CATEGORIE TECHNIQUE
Quai Gloesener, 6 - 4020 Liège

Penetration testing of commercial drones and realisation of a drone pentesting framework

Yannick PASQUAZZO

Travail de fin d'études présenté en vue de l'obtention du grade de Master en sciences de l'ingénieur industriel, orientation informatique

Année académique : 2018 - 2019

Abstract

Some text.

Some other text.

In this master thesis, we propose something.

Foreword

"A nice quote."

John Doe [**manual-identifier**]

Some text.

Some other text.

Acknowledgements

I would like to express my gratitude to my supervisor, Captain Ir Alexandre D'Hondt, whose comments, advices and engagement through the period of my master thesis helped me to direct my efforts on rewarding matters. I would like to thank my tutor, Mr Ludovic Kuty, and Ir Pierre de Fooz for their guidance on troubleshooting some issues.

I would also like to thank my readers, Dr Ir Cyrille Mosbeux and Ing Thomas Guérin, who have kindly devoted their precious time for reading this thesis.

Furthermore, I would like to thank my entourage, for having supported me during this long and laborious process.

Table of Contents

List of Acronyms iv

List of Figures v

Chapter 1 Introduction 1

1.1 Problem Statement . . . 2
1.2 Objectives . . . 2
1.3 Approach . . . 3
1.4 Content . . . 3
1.5 Conventions & Reading Advices . . . 4

Chapter 2 Background 5

2.1 Terminology. . . . 6
2.1.1 Vulnerability VS Weakness . . . 6
2.1.2 Black VS White hat hacker . . . 6
2.1.3 Kinds of assessment . . . 6
2.2 Types of penetration test. . . . 7
2.2.1 Blackbox . . . 8
2.2.2 Whitebox . . . 8
2.2.3 Greybox . . . 9
2.3 Methodologies & Techniques. . . . 9
2.3.1 Penetration Testing Execution Standard . . . 9
2.3.2 Wireless penetration testing . . . 12
2.3.3 Reverse engineering . . . 12
2.3.4 Hardware hacking . . . 12
2.4 Tools & Resources . . . 12
2.4.1 Pentesting platforms . . . 12
2.4.2 Intelligence Gathering . . . 13
2.4.3 Vulnerability Analysis . . . 15
2.4.4 Exploitation . . . 16

2.4.5 Reverse engineering 18

Chapter 3 Scope **19**

3.1 Literature review **20**

3.1.1 Parrot AR Drone 20

3.1.2 Known vulnerabilities 21

3.2 Scope definition **21**

3.2.1 Scope limitation 21

3.2.2 Flitt Selfie Cam 22

3.2.3 C-me Selfie Drone 22

3.3 Intelligence gathering **23**

3.4 Vulnerability analysis **23**

Chapter 4 Exploits **24**

Chapter 5 Framework **25**

Chapter 6 Conclusion **26**

6.1 Summary **27**

6.2 Objectives **27**

6.3 Matched Criteria **27**

6.4 Future Works **27**

References **28**

List of Acronyms

APT	Advanced Persistent Threat
CNA	CVE Numbering Authorities
CVE	Common Vulnerabilities and Exposures
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
IS	Information System
PoC	Proof-of-Concept
PT	Penetration Test
PTES	Penetration Testing Execution Standard
VA	Vulnerability Assessment

List of Figures

2.1 Types of penetration test in function of the prior degree of knowledge 8
2.2 The 7 steps of PTES 10

3.1 User interface for piloting the Flitt Selfie Cam from a smartphone 22
3.2 User interface for piloting the C-me Selfie Drone from a smartphone 23

1 Introduction

"When functionality is all that matters, security is often overlooked."

Alexandre D'Hondt

Cybersecurity expert at the Belgian Defense

Internet of Things is on the rise with more than 30 billion devices connected worldwide expected by 2020. Today, controlling a device remotely has become the norm, especially through wireless protocols. Therefore, it is common to find light commercial drones which are pilotable with a simple smartphone. As a result, malicious individuals might be tempted to leverage common security flaws in this field.

From general to particular, this introduction reduces the scope to the field of security assessment, more exactly penetration testing, through the problem statement, higlights the desired objectives and approach and dissects the remainder of this document, that is, what we were able to investigate and perform during our master thesis, using a few conventions to be kept in mind while reading it.

1.1 Problem Statement 2
1.2 Objectives 2
1.3 Approach 3
1.4 Content 3
1.5 Conventions & Reading Advices . 4

Domain	Vulnerability Assessment & Penetration Testing
Scope	Internet of Things : light commercial drones
Audience	Vulnerability Hunters
Purpose	Study the security of common light commercial drones and build a penetration testing framework based on the acquired knowledge

1.1 Problem Statement

The past few years, more and more connected devices have invaded our daily lives. Although this generally allows us to improve our living environment, the proliferation of these connected gadgets is not necessarily without consequences. Indeed, each of these devices can be remotely controlled, either from the Internet or in their vicinity, which implies obvious security risks. More specifically, this work focuses on some ways how an attacker could break into light commercial drones and the impact it could have. To name a few, a malicious person might be able to eavesdrop the video of a device, steal or even crash it.

Some companies have already developed some commercial products in order to provide protection against drones threatening safety, security and privacy. Some of these solutions are as simple as firing a net to catch a device or disrupting the signal by emitting interference and thus making a drone inoperable. Some more advanced technologies also tackle the problem by sending specific commands to force a landing or make a drone go back to a certain point.

But, as far as we know, in the scope of drone software security, there still lacks a convenient open-source solution for gathering and coordinating exploits, one toolkit such as Metasploit, which is already well-established regarding OS penetration testing. That is what we propose in this master thesis; we try, first, to develop several exploits working on the drones we were provided, then we create an open source framework that we design to be modular and easy to contribute to.

1.2 Objectives

Our objectives are four-fold :

1. State the **background**.
   - A – Review the current literature about IT security and especially IoT security.
   - B – Search for processes and methodologies for hacking systems.
   - C – Browse some existing solutions and tools and select relevant ones for exploitation.
2. Narrow our **scope**.
   - A – Select some models of light commercial drones based on their technology.
   - B – Filter out WiFi-based drones and understand their working.
3. Build some **exploits** for breaking into the selected drones.
   - A – Find attack chains for the selected models of drones.
   - B – Design and implement short scripts for exploiting found security holes.
4. Put it altogether in a **framework**.
   - A – Set the basis for the framework.
   - B – Turn the exploits into reusable modules.

1.3 Approach

The school provides some criteria related to the scientific and technological content that are worth being parsed regarding our approach. Succinctly, these criteria match our approach like follows :

1. **Problem analysis** : From general to particular, we narrow our scope to a few targets and clearly state the requirements for the delivrables.
2. **Solution provided** : We implement the requirements into exploit scripts and ultimately a penetration testing framework tailored to drone hacking.
3. **Rigor of the approach** : We segment our approach from the state-of-the-art knowledge to measurable and assessable practical outcomes.
4. **Innovation** : We provide a brand new solution, gather and leverage the best of the parsed and acquired knowledge.
5. **Personal contribution** : We develop exploit scripts and modules for the new penetration testing framework.
6. **Avenues for future development** : We provide an extensible solution that could stir up the curiosity of drone hacking enthusiasts.

Regarding the skills acquired during our formation at the school, this project mainly applies, directly or indirectly, the following courses :

- [**B38**] Operating systems and introduction to IoT : By learning the basics of the Linux operating system and therefore allowing to have a better understanding of the architecture of the drones.
- [**M18**] Network programming and software security : By using the acquired knowledge and tools related to network protocols to develop exploits that can be used from a distance. Cryptography knowledge were also a must-have.
- [**M18**] Internet of Things : Because it is the main subject of this thesis, and, in addition to this, the course strongly insisted on penetration testing and security.
- [**M18**] Network security : By having prior knowledge of the specifics of network security, such as safe connection to a remote host.
- [**M28**] Study of wireless networks : By applying the knowledge of the 802.11 protocol I order to successfully capture and decrypt WiFi transmissions.
- [**M18+M28**] Communication and language : By writing the thesis in English, thus increasing the scope of readers.

1.4 Content

The remainder of this document is structured as follows :

- **Chapter 2 – Background** provides background information in the field of IT security and especially drone hacking. It explains some relevant methodologies and processes and outlines a few existing solutions, either commercial or open-source.
- **Chapter 3 – Scope** presents some models of drones and their overall working, fixing the scope of this thesis to a few targets.
- **Chapter 4 – Exploits** explains the applied hacking techniques and their related exploit scripts.

- **Chapter 5 – Framework** presents the drone penetration testing framework and its modules, developed from the aforementioned exploit scripts.
- **Conclusion** closes this introduction by presenting a general summary, by parsing the achieved objectives and outcomes of this work and by providing ways ahead and ideas for future works.

1.5 Conventions & Reading Advices

This document is organized such that it can be read mostly using a method in three passes. Indeed, the reader who wants to spare time can get an insight of this work, as a first pass, by simply reading the chapter cover pages. The reader who can take a bit more time for this reading, as a second pass, can directly jump to the end of the chapters for reading the summaries and discussions. Ultimately, the interested reader, as a third pass, can read the entire content.

Why such a layout ?
In order to get this document as much attractive as possible, we designed it with an unusual style, starting each chapter with a cover page providing a quotation from an IT professional and introducing the chapter matter bouncing from the quotation. We hope you will enjoy reading it !

Want to spare time ?
Check the summaries and related discussions at the end of each chapter, they contain information enough so that you can quickly get the main thread !

More focused on Drone Hacking ?
Check chapters **2** and **3** then look at the summaries and related discussions of chapters **4** and **5**.

More focused on the Exploitation Framework ?
Check chapters **4** and **5** and their related appendices.

2 Background

"I believe that the challenges of IT security will continue to increase, even if the operating systems and applications drastically improve."

H. D. Moore

Cybersecurity expert, developer of Metasploit

IT security is nowadays a very flourishing field with the evolution of technologies, bringing new challenges and techniques, especially when trying to break into systems.

This chapter presents some basics about penetration testing, the ultimate science of breaking into systems, starting with defining what a penetration test is, then explaining the different test levels, methodologies and techniques as of the current state-of-the-art. It then lists a few existing tools in this field.

2.1 Terminology 6
2.1.1 Vulnerability VS Weakness . 6
2.1.2 Black VS White hat hacker . 6
2.1.3 Kinds of assessment 6
2.2 Types of penetration test 7
2.2.1 Blackbox 8
2.2.2 Whitebox 8
2.2.3 Greybox 9
2.3 Methodologies & Techniques . . 9
2.3.1 Penetration Testing Execution Standard 9
2.3.2 Wireless penetration testing 12
2.3.3 Reverse engineering 12
2.3.4 Hardware hacking 12
2.4 Tools & Resources 12
2.4.1 Pentesting platforms 12
2.4.2 Intelligence Gathering 13
2.4.3 Vulnerability Analysis 15
2.4.4 Exploitation 16
2.4.5 Reverse engineering 18

2.1 Terminology

Several notions should be known before continuing. This section defines the most important ones with their distinctions when relevant.

2.1.1 Vulnerability VS Weakness

Vulnerability

Weakness

Common Vulnerabilities and Exposures (CVE) ® is a list of common identifiers for publicly known vulnerabilities.

Use of CVE entries, which are assigned by CVE Numbering Authorities (CNA) from around the world, ensures confidence among parties when used to discuss or share information about a unique software or firmware vulnerability, provides a baseline for tool evaluation, and enables automated data exchange.

By definition, CVE is :

- One identifier for one vulnerability or exposure
- One standardized description for each vulnerability or exposure
- A dictionary rather than a database
- How disparate databases and tools can "speak" the same language
- The way to interoperability and better security coverage
- A basis for evaluation among services, tools, and databases
- Free for public download and use

2.1.2 Black VS White hat hacker

Blackhat hacker (the Bad) This profile is the malicious guy who wants to break into systems for his own benefit, in general for financial reasons.

Whitehat hacker (the Good) This profile pertains to the security expert and/or researcher who searches for security issues in IS' and whose goal is to fill the security gaps.

In the scope of this work, we act as whitehat hackers as our project aims to identify and help fix security issues on light commercial drones.

2.1.3 Kinds of assessment

Penetration Testing *... can be defined as a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure.* [1] The main idea is to find security issues just like an attacker would do so that these can be mitigated before a real attack.

Vulnerability Assessment (VA) This kind of security assessment is a system review aimed to find potential security issues, generally performed in a live environment. The important point here is that it stops after having found vulnerabilities and reported them for mitigations, it does not assess their exploitability yet. At some point, when reported vulnerabilities are too numerous, it could be helpful to focus attention on these that could be effectively exploited for breaking into the related IS and direct the available resources on fixing these issues instead of trying to fix all the reported vulnerabilities (whose most of them are often informational or very difficult to exploit in practice). This is what distinguishes a Vulnerability Assessment from a Penetration Test.

Penetration Test (PT) This kind of security assessment encompasses the VA but goes far beyond, demanding more resources (in time and efforts) to check for the exploitability of the found vulnerabilities in order to determine with more precision their impact on IS' environment. The outcome of such a test is a **Proof-of-Concept (PoC) attack** that demonstrate the exploitability. This can lead to the discovery of security holes opening the way to pervasive actions like listed hereafter. The most important point here, and what distinguishes a penetration tester from a real attacker, is the **permission** (nonwithstanding the difference in motivation).

Multiple effects can result from the exploitability of a vulnerability :

- Sensitive or even critical business-related data exfiltration.
- Potential for service and system disruption, that could lead to heavy financial impact for the host organization.
- Pivoting inside organization's network to impact other systems than the original target IS.

Ultimately, the main purpose of these kinds of security assessments is to report on security holes to the management, to provide mitigations and recommendations and to prioritize them in order to prevent real attacks from being led.

Red teaming This notion pertains to simulating what we could begin to describe as an Advanced Persistent Threat (APT). In this context, the PT is extended over a much larger period (several months instead of a few days or weeks). This kind of scenario is more rare, as it demands far more resources to be carried out. However, this reflects even more closely the context of an external attack in which hackers increasingly tend to take their time to study the targeted IS and also to hide their presence.

Vulnerability hunting This notion also refers to system or software review, but generally on a product outside its operating environment. It pertains to studying an asset in order to demonstrate its potential security flaws and help the manufacturer fix them before or yet after its deployment on the market.

In our case, we use the penetration testing approach and its related techniques to achieve vulnerability hunting on multiple drone models from various manufacturers.

2.2 Types of penetration test

A penetration test can take a different form according to the requirements of the target organization, the assessment depth and the degree of starting information. Figure 2.1 depicts the different types in function of the degree of prior knowledge.

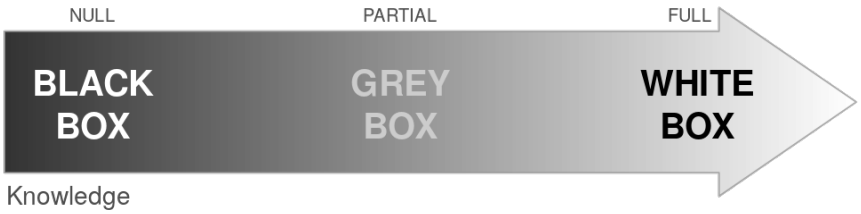

Figure 2.1: Types of penetration test in function of the prior degree of knowledge

2.2.1 Blackbox

A black box is the representation of a system taking inputs and giving outputs, with no consideration of its internal functioning. This working is either inaccessible (simply not available or lost), or deliberately omitted (in order to simulate the conditions in which a real attacker would operate).

In the *Blackbox* context, the pentester really puts himself in the shoes of an external attacker and starts his penetration test with as little information as possible on his target (his target then being the company having requested the assessment). Indeed, when the tester begins his attack, he does not have (or rarely) the complete map of the IS, the list of servers with their IP, and so forth. The *Blackbox* context aims to find and demonstrate the presence of an actionable plan by an external person to take control of the IS or get hold of certain information.

Reasons for black box penetration testing :

- ✓ Simulates the majority of potential attacks facing your organization
- ✓ Helps to identify where your systems are weakest
- ✓ Provides an objective, outsider's view of your systems

2.2.2 Whitebox

In systems theory, a white box, is a module of a system that can be expected to work internally because we know the operating characteristics of all elements that compose it. In other words, a white box is a module that has as few black boxes as possible.

In this case, the pentester works in close collaboration with the DSI, the RSSI and the technical team of the information system. The goal is to obtain 100% information on the information system and to support the CIO in the detection of vulnerability. One of the advantages of the White Box mode is that it is then possible to detect security flaws in a wider way and that the Black Box mode would not have made it possible to detect, for example if the pentester had not reached a certain stage of the intrusion. In addition, the White Box mode is more easily integrated into the IS lifecycle, sometimes at each stage of its evolution.

Reasons for white box penetration testing:

- ✓ Efficient use of ethical hacker's time
- ✓ Provides "full disclosure" insights
- ✓ Allows for objective scrutiny of internal systems

2.2.3 Greybox

An engagement that allows a higher level of access and increased internal knowledge falls into the category of gray-box testing. Comparatively, a black-box tester begins the engagement from a strict external viewpoint attempting to get in, while the gray-box tester has already been granted some internal access and knowledge that may come in the form of lower-level credentials, application logic flow charts, or network infrastructure maps. Gray-box testing can simulate an attacker that has already penetrated the perimeter and has some form of internal access to the network.

By providing some form of background to the security consultants undertaking the assessment, it helps to create a more efficient and streamlined approach. This saves on the time (and money) spent on the reconnaissance phase, allowing the consultants to focus their efforts on exploiting potential vulnerabilities in higher-risk systems rather than attempting to discover where these systems may be found.

Reasons for grey box penetration testing :

- ✓ Focus the attention on more highly-valuable areas within the network
- ✓ Increases the attack coverage and efficiency
- ✓ Cost effective

The point here is that the type of testing we will achieve will depend on the target, as only a few information or even none could be available. We will see that some models of target and their related manufacturers are not always very transparent, then forcing the test to be limited to blackbox.

2.3 Methodologies & Techniques

This section presents some methodologies and techniques that will be used in the remainder of this work.

2.3.1 Penetration Testing Execution Standard

After having introduced what is a pentest, let us look at a popular methodology for organizing a pentest.

But why do penetration testers need to follow a methodology ?

- The security world is changing fast, techniques and technologies are constantly changing, vulnerabilities are discovered on a daily basis. Faced with this, a bit of **structure** is needed. The methodologies allow to provide a fixed framework on the progress of a penetration test.
- Faced with an IS or many, a pentester has to look into each system, service, port or IP which can hide a vulnerability, see a path all traced until the takeover of the system. This multiplied by the very large amount of testing to be done depending on the service or technology used, makes it is easy to get lost. A framed and fixed methodology then makes it possible to follow a **canvas** and thus to organize the penetration test so that it is as **complete** as possible and **repeatable**.
- In the professional environment, **team working** is essential. This is also the purpose of common methodologies, norms and standards. A methodology known by several pentesters will facilitate and make more efficient the work within the team. Also, the report generated and the documentation will be made following a known frame : that of the methodology used.

First and foremost, a methodology will make it possible to not forget anything during the penetration test and to make sure that it is as complete as possible. During said pentest, the methodology acts as a way to follow, facilitating the organization of the pentester.

Penetration Testing Execution Standard (PTES) [2] Let's take a look at this popular methodology that is often used by most pentesters. Mostly, this methodology is used as a basis in the pentest teams who customize them with their own methods, tools and techniques, which is what will make the difference between one team of pentest and another. The experience of the field is then added to the theory of the methodology. It should also be noted that each pentest follows a path of its own according to the IS to be tested. This methodology follows the steps as depicted in 2.2 and explained hereafter.

Figure 2.2: The 7 steps of PTES

1. **Pre-engagement Interactions** : Obviously, each intrusion test starts with a negotiation, an understanding of customer needs and finally a contractualization that puts all of this into paper form. The establishment of the contract is an opportunity for the company requesting the security test to specify several things concerning the scope of the intrusion test to be performed. In terms of time (days, weeks, months), in terms of technologies, in geographical terms (range of IP to test and not to test for example). Also, this is an opportunity to specify the type of test to be provided: can testers set up DoS attack phases ? Will they have to go to the end of the operation even if they have to open sensitive documents ? The contract also aims to establish a legal link and a protection for the pentester that could fall on sensitive documents, but also to negotiate and inform the companies collaborating with the customer (ISP, host, business application, etc.).
2. **Intelligence Gathering** : The second step is that of "Intelligence Gathering" or "Information Gathering". Understandably the collection of information. Here, the pentester will try to list as much

information about the SI tested, and this via several methods. The goal is therefore to map the attack surface from the starting position, to obtain information on the services, servers and active elements used, as well as their version and security systems in place (VPN, Firewall, Anti-Ddos, DMZ, etc.). The Intelligence Gathering part can be done passively (without direct interaction with the IS of the target) and / or actively (by going to communicate with the targeted servers). This also includes the concepts of OSINT (Open Source Intelligence) and HUMINT (Human Intelligence, understand social engineering).

3. **Threat Modeling** : The third step is the Threat Modeling, in which the pentester is going to look for a list of threats for the company. What can the company have to deal with ? He will try to evaluate what is critical for the company (e.g. its manufacturing secret), to establish the potential impact of the loss of this secret and then see what vulnerabilities can lead to a loss of manufacturing secrecy. Also, one can go through an analysis phase in terms of security of competitors / similar companies. Have they been compromised recently ? What was the impact of the attack ? An analysis of the company's process and its human assets can also be performed as well as an analysis of the IS: is there an encryption mechanism for salespeople communicating with the inside of the IS on the business application ? Is a VPN in place for the remote administration of the IS ? The model should be clearly documented, and be delivered as part of the final report as the findings in the report will reference the threat model in order to create a more accurate relevance and risk score that is specific to the organization (rather than a generic technical one).

4. **Vulnerability Analysis** : The information collected previously will be used to analyze the systems, staff and processes in place to search for exploitable vulnerabilities. Thus, it goes through a manual and / or automatic search process of vulnerabilities. The objective here is to draw up a list of what could be exploitable and build a "tree of attack", i.e. a continuation of attack that can lead to the discovery of other vulnerabilities, tracing a path towards the takeover of the IS or the possibility of establishing acts of espionage or sabotage. This phase is systematically based on research on the bases of public exploits, verification of the patches of the scanned systems and the use of scans tools (Ex: Nessus, OpenVAS, Nmap, etc.). If the vulnerability analysis phase was properly completed, a high value target list should have been complied. Ultimately the attack vector should take into consideration the success probability and highest impact on the organization.

5. **Exploitation** : The fifth stage is often the most awaited of the pentesters, it is the moment to go to the attack: the exploitation. Here, he will try to test the vulnerabilities found in the previous phases and to advance in the intrusion of the information system. He will also try to test the barriers and security systems in place (evasion of an IDS, bypassing a security, bypass authentication). This will be followed by a validation process of the potential vulnerabilities found. It is important here to find ways to exploit vulnerabilities and also security systems. If the pentester advances in the SI, it may have to go back to step 4 in order to perform a new vulnerability analysis in the new zone (let's say if it manages to switch from the DMZ to the LAN). Note that in the pre-engagement interaction phase with the customer, a clear definition of the overall objectives of the penetration test should have been communicated. In the case of the exploitation phase, the biggest challenge is identifying the least path of resistance into the organization without detection and having the most impact on the organizations ability to generate revenue. By performing the prior phases properly, a clear understanding of how the organization functions and makes money should be relatively understood. From the exploitation phase and into the post-exploitation phase, the attack vectors should rely solely on the mission of circumventing security controls in order to represent how the organization can suffer substantial losses through a targeted attack against the organization.

6. **Post Exploitation** : The purpose of the Post-Exploitation phase is to determine the value of the machine compromised and to maintain control of the machine for later use. This step has several interesting phases, like a real attacker, the pentester will have to erase his tracks so that these actions are as discreet as possible (clearing logs for example). Also, it will have to seek to establish a sustainable way in the information system by inserting backdoors, by creating a VPN account or by installing a

malware / rootkit. The post-exploitation phase is also the last technical phase of the intrusion test, so it is time to target the information to be stolen (also referred as “Pillaging”), the interesting people of the company (trapping a high placed target for example). The idea here is to really show what mischief an attacker could accomplish: espionage, sabotage, exfiltration of data, deployment of a botnet, decommissioning part of the IS, etc.

7. **Reporting** : The Pentest approach described in the PTES ends with the reporting phase, and this is the most important phase because it consists in exposing the progress of the intrusion test and its fruits to the client. Reporting will be both technical and non-technical. The pentester will then detail the information retrieved during the Intelligence Gathering phase, expose the vulnerabilities listed and exploited and describe the misdeeds that have / could have been perpetrated in the Post-Exploitation phase. This phase is often oral but is always accompanied by a written deliverable containing all this information. Finally, the pentest team is often called upon to draw up a "Remediation Roadmap", i.e. a list of countermeasures, corrections and protections to be put in place so that the company has tracks to start on following the securing of his IS.

2.3.2 Wireless penetration testing

2.3.3 Reverse engineering

2.3.4 Hardware hacking

For this work, we essentially apply the penetration testing methodology PTES to wireless devices, using reverse engineering and hardware hacking to complement the assessment with in depth target understanding.

2.4 Tools & Resources

In the past, hacking was difficult and required a lot of manual tinkering. Today, hackers have complete suites of automated test tools and processing capabilities that allow them to conduct much more sophisticated attacks, on much larger scales and much faster. In order to end framing the background, various existing security solutions are presented in order to get an overview of what could be usable in the implementation of a penetration test. The remaining of this section presents useful and popular tools distinguished by categories, relevant to penetration testing but not necessarily in our scope. For each tool, it is mentioned if it was used in this project or not.

2.4.1 Pentesting platforms

Two Linux distributions seem to be among the leaders regarding penetration testing ; Kali Linux [3] and Parrot OS [4].

Type	Operating System
Purpose	Reconnaissance, vulnerability analysis, wireless attacks, web application attacks, sniffing and spoofing, reverse engineering, digital forensics, exploit creation, ...
Pros	Open-source Community supported Various command-line tools
Used	Yes

Kali Linux [3] is a Debian-based distribution created by Offensive Security [5] (another reference in security trainings and certifications) with advanced penetration testing features and tools. This is currently one of the most complete and powerful existing pentesting distributions. It also makes available various command-line tools that are appropriate for scripting and automation. A complete list of the available tools can be consulted at [6]. One can point out the Metasploit framework [7] as the most interesting and sophisticated tool to perform penetration tests and Volatility [**volatility**] for digital forensics (essentially the analysis of memory dumps).

Type	Operating System
Purpose	Reconnaissance, vulnerability analysis, wireless attacks, web application attacks, sniffing and spoofing, reverse engineering, digital forensics, exploit creation, ...
Pros	Open-source Lightweight Various command-line tools
Used	No

Parrot OS [4] is a free and open source GNU/Linux distribution based on Debian Testing designed for security experts, developers and privacy aware people. Originally developed as part of Frozenbox, the effort has grown to include a community of open source developers, professional security experts, advocates of digital rights, and Linux enthusiasts from all around the globe. It is intended to provide a suite of penetration testing tools to be used for attack mitigation, security research, forensics, and vulnerability assessment.

2.4.2 Intelligence Gathering

As previously evoked, the reconnaissance is the very first of the phases, and it is the one where the pentester will collect the maximum of data on its target before moving on to the following phases. There are a lot of different tools available to the attacker, here are presented a few of the most representative.

Type	Search Engine
Purpose	Advanced search on Internet
Pros	Free Easy to use Finds information that is not readily available on a website
Used	Yes

Google Hacking [8] is a technique that relies on the search power of the famous search engine to find accurate information that can help navigate a security breach. To use it, one has to use the Google Dork, they are specific search operators that allow to find accurate data, unlike every day searches, one types keywords in the search bar that can correspond to a text, a title, an image, a meta tag, an alternative text of an image and many others.

Type	Search Engine
Purpose	Find vulnerable devices
Pros	Easy to use Indexes all the things connected to the internet
Used	No

Shodan [9] is a website specialized in finding objects connected to the Internet, and therefore having a visible IP address on the network. It allows to find a variety of web servers, routers and many devices such as printers or cameras. Such a request is processed with a simple analysis of the HTTP header returned by the device or server. It is then possible to retrieve lists of specific elements. For each result, it finds the IP address of the server as well as other types of sensitive but accessible information.

Type	Data Mining tool
Purpose	Providing a library of transforms for discovery of data from open sources, and visualizing that information in a graph format, suitable for link analysis and data mining
Pros	Highly customizable Graph export options Runs on Windows, Mac and Linux
Used	No

Maltego [10] can be used to determine the relationships between people, websites, documents and files... Connections between these pieces of information are found using open source intelligence (OSINT) techniques by querying sources such as DNS records, Whois records, search engines, social networks, various online APIs and extracting meta data. Maltego provides results in a wide range of graphical layouts that allow for clustering of information which makes seeing relationships instant and accurate – this makes it possible to see hidden connections even if they are three or four degrees of separation apart.

2.4.3 Vulnerability Analysis

New vulnerabilities are emerging every day, within networks, applications, databases. A vulnerability scanner is therefore a computer program designed to assess them for known weaknesses. They are utilized in the identification and detection of vulnerabilities arising from mis-configurations or flawed programming within a network-based asset such as a firewall, router, web server, application server, etc. Modern vulnerability scanners allow for both authenticated and unauthenticated scans. Once again, let's have a look at some of the most popular ones.

Type	Security Scanner, Port Scanner, Network Exploration Tool
Purpose	Identifying what devices are running on a network, discovering hosts that are available and the services they offer, finding open ports and detecting security risks
Pros	Free Well documented Includes many port scanning mechanisms
Used	Yes

Nmap [11] ("Network Mapper") is a free and open source utility for network discovery and security auditing. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts.

Type	Packet Analyzer
Purpose	Enables live data reading and analysis for a wide range of network protocols
Pros	Open Source Display filters are used to filter and organize the data display New protocols can be scrutinized by creating plug-ins
Used	Yes

Wireshark [12] intercepts traffic and converts that binary traffic into human-readable format. This makes it easy to identify what traffic is crossing a network, how much of it, how frequently, how much latency there is between certain hops, and so forth. While Wireshark supports more than two thousand network protocols, many of them esoteric, uncommon, or old, the modern security professional will find analyzing IP packets to be of most immediate usefulness. The majority of the packets on your network are likely to be TCP, UDP, and ICMP.

Type	Vulnerability Scanner
Purpose	Identifying what devices are running on a network, discovering hosts that are available and the services they offer, finding open ports and detecting security risks
Pros	Highly customizable Covers a wide range of technologies Automated reports
Cons	Full version is expensive
Used	No

Nessus [13] is a proprietary vulnerability scanner developed by Tenable. Nessus detects live machines on a network, scans open ports, identifies active services, their version, and then attempts various attacks. He then points out the potential or proven weaknesses on the tested machines on a global report. Nessus scans cover a wide range of technologies including operating systems, network devices, hypervisors, databases, web servers, and critical infrastructure. Since version 3, it is licensed, but still free for personal use. Version 2 is maintained. There is also a fork of Nessus 2 still under GPL license called OpenVAS.

2.4.4 Exploitation

With a map of all possible vulnerabilities and entry points, the pentester begins to test the exploits found within your network, applications, and data. The goal is for the ethical hacker is to see exactly how far they can get into your environment, identify high-value targets, and avoid any detection. These actions are once again made easier with the help of multiple tools, some of which are discussed below.

Type	Penetration Testing Framework
Purpose	Tool for developing and executing exploit code against a remote target machine
Pros	Rather intuitive Currently has over 1600 exploits Large user community
Used	No

The Metasploit Framework [7] is a Ruby-based, modular penetration testing platform that enables to write, test, and execute exploit code. The Metasploit Framework contains a suite of tools that can be used to test security vulnerabilities, enumerate networks, execute attacks, and evade detection. At its core, the Metasploit Framework is a collection of commonly used tools that provide a complete environment for penetration testing and exploit development. More specifically, the MSFconsole provides a command line interface to access and work with the Metasploit Framework. The console lets you do things like scan targets, exploit vulnerabilities, and collect data.

Type	Offline Password Cracker
Purpose	Test the security of a password, crack password hashes
Pros	Open source Extensible (for new crackable hash types) Can leverage graphical card's GPU power for a better performance
Used	Yes

John the Ripper [14] is a free software for breaking a password, used in particular to test the security of a password. First developed to run under UNIX-derived systems, the program now operates under fifty different platforms, such as BeOS, BSD and its derivatives, DOS, Linux, OpenVMS, Win32. John is one of the most popular password cracking software because it includes the autodetection of hash functions used to store passwords, the implementation of a large number of cracking algorithms, by the fact that it is very easily modifiable, and also that it is possible to resume an attack after a pause.

Type	Online Password Cracker
Purpose	Test the security of a password, crack passwords on live services
Pros	Open source Supports a large set of different password hash types Possible to restore a previous aborted/crashed session
Used	Yes

Hydra [15] is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add. Hydra is often John's accomplice. It can take over to break a password online, for example an SSH or FTP, IMAP, IRC, RDP and more. Just point Hydra to the service you want to hack, provide it a list of words, and launch it. Tools like Hydra point out that limiting the password rate and disconnecting users after several unsuccessful login attempts are effective defensive measures against attackers.

Type	Web Application Testing
Purpose	Identify vulnerabilities and verify attack vectors that are affecting web applications
Pros	Can be used to modify requests to the server, resend them, and observe the results Reported vulnerabilities contain detailed custom advisories
Used	No

Burp Suite [16] is a Java based Web Penetration Testing framework. It has become an industry standard suite of tools used by information security professionals. Burp Suite helps you identify vulnerabilities and verify attack vectors that are affecting web applications. In its simplest form, Burp Suite can be classified as an Interception Proxy. While browsing their target application, a penetration tester can configure his internet browser to route traffic through the Burp Suite proxy server. Burp Suite then acts as a (sort of) "Man in The Middle" by capturing and analyzing each request to and from the target web application. Penetration

testers can pause, manipulate and replay individual HTTP requests in order to analyze potential parameters or injection points. Injection points can be specified for manual as well as automated fuzzing attacks to discover potentially unintended application behaviors, crashes and error messages.

	**Type**	WiFi attacks toolkit
	Purpose	Packet sniffing and injection, WEP encryption key recovery
	Pros	Focuses on different areas of WiFi security: monitoring, attacking, testing and cracking All tools are command line which allows for heavy scripting
	Used	Yes

Aircrack-ng [17] is a complete suite of tools to assess WiFi network security. It focuses on different areas of WiFi security :

- Monitoring : Packet capture and export of data to text files for further processing by third party tools
- Attacking : Replay attacks, deauthentication, fake access points and others via packet injection
- Testing : Checking WiFi cards and driver capabilities (capture and injection)
- Cracking : WEP and WPA PSK (WPA 1 and 2)

2.4.5 Reverse engineering

ILSpy

Hopper Disassembler

3 Scope

"Drones cause problems for more and more types of secure sites, whether it's a matter of irresponsible or nefarious operators [...] taking photos where you shouldn't be taking photos or putting people on the ground at risk."

NIMO SHKEDY

CEO of ApolloShield

...

3.1 Literature review 20

3.1.1 Parrot AR Drone 20

3.1.2 Known vulnerabilities 21

3.2 Scope definition 21

3.2.1 Scope limitation 21

3.2.2 Flitt Selfie Cam 22

3.2.3 C-me Selfie Drone 22

3.3 Intelligence gathering 23

3.4 Vulnerability analysis 23

Now that we have set the basics of what is penetration testing, we can move to the actual application of the concept to the drones.

The first phase of the work consists in gathering as much information as possible regarding the current state-of-the-art on drone security.

3.1 Literature review

This section focuses on showing a compilation of drone vulnerability testing/exploit methodologies. As part of that effort, the following emphasizes on giving a ready reference of one particular vulnerable drone and associated open source attack tools that have already been developed. This compilation should provide the reader with a better understanding of how drone vulnerability is currently exploited, and how future drone will take advantage of improvements in available vulnerability research data.

3.1.1 Parrot AR Drone

The Parrot AR Drone 2.0 is one of the most popular quadcopters produced. This is a cheap drone that is both quick and fast. It is controlled by an iOS or Android smartphone or tablet and allows 720p live high-definition video streaming and recording in flight.

This device is a good reference on how one of the most popular commercial drones totally lacks security. Security breaches are numerous, let's review them :

- **Open FTP on port TCP/21** : Simply just connect to the IP without any username or password, and have access to the directory of the drone, where it stores the recorded videos.
- **Open Telnet on port 23** : Once again, simply just connect to the IP without any username or password, and you are logged in. Furthermore, it's running Linux and it is a root access to the device! At this point the attacker could simply issue a shutdown and watch the drone fall to the ground.
- **Unencrypted communications** : A simple capture of the communication packets between the drone and the controller allows the attacker to have an easy view of the protocol. Once he has successfully analyzed the main commands, he can easily replay them – for example using the python Scapy library – in order to hijack the drone.
- **WiFi controlled** : The drone is controlled with an app through WiFi, and by default, it isn't even password-protected ! This allows any user running the application to control the drone. Even though a security can be enabled, called *Pairing*, which will make the drone drop the packets if the MAC address sending them is not the one it is paired with. It is nonetheless easy for the attacker to spoof the source MAC on the packets.

More information can be found on the subject at [18] as well as some investigations for improving the security at [19].

3.1.2 Known vulnerabilities

As defined in Subsection 2.1.1, CVE's are identifiers for publicly known vulnerabilities. It is therefore useful, prior to starting any pentest on drones, to look for eventual known vulnerabilities on the field. In the same way, CVE may be used once a certain service is identified on a device.

As for now, only one entry is available and concerns the DBPOWER U818A WIFI quadcopter drone :

- **CVE-2017-3209** : *The quadcopter drone provides **FTP** access over its own local access point, and allows full file permissions to the anonymous user. The DBPower U818A WIFI quadcopter drone runs an FTP server that by default **allows anonymous access without a password**, and provides full filesystem read/write permissions to the anonymous user. A remote user within range of the open access point on the drone may utilize the anonymous user of the FTP server to read arbitrary files, such as images and video recorded by the device, or to replace system files such as /etc/shadow to gain further access to the device. Furthermore, the DBPOWER U818A WIFI quadcopter drone uses BusyBox 1.20.2, which was released in 2012, and may be vulnerable to other known BusyBox vulnerabilities.*

It goes without saying that a single result is terribly weak, for example, we get nearly 1,500 results just for the Apache web server (as August 2019). This tends to show that a substantive work remains to be done in this area, but also that the door is open for real progress.

Here insert a note on some commercial projects such as:

- ApolloShield: https://www.apolloshield.com/
- Talk about the many "non-hacking" solutions

3.2 Scope definition

This section presents some models of drones selected for the study and the beginning of the application of the penetration testing methodology, that is, the *intelligence gathering*.

3.2.1 Scope limitation

The drones were provided by the professional supervisor within the limits of his allocated budget to cover a small spectrum of drone-related technologies in order to establish the basis of a framework (as it is explained in Chapter 5). For a question of scope, only a few models are selected to match some technical specifications as we focus on wireless penetration testing.

For this work, we received the following six different drones. Each of them comes from a different brand so that we can cover the wider spectrum of technologies possible.

- Drone S9
- Jamara Skip 3D Quadrocopter
- NINCOAIR QUADRONE MINI
- UdiR/C Free Loop U27
- Flitt Selfie Cam
- C-me 1080P WiFi FPV GPS Selfie Drone

In the initial problem statement, it was decided to develop plugins for at least 3 different popular commercial drones on the framework according to the exploits that could be discovered. Rather quickly, we excluded the first four since they were not WiFi- but radio-controlled and should require the use of a different technology than the one aimed in this work. It leaved us with the last two.

3.2.2 Flitt Selfie Cam

Flitt Flying Camera is a pocket-sized flyer that features an adjustable CMOS camera that can record 720p video at 30 fps and shoot 1.3MP photos. The Flitt uses 2.4 GHz Wi-Fi to communicate with either an Android or IOS free companion app. From the app, you can fly the Flitt, take pictures and video, adjust camera pitch, change the photo mode, and more.

Figure 3.1: User interface for piloting the Flitt Selfie Cam from a smartphone

The device must be unlocked in order to take off, auto-regulates its high and has an emergency landing feature in case of loss of control.

3.2.3 C-me Selfie Drone

C-me flying camera is designed to be the ultimate flying "selfie stick" for capturing life's memorable events thanks to its 8MP HD 1080p camera. It comes with an intuitive app control that interfaces with iOS and most Android phones and tablets. It communicates through 2.4 GHz Wi-Fi. Its size allows to easily fit in a pocket, purse or backpack.

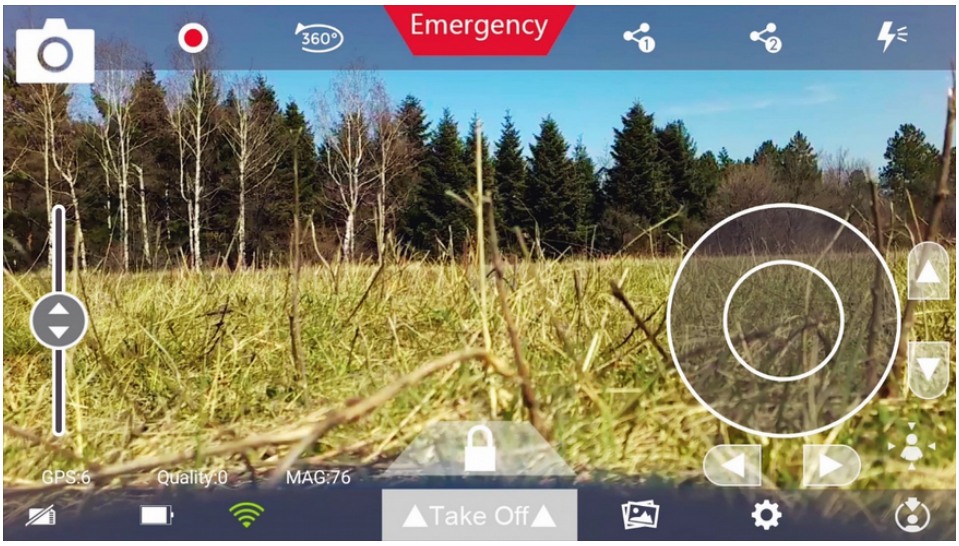

Figure 3.2: User interface for piloting the C-me Selfie Drone from a smartphone

Once again, the device must be unlocked in order to take off, auto-regulates its high and has an emergency landing feature in case of loss of control.

It is quite obvious that the interfaces are quite similar, and after further research, it turns out that the two drones are produced by the same manufacturer, i.e. Hobbico. This means that there will probably be similarities between the two devices, even though the C-me has more functionalities since it's a little bit more expensive than the Flitt (depending on the reseller, in the 150range versus 80). This will be a good occasion to compare them and see if one is more secure than the other.
Hobbico, Inc. was a manufacturer and distributor of hobby products including radio control airplanes, boats, cars, helicopters and drones. Unfortunately, on 2018, it was announced that Hobbico had filed for Chapter 7 bankruptcy and went into liquidation1, and the company was later bought by Horizon Hobby2. There website, hobbico.com, is no longer available online and there is no reference to any of the drones on the Horizon Hobby website meaning that there is no support anymore for them.

3.3 Intelligence gathering

3.4 Vulnerability analysis

4 Exploits

"I get hired by companies to hack into their systems and break into their physical facilities to find security holes. Our success rate is 100%; we've always found a hole."

Kevin Mitnick

Cybersecurity consultant

...

5 Framework

"Any program is only as good as it is useful."

LINUS TORVALDS

Creator of Linux

DRONESPLOIT, the framework we propose to build, is an attempt to mechanize drone hacking techniques in particular in a user-friendly way, exactly like Metasploit does for network penetration testing in general.

6 Conclusion

6.1 Summary 27
6.2 Objectives 27
6.3 Matched Criteria 27
6.4 Future Works. 27

"Device makers, especially consumer-focused ones, have been the Achilles' heel of IoT security. These vendors have often viewed proper security implementations as extra cost, complexity, and time-to-market burdens with an unclear payoff."

Maciej Kranz

Vice President of Strategic Innovation at Cisco Systems

...

6.1 Summary

...

6.2 Objectives

Our objectives were four-fold :

1. The **background** is fully stated.
   - A – We reviewed the current literature regarding IoT security, especially in the field of light commercial drones.
   - B – We found some methodologies and processes for hacking, especially the penetration testing process.
   - C – We presented some existing solutions and used a few ones in the exploits and the framework.
2. The **scope** was narrowed to only two drones.
   - A – Multiple provided drones were not suitable for WiFi exploitation and then withdrawn from the scope.
   - B – The working of the selected drones was studied.
3. Some **exploits** could be written and tested.
   - A – We found a few attacks chaining multiple hacking techniques.
   - B – We wrote a few exploit scripts proven to be effective.
4. A new **framework** is born, tailored to drone hacking.
   - A – The new framework was made on top of SploitKit and some scanning modules could be included.
   - B – The exploit scripts were turned into exploitation modules for DroneSploit.

6.3 Matched Criteria

The level of fulfilment of the evaluation criteria can be assessed as follows :

1. ...

6.4 Future Works

DroneSploit opens some new avenues of improvement :

- ...

References

[1] Patrick Engelbretson. *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy*. Syngress, 2011. ISBN: 978-0-12-411644-3.

[2] *Penetration Testing Execution Standard*. 2012. (Visited on 08/10/2019).

[3] Offensive Security. *Kali – Our Most Advanced Penetration Testing Distribution, Ever*. URL: https://www.kali.org (visited on 08/12/2019).

[4] Parrot Team. *Parrot OS – The advanced system for security experts, developers and crypto-addicted people*. URL: https://www.parrotsec.org/ (visited on 08/13/2019).

[5] Offensive Security. *Training, Certifications and Services*. URL: https://www.offensive-security.com (visited on 08/12/2019).

[6] *Kali Linux Tools Listing*. URL: http://tools.kali.org/tools-listing (visited on 08/12/2019).

[7] Rapid7. *Metasploit – Penetration Testing Software, Pen Testing Security*. URL: https://www.metasploit.com (visited on 08/04/2019).

[8] Johnny Long, Bill Gardner, and Justin Brown. *Google Hacking for Penetration Testers*. Third. Syngress, 2015.

[9] Shodan. *The world's first search engine for Internet-connected devices*. URL: https://www.shodan.io/ (visited on 08/10/2019).

[10] Paterva. *Maltego Clients*. URL: https://www.paterva.com/buy/maltego-clients.php (visited on 08/10/2019).

[11] Gordon Lyon. *Nmap: the Network Mapper – Free Security Scanner*. URL: https://nmap.org/ (visited on 08/01/2019).

[12] Wireshark Foundation. *Wireshark*. URL: https://www.wireshark.org/ (visited on 08/03/2019).

[13] Tenable. *Nessus – Comprehensive Vulnerability Assessment Solution*. URL: https://www.tenable.com/products/nessus (visited on 08/01/2019).

[14] Openwall. *John the Ripper password cracker*. URL: https://www.openwall.com/john (visited on 08/04/2019).

[15] THC. *Hydra*. URL: https://github.com/vanhauser-thc/thc-hydra (visited on 08/04/2019).

[16] PortSwigger. *Burp Suite Scanner*. URL: https://portswigger.net/burp (visited on 08/04/2019).

[17] Aircrack-ng. *Aircrack-ng*. URL: https://www.aircrack-ng.org/ (visited on 08/03/2019).

[18] Mark Szabo-Simon. *Let's hack a drone!* URL: https://github.com/markszabo/drone-hacking (visited on 07/27/2019).

[19] Johann Pleban, Ricardo Band, and Reiner Creutzburg. "Hacking and securing the AR.Drone 2.0 quadcopter - Investigations for improving the security of a toy". In: ResearchGate, 2014. DOI: 10.1117/12.2044868. URL: https://www.researchgate.net/publication/260420467.