GRYPHON: Drone Forensics in Dataflash and Telemetry Logs

Evangelos Mantas and Constantinos Patsakis

Department of Informatics,University of Piraeus

Abstract. The continuous decrease in the price of Unmanned Aerial Vehicles (UAVs), more commonly known as drones, has pushed their adoption from military-oriented to a wide range of civilian and business applications. Nevertheless, the many features that they offer have started being maliciously exploited. The latter coupled with the fact that accidents or malicious acts may occur to drones has sparked the interest towards drones forensics.
Trying to fill in the gap of the literature, this work focuses on a particular field of drone forensics that of forensics on the flight data logs. Therefore, we investigate one of the most widely used platforms, `Ardupilot` and the dataflash and telemetry logs. In this work, we discuss a methodology for collecting the necessary information, analysing it, and constructing the corresponding timeline. In this regard, we have developed an open source tool that is freely available and tested it on data provided by VTO Labs.

Keywords: UAV, Drone, Forensics, Ardupilot, Log files

1 Introduction

The rise of technology in the last decade contributed to the development of new concepts that may change the world. Unmanned Aerial Vehicle (UAV) refers to any reusable air vehicle that does not have a pilot on board. These vehicles are already changing the modern day society, from the battlefield to everyday commercial use. UAVs, or simply drones, are currently used for military applications, engaging targets, collecting intelligence or used for surveillance assisting the ground troops. Commercial applications are more challenging since they fall under government regulations and require special operation licensing. Infrastructure inspection, package delivery, crop dusting, first aid, emergency response, and civilian transportation were just concepts in the past which are now becoming a reality.

The use of drones in civilian and business applications is steadily increasing. Globally, drone market volume is forecast to reach 4.7 million[1] units by

[1] `https://www.businesswire.com/news/home/20160509005554/en/Unmanned-Aerial-Vehicles-UAV-Market-Forecast-2020`

2020 (other estimates are even more optimistic), with the market for commercial application of UAV technology estimated to soar from \$2bn to \$127bn[2]. This growth projects that over time drones are becoming cheaper and easier to use, as well as regulatory progress. Therefore, more and more drones will be flying in public airspace. The trend is shifting from quadcopters with simple flight boards, to machines with computers that run Operating Systems (OS) and execute complex tasks. As a result, a vulnerability in these systems can be exploited with fatal consequences. Since the flight in densely populated areas implies several risks directly [24,4] or indirectly [22], public safety must be taken into consideration . Even if an accident happens or a drone "falls to the wrong hands", it must be ensured that the responsible for this action will be identified and held accountable. Thus, a forensic analysis of the drone flight is crucial.

It should be highlighted that regardless of how far-fetched UAV criminality may sound, it has constantly been rising leading us to consider far more scenarios than the typical military-oriented. Unfortunately, the list of crimes committed through the use of drones is increasing, not only in terms of numbers but in terms of sophistication. The recent three-day confusion at Gatwick airport due to an unauthorised flight of drones affected almost 140,000 passengers of 1,000 flights[3] in fear of public safety. While this can be considered as the most significant, as it affected the most people and due to the delays it had a huge cost there was no physical harm. Beyond "traditional" privacy violations, drones have been used to smuggle drugs in prisons[4] and across borders[5], spy on houses for burglary[6], record PINs on ATMs[7]. Finally, Hartmann and Giles [11] discuss how exploitation of UAVs can be used to realise cyber power with effect in the real world.

1.1 Scope of this work

The continuous growth of the autonomous flying vehicles led to many different flight firmware. One of the first was `Ardupilot`[8], initially developed by amateur hobbyists to control their RC model airplanes. It soon grew to an open source software suite, widely adopted by the airspace industry and used by many enterprises because of its flexibility to support different types of autonomous vehicles

[2] `https://press.pwc.com/News-releases/global-market-for-commercial-applications-of-drone-technology-valued-at-over--127-bn/s/ac04349e-c40d-4767-9f92-a4d219860cd2`

[3] `https://www.nytimes.com/2018/12/23/world/europe/gatwick-airport-drones.html`

[4] `https://www.bbc.com/news/uk-england-43413134`

[5] `https://www.rt.com/news/225051-drone-meth-crash-tijuana/`

[6] `https://www.telegraph.co.uk/news/uknews/crime/11613568/Burglars-use-drone-helicopters-to-identify-targe-homes.html`

[7] `https://www.belfasttelegraph.co.uk/news/northern-ireland/drone-filmed-peoples-pin-codes-at-co-antrim-atm-34945847.html`

[8] `http://ardupilot.org/`

making it mainstream in the field. Drone companies like DJI may hold the majority of the market share [1] but the fact that their proprietary software is not openly available and offers limited flexibility for developers through its SDK, makes it less appealing for enterprises that seek customization, high flexibility and robustness in their flying vehicles autopilot firmware, in contrast to the civilian market where consumers opt for a ready-to-fly quadcopter[9].

This work focuses on drone forensics performed on the ground control station of the UAV under investigation, and the logfiles found on the internal memory. More precisely, the research is focused on the investigation of two specific logs for forensic evidence, the dataflash and telemetry logs.

It is clear, that by capturing a drone one may collect a wide range of forensic evidence, from the serial number of the frame of proprietary drones leading to the person who made the purchase, to fingerprints of the pilot or other ground crew. Indeed, a drone might have other connected devices that store data in an SD card and can be used as forensic evidence. A typical example can be considered a camera. Apparently, a camera and the stored data may be a rich source of forensic evidence since the recorded footage might contain not only visual information but metadata in the form of EXIF data [5]. Nevertheless, the above scenario is considered as out of scope and we only focus on forensic's data from the onboard SD card and the ground control station of a drone with the "minimum" setup. Figure 1 illustrates the different aspects that can be used for drone forensics of `Ardupilot` compliant UAVs.

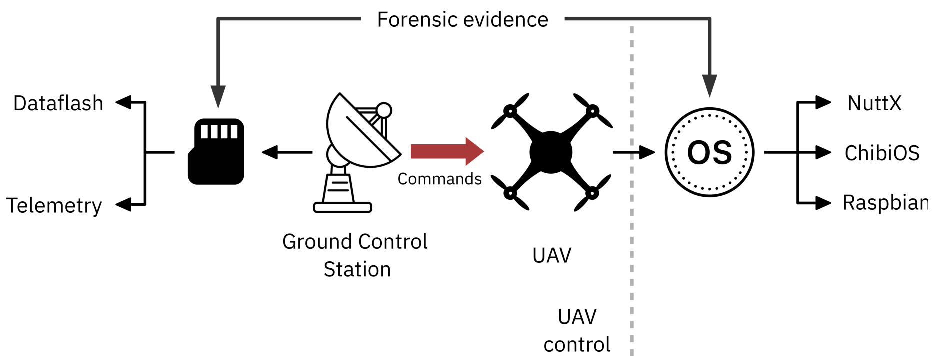


Fig. 1: Sources of forensic evidence of UAVs compliant with `Ardupilot`.

1.2 Main contributions

To the best of our knowledge, there is currently no forensics tool for Dataflash log available, nor a detailed methodology on what data need to be collected and

[9] `https://globaluavtech.com/news-media/blog/open-source-ardupilot-software-vs-dji-software/`

how to treat them. To this end, a new tool was created and is available available as an Open Source Project on Github[10]. Existing solutions such as `MAVExplorer` mainly focus on displaying data with no flexibility, serving more like diagnostic tools. Since the underlying technology is not stable enough to guarantee that hard changes will not be pushed in the new versions, our forensics tool considers other aspects such as maintainability, flexibility, and sustainability. Therefore, a future update of the `MAVlink` protocol would have a limited impact in the code compatibility and additions can improve the efficiency and functionality of the forensic research with minimal changes.

1.3 Organisation of this work

The rest of this work is organised as follows. In the next section, we give a brief overview of the related work in drone forensics. Section 3 describes of the UAV architecture, discussing some specific details regarding the `Ardupilot` logs. Then, in Section 4 we detail our methodology for drone forensics in dataflash and telemetry logs. Finally, the paper concludes, summarising our contributions and discussing open issues and future work.

2 State of the art

As already discussed, drone forensics can be performed either in the drone or its corresponding ground control station. Due to the fragmentation of the market, the UAVs come in two main families. One running proprietary firmware and one running open-source. The proprietary firmware share is dominated by companies such as `DJI`, `Yuneec`, `Parrot` and `3DR` with its proprietary frame and open source autopilot firmware. In the open source share, the vast majority is using `Ardupilot` and the operating system on the UAV may come in different flavours, such as `Raspbian`[11], `ChibiOS`[12], and `NuttX`[13].

Kim et al. [17] were the first ones to provide an overview of the attack surface of a UAV. An initial risk assessment of UAVs was performed by Hartmann and Steup [12] and later extended by Hartmann and Giles [11]. Due to their nature, UAVs may suffer many networks attacks for which there is already some work on trying to detect them [21], simulate them [16], and reinforce their security while in flight [20,13,8,3,2]. The main reason behind this is that drones have low processing power, so the use of cryptographic primitives implies a significant computational effort. In general, the threats UAVs are exposed to include but are not limited to jamming, interception, and spoofing of the GPS and communications, denial of service, open ports that allow for arbitrary access, injection

[10] `http://github.com/emantas/GRYPHON_dft`

[11] `http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html`

[12] `http://ardupilot.org/copter/docs/common-loading-chibios-firmware-onto-pixhawk.html`

[13] `http://ardupilot.org/dev/docs/interfacing-with-pixhawk-using-the-nsh.html`

of forged sensor data [19,23]. For a more detailed overview of these threats for civilian UAVs, the interested reader may refer to [4].

As outlined in [25] IoT forensics imply several new challenges for the forensics investigator as she has to search for evidence in diverse data formats in devices with different firmware, often proprietary, where physical access to the storage and processing units can be concealed or access-protected. Moreover, digital evidence has a short survival period and many of them reside in the cloud. Bouafif et al. [9] highlight other issues including non-standardization of firmware, hardware, and software, loss of evidence stored in volatile memory due to crash or battery failure, lack of attribution of ownership, and lack of existence of forensics tools or support from mainstream tools.

In the literature, some papers are focusing on data exfiltration from `DJI` [10,6,25] and `Parrot` [14,7,9] drones, but the proposed procedures apply only to specific market models. This paper focuses on a broader spectrum of flying vehicles proposing a forensic methodology applied not only for a civilian hobbyist but also powerful commercial drones. Since the application of drones in everyday life is expected to be expanded, the forensic approach should not be focused on proprietary software, but shifted to open source flight stacks like `Ardupilot`. This means it is cheaper to acquire or build an open source flying vehicle, making harder to track since there is no financial evidence of a purchase, in contrast to ready-to-fly proprietary drones, or even programming complex flight missions like package deliveries, in the event of an accident or a dreadful scenario.

Other papers focus on technical data, the architecture of components of drones and physical acquisition of the hardware [15]. Although a hardware analysis of the flying vehicles is essential, no hard evidence of malicious behaviour can be extracted, e.g. flying on a no-fly zone. This evidence falls into the scope of software forensics, a scope that this research covers providing a tool to assist the forensic researcher.

There were tries to include the forensic data of open source autopilots, using the telemetry logs [18] provided by the GCS connected to the drone and extracting the flying vehicle's autopilot parameters, but no other data were recovered to perform a thorough forensic examination. However, as it is clearly stated "*GCS and post processing systems, analyzing the sensor data tell 80% of the story*". This should be the researcher's main focus, exfiltrating the flight data from the log files and if possible acquiring the data from the mobile device's GCS, having the full picture in constructing the forensic's case.

3 Drone Basics

At the very core of a drone is its *flight controller* which is in charge of its avionics, communications, sensor management and other actuators. Due to the complexity of the tasks that have to be performed, the flight controller has changed from simple micro-controllers to small computers, running on a real-time operating system, using a limited processing unit. Depending on the needs, the communication of the drone can be performed over short-range channels including WiFi and Bluetooth, or long-range including radio waves and satellite transmissions.

Before going further, it is essential to distinguish the flight of a UAV from, e.g. a hobbyist's quadcopter. The main difference is the onboard computer, the autopilot. The autopilot manages and orchestrates all the flight components, receives the pilot input from a radio transmitter, and is capable of executing complex commands or taking into action the fail-safe commands automatically, without any manual intervention. For example, a drone can return to the takeoff point, if the autopilot detects that battery level is running low. In this scenario, the drone did not receive any command from a human; the autopilot took control of the flying vehicle. On the contrary, a quadcopter with a flight board receives only the signal from the pilot's radio transmitter. It has limited flight capabilities that depend on the pilot skills. It should be noted that there is a hybrid approach in which the UAV carries out its mission with an operator supervising the progress and intervening when deemed necessary.

Forensic analysis on non-automated vehicles is almost impossible since no events are recorded beyond, e.g. the onboard camera, if one is installed. On the other hand, during the flight of a drone with `Ardupilot` autopilot, two different types of flight logs are generated, the *Dataflash logs* and the *Telemetry logs*. Dataflash logs are generated from the autopilot and are stored on the onboard flash memory, usually an SD card. Telemetry logs are generated from the Ground Control Station (GCS), software to monitor and command the drone, should a connection between the flying vehicle and the GCS be established[14].

When a drone is ready to fly, the pilot should "arm" the system. This command can be given either through a radio transmitter or through a GCS (e.g. Mission Planner, QGroundControl). This command is the "trigger" to generate the Dataflash log. This type of log contains everything that happens during the flight of the drone. More precisely, it contains the values of the RC (Radio Channel) input when the pilot flies manually the drone, the GPS location, values of the onboard sensors, commands sent from the GCS and other data. When the autopilot is connected to a GCS, a Telemetry log is generated and records the flight path, which can be replayed using the Mission Planner GCS.

The autopilot communicates with all the flight components of the drone and transmitters, through the MAVlink (Micro Air Vehicle) Communication Protocol. A MAVlink network is made up of systems (vehicles, GCS, antenna trackers etc.) which are themselves made up of components (Autopilot, camera system, etc.). The protocol defines two IDs that can be specified in messages to control routing of the command to the necessary system and component: the system which will execute the command, called the target system and the component which will execute the command called the target component [15]. MAVLink follows a modern hybrid publish-subscribe and point-to-point design pattern. Data streams are published as topics while configuration sub-protocols such as the

[14] `http://ardupilot.org/copter/docs/common-diagnosing-problems-using-logs.html`
[15] `https://mavlink.io/en/`

mission protocol or parameter protocol are point-to-point with retransmission. These messages are recorded by the autopilot in the Dataflash and Telemetry logs. A proper understanding of this protocol architecture is required to extract the flight data and analyse the recorded events. A MAVLink message is comprised of the following features[16]:

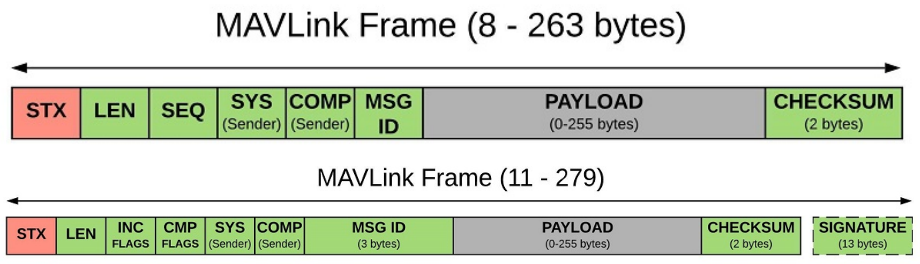


Fig. 2: MAVlink packet architecture. Top: MAVlink v1, Bottom: MAVlink v2. Source: `https://mavlink.io/en/guide/serialization.html`

– A 24-bit message ID - Allows over 16 million unique message definitions in a dialect (MAVLink 1 was limited to 256)
– Packet signing - Authenticate that messages were sent by trusted systems.
– Message extensions - Add new fields to existing MAVLink message definitions without breaking binary compatibility for receivers that have not updated.

Knowing the structure of each message facilitates the task of a forensic researcher in extracting the necessary data and diagnosing the critical events of the UAV flight.

4 Dataflash and Telemetry Forensics

In what follows we consider that the forensics investigator had access to the ground base station and collected both the telemetry and dataflash logs in a forensics sound manner. Therefore, we discuss how the collected information should be processed and what kind of evidence can be extracted.

4.1 Logfile Acquisition

It is clear that a forensic analysis of a UAV flight requires the generated log files. Although enterprises may come with their own custom logfile genesis solutions for their flying vehicles, the generation of Dataflash and Telemetry logs stays the same.

[16] `https://mavlink.io/en/guide/mavlink_2.html`

Telemetry logs, as mentioned before, are automatically generated from the GCS in the computer that is connected to the drone. This type of log file is saved in *.kmz* format and contain the flight path of the vehicle which is replayed, using the GCS built-in functionality.

Dataflash logs can be downloaded through a GCS using its built-in functionality or SSH connection to the flying vehicle. The autopilot automatically creates a log file in *.bin* format, after the pilot has armed the drone, saved in the SD card mounted on the autopilot dedicated slot.

It is easily understandable that in the event of a catastrophic failure, e.g. crash-landing, components of the flying vehicle may be damaged or destroyed, dropping the connection between the drone and the GCS. Thus, no log files can be downloaded, and the incidence response will suffice in acquiring the SD card of the autopilot. It is crucial that all the necessary precautions are taken into account in order not to "disturb" the forensic chain of command. This means that an incident response officer should follow all the procedures in ensuring that no key component has been alerted in any way during the acquisition - intentionally or not. For this reason, cloning of the SD card is deemed essential. A further study in the UAV forensic acquisition seems to be decisive in establishing a formal procedure, but this is out of the scope of this research.

4.2 Dataflash log analysis methodology

This research methodology for analysing the Dataflash logs consists of 6 steps. Their scope starts from firmware integrity and goes through trajectory, execution, and error analysis to reach low-level hardware logs and finish with timeline analysis. More precisely, the steps of our methodology are the following:

1. **Check integrity of the UAV:** The goal of this step is to determine whether the firmware running on the drone has been tampered with.
2. **Trajectory analysis:** By visualising the trajectory of the UAV one may determine possible differentiation in the course and decrease the timeline that has to be analysed. For example, one may notice that the trajectory of the UAV is not the expected one after a specific timeframe. Moreover, by detecting anomalies in the trajectories, one may gain further insight on an incident. For instance, a sudden variation in the height of the UAV may imply collision with an object.
3. **Command verification:** The goal of this step is to determine whether all the commands that were submitted by the pilot have been executed.
4. **Error analysis:** In this step, all errors reported by MAVlink are collected to determine whether any of the reported errors resulted in fatal errors or warnings that affected the estimated flight capability.
5. **Analysis of sensors measurements:** The goal of this step is to determine whether all the measurements from the embedded sensors can be considered within the expected range. Anomalies in the sensor measurements may imply a hardware problem in the UAV. Note that such issues may not trigger errors.

6. **Timeline analysis:** A Timeline Analysis is the process of chronologically arranging data of the flight, a crucial part of any digital forensics examination. This part enables the forensic investigator to correlate the found evidence and understand what has happened in the case under investigation.

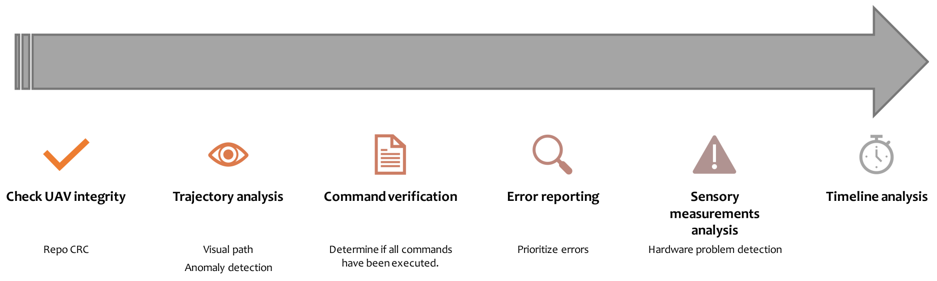


Fig. 3: Brief overview of the proposed methodology.

Based on the above, `GRYPHON Drone Forensics Tool` provides the following functionality.

– **Extraction of flight data to find anomalies:** The main functionality of `GRYPHON` is data exfiltration. Since the Dataflash log is in binary format, the recorded flight events must be converted to the corresponding MAVlink format. Multiple independent functions extract and display the data, giving the forensic investigator the ability to monitor the UAV flight behaviour. Each event is displayed with the corresponding timestamp and the value of the MAVlink message recorded. To facilitate reading, we opted for the use of colour coding. Therefore, error information appears in red, whereas warning or medium importance data appear in orange.

– **GPS coordinates mapping:** During the flight, the location of the drone is transmitted using the GPS MAVlink message, which is recorded in the Dataflash logfile. `GRYPHON` extracts the timestamp, GPS signal strength (e.g. good fix, no signal by corelating numerical value to a readable string output), latitude and longitude as well as the relative altitude (altitude from the ground) and the absolute altitude (altitude from the sea level). These data are displayed along with a coloured output for the GPS signal reception. All data are displayed on a map, using the functionality of `mavflightview`[17] embedded in the script, revealing the full trajectory of the UAV.

– **Unexpected altitude variation detection:** During the flight, the drone is expected to cruise at a determined altitude. A sudden variation in the altitude within a specific timeframe (from timestamp to timestamp) may imply that the drone is losing altitude from an undetermined factor (e.g.

[17] `https://github.com/ArduPilot/MAVProxy/blob/master/MAVProxy/tools/mavflightview.py`

obstacle hit, propeller breakdown). GRYPHON extracts the relative altitude from the GPS and determines if such an unexpected event has occurred.

– **Determine whether a CMD command was executed:** As already discussed, the UAV can execute complex commands given by the pilot through the GCS. A MAVlink CMD message is stored in the Dataflash log, commanding the drone to fly at the coordinations given at a specific altitude. The autopilot directly executes commands upon receiving them. Practically, this means that newer commands would discard the previous ones if there is a conflict, e.g. fly to new coordinates. However, conflicting commands may indicate an attack which leads to rejection of commands as a result of a breach by a perpetrator, or a jamming attack. GRYPHON extracts the received commands and verifies that the drone has flown at the desired coordinates within a margin of 10 centimetres. Failure to do so is reported to the user with a critical error.
– **Data specific extraction errors:** During the flight of the drone, errors or crashes may occur in different flight components (e.g. sensors, GPS, compass). A MAVlink ERR message is stored in the log file and contains the timestamp and the code of the component and the error code. GRYPHON extracts all these error events and performs colour coding, displaying them in red to allow the forensic investigator to distinguish them faster.
– **AC/DC measurements analysis:** A MAVlink CURR message records the battery and flight board voltage and current information. During the flight it is possible that a short circuit or a battery voltage malfunction may occur, resulting in a crash-landing. Extracting the battery voltage and the circuit current consumption gives the forensic researcher the ability to investigate if the values exceeded a 10% acceptable threshold of the median voltage and current value. The latter values are automatically extracted from the PARM messages containing the battery voltage and capacity. These parameter values are given from the pilot of the flying vehicle before the take-off, to ensure that the autopilot can adequately manage the components of the drone.
– **Check CRC from repositories:** The Dataflash log contains the firmware name and its current installed version. A MAVlink MSG message records the firmware info and the vehicle type (e.g. quadrotor, VTOL plane). GRYPHON tries to cross-validate the checksum hash of the firmware of the vehicle with the Ardupilot official Github repository [18]. To achieve this GRYPHON checks whether the collected checksum matches any of the official ones. It should be noted that this indicator can only be considered if a negative result is received. More precisely, if the reported value is aligned with the checksums of the repository, this does not guarantee that the firmware has not been tampered with. Clearly, a tampered firmware can report any value it wants in the log file. However, if the checksum does not match any of the official checksums of the repository, then one can safely assume that the firmware has

[18] https://github.com/ArduPilot/ardupilot

been tampered with. Unfortunately, vendors do not provide signed hashes of their firmware to facilitate these checks.
- **Timeline Analysis:** As mentioned before the Dataflash log contains all the events recorded during the flight. Only a specific type of messages are useful for forensic analysis. MAVlink messages like GPS, ERR, CURR, MSG, CMD and specific parameters of the flight components (e.g. battery capacity and voltage) have high importance in determining the cause of a failure, or verify that a malicious action took place. After extracting the flight event data, `GRYPHON` sorts the events according to their timestamps, creating a timeline analysis for the forensic investigator. This analysis is generated in a new text file for later use, under the name `logfile_name.bin.analysis`.

4.3 Replaying data from the Telemetry log

A useful companion on the forensic research of a flying vehicle is the Telemetry log. When a drone is connected to a GCS on the pilot's device (e.g. a computer, a tablet, a mobile phone), the Telemetry log is generated on the device once the connect button on the GCS is pressed. It contains all the flight data sent to the drone like the Dataflash logfile, but it is a replay log. This means that acquiring and replaying this type of log gives the forensic analyst a better view of the flight. The replay displays all the flight data as if the drone was flying at the moment. Using the built-in functionality of state of the art GCS like the Mission Planner[19], the Telemetry log offers a "real-time" behavioural analysis, from the flight path to the command request and execution. Mission Planner's graphs and other tools like 3D flight path generation for Google Earth, parameters extraction make the visualisation of Telemetry log data easier, multiplying the forensic value of the logs. It must be noted that in the event of a drone crash-landing, the drone should be connected to an assosiated device GCS to have the Telemetry log in possession. This means that in the dreadful scenario of a drone attack or malicious behaviour (e.g. restricted area flight), the Telemetry log cannot be acquired. Therefore, finding the perpetrator's device connected to the drone lies to the Authorities which have to make a forensic examination to extract the telemetry logs. In any case, the telemetry log is useful for enterprises that maintain fleets of drones to facilitate the forensic examination in case of accidents.

5 Experimental Results

Using the dataset provided from VTO Labs in the Drone Forensics Project[20], we tested the compliance of our tool with the provided binary to determine whether it satisfies the functionalities described in the previous section. More precisely, logs of the `ArduPilot DIY drone` were used since it was the only `Ardupilot`

[19] `http://ardupilot.org/planner/`
[20] `https://www.droneforensics.com`

compliant drone available, then we used the tool to another private log file from a drone crash. GRYPHON successfully collected the necessary information from the binaries, extracting the data of various flights. One particular flight serves as an example of the capabilites of the GRYPHON forensic tool. The events of the flight are illustrated in Figure 4c screenshots of the trajectories of four flights. During the flight of an autonomous flying vehicle a battery failure occurred, resulting in a crash land. The forensic analysis of the events seems to solve the case, see Figure 4.

Apart from mapping the trajectories and creating the corresponding timelines, GRYPHON managed to find several events in each flight but no evidence of a malicious act. Each flight mode is displayed in a different colour to facilitate recognition (image flightpath1). GRYPHON reported that during the flight a battery voltage, see Figure 4, an anomaly caused a crash land. Since the autopilot verified the voltage problem the Return To Land (RTL) mode was automatically triggered. The drone did not manage to return safely to the ground as displayed on and resulted in a crash landing. Additionally, the firmware version of the flying vehicle appears to match the checksum of the `Ardupilot` Github repository. Therefore, there is no obvious tampering of the firmware.

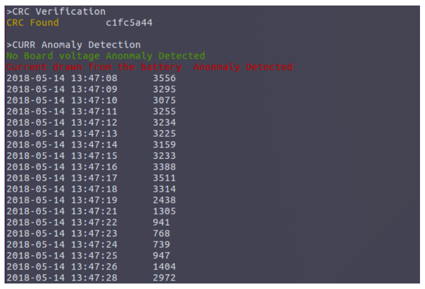
6 Conclusions

The upcoming rise of the unmanned flying vehicles posses new threats to everyday life. A careful examination of the forensic artifacts should ensure that the perpetrators will face justice or identify the issues that caused e.g. a collision. Since there is no established procedure of drone forensic research to date, this research proposes a detailed plan on how to approach such a challenge from the perspective of the ground control station. We argue that drone forensic examinations should focus on the topics described in this work, despite the wide variety of the underlying autopilot firmware and operating systems. In the future, we plan to extensively test the tool to determine its effectiveness for a proper forensic investigation, leading to a more robust and effective toolkit for a forensic researcher. Nevertheless, we should highlight the lack of available datasets that could be used as baseline for further extension of such work.

Acknowledgements

This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the project YAKSHA (Grant Agreement no. 780498) and is based upon work from COST Action CA17124: *Digital forensics: evidence analysis via intelligent systems and practices* (European Cooperation in Science and Technology).

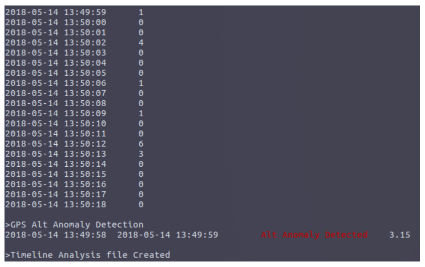
This paper utilised datasets from droneforensics which is based on research completed by VTO Labs (Colorado, USA); sponsored by the United



```
>CRC Verification
CRC Found        c1fc5a44

>CURR Anomaly Detection
No Board voltage Anonmaly Detected
Current drawn from the battery  Anonmaly Detected
2018-05-14 13:47:08      3556
2018-05-14 13:47:09      3295
2018-05-14 13:47:10      3075
2018-05-14 13:47:11      3255
2018-05-14 13:47:12      3234
2018-05-14 13:47:13      3225
2018-05-14 13:47:14      3159
2018-05-14 13:47:15      3233
2018-05-14 13:47:16      3388
2018-05-14 13:47:17      3511
2018-05-14 13:47:18      3314
2018-05-14 13:47:19      2438
2018-05-14 13:47:21      1305
2018-05-14 13:47:22      941
2018-05-14 13:47:23      768
2018-05-14 13:47:24      739
2018-05-14 13:47:25      947
2018-05-14 13:47:26      1404
2018-05-14 13:47:28      2972
```

(a) Firmware verification and voltage analysis.

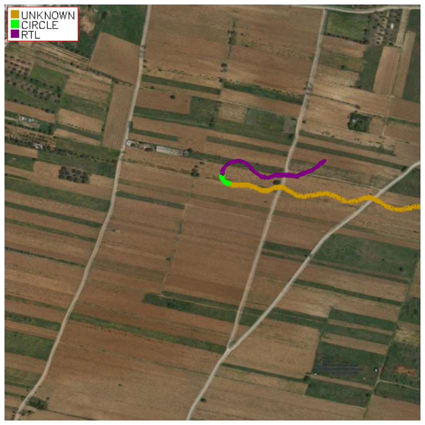


```
2018-05-14 13:49:59      1
2018-05-14 13:50:00      0
2018-05-14 13:50:01      0
2018-05-14 13:50:02      4
2018-05-14 13:50:03      0
2018-05-14 13:50:04      0
2018-05-14 13:50:05      0
2018-05-14 13:50:06      1
2018-05-14 13:50:07      0
2018-05-14 13:50:08      0
2018-05-14 13:50:09      1
2018-05-14 13:50:10      0
2018-05-14 13:50:11      0
2018-05-14 13:50:12      6
2018-05-14 13:50:13      3
2018-05-14 13:50:14      0
2018-05-14 13:50:15      0
2018-05-14 13:50:16      0
2018-05-14 13:50:17      0
2018-05-14 13:50:18      0

>GPS Alt Anomaly Detection
2018-05-14 13:49:58  2018-05-14 13:49:59        Alt Anomaly Detected    3.15

>Timeline Analysis file Created
```

(b) Battery failure led to voltage and altitude error detection.



(c) Flight trajectory with colored flight modes to be easily distinguished.

Fig. 4: Screenshots from the trajectory of the crash landing.

States Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD) via contract number HHSP233201700017C.

References

1. 2018 drone market sector report. `http://droneanalyst.com/research` (2018)
2. Abbaspour, A., Yen, K.K., Forouzannezhad, P., Sargolzaei, A.: A neural adaptive approach for active fault-tolerant control design in uav. IEEE Transactions on Systems, Man, and Cybernetics: Systems (99), 1–11 (2018)
3. Abbaspour, A., Yen, K.K., Noei, S., Sargolzaei, A.: Detection of fault data injection attack on uav using adaptive neural network. Procedia computer science **95**, 193–200 (2016)
4. Altawy, R., Youssef, A.M.: Security, privacy, and safety aspects of civilian drones: A survey. ACM Transactions on Cyber-Physical Systems **1**(2), 7 (2017)

5. Alvarez, P.: Using extended file information (exif) file headers in digital evidence analysis. International Journal of Digital Evidence **2**(3), 1–5 (2004)
6. Barton, T.E.A., Azhar, M.A.H.B.: Open source forensics for a multi-platform drone system. In: Matoušek, P., Schmiedecker, M. (eds.) Digital Forensics and Cyber Crime. pp. 83–96. Springer International Publishing, Cham (2018)
7. Barton, T.E.A., Azhar, M.H.B.: Forensic analysis of popular uav systems. In: Emerging Security Technologies (EST), 2017 Seventh International Conference on. pp. 91–96. IEEE (2017)
8. Birnbaum, Z., Dolgikh, A., Skormin, V., O'Brien, E., Muller, D., Stracquodaine, C.: Unmanned aerial vehicle security using recursive parameter estimation. Journal of Intelligent & Robotic Systems **84**(1-4), 107–120 (2016)
9. Bouafif, H., Kamoun, F., Iqbal, F., Marrington, A.: Drone forensics: Challenges and new insights. In: New Technologies, Mobility and Security (NTMS), 2018 9th IFIP International Conference on. pp. 1–6. IEEE (2018)
10. Clark, D.R., Meffert, C., Baggili, I., Breitinger, F.: Drop (drone open source parser) your drone: Forensic analysis of the DJI Phantom III. Digital Investigation **22**, S3–S14 (2017)
11. Hartmann, K., Giles, K.: Uav exploitation: A new domain for cyber power. In: Cyber Conflict (CyCon), 2016 8th International Conference on. pp. 205–221. IEEE (2016)
12. Hartmann, K., Steup, C.: The vulnerability of uavs to cyber attacks-an approach to the risk assessment. In: Cyber Conflict (CyCon), 2013 5th International Conference on. pp. 1–23. IEEE (2013)
13. Hooper, M., Tian, Y., Zhou, R., Cao, B., Lauf, A.P., Watkins, L., Robinson, W.H., Alexis, W.: Securing commercial wifi-based uavs from common security attacks. In: Military Communications Conference, MILCOM 2016-2016 IEEE. pp. 1213–1218. IEEE (2016)
14. Horsman, G.: Unmanned aerial vehicles: A preliminary analysis of forensic challenges. Digital Investigation **16**, 1–11 (2016)
15. Jain, U., Rogers, M., Matson, E.T.: Drone forensic framework: Sensor and data identification and verification. In: Sensors Applications Symposium (SAS), 2017 IEEE. pp. 1–6. IEEE (2017)
16. Javaid, A.Y., Sun, W., Alam, M.: Uavsim: A simulation testbed for unmanned aerial vehicle network cyber security analysis. In: Globecom Workshops (GC Wkshps), 2013 IEEE. pp. 1432–1436. IEEE (2013)
17. Kim, A., Wampler, B., Goppert, J., Hwang, I., Aldridge, H.: Cyber attack vulnerabilities analysis for unmanned aerial vehicles. In: Infotech@ Aerospace 2012, p. 2438 (2012)
18. Kovar, D., Dominguez, G., Murphy, C.: Uav (aka drone) forensics. Slides of a talk given at SANS DFIR summit in Austin, TX July **7** (2015)
19. Pleban, J.S., Band, R., Creutzburg, R.: Hacking and securing the ar. drone 2.0 quadcopter: investigations for improving the security of a toy. In: Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2014. vol. 9030, p. 90300L. International Society for Optics and Photonics (2014)
20. Schumann, J., Moosbrugger, P., Rozier, K.Y.: R2u2: monitoring and diagnosis of security threats for unmanned aerial systems. In: Runtime Verification. pp. 233–249. Springer (2015)
21. Sedjelmaci, H., Senouci, S.M., Messous, M.A.: How to detect cyber-attacks in unmanned aerial vehicles network? In: Global Communications Conference (GLOBECOM), 2016 IEEE. pp. 1–6. IEEE (2016)

22. Solodov, A., Williams, A., Al Hanaei, S., Goddard, B.: Analyzing the threat of unmanned aerial vehicles (uav) to nuclear facilities. Security Journal **31**(1), 305–324 (2018)
23. Valente, J., Cardenas, A.A.: Understanding security threats in consumer drones through the lens of the discovery quadcopter family. In: Proceedings of the 2017 Workshop on Internet of Things Security and Privacy. pp. 31–36. ACM (2017)
24. Vattapparamban, E., Güvenç, İ., Yurekli, A.İ., Akkaya, K., Uluağaç, S.: Drones for smart cities: Issues in cybersecurity, privacy, and public safety. In: Wireless Communications and Mobile computing Conference (IWCMC), 2016 International. pp. 216–221. IEEE (2016)
25. Yaqoob, I., Hashem, I.A.T., Ahmed, A., Kazmi, S.A., Hong, C.S.: Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges. Future Generation Computer Systems **92**, 265–275 (2019)