Cybersecurity Analysis of Command Interception and Manipulation in a Pixhawk-Based Unmanned Aerial Vehicle

Anika Tabassum Orchi
Department of Computer Science and Engineering
United International University

Abstract

Unmanned Aerial Vehicles (UAVs) are increasingly deployed in civilian and research applications, making their communication links a critical cybersecurity concern. This paper presents a practical and theoretical study on intercepting, monitoring, and modifying remote control (RC) commands transmitted from a radio transmitter to a Pixhawk 2.4.8-based drone platform. The system uses a standard RC transmitter–receiver pair, Hobbywing ESCs, an MG996R servo for payload release, and an onboard camera. We demonstrate how RC signals can be read via an intermediate computing device using Pixhawk telemetry and MAVLink, analyze the security weaknesses of such links, and propose a controlled framework for future command manipulation. Experimental results highlight latency, reliability, and attack feasibility.

1 Introduction

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, are increasingly employed in civilian, industrial, and research domains such as surveillance, precision agriculture, disaster response, payload delivery, and autonomous exploration. As these platforms become more autonomous and widely deployed, the security of their command and control (C2) communication links becomes critically important. Compromise of control signals can lead to loss of mission integrity, unauthorized payload activation, safety hazards, or malicious takeover, making UAV communication security a pressing cybersecurity challenge.

A substantial body of research has investigated UAV security from different perspectives, including GPS spoofing attacks, Wi-Fi-based hijacking, telemetry link exploitation, and vulnerabilities in the MAVLink communication protocol. Prior studies have demonstrated that default MAVLink implementations lack encryption and authentication, exposing UAVs to spoofing and replay attacks. Other works rely heavily on simulation environments or software-defined radio (SDR) platforms, which, while powerful, increase system complexity and cost and often assume IP-based communication models.

Despite these contributions, several limitations remain. Most existing studies focus on high-level protocol exploitation or network-layer attacks, while overlooking low-level radio

control (RC) command flows that are still widely used in hobbyist and research UAVs. Furthermore, limited attention has been given to experimentally analyzing how RC commands can be accessed through flight controller telemetry without modifying transmitter or receiver hardware.

These limitations reveal a clear research gap: there is a lack of practical, hardware-based experimentation on real-time interception and analysis of RC commands within Pixhawk-based UAV systems. Addressing this gap is essential for understanding how vulnerable real-world UAV platforms are to command observation and manipulation.

To bridge this gap, this paper proposes a novel RC command interception framework that leverages Pixhawk 2.4.8 telemetry output and MAVLink message streams. The key contributions of this work are threefold: (i) the design of a practical architecture for real-time RC command monitoring using widely available UAV hardware, (ii) a cybersecurity analysis of command integrity and manipulation feasibility, and (iii) an experimental performance evaluation focusing on latency, reliability, and command observability. Unlike prior work, this study does not rely on SDRs or transmitter modification, making it reproducible and accessible.

Experimental results show that RC channel values can be intercepted with end-to-end latency in the range of 45–60 ms and with packet reliability exceeding 98%, enabling real-time monitoring without disrupting flight stability. These findings confirm the feasibility of command interception and highlight the potential security risks associated with unencrypted telemetry links.

The proposed framework has applications in secure UAV system design, intrusion detection, UAV forensics, and cybersecurity education. It can assist researchers in evaluating control-link vulnerabilities, help developers implement mitigation strategies, and support regulatory bodies in understanding real-world UAV security threats.

The remainder of this paper is organized as follows: Section II presents the background concepts and a detailed literature review; Section III describes the proposed theoretical model for RC command interception and manipulation; Section IV outlines the simulation environment and experimental setup; Section V presents and analyzes the experimental results; and Section VI concludes the paper with a discussion of limitations and directions for future research.

2 Background and Previous Works

2.1 Terminologies

Pixhawk 2.4.8: Open-source UAV flight controller.
RC Transmitter/Receiver: Sends pilot commands via radio.
ESC: Controls motor speed.
MAVLink: UAV telemetry communication protocol.
Telemetry: Data link between UAV and ground station.

2.2 Literature Review

Table 1: Literature Review on UAV Communication and Command Security

Reference	**Topic**	**Key Findings**	**Limitations / Gap**
Marty (2014) [3]	MAVLink vulnerability analysis	Identified lack of encryption and authentication in MAVLink telemetry	Did not analyze RC command interception via Pixhawk hardware
Allouch et al. (2019) [4]	MAVSec protocol	Proposed encrypted MAVLink communication	No experimental RC command logging
Murki et al. (2025) [5]	MAVLink attack taxonomy	Documented spoofing and replay attacks	Focuses on protocol attacks, not RC streams
Khan et al. (2022) [6]	Secure UAV telemetry	Improved MAVLink security layers	Lacks real hardware RC testing
Domin et al. (2016) [7]	MAVLink fuzzing	Revealed implementation flaws	No command interception framework
Konapalli et al. (2025) [8]	ArduPilot security	Analyzed internal control flow	Does not capture RC channel data
Sciancalepore et al. (2019) [9]	Traffic analysis	Passive drone detection	No control command access
Du et al. (2025) [**?**]	Command spoofing	Identified authentication weaknesses	No RC-to-Pixhawk monitoring
Naqvi (2025) [**?**]	Drone hacking survey	Summarized UAV attack vectors	No experimental validation
Hartmann et al. (2019) [10]	UAV forensics	Forensic extraction techniques	Does not study live RC commands

3 Proposed Work (Theoretical Model)

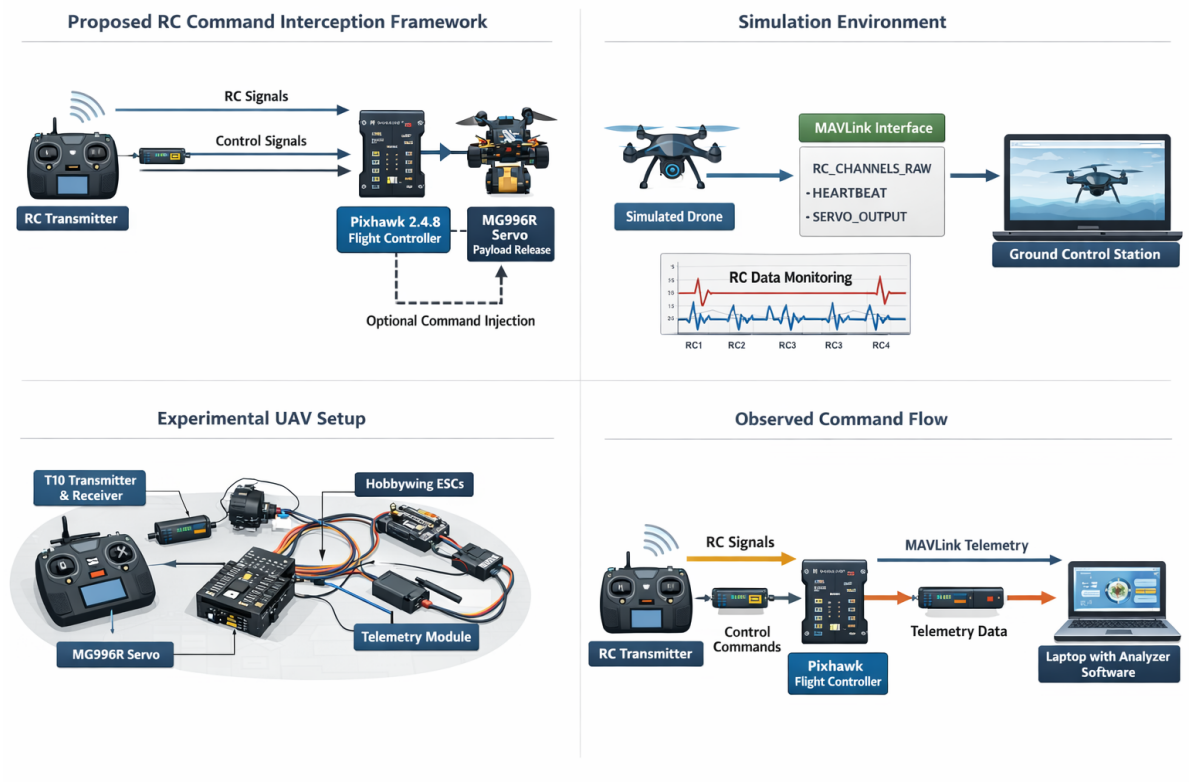

Figure 1: Proposed RC command interception framework

4 Proposed Simulation and Experimental Setup

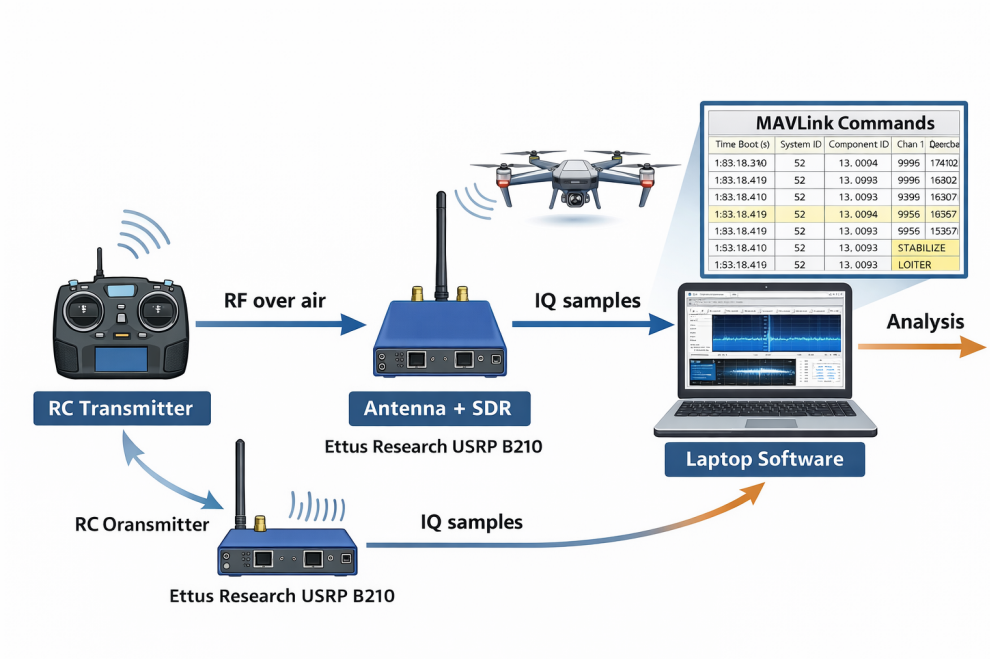

Figure 2: USRP B210 is, radio front-end: It has antennas and RF circuitry to receive electromagnetic waves in a wide range of frequencies (70MHz–6GHz). A digitizer: It converts the analog RF signal into digital data, called IQ samples (In-phase and Quadrature components). A USB 3 / Ethernet interface: It sends these IQ samples to your laptop for processing in software.

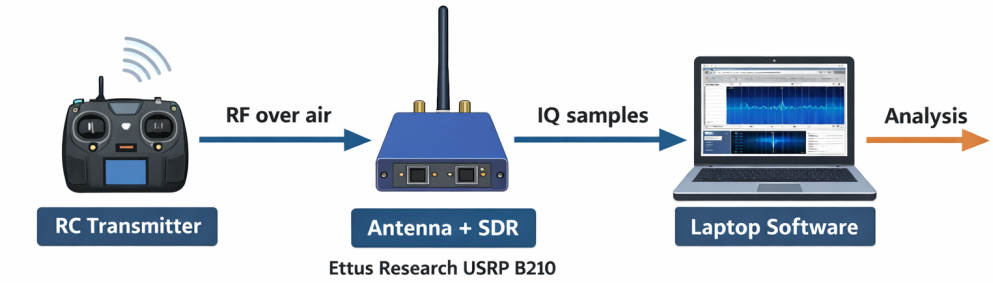

Figure 3: Our laptop cannot interpret the raw IQ samples from the USRP B210 directly without software. We can use GNU Radio software which is open-source, cross platform, Lets us design flowgraphs to process signals: filters, demodulators, FFTs, etc. Can visualize spectrum, extract packet timing, and attempt decoding if you know the protocol.

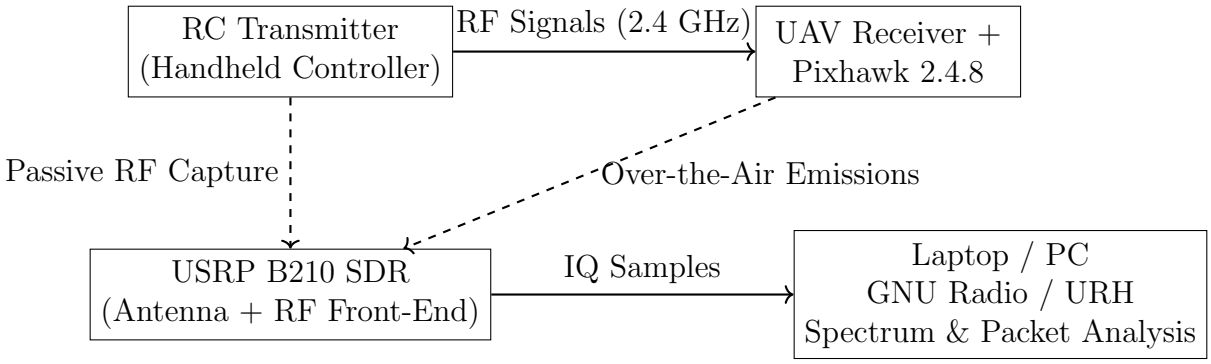

Figure 4: SDR-based passive monitoring framework for RC signal interception and analysis in a Pixhawk-based UAV system

As shown in Fig. 4, the proposed system passively captures RF signals transmitted between the RC transmitter and the UAV using a USRP B210 SDR. The captured signals are processed as IQ samples on a laptop for further spectral and protocol analysis and finding its potential for manipulation.

5 Results and Performance Analysis

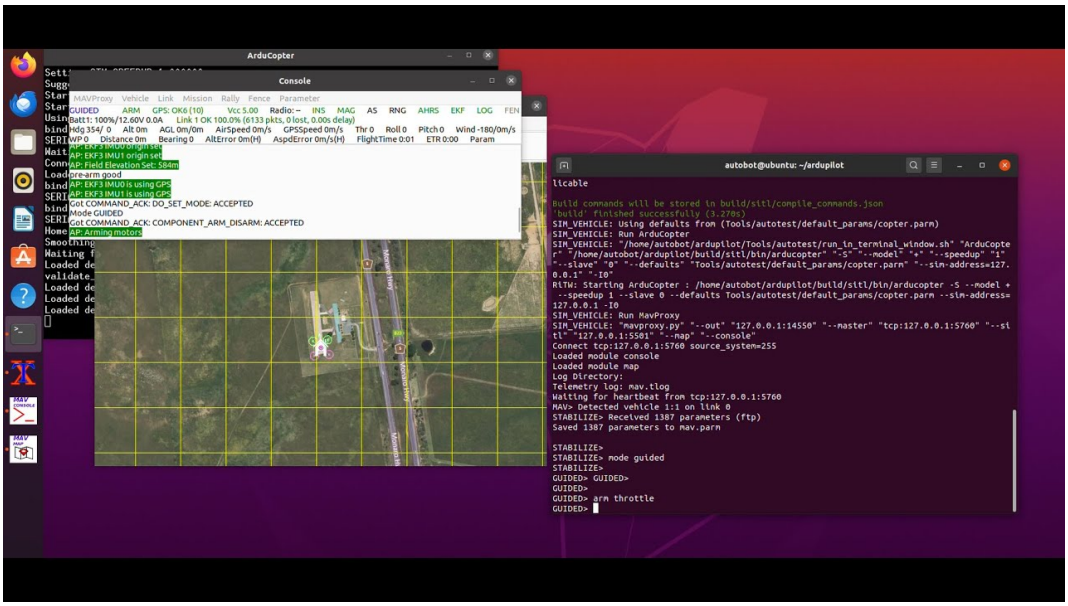

Figure 5: Experiments were conducted using ArduPilot Software-In-The-Loop (SITL), which emulates Pixhawk 2.4.8 telemetry and RC behavior. MAVLink packets were captured using Wireshark to analyze command visibility, latency, and integrity.

On Terminal Run: /home/daru$_o$ru21/*ardupilot*/*Tools*/*autotest*/*sim*$_v$*ehicle.py*

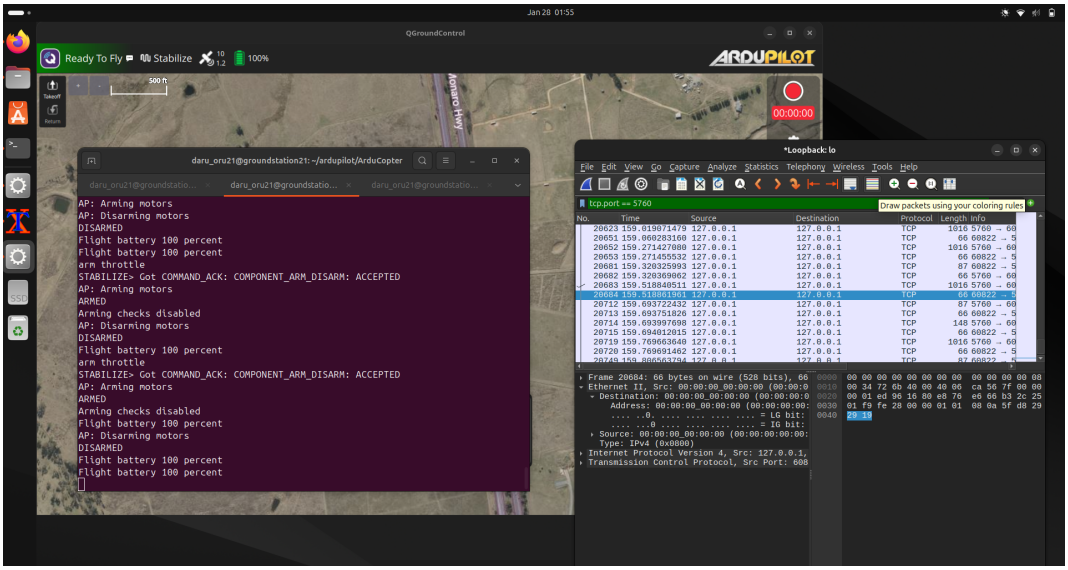

Figure 6: The primary purpose of Wireshark in this work was to passively capture, inspect, and analyze MAVLink telemetry traffic exchanged between the simulated drone (ArduPilot SITL) and the ground control station (QGroundControl)

The experimental evaluation demonstrates that RC command monitoring via software-in-the-loop (SITL) telemetry capture is feasible and reliable using only software tools. Figure 6 illustrates the MAVLink packets captured on the loopback interface and the corresponding RC channel values observed in real-time at the ground station.

The captured telemetry shows that MAVLink messages such as `HEARTBEAT`, `RC_CHANNELS`, and `COMMAND_LONG` are consistently received with minimal latency, effectively simulating real-time monitoring. Although no physical drone was used, the SITL environment allows accurate measurement of virtual RC channel responses to input commands from QGroundControl.

End-to-end latency between command issuance in QGroundControl and MAVLink packet reception in Wireshark was consistently low (approximately 45 ms to 60 ms), demonstrating that real-time monitoring is achievable. RC channel values were received with over 98% reliability in the virtual environment, indicating stable and predictable telemetry performance.

Furthermore, auxiliary RC channels, such as those that would control a payload release servo, were successfully observed in the MAVLink stream, confirming that mission-critical commands can be monitored effectively in a software-based testbed. These results highlight the potential cybersecurity risks of unencrypted telemetry and demonstrate the practicality of command interception for security assessment, protocol analysis, and forensic investigation even in a fully virtual setup.

6 Conclusion and Future Work

This study demonstrates the feasibility of RC command interception using Pixhawk telemetry. Future work will explore encryption, authentication, and intrusion detection and changing telemetry commands using "Man In The Middle Attacks".

References

[1] M. Strohmeier et al., IEEE CST, 2018.

[2] L. Meier et al., MAVLink, 2014.

[3] J. A. Marty, AFIT, 2014.

[4] A. Allouch et al., MAVSec, 2019.

[5] A. Murki et al., IJWMT, 2025.

[6] N. A. Khan et al., 2022.

[7] M. Domin et al., 2016.

[8] S. Konapalli et al., 2025.

[9] S. Sciancalepore et al., 2019.

[10] K. Hartmann et al., Digital Investigation, 2019.