KAUST and NCA Cybersecurity Program

Drones Security Testing Framework

Anas Alzahrani
Anas Alamri
Asim Barnawi
Mohammad Almahallawi

Table of Contents

1. Introduction 3
2. Problem Statement 3
3. Objectives 3
4. System Architecture 4
5. Attack Modules 4
6. Chain of Attacks 5
7. Contribution 6
8. Implementation Details 6
9. Testing and Validation 7
10. Future Work 7
11. Conclusion 8
12. References 8
13. Acknowledgements 8

Drone Security Testing Framework

1. Introduction

The rapid expansion of the global drone market has heightened the need for effective security measures to protect unmanned aerial vehicles (UAVs) from potential cyber threats. Drones are increasingly being utilized in critical sectors such as logistics, surveillance, and agriculture. However, their growing prevalence has also made them an attractive target for attackers. Despite this, there is a significant lack of accessible tools for safely and realistically testing drone vulnerabilities. Traditional testing methods are not only costly but also pose risks of damaging the drones themselves. This project, titled 'Drone Security Testing Framework,' is designed to address these security gaps by providing a budget-friendly and risk-free solution for assessing and enhancing the security of UAVs.

2. Problem Statement

As drones become more integrated into daily operations across various industries, their vulnerabilities are becoming more apparent. The current methods for testing and addressing these security concerns are either prohibitively expensive or inherently risky, leading to potential damage to the drones being tested. Additionally, the shortage of accessible tools for vulnerability testing exacerbates the problem, leaving critical security flaws unaddressed. The demand for practical and effective security solutions is growing, especially as drones are deployed in more critical roles.

3. Objectives

The primary objectives of the Drone Security Testing Framework are as follows:

- Assess UAV Vulnerabilities: To identify and analyze common security weaknesses in drones, focusing on both hardware and software components.

- Simulate Real-World Attacks: To create realistic attack scenarios that mirror potential threats faced by UAVs in real-world applications.

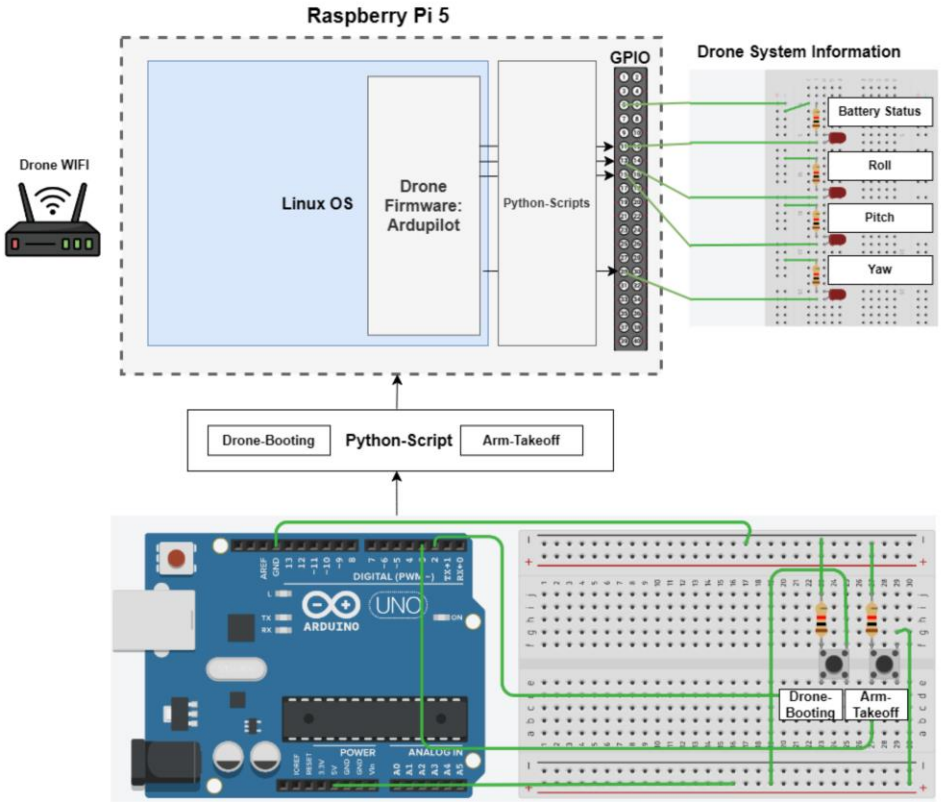
- Develop Mitigation Strategies: To design and implement countermeasures that can effectively prevent or mitigate the impact of identified vulnerabilities.

- Provide a Safe Testing Environment: To ensure that all tests are conducted in a controlled and risk-free environment, minimizing the likelihood of damaging the drone or other systems.

- Enable Accessibility: To create a framework that is both cost-effective and accessible to cybersecurity professionals, hobbyists, and researchers alike.

4. System Architecture

The Drone Security Testing Framework leverages a Raspberry Pi 5 as the core processing unit, running a Linux operating system and the ArduPilot firmware in a System-in-the-Loop (SITL) environment. The Raspberry Pi interfaces with an Arduino-based remote controller, which includes two essential functions: drone boot/reboot and takeoff. The Arduino is connected to the Raspberry Pi's GPIO pins, which are used to visually represent key drone metrics such as roll, pitch, yaw, and battery status through LEDs. This setup provides a tangible, real-time indication of the drone's state, making the testing process both intuitive and cost-effective.



5. Attack Modules

The framework incorporates a comprehensive suite of attack modules that target various vulnerabilities in drone systems. These modules cover a wide range of attack types, including:

- Wi-Fi Deauthentication: Disrupting the drone's connection to its control station by flooding the network with deauthentication frames.

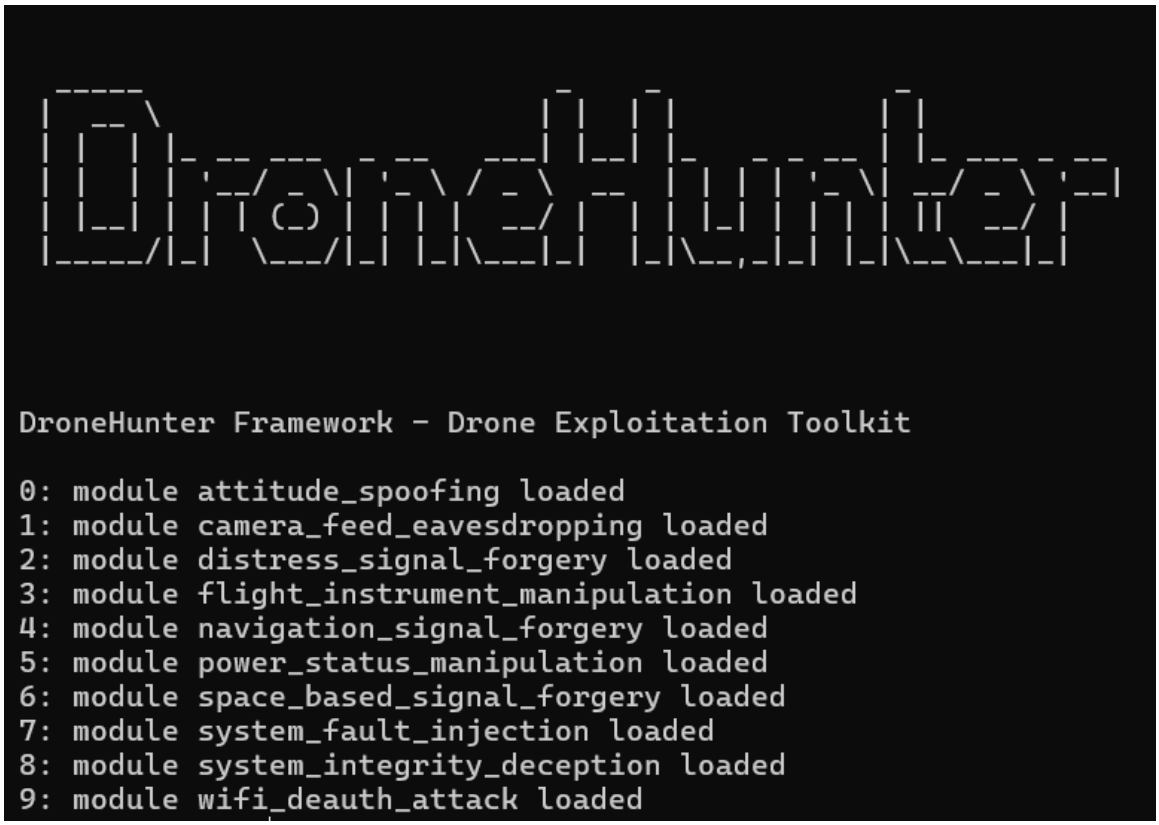
- Attitude Spoofing: Manipulating the drone's perceived orientation by injecting false data into the sensor feed, leading to erratic movements.

- System Fault Injection: Intentionally introducing errors into the drone's software to observe how it handles unexpected conditions.

- Camera Feed Eavesdropping: Intercepting the video feed from the drone's camera, allowing attackers to gain unauthorized access to visual data.

- Battery Information Manipulation: Falsifying the drone's battery status to cause unexpected shutdowns or forced landings.

- Satellite Information Manipulation: Altering GPS data to mislead the drone's navigation system.



6. Chain of Attacks

The framework's attack modules are designed to be executed in a sequence, forming a chain of attacks that progressively escalate the level of control over the drone. This chain of attacks typically follows these stages:

1. Initial Access: The attacker gains access to the drone's network, typically through Wi-Fi or LoRa.

2. Exploit the drone : After gaining network access, the attacker get control of the drone by exploiting vulnerabilities in the drone's firmware or communication protocols.

3. Persistence: The attacker establishes persistence by installing backdoors or manipulating the drone's settings to maintain control even after reboots.

4. Command and Control: With full access, the attacker can now issue commands to the drone, leading to various outcomes such as erratic movement, unauthorized data capture, or system shutdown.



Figure 3

7. Contribution

The framework is designed to achieve a balance between cost-effectiveness and risk mitigation. By using a Raspberry Pi 5 as the central platform, the project minimizes the need for expensive, specialized hardware. The use of LEDs to represent drone metrics provides a visual and interactive means of understanding the impact of various attacks without the need for actual drone hardware, further reducing the risk of damage. Additionally, the inclusion of the Drone Hunter Framework—a drone exploitation toolkit—enables users to simulate and explore various security vulnerabilities, offering a practical and comprehensive testing environment.

8. Implementation Details

The implementation of the Drone Security Testing Framework involves several key components:

- Raspberry Pi 5: Serving as the main computational unit, the Raspberry Pi runs the ArduPilot firmware in a SITL environment. It interfaces with the Arduino to control the physical indicators (LEDs) representing drone status.

- Arduino Uno: The Arduino controls the LEDs and receives commands from the Raspberry Pi for drone boot/reboot and takeoff functions. It is programmed to interpret signals from the Raspberry Pi's GPIO pins.

- LED Indicators: Connected to the Arduino, the LEDs provide real-time feedback on the drone's status, including roll, pitch, yaw, and battery life.

- Wi-Fi and LoRa Modules: These modules enable communication between the drone and its control station, simulating real-world data-link connections.

- Attack Scripts: Written in Python, these scripts automate the execution of various attack modules, allowing users to conduct tests with minimal manual intervention.

9. Testing and Validation

The effectiveness of the Drone Security Testing Framework was evaluated through a series of tests that simulated real-world attack scenarios. The framework was tested against several popular drone models, including those running the ArduPilot firmware. The following aspects were validated:

- Accuracy of Attack Simulations: Ensuring that the attack modules accurately replicate the intended vulnerabilities.

- System Stability: Assessing the stability of the framework under continuous operation and during the execution of complex attack chains.

10. Future Work

Future developments will focus on enhancing the framework's capabilities, particularly in the areas of communication security and long-range attack simulations. Planned improvements include:

- Robust Countermeasures: Implementing advanced encryption techniques to secure drone communications and mitigate the impact of attacks.

- LoRa Integration: Incorporating LoRa modules into both the Raspberry Pi and Arduino components to explore long-range communication capabilities and extend the framework's testing range.

- Advanced Attack Scenarios: Expanding the suite of attack modules to include more sophisticated and emerging threats, such as AI-driven attacks and autonomous swarm manipulation.

- Open Source Publication: To contribute to the broader cybersecurity community, the project will be released as open-source software, allowing professionals to build upon and refine the framework.

11. Conclusion

The Drone Security Testing Framework provides a practical, cost-effective solution for exploring and mitigating vulnerabilities in UAV systems. By simulating real-world attack scenarios in a controlled environment, the framework helps users understand the potential risks associated with drone operations and develop effective countermeasures. As drones continue to play a more significant role in various industries, the need for robust security measures becomes increasingly urgent. This framework represents a step forward in addressing these challenges and ensuring the safe and secure operation of drones in critical applications.

12. References

- ArduPilot Drone Firmware Build

- GGroundControl Documentation

- Raspberry Pi Documentation

- GitHub Repository: DroneHunterFramework

13. Acknowledgements

This project was developed as part of the KAUST-NCA Cybersecurity Program, with support from the National Cybersecurity Authority and King Abdullah University of Science and Technology. Special thanks to the program mentors and collaborators for their guidance and support throughout the development of this framework.