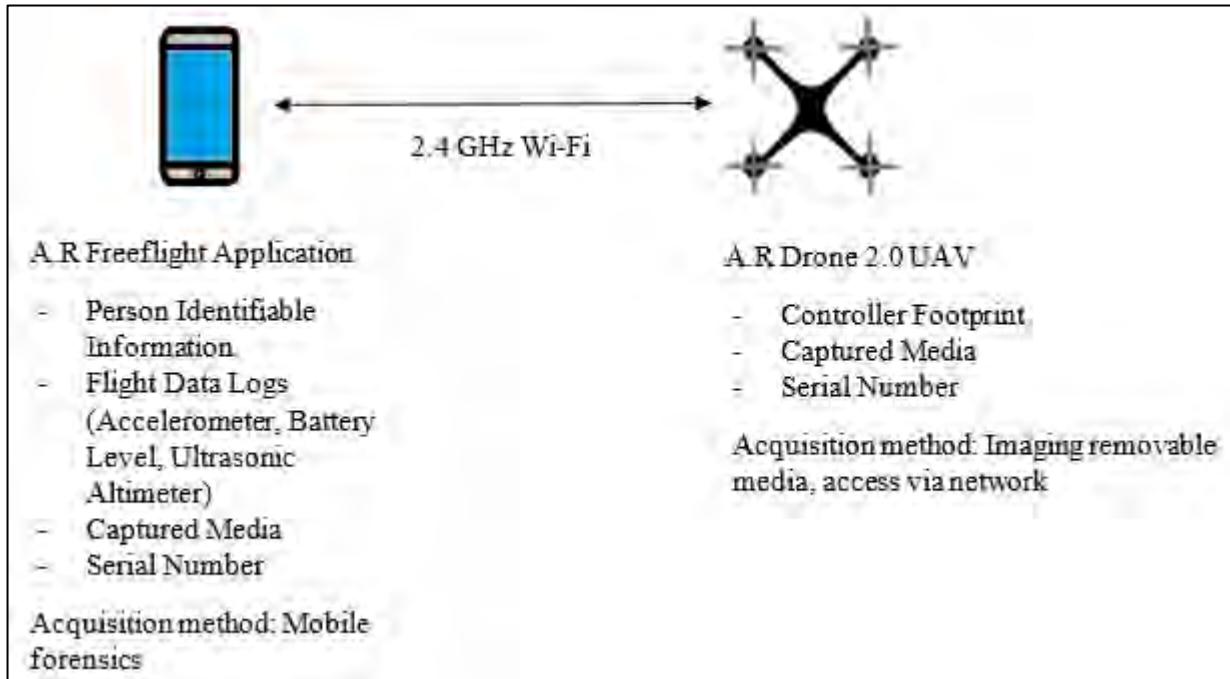


Figure 7. A.R. Drone 2.0 Power edition operation and potential artefacts

```

GNU nano 2.5.3 File: parrotAdvanced initial
Script started on Sun 26 Mar 2017 18:15:39 BST
^[]0;root@lab: ~/drones^G^[]01;31mroot@lab^[]00m:^[]01;34m~/drones^[]00m# arp -a
gateway (192.168.1.1) at 90:03:b7:92:53:38 [ether] on wlan0
^[]0;root@lab: ~/drones^G^[]01;31mroot@lab^[]00m:^[]01;34m~/drones^[]00m# nmap 192.168.1.1

Starting Nmap 7.12 ( https://nmap.org ) at 2017-03-26 18:15 BST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or speci$
Nmap scan report for 192.168.1.1
Host is up (0.0094s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
5555/tcp  open  freeciv
MAC Address: 90:03:B7:92:53:38 (Parrot)

Nmap done: 1 IP address (1 host up) scanned in 0.67 seconds
^[]0;root@lab: ~/drones^G^[]01;31mroot@lab^[]00m:^[]01;34m~/drones^[]00m# telnet 192.168.1.1
Trying 192.168.1.1...

```

Figure 8. Identification and connection to UAV

Once connected to the A.R.'s access point, querying the forensic workstations' Address Resolution Protocol (ARP) table using the command "arp -a" shows that the UAV has the local address "192.168.1.1." Running a port

scan against this address using the "nmap" (nmap) tool reveals three open ports, 21, 23 and 5555. Service scans from the "nmap" tool showed they were File Transfer Protocol (FTP) and telnet ports respectively, with port

5555 being erroneously identified as “freeciv,” an online gaming protocol. Attempting to connect to the telnet port using an anonymous username gave instant root access to the UAV’s underlying operating system. The log for this process is seen in Figure 8.

Upon connection the telnet welcome message identified as running “busybox” version 1.14.0, a compiled set of binaries that provides a number of Linux utilities and is usually deployed on embedded devices where space is a premium. Running the “uname -r” command showed the UAV was running Linux version 2.6.32., which was released in 2009 (Kernel, 2009). The “mount” command showed two partitions of interest; “/tmp/udev/dev/sda1” was identified as being the removable storage media of the UAV, mounted using the Virtual FAT (VFAT) filesystem which provides backwards compatibility with FAT devices, and the “ubi2:data” partition formatted in the Unsorted Block Image file system (UBIFS). The use of UBIFS presents a challenge for forensic imaging because it is a “raw” file system that does not support logical block addressing in the same way that file system such as FAT and NTFS do. UBIFS uses the same flash hardware as other block devices, but without hardware logical block addressing, the partition could not be imaged directly using the “dd” tool as it is based on logical blocks. A possible solution to this problem involved compiling a UBIFS block interface on the UAV. After several failed cross-compilation efforts from the forensic workstation to the UAV, an attempt was made to compile the block interface on the UAV directly. However, the lack of a suitable build environment, including a package manager, configuration tools and even compiler in the UAV system meant that this option would require extensive changes to data within the UAV and was

discontinued in favour of logical level acquisition. This was performed by mounting a forensic mass storage device to the UAV and copying files from the mounted “/data” partition using the “cp” command.

4. RESULTS

This section covers the key findings from the analysis described in Section 3. The results are broken down into three different areas of interest; the removable storage used by the UAV, the internal storage of the UAV and the results of the mobile forensic analysis on the DJI GO application in DJI Phantom 3 Professional and A.R Freeflight application in A.R Drone 2.

4.1 DJI Phantom 3 Professional

1) SD Card: As described in section 3.1, images acquired on DJI Phantom micro SD card was mounted to the forensic workstation and output from the “tree” command, shows two directories, DCIM and MISC as shown in Figure 9. The DCIM directory contains a wealth of .JPG, .DNG and .MP4 files, all of which are common media file formats. 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” (exiftool) 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 create a timestamped GPS flight log, a simple script was created, as shown in Figure 10.


```

tree
.
├── 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
├── EXIF
├── LOG
├── MISC
├── THM
├── XCODE
├── P3S_FW_RESULT_AB.txt
├── P3S_FW_V01.10.0090.bin
├── P3S_FW_LOG_AB.txt
└── 8 directories, 61 files

```

Figure 9. Output of tree command

```

GNU nano 2.5.3 File: /root/drones/d
exiftool * -c "%.6f %.6f %.6f" | egrep 'GPS Position|Create Date'

```

Figure 10. Script to retrieve GPS data from media EXIF information

```

root@lab:/mnt/analysis/DCTR/100MEDIA# ~/drones/dji/script.sh
Create Date      : 2017:04:01 14:07:30
GPS Position     : 51.000000 15.000000 28.380300 N, 0.000000 36.000000 53.406800 E
Create Date      : 2017:04:01 14:07:30
GPS Position     : 51.000000 15.000000 28.380800 N, 0.000000 36.000000 53.412300 E
Create Date      : 2017:04:01 14:07:46
Track Create Date : 2017:04:01 14:07:46
Media Create Date : 2017:04:01 14:07:46
GPS Position     : 51.000000 15.000000 28.378800 N, 0.000000 36.000000 53.391600 E
Create Date      : 2017:04:01 14:09:10
GPS Position     : 51.000000 15.000000 27.342900 N, 0.000000 36.000000 54.332000 E
Create Date      : 2017:04:01 14:09:10
GPS Position     : 51.000000 15.000000 27.347600 N, 0.000000 36.000000 54.334400 E

```

Figure 11. Sample of output from EXIF GPS extractor script

The script executed “exiftool” on all files in the directory, formatting the GPS data to 6 decimal places. The output was then filtered to only contain the ‘GPS Position’ and ‘Create Date,’ which denotes when the picture or video was taken. The output of this script is shown in Figure 11.

2) Internal Storage: The files extracted from the internal storage of the DJI Phantom were analysed using “CsvView” tool (Csv, 2017). The DJI Phantom 3 operating system began 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” tool, 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 section 3.2. A copy of this visualisation is shown in Figure 12, with each flight, waypoints 1-4 and point of interest (POI) highlighted. It is worth noting the supreme accuracy displayed by the automatic flight function in flight 3, in comparison to the other manual flights. The DJI Phantom was able to compensate for wind and other external

factors flying at a constant altitude with minimal deviance from the set path.

Because it is constantly recorded, the GPS data alone is not enough to distinguish between individual flights. The DJI Phantom flight recorder produces a host of other artefacts. Plotting these artefacts against each other using the “CsvView” tool provides a comprehensive understanding the actions taken by the drone. Figure 3.1.2.2 shows the flight

time (green), which remains constant under periods of non-activity, 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.



Figure 12. Annotated visualisation of flights 1-3

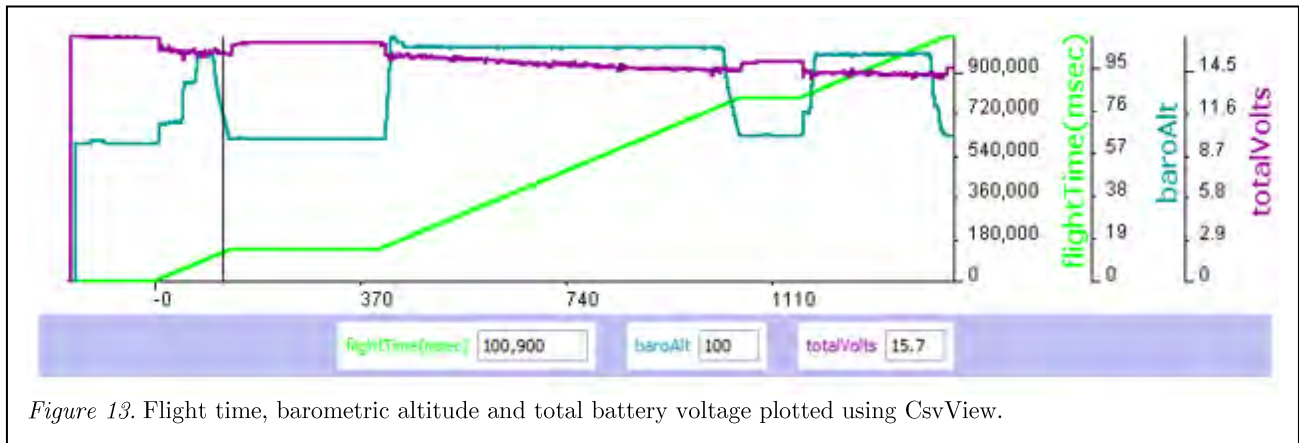


Figure 13. Flight time, barometric altitude and total battery voltage plotted using CsvView.



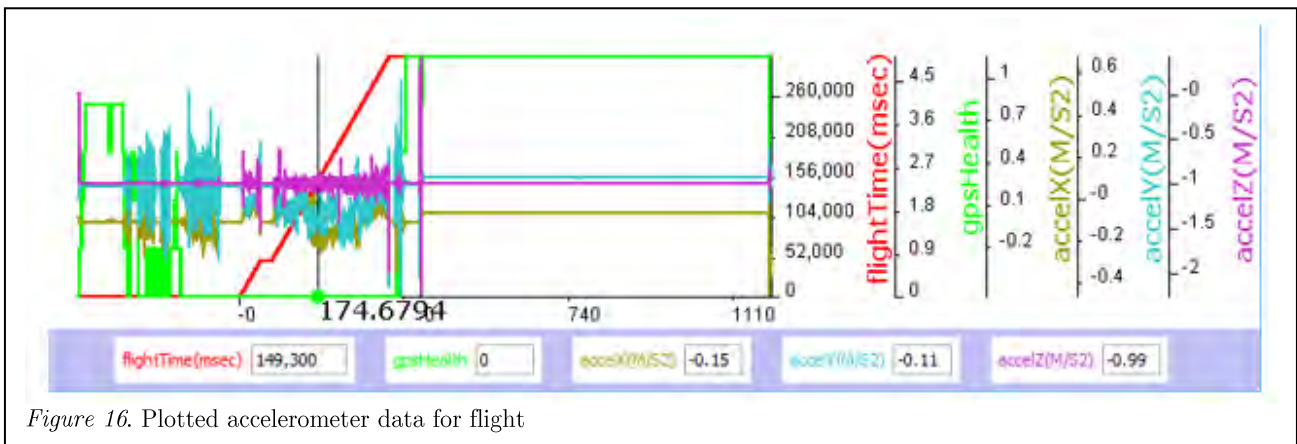
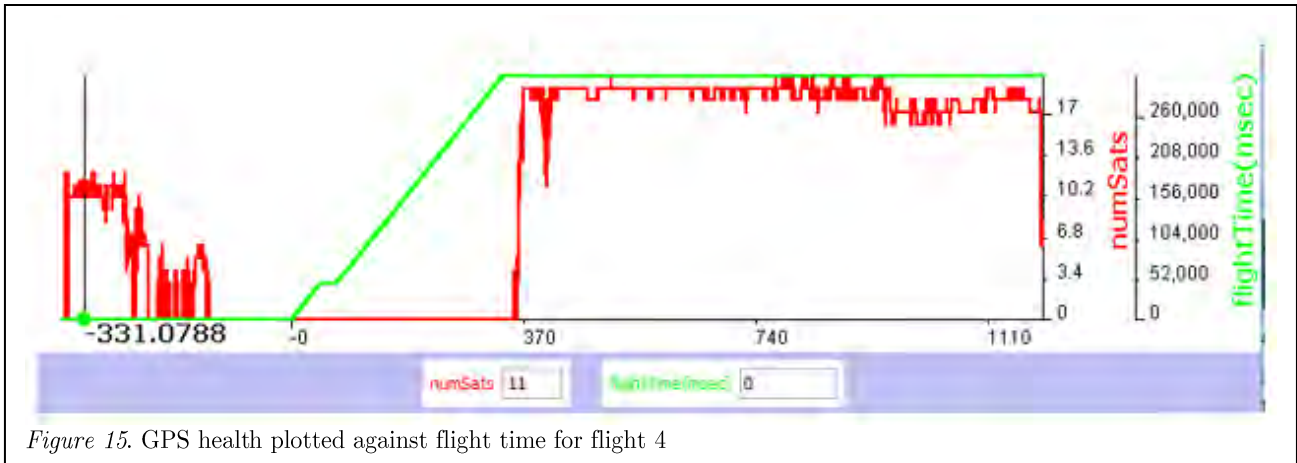
Figure 14. Garbage GPS data from flight 4

The file “FLY014.DAT” file was identified as being the log for 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 Figure 14. According to the Operator’s previous experience, the recommended amount of GPS signals was about 11, but with the foil obstructing the unit, the DJI 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 Figure 15. The foil was removed after the flight due to fears of overheating the drone through obstruction of the cooling vents.

This confirms findings from related work (Maarse et al., 2016) 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/second². Figure 16 shows these readings when plotted against the flight data shown in the previous figure.

These measurements can be used to reconstruct a flight in 3D space, relative to an arbitrary home point. While it would be possible to perform this analysis manually, the frequency of measurements taken by the Phantom make it unreasonable, and it would be better to develop a tool to do this automatically.



3) DJI GO Application: Artefacts from the DJI GO application were located in different locations within the “userdata” partition of the

android test platform, which was acquired using methods described in section 3.1. A list of these useful directories is shown in table 5.

Table 5
List of useful directories from 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 containing information on satellite data not being available.
/media/0/DJI/dji.pilot/DJI_RECORD	Media	A number of videos 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. This contains GPS data, manufacturing information and capture dates.
/media/0/DJI/dji.pilot/Flight Record	Flight Data, Person Identifying Information, UAV Serial Number	This directory contains flight data relating to a number of flights. A string search of these files 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 flight record files extracted from the "FlightRecord" directory were analysed using the "CsvView" 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 1Kb and 1Mb, 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 Figure 17.

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

Figure 17. Metadata from DJI GO application flight log

As well as this metadata, several other streams of flight data relating to use of the DJI GO application were available. The “flyCState” attribute described whether the Phantom was in manual or automatic mode. Figure18 shows the distance of the UAV from the home point plotted against the “flyCState” attribute during flight 3. The automatic POI function generated a clearly visible sine wave in the distance measurements during the time when the UAV was in automatic flight mode. This may be a useful artefact in identifying when the POI function has been used.

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.

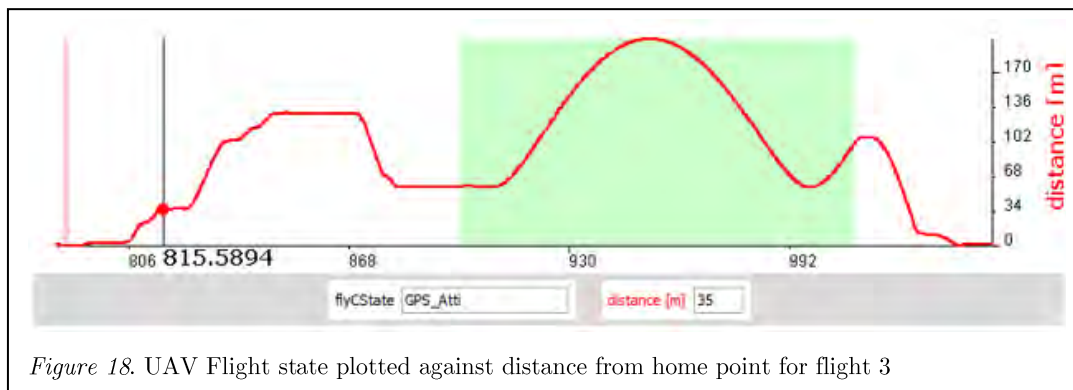


Figure 18. UAV Flight state plotted against distance from home point for flight 3

4.2 A. R Drone 2.0

1) Internal Storage: The mass storage device was mounted to the forensic workstation for examination of the files acquired from the

UAV. A number of files of interest were located, as listed in Table 6.

Table 6
List of files acquired from A.R Drone 2.0 Internal Storage

Path	Type	Description
/data/syslog.bin	System log, containing details of various software and hardware events from the UAV's internal operating system.	Version information, configuration data, mount information, file creation logs
/data/config.ini	Configuration file for the UAV.	Drone serial number, software version, drone name, access point SSID
/data/emergency.bin	Unidentified binary file. Further work should identify the importance of this file and it's cybersecurity implications.	n/a
/data/custom.configs/sessions/	Directory containing several files named "config.xxxxxxxx.ini"	GPS data. The UAV does not have a GPS sensor installed so it likely originated from the A.R Freeflight application.
/data/custom.configs/profiles/	Directory containing a file named "config.xxxxxxxx.ini."	Contains a footprint from the controlling application with name of the mobile platform, "Mororola_MotoG3" and a serial number – "PS721003AJ4K103341."

The amount of data present in the system log located at "/data/syslog.bin" meant scripts were used to analyse threads of output. For example, the A.R drone's operating system uses a set of processes to access the external storage with the prefix "UsbKey," including "UsbKeyMonitor," "UsbKeyWriter" and "UsbKeyRepairer." The logs from these methods were retrieved using the command

"cat syslog.bin | grep UsbKey" and then further filtered. "UsbKeyMonitor" prints the serial number when a new USB device is attached, so filtering using the word "Serial" produced a history of all the USB keys attached to the UAV, as shown in Figure 19. Other information such as the vendor and product ID was also available through this method.

```
root@lab:~/drones/parrot/acquisition# cat syslog.bin | grep "UsbKey" | grep "Serial"
2.599151 UsbKeyMonitor 6 905 USB Mass Storage Serial = '076511B10BAC'
2.461425 UsbKeyMonitor 6 912 USB Mass Storage Serial = '20020501A5BCF703'
2.464935 UsbKeyMonitor 6 915 USB Mass Storage Serial = '20020501A5BCF703'
2.690795 UsbKeyMonitor 6 918 USB Mass Storage Serial = '20020501A5BCF703'
2.463745 UsbKeyMonitor 6 914 USB Mass Storage Serial = '20020501A5BCF703'
2.451904 UsbKeyMonitor 6 914 USB Mass Storage Serial = '20020501A5BCF703'
2.657562 UsbKeyMonitor 6 910 USB Mass Storage Serial = '0000177BE961C012'
2.453735 UsbKeyMonitor 6 898 USB Mass Storage Serial = '078A01110998'
root@lab:~/drones/parrot/acquisition#
```

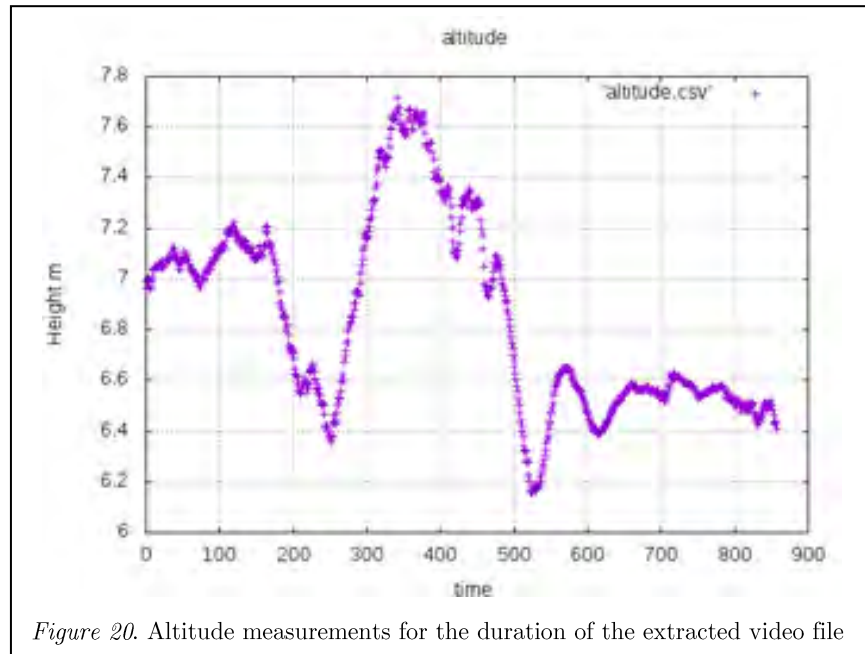
Figure 19. USB serial number history of UAV

Filtering the system log for all "UsbKeyWriter" outputs gave a history of all files created on the removable storage, with system times in the Linux log epoch format of seconds. Filtering for the "Video" outputs gives

a log of the use of both the UAV's internal cameras, which video codec is being used and other details including resolution. Examination of the "syslog.bin" file give a comprehensive overview of actions carried out by the UAV's

operating system. Values found also reflected values in the “config.ini” files examined in Table 6. Future work should identify whether modification of the “config.ini” files would change data in the system log, for anti-forensics purposes.

2) Removable Storage: Based on artefacts from the system logs, the removable storage media only serves the purpose of storing media files captured from the UAV. A list of all files in the partition using the “tree” command confirms this to be the case.



All the files were videos, stored with the “mp4” file extension. Photos were taken during the flight; however, none were present on any of the UAV’s storage media. This is likely due to the photos being stored on the mobile device, and videos being too large to transmit over the network. Examination of the video files using “exiftool” revealed a number of artefacts, including creation time, the device name “Parrot AR.Drone.” The “ARDroneTelemetry” tag was extracted from one of the videos using the “-b” option. This appeared as a set of floating point numbers and integers, with no labels or column headers. Using heuristics based on knowledge of the phantom system and known flight data, it was deduced that the first of the floating points was a timestamp, as it increased in regular

increments. Also deduced was that the last floating point was the altitude of the UAV during flight, as the values steadily changed, and matched the approximate value of the flight. The telemetry data was dumped to a file for analysis with the command “exiftool -b -ARDroneTelemetry media20170401_150213/video_20170401_150249.mp4>~/drones/parrot/gnuplot/telemetry.” A basic script was created to convert the data to a comma-separated value file, which could then be visualised using the “gnuplot” tool for Linux (gnuplot, 2017). The altitude was plotted over the period of the whole video, as shown in Figure 20. Further work should be carried out to confirm these hypotheses regarding the telemetry data for the A.R Drone 2.0.

3) A.R Freeflight Application: The “userdata/ data/com.parrot.freeflight” directory contained several “.xml” files, with names in the format of “<MAC Address of mobile platform>_ <Timestamp>.” These appear to correlate with sessions of activity on the UAV. Each file contains a number of flight and application session records, with each XML (eXtensible Markup Language) block being named accordingly. The “FLIGHT_DRONE_SERIAL” tag displays a matching serial number to the one listed in section 2, meaning these artefacts both exist on the UAV and the mobile platform and can be used to connect the two during an investigation. Another XML file, located in “userdata/com.parrot.freeflight/shared _prefs/Preferences.xml” held a number of important artefacts, including the GPS coordinates of the last flight, the email address of the google account used to download the application, and when the application was last opened. This is the first artefact found that directly links the use of a drone to a user account, which can later be used to trace a user.

The A.R Freeflight application has a media storage location in the platform’s “userdata/media/0/DCIM” (Digital Camera Image) directory, which contains all the media captured by the UAV’s cameras. EXIF data for these files varies, some containing GPS information which matches the operator location during the flight, and some only containing a few details such as the creation date – it is unclear whether this is due to file corruption. Again, all GPS readings most likely originated from the mobile platform as the UAV does not natively possess a GPS capability.

5. CONCLUSION

Two multi-platform UAV systems – the DJI Phantom 3 Professional and Parrot A.R Drone

2.0 Power edition, have been investigated in this paper. Although, there is operational difference between the drones, a number of common methods were utilised to recover data from drones and controlling devices using open source tools. In comparison to the A.R, the DJI phantom was found to have an extraordinarily large number of artefacts associated with it, because having more sensors and a higher resolution of data capture, which comes from its status as a professional device. In both cases, it was necessary to interpret flight data collected by the UAVs. At its most simple, this involved interpreting the movements of the UAVs in three-dimensional space. This was simpler with the DJI as it records GPS automatically and work had already been done in that area to develop a tool for easy interpretation of the flight data. Most of the potential artefacts listed in Section 3 were found on the UAVs or their controlling systems, and were successfully extracted to identify a suspect, recreate a flight and capture media from the devices. Some anti-forensics methods were also successfully tested. Future work should look at the automation of drone forensics, and explore methods discussed in this paper to other drone models such as DJI Phantom 4 Pro, DJI Mavic Pro or Parrot Bebop and different mobile platforms such as Windows and iOS and especially, integration of the methods discussed here into commercial forensic tool-kits.

REFERENCES

- ACPO. (2012). ACPO Good Practice Guide ACPO Good Practice Guide for Digital Evidence for Digital Evidence.
- Android. (2017). Android Debug Bridge (adb) | Android Studio. Retrieved 24 March 2018, from <https://developer.android.com/studio/command-line/adb.html>
- Azhar, M. A. H. Bin, & Barton, T. E. A. (2016). Forensic Analysis of Secure Ephemeral Messaging Applications on Android Platforms (pp. 27–41). Springer, Cham. https://doi.org/10.1007/978-3-319-51064-4_3
- Barrett, D. (2015). Burglars use drone helicopters to target homes - Telegraph. Retrieved 23 March 2018, from <https://www.telegraph.co.uk/news/uknews/crime/11613568/Burglars-use-drone-helicopters-to-identify-targe-homes.html>
- Barton, T. E. A., & Azhar, M. A. H. Bin. (2018). Open Source Forensics for a Multiplatform Drone System (pp. 83–96). Springer, Cham. https://doi.org/10.1007/978-3-319-73697-6_6
- BBC. (2016). Big rise in drone jail smuggling incidents - BBC News. Retrieved 23 March 2018, from <http://www.bbc.co.uk/news/uk-35641453>
- CAA. (2015). Airspace restrictions for unmanned aircraft and drones | UK Civil Aviation Authority. Retrieved 23 March 2018, from <https://www.caa.co.uk/Consumers/Unmanned-aircraft/Our-role/Airspace-restrictions-for-unmanned-aircraft-and-drones/>
- Carrier, B. (2002). Open Source Digital Forensics Tools: The Legal Argument.
- Csv. (2017). CsvView Downloads. Retrieved 24 March 2018, from <https://datfile.net/CsvView/downloads.html>
- CyanogenMod. (2017). CyanogenMod android operating system. Retrieved 24 March 2018, from <https://github.com/CyanogenMod>
- Dinan, S. (2017). Mexican drug cartels using drones to smuggle heroin, meth, cocaine into U.S. - Washington Times. Retrieved 23 March 2018, from <https://www.washingtontimes.com/news/2017/aug/20/mexican-drug-cartels-using-drones-to-smuggle-heroi/>
- DJI. (2018). Phantom 3 Professional - Specs, FAQ, Tutorials, Downloads and DJI GO - DJI. Retrieved 23 March 2018, from <https://www.dji.com/phantom-3-pro/info#specs>
- Gartner. (2018). Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017. Retrieved 24 March 2018, from <https://www.gartner.com/newsroom/id/3859963>
- gnuplot. (2017). gnuplot tool. Retrieved 24 March 2018, from <http://www.gnuplot.info/download.html>
- Hambling, D. (2017). Islamic State Now Using Off-the-Shelf Drones | Defense content from Aviation Week. Retrieved 23 March 2018, from <http://aviationweek.com/defense/islamic-state-s-new-weapon-choice-shelf-drones>
- Horsman, G. (2016). Unmanned aerial vehicles: A preliminary analysis of forensic challenges. *Digital Investigation*, 16, 1–11. <https://doi.org/10.1016/J.DIIN.2015.11.002>

- Jain, U., Rogers, M., & Matson, E. T. (2017). Drone forensic framework: Sensor and data identification and verification. In *2017 IEEE Sensors Applications Symposium (SAS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/SAS.2017.7894059>
- Karlsson, K.-J., & Glisson, W. B. (2014). Android Anti-forensics: Modifying CyanogenMod. In *2014 47th Hawaii International Conference on System Sciences* (pp. 4828–4837). IEEE. <https://doi.org/10.1109/HICSS.2014.593>
- Kernel. (2009). Linux_2_6_32 - Linux Kernel Newbies. Retrieved 24 March 2018, from https://kernelnewbies.org/Linux_2_6_32
- KingoApp. (2017). KingoRoot APK, the Best One-Click Android Root app for free. Retrieved 24 March 2018, from <https://root-apk.kingoapp.com/>
- Kovar, D., Dominguez, G., & Murphy, C. (2015). UAV (aka drone) Forensics, 7.
- Lee, C., Sim, D., Hwang, J.-Y., Cho, S., Development, S. / W., & Business, T. M. (2015). F2FS: A New File System for Flash Storage.
- Luo, A. (2016). Drones Hijacking. *Defcon*.
- Maarse, M., Sangers, L., Ginkel, J. Van, & Pouw, M. (2016). *Digital forensics on a DJI Phantom 2 Vision+ UAV*.
- Majendie, A., & Chia, K. (2018). The Future of Flying Is All About Drones - Bloomberg. Retrieved 16 March 2018, from <https://www.bloomberg.com/news/articles/2018-02-08/in-the-global-game-of-hide-and-peek-the-drones-are-winning>
- Mikelionis, L. (2018). Drug cartels using drones to smuggle drugs at border. *Fox News*.
- Moskwa, W. (2016). World Drone Market Seen Nearing \$127 Billion in 2020, PwC Says - Bloomberg. Retrieved 16 March 2018, from <https://www.bloomberg.com/news/articles/2016-05-09/world-drone-market-seen-nearing-127-billion-in-2020-pwc-says>
- Parrot. (2017). Quadcopter AR Drone 2.0 Power Edition | Parrot Store Official. Retrieved 23 March 2018, from <https://www.parrot.com/uk/drones/parrot-ardrone-20-power-edition/#parrot-ardrone-20-power-edition-details>
- Pleban, J.-S., Band, R., & Creutzburg, R. (2014). Hacking and securing the AR.Drone 2.0 quadcopter: Investigations for improving the security of a toy. In R. Creutzburg & D. Akopian (Eds.) (Vol. 9030, p. 90300L). International Society for Optics and Photonics. <https://doi.org/10.1117/12.2044868>
- Siddique, H. (2017). Drone complaints soar as concerns grow over snooping | Technology | The Guardian. Retrieved 23 March 2018, from <https://www.theguardian.com/technology/2017/apr/03/drone-complaints-soar-as-concerns-grow-over-snooping>
- Trujano, F., Chan, B., Beams, G., & Rivera, R. (2016). Security Analysis of DJI Phantom 3 Standard.
- UAV. (2018). Tarot T-18 Ready To Fly Drone | UAV Systems International. Retrieved 23 March 2018, from <https://www.uavsystemsinternational.com/product/tarot-t-18-ready-fly-drone/>
- Valentak, Z. (2017). Drone market share analysis & predictions for 2018 - DJI dominates, Parrot and Yuneec slowly catching up - DronesGlobe.com. Retrieved 23 March 2018, from <http://www.dronesglobe.com/news/drone-market-share-analysis-predictions-2018/>
- Waters, N. (2017). bellingcat - Death From Above: The Drone Bombs of the Caliphate - bellingcat. Retrieved 23 March 2018, from

<https://www.bellingcat.com/news/mena/2017/02/10/death-drone-bombs-caliphate/>

Yeung, P. (2016). Drone reports to UK police soar 352% in a year amid urgent calls for regulation | The Independent. Retrieved 16 March 2018, from <http://www.independent.co.uk/news/uk/home-news/drones-police-crime-reports-uk-england-safety-surveillance-a7155076.html>

Zanero, S., & Huebner, E. (2010). The Case for Open Source Software in Digital Forensics. In *Open Source Software for Digital Forensics* (pp. 3-7). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-5803-7_1