Defending Against Consumer Drone Privacy Attacks: A Blueprint for a Counter Autonomous Drone Tool

Lanier Watkins, PhD
Chair of Computer Science and Cybersecurity Programs
Engineering for Professionals
Johns Hopkins University
Whiting School of Engineering
Objective

• To perform an initial security assessment on the sensors, wireless network, and GPS of autonomous drones looking for “Hard-to-Patch” Vulnerabilities

• To use these “Hard-to-Patch” Vulnerabilities to design a novel Counter Autonomous Drone Tool
Motivation

Drone Industry Faces Issues On All Fronts

• Privacy
  ➢ Drones can be used to spy on you and your family

• National Security
  ➢ Drones can be used to kill

• Consumer Safety
  ➢ Vendors do not sufficiently warn consumers of security risks
Agenda

• Introduction to the Rouge Drone Problem
• Notional Autonomous Drone
• Our Approach: Finding Hard-to-Patch Vulnerabilities
• Related Works
• Experimental Evaluation
• Results and Discussion
• Counter Autonomous Drone Tool Design
• Conclusion and Future Work
Introduction

Rouge Drone Problem (2015 – Present)
• Last past 5 years this problem has been exacerbating
  ➢ Current issue, user controlled drones
  ➢ Autonomous drones, future issue
  ➢ Endangering critical infrastructure and private citizens
• Don’t take my word for it, let’s hear from government officials, journalist, and experts [1][2][3][4]
Notional Autonomous Drone

4 Levels of Autonomy [5]:

- Level 0: fully user controlled – manual
- Level 1: semi-autonomous (low) - user makes the rules, drone follows them
- Level 2: semi-autonomous (high) - drone makes its own rules, user approves them
- Level 3: fully autonomous - drone makes its own rules and executes them at will

Autonomous drones have embedded systems that can:

- Communicates with the drone’s:
  - Wireless network
  - Rotors
  - Sensors (camera, collision avoidance, inertial unit)

- Execute code for:
  - Autonomy – manages systems in drone to achieve goals
  - Mission Planner - provides an overall goal for drone
  - Flight Planner – interfaces with GPS to produce coordinates
DJI Autonomous Drones

DJI Active Track [6]
• Level 1: semi-autonomous (low) - user makes the rules, drone follows them
  ➢ Allows user to select a target to track and record
  ➢ Using the camera and sensors, drone autonomously follows and records target while avoiding obstacles

DJI Spark Highlights [7]
➢ User can connect using smartphone and DJI Go app over Wi-Fi
➢ Active Track
➢ Infrared collision avoidance
➢ Camera vision tracking
➢ GPS

DJI Phantom 4 Highlights [8]
➢ User can connect using smartphone and DJI Go app over RF
➢ Active Track
➢ GPS
➢ Camera vision tracking and collision avoidance
1. Develop UAS Security Focused Taxonomies
   - Our approach is to classify sUAS in terms of its main components (i.e., potential attack surfaces):
     1. wireless network
     2. embedded system
     3. GPS
     4. navigational system
     5. autonomy
   - Taxonomies facilitates penetration testing
2. Consider existing autonomous sUAS vulnerabilities
3. Perform zero-day penetration testing on multiple autonomous sUAS
4. Document successful exploit attack trees
5. Look across attack trees for multiple autonomous products
6. Build counter sUAS tool using Hard-to-Patch vulnerabilities
   - Hard-to-Patch vulnerabilities are likely cross vendor and based on financial infeasibilities (i.e., doesn't make financial sense to fix)
Related Work: User-Controlled Drone Security Assessments

- Watkins et al. [9]
  - Assessed the security of user-controlled drones by focusing on the major components
    - They broke COTS drones into 4 components:
      - wireless network
      - GPS
      - navigational system
      - embedded system.
  - They performed a security assessment of multi-vendor drones, found vulnerabilities, verified “Hard-to-Patch” with vendor, and weaponized vulnerabilities to produce a counter drone tool.
    - Counter drone tool was based on Wi-Fi de-authentication and fingerprinting

Our approach is similar, but the distinction is that we:
- Look solely at autonomous drones
- Propose a design for a counter autonomous drone tool

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>DJI Phantom 3 Response</th>
<th>Parrot Bebop II Response</th>
<th>3DR Solo Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARP Replay Attack*</td>
<td>Mobile Device Disconnect</td>
<td>Mobile Device Disconnect</td>
<td>Wi-Fi Controller Disconnect</td>
</tr>
<tr>
<td>MDNS Replay Attack</td>
<td>Not Vulnerable</td>
<td>Mobile Device Disconnect</td>
<td>Not Vulnerable</td>
</tr>
<tr>
<td>MAVLink Command Injection Attack</td>
<td>Not Vulnerable</td>
<td>Subverts Primary Controller</td>
<td>Subverts Wi-Fi Controller</td>
</tr>
<tr>
<td>Aircrack-ng Deauthentication Attack*</td>
<td>Mobile Device Disconnect</td>
<td>Mobile Device Disconnect</td>
<td>Wi-Fi Controller Disconnect</td>
</tr>
<tr>
<td>Bebop I Denial of Service Attack</td>
<td>Not Vulnerable</td>
<td>Not Vulnerable</td>
<td>Not Vulnerable</td>
</tr>
<tr>
<td>Bebop I Buffer Overflow Attack</td>
<td>Not Vulnerable</td>
<td>Not Vulnerable</td>
<td>Not Vulnerable</td>
</tr>
<tr>
<td>802.11 Protocol Stack Fingerprinting*</td>
<td>Uniquely identifies sUAS</td>
<td>Uniquely identifies sUAS</td>
<td>Uniquely identifies sUAS</td>
</tr>
</tbody>
</table>

*Hard-to-patch vulnerabilities (affect all top vendors) are highlighted in red
Related Work: User-Controlled Drone Security Assessments

• Birnbach et al. [10]
  ➢ Focused on privacy violation use cases
    • “Peeping Tom” drones
  ➢ Counter drone solution born from analysis of commonality of popular drones
    • Counter drone tool was based on Wi-Fi detection and tracking

Our approach is similar, but the distinction is that we:
• Look solely at autonomous drones
• Propose a design for a counter autonomous drone tool
Related Work: Autonomous Drone Security Assessments

- Apvrille et al. [11]
  - Short paper proposes to use SysML-Sec environment via TTool:
    - to preserve security and privacy in autonomous drone embedded system design
    - for formal verification of design
  - Demonstrates feasibility using autonomous Parrot drone

Our approach is similar, but the distinction is that we:
- Perform actual penetration testing on actual autonomous drones
  - Authors likely did not penetration test prototype
Experimental Setup

- **Autonomous Drones**
  - DJI Phantom 4
  - DJI Spark

- **Hardware**
  - Attack laptop
  - HackRF One
  - 1.5-foot Yagi 1.58GHz antenna
  - Smartphone
  - 1,220 Lux Multi-color LED Floodlight
  - 850 nm infrared spotlight
  - Indoor test facility

- **Software**
  - Kali Linux
  - Custom Python scripts
Experimental Procedure

• In our experimental procedure we:
  1. Performed remote security assessment on the sensors, wireless network, and GPS of each drone, looking for *Hard-to-Patch* vulnerabilities
  2. Developed exploits for each vulnerability found
  3. Communicated vulnerabilities to vendor and verified they would not patch vulnerabilities
  4. Designed a counter autonomous drone tool by using only *Hard-to-Patch* vulnerabilities
Normal DJI Active Track Behavior Experiment

### TABLE I. NORMAL ATRAK FLIGHT PLAN DATA

<table>
<thead>
<tr>
<th>OSP.ctrl Device</th>
<th>OSP.flight Action</th>
<th>OSP.fly cState</th>
<th>APP WARN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onboard Device</td>
<td>None</td>
<td>Nav/SubMode_Tracking</td>
<td></td>
</tr>
</tbody>
</table>

- **Device Controlling Drone**: None
- **Current Flight Mode**: Nav/SubMode_Tracking
- **Pre-programmed Flight Action**: None
- **Current Warnings**: None
Attacking Optical Sensor Experiment

TABLE II. ATrak Bright Light Attack Flight Plan Data

<table>
<thead>
<tr>
<th>OSD_ctrl Device</th>
<th>OSD_flight Action</th>
<th>OSD_flight eState</th>
<th>APP WARN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onboard Device</td>
<td>None</td>
<td>NaviSubMode_Tracking</td>
<td></td>
</tr>
<tr>
<td>RC</td>
<td>None</td>
<td>GPS_Atti</td>
<td>Subject Lost</td>
</tr>
</tbody>
</table>

Denotes abrupt change in control device
Attacking Collision Avoidance Sensor Experiment

TABLE III. ATrak Infrared Attack Flight Plan Data

<table>
<thead>
<tr>
<th>OSD.ctrl Device</th>
<th>OSD.flight Action</th>
<th>OSD.fly cState</th>
<th>APP WARN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onboard Device</td>
<td>None</td>
<td>NaviSubMode_Tracking</td>
<td></td>
</tr>
<tr>
<td>RC</td>
<td>None</td>
<td>GPS_Atu</td>
<td></td>
</tr>
</tbody>
</table>

Denotes abrupt change in control device
Attacking GPS Experiment

Drone forced out of autonomous mode
De-authenticating drone’s controller breaks Active Track

Drone forced out of autonomous mode

<table>
<thead>
<tr>
<th>OSD.ctrl Device</th>
<th>OSD.flight Action</th>
<th>OSD.fly cState</th>
<th>APP WARN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Onboard Device</td>
<td>None</td>
<td>NaviSubMode_Tracking</td>
<td></td>
</tr>
<tr>
<td>RC</td>
<td>GoHome</td>
<td>AutoLanding</td>
<td></td>
</tr>
</tbody>
</table>
Summary of Results

Risks Associated With These Vulnerabilities

- **The Bad**
  - Consumer Safety
    - While in Active Track Mode, thieves could steal drone

- **The Good**
  - National Security & Citizen Privacy
    - Weaponized vulnerabilities could be used to neutralize threats

### TABLE II. SUMMARY OF AUTONOMOUS DRONE VULNERABILITIES

<table>
<thead>
<tr>
<th>Drone</th>
<th>Component</th>
<th>Vulnerability</th>
<th>Range</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>P4/Spark</td>
<td>Optical Sensor</td>
<td>1,220 Lux White Light</td>
<td>≤ 3m*</td>
<td>Breaks Autonomy Code and Hovers</td>
</tr>
<tr>
<td>P4/Spark</td>
<td>GPS</td>
<td>GPS Spoofing</td>
<td>≤ 3m@</td>
<td>Breaks Autonomy Code and Lands</td>
</tr>
<tr>
<td>Spark</td>
<td>Wireless Network</td>
<td>Wi-Fi Deauth.</td>
<td>≤ 20m</td>
<td>Break Autonomy Code and Lands</td>
</tr>
<tr>
<td>Spark</td>
<td>IR Sensor</td>
<td>850nm IR Light</td>
<td>≤ 3m*</td>
<td>Breaks Autonomy Code and Hovers</td>
</tr>
</tbody>
</table>

*Extended by increasing intensity
@Extended by using better antenna
Counter Autonomous Drone Tool Design

Autonomous Drone Tool Design:

1. Detect autonomous drones using HackRF One
   - Major challenge
   - Discern between DJI drone and local networks Wi-Fi
   - Non-Wi-Fi DJI drones operate in 2.4GHz frequency band just like Wi-Fi drones

2. Mitigate autonomous drones using weaponized vulnerabilities
Future Work

• In future work, we plan to:
  1. Collaborate with RF Engineers to build Counter Autonomous Drone Tool
  2. Test and refine Counter Autonomous Drone Tool
  3. Work with DJI to reduce security risks for consumers
1. https://www.youtube.com/watch?v=SCJDlzayPMk
2. https://www.youtube.com/watch?v=BwjRY5oQtaA
3. https://www.youtube.com/watch?v=uh3iHa33kQY
4. https://www.youtube.com/watch?v=boPzM0YW53A
Questions?

Lanier Watkins, PhD
JHU EP Program Chair, Computer Science and Cybersecurity
The Johns Hopkins University
Lanier.Watkins@jhuapl.edu
404-406-5426