A Security Assessment for Consumer Wifi Drones

Joshua Gordon*, Victoria Kraj†, Ji Hun Hwang∓, Ashok Raja*
* Department of ECSSE, Embry-Riddle Aeronautical University
† Department of Literature, Media, and Communications, Georgia Institute of Technology
∓ Department of Mathematics and Statistics, University of Massachusetts Amherst
Emails: *{gordon32,rajaa}@my.erau.edu, †victoria.kraj@gatech.edu,∓jihunhwang@umass.edu

***Abstract*—Small-scale unmanned aerial vehicles (UAVs) have become an increased presence in recent years due to the their decreasing price and ease of use. Similarly, ways to detect drones through easily accessible programs like WireShark have raised more potential threats, including an increase in ease of jamming and spoofing drones utilizing commercially of the shelf (COTS) equipment like software defined radio (SDR). Given these advancements, an active area of research is drone security. Recent research has focused on using a HackRF SDR to perform eavesdropping or jamming attacks; however, most have failed to show a proposed remediation. Similarly, many research papers show post analysis of communications, but seem to lack a conclusive demonstration of command manipulation. Our security assessment shows clear steps in the manipulation of a WiFi drone using the aircrack-ng suite without the need for additional equipment like a SDR. This shows that anyone with access to a computer could potentially take down a drone. Alarmingly, we found that the COTS WiFi drone in our experiment still lacked the simple security measure of a password, and were very easily able to take over the drone in a deauthorization attack. We include a proposed remediation to mitigate the preformed attack and assess the entire process using the STRIDE and DREAD models. In doing so, we demonstrate a full attack process and provide a resolution to said attack.**

***Index Terms*—UAS, cybersecurity, deauthentication, deauthorization, denial-of-service.**[1]

I. Introduction

Due to the decreasing prices, increasing ease of use, and increasing commercially off the shelf (COTS) availability, small-scale unmanned aerial vehicles (UAVs) have gained a lot of attention in the public eye. According to the US Federal Aviation Administration (FAA), small-scale UAVs are going to reach 3.17 million by 2022 [1]. Increases in consumer size drones, have lead to instances such as a drone landing on the lawn of the United States' White House, and stir questions surrounding topics of: security, defense, and drone detection. Drone detection has been an active area of research as seen through [2]–[10].

Such increase in popularity has made UAVs an interesting subject of research. Drones, especially the commercial ones, are now classified as an Internet of Things (IoT); WiFi network is used make a connection between a UAV and its controller, allows users to control drones from devices such as laptops or smartphones. As the use of drones can pose threats to the third party, researches on protecting ones from malicious drones/users are actively ongoing today. Such protection system includes, but not limited to, detecting drones using laser signal and physically shooting down, emitting a high energy shock wave that is strong and effective enough to jam incoming drones' communication system. However, there is a reverse side of such development in UAV prevention. Malicious users can purposefully use the newly developed prevention technology to steal or hijack innocent UAVs.

The main purpose of this paper is to emphasize the vulnerability, in terms of cybersecurity, of commercial WiFi drones that are currently being used widely among civilians. This paper describes how the deauthentication (deauth) attack can be done against an airborne UAV and how effective it could be. Deauth attack was specifically chosen for this research, as it is one of the easiest form of cyber attack that is very accessible (e.g. aircrack-ng) and easy to comprehend. The attack is then be analyzed under STRIDE and DREAD models. This paper then discusses what can be improved as remediation to this vulnerability.

A. WiFi Drones and IoT

IoT or the "Internet of Things" is the interconnection of devices that do not follow a traditional computer, server, or network infrastructure format. These devices include smart devices such as televisions, thermostats, refrigerators, and even door locks. Therefore, WiFi based UAS would also fall under this categorization. One of the main downfalls of IoT is competitive pricing. Securing a system will always cost company resources and funds to perform. Should this process threaten the bottom line of a product, the company will choose the bottom line over security almost every time. Since IoT is designed to be affordable for all consumers, security is typically an after thought and sometimes not even apart of the development process.

1) *Message Integrity*

   Message integrity can be used to measure the validity of the source; whether the message has came from a reliable user or not. In cryptography, hashing algorithm is widely used for message integrity as it is mathematically almost impossible for one to decrypt a message to its original message as hash functions cannot be one-to-one. Hashing algorithms are also Shannon secure as a slight change on a message could cause a dramatic change once encrypted.

   Arora in [11] provides a new method of implementing a message integrity using universal unique identifier

[1] Project Video https://youtu.be/aPtElNXoY6k

(UUID) and a hashing algorithm SHA-512. Upon authentication, the client generates a UUID u_1, hashes it using SHA-512 and call it h_1, and include it in the association request frame (ARF). Once the access point receives the ARF, it checks whether SHA-512 is already in its memory. If it does, then the AP randomly generates a UUID u2 and stores it in the memory, hash it (call it h2), and then includes h2 in the association response frame and sends it to the AP. If it does not, the AP rejects ARF and considers it as a replay attack.

2) *MAC Address Filtering*
Liu *et al*. in [12] had experimentally proven that MAC address filtering (MAF) can be used to stop authentication request flooding (AUTHRF) or association request flooding (ASSRF) attack by comparing the MAC address of a malicious user's with saved MAC addresses in ap_control table. It is effective and easy to implement. However, it is vulnerable to the attackers who sniff or spoof MAC address [12].
However, Wright in [13] and [14] argues that using MAC spoofing can be detected using sequence number analysis. Basically, the network is monitored to find abnormalities in the selection of sequence numbers. Spoofed MAC that does not match with the pattern can be identified as anomalous, and then our access point transmits a deauthentication packet with a spoofed source MAC.

3) *Intrusion Detection System*
The biggest disadvantage of a DoS type attack, compared to other kinds of cyber attacks, is that it is easy to be detected, although it may be difficult to be prevented. There are many ways to determine whether a new client is a malicious intruder or not, including the MAC address filtering mentioned above.
IDS can sustain a DoS attack against a target (MAC address) and prevent access to the network by repeating the transmission of these frames; when IDS identifies an unauthorized station on a wireless network, it may attempt to prevent the station from accessing network resources [14]. Wright in [13] also suggests a way of detecting an unusual MAC address by observing and comparing the signal strength.
Agarwal *et al*. in [15] had used a machine learning (ML) technique to design a new intruder detection system (IDS) with accuracy of approximately 95%; it also helps detecting the existence of DoS attack, and can be used for both open and encrypted networks. The IDS made by Agarwal *et al*. in [15] were trained based on 18 different features to identify flooding attacks: number of TCP frames, inter-frame distance, DNS frames, association request frames, UDP frames, etc. Also, it does not require protocol modification or encryption algorithms.

4) *WatchDog Timer*
The main point of DoS attack is to freeze a machine by exhausting its finite memory. In [16], Hooper *et al*. suggests that installing a timer that limits the runtime of a CPU could reduce the damage from DoS attack. To be specific, 'watchdog timer' will measure the time a CPU has spent for non-navigational processing, temporarily stops the CPU's processing once it finds out that the CPU is using too much of its power for something not productive, such as processing a flood of meaningless deauthentication packets.

5) *Security Channel*
Another way of preventing a network from DoS attacks is to design a private, secured network/channel that no one else can access into. Yoon *et al*. in [17] proposed a new UAV network by utilizing two different communication channels between a UAV and its paired controller. Primary channel is for regular communication and security channel is for current time and public key before take-off, and for key-table values and an index (the location of the public key in the array) after take-off. Before take off, the ground station (GS) computes a public key using AES based on time, and transmits an array of random values with the public key inside to the UAV. GS will monitor UAV continuously and authentication between UAV and GS starts and repeats periodically every 60 seconds (authentication uses the values in key-table for its keys). Once GS detects an unusual activity, GS immediately disables the primary channel that was used to communicate with the UAV.

II. Related Works

A. Line-of-Sight

A wireless network, like the ones set up for wifi drones, requires both a transceiver and transmitter. An UAV controller sends a packet of signals to its paired UAV. Once the UAV receives and processes the signal, the UAV then sends a signal back to the controller acknowledging that signal has read. Line-of-Sight (LOS) propagation is when the transmitter has a visual on its paired transceiver, thus signal packets can propagate from transmitter to transceiver through a direct path. Non-Line-of-Sight (NLOS) propagation is when a presence of an obstacle is blocking the line of sight from the transmitter to transceiver. In most cases, NLOS propagation faces more challenge than LOS due to diffraction and interference. Such issues can come from electromagnetic fields around electric cables, which makes NLOS of WLAN networks especially challenging.

B. Signal Modulation

Signal modulation is a technique of converting a electrical signal into a wave conveying the same information and can be transmitted conveniently, by modulating the basic characteristics of original carrier wave. Carrier waves is a wave with high frequency, persistent and steady; hence, information can be written on to the carrier wave by segmentally editing the parameters of the given carrier waves (such as amplitude, phase, and frequency), based on the information one wish to send.

Signal modulations are not restricted to encoding a signal onto a carrier wave. UAVs, in order to efficiently maneuver and communicate with its ground controller, utilize the following modulations as well.

Orthogonal Frequency Division Multiplexing (OFDM) is classified as a digital modulation. In contrary to conventional modulations where a signal is encoded on a carrier waves, OFDM uses multiple carrier frequencies called subcarriers in order to transmit larger amount of digital data via a RF wave. OFDM decomposes a carrier into a group of subcarriers and each subcarrier is modulated by QPSK or QAM individually [18]. An OFDM signal contains more than one modulated signals, each orthogonal to each other, transmitted simultaneously in parallel [19]; an OFDM signal consists of N sinuoids with spacing $1/T$ where T is a the symbol period [20]. This allows subcarriers be orthogonal to each other, and therefore signals can be overlapped with each other without being interfered.

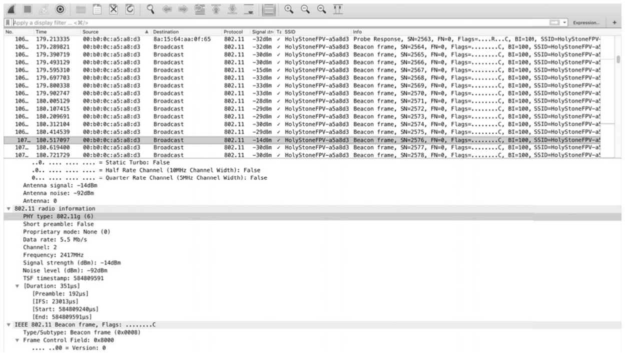
Using OFDM reduce the effects of interference, and OFDM signals can easily be modulated/demodulated efficiently by using Fast Fourier transform algorithm [20]. OFDM, on the other hand, is known to be vulnerable to Doppler effect. However, its effect on WLAN networks (2.4GHz or 5GHz) is very minimal since it travels as fast as light travels. Thus, it is popularly used for WiFi drones.

C. Listening to a WiFi Drone

Due to the lowering prices of software defined radio and open platforms in utilizing it, such as, GNURadio, we utilized an Universal Software Radio Peripheral (USRP) B210 [21] to receive packets through Wireshark [22] 1. A USRP B210 was used to conduct this research due to its superior range and power compared to other COTS SDRs This research utilizes GNU Radio since it is a widely used, free, open source software (OSS) development kit [23]. GNU Radio is mainly developed and tested on Linux distributions, keeping the Linux distributions more up to date and bug free [24]. For this reason, the researchers used Linux Ubuntu 18.04. It was also noted that dual booting a computer with the Linux distribution, works best for the software in this project. This in part because running virtual environments, like VMware Fusion and VirtualBox, was found to be too slow to be able to process signals. This is due to USB drivers being one of numerous important contributors to latency [25]. This is important because SDRs connect to computers via USB, thus virtual environments created a lag in signal capturing that is not seen in dual booted computers.

To check that the computer is able to detect the USRP, run the terminal command: "`uhd_find_devices`." Next, we recommend running a simple FM radio station GRC code to make sure that the system is capturing packets with limited noise and lag. This file is contained in the GNU Radio download and can be located at the download path inside: "gnuradio/gr-uhd/example/grc/uhd_wbfm_receive.grc" [23].

Wireshark is a network packet analyzer and perfectly suited for analyzing traffic on the 2.4GHz bandwidth. It is capable of monitoring specified media such as a network interface cards or serial ports to log all traffic it is connected to regardless if intended for the hosting computer. This log is saved as a packet capture and can then be analyzed offline with built-in or third-party tools. Utilizing IEEE802.11's github example "rx_demo.sh", which is also installed with GNURadio in a path like: "/usr/share/gnuradio/digital/ofdm/rx_receiver," we were able to capture the packets coming over WiFi from the drone. While there has been research done in the demodulation of OFDM, and githubs like gr-ofdm [26], when we tried to utilize these they were either: decrepit, had code missing that the author did not share, or the researchers in the paper did not include their process of how they successfully demodulated OFDM. Due to this, we decided to chose a more succinct path into hacking a drone that could be successfully accomplished by almost anyone with access to a computer.



```
No.    Time        Source              Destination        Protocol Signal str Tx SSID                 Info
 106…  179.213335  00:b0:0c:a5:a8:d3   8a:15:64:aa:0f:65  802.11   -32dBm  ✓  HolyStoneFPV-a5a8d3  Probe Response, SN=2563, FN=0, Flags=....R...C, BI=100, SSID=HolyStoneFPV-
 106…  179.289821  00:b0:0c:a5:a8:d3   Broadcast          802.11   -32dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2564, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  179.390719  00:b0:0c:a5:a8:d3   Broadcast          802.11   -30dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2565, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  179.493129  00:b0:0c:a5:a8:d3   Broadcast          802.11   -30dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2566, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  179.595310  00:b0:0c:a5:a8:d3   Broadcast          802.11   -15dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2567, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  179.697703  00:b0:0c:a5:a8:d3   Broadcast          802.11   -33dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2568, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  179.800338  00:b0:0c:a5:a8:d3   Broadcast          802.11   -33dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2569, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  179.902747  00:b0:0c:a5:a8:d3   Broadcast          802.11   -33dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2570, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  180.005129  00:b0:0c:a5:a8:d3   Broadcast          802.11   -28dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2571, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  180.107415  00:b0:0c:a5:a8:d3   Broadcast          802.11   -29dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2572, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  180.209691  00:b0:0c:a5:a8:d3   Broadcast          802.11   -29dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2573, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  180.312104  00:b0:0c:a5:a8:d3   Broadcast          802.11   -30dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2574, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 106…  180.414539  00:b0:0c:a5:a8:d3   Broadcast          802.11   -29dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2575, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
  107… 180.517097  00:b0:0c:a5:a8:d3   Broadcast          802.11   -14dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2576, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 107…  180.619400  00:b0:0c:a5:a8:d3   Broadcast          802.11   -30dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2577, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
 107…  180.721729  00:b0:0c:a5:a8:d3   Broadcast          802.11   -30dBm  ✓  HolyStoneFPV-a5a8d3  Beacon frame, SN=2578, FN=0, Flags=........C, BI=100, SSID=HolyStoneFPV-a5
        ..0. .... .... .... = Static Turbo: False
        .0.. .... .... .... = Half Rate Channel (10MHz Channel Width): False
        0... .... .... .... = Quarter Rate Channel (5MHz Channel Width): False
      Antenna signal: -14dBm
      Antenna noise: -92dBm
      Antenna: 0
  ▼ 802.11 radio information
      PHY type: 802.11g (6)
      Short preamble: False
      Proprietary mode: None (0)
      Data rate: 5.5 Mb/s
      Channel: 2
      Frequency: 2417MHz
      Signal strength (dBm): -14dBm
      Noise level (dBm): -92dBm
      TSF timestamp: 584809591
    ▼ [Duration: 351µs]
        [Preamble: 192µs]
        [IFS: 23013µs]
        [Start: 584809240µs]
        [End: 584809591µs]
  ▼ IEEE 802.11 Beacon frame, Flags: ........C
      Type/Subtype: Beacon frame (0x0008)
    ▼ Frame Control Field: 0x8000
        .... ..00 = Version: 0
```

Fig. 1. This figure shows WireShark listening to the packet exchange between the HolyStone and its remote controller. In the header information, WireShark tells us that the physical layer type is 802.11g, and that the drone is found on channel 2.

III. Proposed Method

A. Target Drone Architecture

For this research, we utilized a COTS HolyStone WiFi Drone. According to the manufacturerI, we know that the drone works with 802.11b, 802.11g and 802.11n. We also know that it operates on channel number 11 and at frequencies 2412 MHz - 2462 MHz. There is a 5 MHz separation for the channels and a 3.0 dBi antenna gain. The drone itself acts as the wireless access point with smart phones connecting as ground station clients. After examining packets coming from the drone with Wireshark, we were able to determine that our target WiFi communications occur using 802.11g with a frequency of 2417 MHz at channel 2. 2

B. Deauthentication and WPA Cracking

A Denial-of-Service (DoS) attack occurs when an adversary disables a system or a network by intentionally flooding it with a huge traffic of commands or messages, instead of harming a system by decrypting and eavesdropping the communication in network. DoS attack has gained its popularity as it does not requires much background knowledge on cryptography or network security. A DoS attack can be done without breaking a password or gaining a privilege to a system. Deauthentication

WiFi			
Supported Type	802.11b	802.11g	802.11n (H20)
Modulation	DSSS for 802.11b OFDM for 802.11g/802.11n (H20)		
Operation Frequency	2412 MHz - 2462 MHz for 802.11b/802.11g/802.11n (H20)		
Channel Number	11 for 802.11b/802.11g/802.11n (H20)		
Channel Separation	5 MHz		
Antenna Type	Integral Antenna		
Antenna Gain	3.0 dBi		

TABLE I
THE MANUFACTURER TEST REPORT FOR THE HOLYSTONE DRONE LISTS ITS MODULATION AS BEING OFDM FOR 802.11G

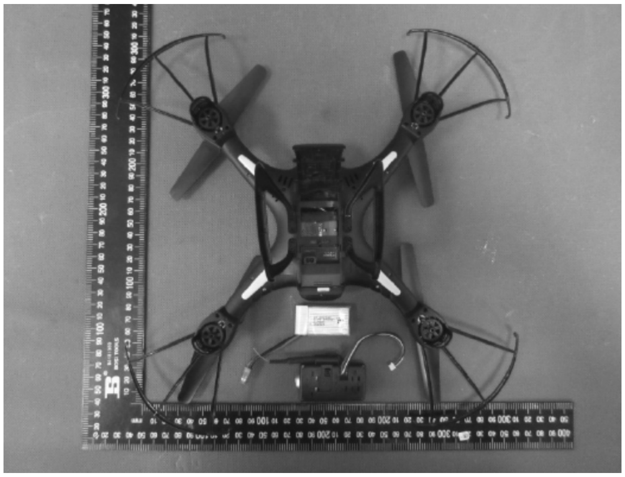


Fig. 2. This figure shows, from bottom to top, the Integral Antenna, the 3.7V battery, and then the drone flipped on its back

(Deauth) attack is a type of DoS attack that targets a user in WLAN connection. A Malicious user launches deauth packets to wireless access point (AP) to deceive the AP to believe it was sent from the real client, or the other way around [11]. It is known especially to be effective against 802.11a, b, g, and n; 802.11ac and w offers partial protection when it is encrypted with password [27]. 802.11i also provides a protecting via AES encryption scheme, providing confidentiality and integrity of the data transferred and received [15]. Deauth attack can often be inefficient if the attacker does not have enough transmit power for generating or sending packets, or the wireless AP does not have a public deauth code and attackers need to send a deauth code directed at selected client [27].

For this attack we will be using the Aircrack-ng suite of tools and will be broken down into a few phases. The first phase of this attack involves using airmon-ng to set our device to monitor mode to capture nearby 802.11 SSIDs. We will then use airodump-ng to find our target and get some more information on it such as the broadcasting MAC address. With this information we can now run airodump-ng again against our target MAC address and channel to capture the WPA/WPA2 handshakes. In order to force a handshake to be captured, we will perform a packet injection attack with aireplay-ng with our airodump-ng still listening. Once the deauth is complete, we can simply stop the attack and let the controller reconnect and perform the handshake we are easedropping for. Once we have acquired the handshake, we can load our packet capture into aircrack-ng and actually crack the password back into ASCII text. This can be done through rainbow tables, dictionaries, brute-force or hybrid attacks.

IV. EVALUATION

In order to properly assess the WiFi drone vulnerabilities and risks, we will follow the STRIDE and DREAD models. STRIDE and DREAD were developed by Microsoft and later accepted as a standard for assessment by the Cybersecurity community. The STRIDE Threat Model [28] II is designed to help categorize and define a vulnerability. Whereas, the DREAD Risk Model III is designed to give a qualitative analysis of that vulnerability or any additional risks that may not follow the STRIDE model. In our study, we found that most consumer drones either had no password or had a static default password using WPA encryption. We also found that the WPA protecting the drones was extremely susceptible to password rainbow tables developed from common default credentials.

A. *STRIDE Assessment*

In terms of the STRIDE Threat model, the deauth attack and hijack scenario is most applicable to Denial of Service and Tampering. However, should the drone not record proper logs of MAC addresses associated with the client connections, Repudiation could also be applied. One could also argue that during the hijack phase, we are performing a form of Elevation of Privileges as we are given both a valid client connection and full flight capabilities. Alternatively, if the drone has a password associated with the WiFi such as a WPA password, Information Disclosure could also be applicable should the threat actor crack the password for the WiFi signal. We do not at this time believe that Spoofing is applicable to this attack strategy.

B. *DREAD Assessment*

In terms of the DREAD Risk model, we will be evaluating the full hijack scenario against a drone with a weak WiFi WPA password. It should be noted that this model is being adapted to a UAS and exact definitions are best interpretations. The final outcome of this attack is a full system takeover and we therefore assign Damage Potential (D) as High(3). To reproduce this attack there are several tasks and assumptions

Threat	Definition
S - Spoofing	A threat actor is able to forge their identity
T - Tampering	The threat actor is able to manipulate data within a system
R - Repudiation	The threat actor is able to perform a non-attributable action
I - Information Disclosure	The threat actor is able to disclose private or confidential information
D - Denial of Service	The threat actor is able to prevent a legitimate user from accessing a system or service
E - Escalation of Privilege	The threat actor is able to elevate their current user status by bypassing security measures

TABLE II
STRIDE THREAT MODEL CATEGORY DEFINITIONS ACCORDING TO MICROSOFT[28]

Risk	High [3]	Medium [2]	Low [1]
D (Damage Potential)	The attacker can subvert the security system, get full trust authorization, run as administrator; upload content	Leaking sensitive information	Leaking trivial information
R (Reproducibility)	The attack can be reproduced every time and does not require a timing window.	The attack can be reproduced, but only with a timing window and a particular race situation.	The attack is very difficult to reproduce, even with knowledge of the security hole.
E (Exploitability)	A novice programmer could make the attack in a short time.	A skilled programmer could make the attack, then repeat the steps.	The attack requires an extremely skilled person and in-depth knowledge every time to exploit.
A (Affected Users)	All users, default configuration, key customers	Some users, non-default configuration	Very small percentage of users, obscure feature; affects anonymous users
D (Discoverability)	Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable.	The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use.	The bug is obscure, and it is unlikely that users will work out damage potential.
Final Cumulative Score Ranges	Score: 12-15	Score: 8-11	Score: 5-7

TABLE III
DREAD RISK MODEL USING 5-15 SCALE ACCORDING TO MICROSOFT[28]

that must be taken. The first is that the threat actor is physically in the area of an active flight. Stage 1 requires either already listening before the pairing phase of the drone is complete or, performing a deauth attack and listen to the reconnect. Stage 2 occurs after the handshake is acquired and involves actually cracking the password back into ASCII text. Stage 3 consists of starting another deauth attack and connecting the threat actor's phone to take up the opened socket. With this complete, the threat actor can now fly the drone as if it belonged to them and choose to either continue or stop their deauth attack against the true user's phone. Given this, Low(1) is an appropriate score for Reproducability (R). This attack utilizes the fairly well known Aircrack-ng framework and therefore we give Exploitability (E) a score of High(3). Since there is only one possible user connected to the drone at any time we give Affected Users (A) a score of High(3). This attack strategy is fairly common against typical WiFi but may not be an obvious choice for UAS. Therefore, we give Discoverability (D) a score of High(3). The final overall risk score would then be High(13) for this attack strategy making this an extremely dangerous attack.

V. CONCLUSION

A. Discussion of Prevention Strategies

The ability to stop all possible attacks in Cybersecurity is a paradox. However, creating preventative measures, security controls and best practice solutions is the most feasible defensive measure we can currently implement. As such, we humbly propose a few strategies to help in the securement of consumer UAS.

1) WiFi Encryption Protocols: Currently, many consumer WiFi based drones do not even implement WEP. Our first recommendation is to at the very least have a secured WiFi network with at least WPA encryption scheme. The recommended minimum solution is the implementation of WPA2 as it is more secure and faster than its predecessor. However, the preferred solution is the implementation of the new WPA3 scheme.

2) Best Password Practices: Using one of the standard WiFi encryption standards inherently introduces security flaws within symmetric encryption schemes. We recommend a few options to help strengthen future keys. The first would be to implement unique passwords per individual UAVs not UAV models. These passwords should be at minimum 15-20 characters with a maximum of 63 characters. They should include lower and upper case letters, numbers and special characters. If the minimum recommended requirements are implemented properly, the alphabet length should be 95 characters as WPA2 accepts the ASCII decimal range 32-126 inclusive. We also recommend that WPA2 is implemented with at least 128-bit AES but preferably 256-bit AES encryption. The total possible keys using this format would be equal to the below equation.

Let n be password length and f_k be the number of possible keys:

$$f_k = 95^n \tag{1}$$

It should be noted that password complexity will not prevent a deauthorization attack. This strategy is designed to prevent the reverse engineering of the password should the threat actor obtain the handshake of a connecting client.

VI. Acknowledgment

This research was supported by National Science Foundation under Grant No. CNS-1757781.

References

[1] F. A. A. (FAA), "Unmanned aircraft systems." [Online]. Available: https://www.faa.gov/dataresearch/aviation/aerospaceforecasts/media/UnmannedAircraftSystems.pdf

[2] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-uav detection and classification from rf fingerprints using machine learning techniques," *2019 IEEE Aerospace Conference*, pp. 1–13, 2019.

[3] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective rf-based detection of drones," in *Proceedings of the 2Nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet '16. New York, NY, USA: ACM, 2016, pp. 17–22. [Online]. Available: http://doi.acm.org/10.1145/2935620.2935632

[4] A. V. Raja, "Uncover the power of multipath: Detecting nlos drones using low-cost wifi devices," Ph.D. dissertation, Embry-Riddle Aeronautical University, 2019.

[5] X. Yue, Y. Liu, J. Wang, H. Song, and H. Cao, "Software defined radio and wireless acoustic networking for amateur drone surveillance," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 90–97, April 2018.

[6] S. Birnbach, R. Baker, and I. Martinovic, "Wi-fly?: Detecting privacy invasion attacks by consumer drones," in *NDSS*, 2017.

[7] H. Fu, S. Abeywickrama, L. Zhang, and C. Yuen, "Low-complexity portable passive drone surveillance via sdr-based signal processing," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 112–118, April 2018.

[8] J. Mezei, V. Fiaska, and A. Molnr, "Drone sound detection," *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 333–338, Nov 2015.

[9] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2315–2322, Aug 2018.

[10] J. Mezei and M. Andras, "Drone sound detection by correlation," *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics*, pp. 509–518, 05 2016.

[11] A. Arora, "Preventing wireless deauthentication attacks over 802.11 networks," *ArXiv*, vol. abs/1901.07301, 2019.

[12] L. Chibiao and J. Yu, "A solution to wlan authentication and association dos attacks," *IAENG International Journal of Computer Science*, vol. 34, 01 2007.

[13] J. L. Wright, "Detecting wireless lan mac address spoofing," 2003.

[14] ——, "Weaknesses in wireless lan session containment," 2005.

[15] M. Agarwal, D. Pasumarthi, S. Biswas, and S. Nandi, "Machine learning approach for detection of flooding dos attacks in 802.11 networks and attacker localization," *International Journal of Machine Learning and Cybernetics*, vol. 7, pp. 1035–1051, 2016.

[16] M. Hooper, Y. Tian, R. Zhou, B. Cao, A. P. Lauf, L. Watkins, W. H. Robinson, and W. Alexis, "Securing commercial wifi-based uavs from common security attacks," in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, Nov 2016, pp. 1213–1218.

[17] K. Yoon, D. Park, Y. Yim, K. Kim, S. K. Yang, and M. Robinson, "Security authentication system using encrypted channel on uav network," in *2017 First IEEE International Conference on Robotic Computing (IRC)*, April 2017, pp. 393–398.

[18] "Ofdm." [Online]. Available: https://www.mathworks.com/discovery/ofdm.html

[19] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An ieee 802.11a/g/p ofdm receiver for gnu radio," 08 2013.

[20] T. Pollet, M. Van Bladel, and M. Moeneclaey, "Ber sensitivity of ofdm systems to carrier frequency offset and wiener phase noise," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 191–193, Feb 1995.

[21] "Usrp hardware driver and usrp manual." [Online]. Available: https://files.ettus.com/manual/index.html

[22] G. Combs, "Wireshark," Jul 2019. [Online]. Available: https://www.wireshark.org/

[23] "Gnuradio github," Jul 2019. [Online]. Available: https://github.com/gnuradio/gnuradio

[24] D. C. Tucker and G. A. Tagliarini, "Prototyping with gnu radio and the usrp - where to begin," *IEEE*, 2009.

[25] N. B. Truong, Y.-J. Suh, and C. Yu, "Latency analysis in gnu radio/usrp-based software radio platforms," *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pp. 305–310, 2013.

[26] "gr-ofdm," Mar 2018. [Online]. Available: https://github.com/rwth-ti/gr-ofdm

[27] H. A. Noman, S. M. Abdullah, and H. I. Mohammed, "An automated approach to detect deauthentication and disassociation dos attacks on wireless 802 . 11 networks," 2015.

[28] J. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla, and A. Murukan, "Threat modeling," Jun 2003. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)

[29] B. Liang, *HOLYSTONES01 quadcopter wifi camera Test Report*. Shenzhen Huatongwei International Inspection Co., Ltd., 2017. [Online]. Available: https://fccid.io/2AJ55HOLYSTONES01/Test-Report/Test-report-3381453